MITSUBISHI ELECTRIC

Programmable Controller

MELSEC iQ-R series

MELSEC iQ-R CPU Module User's Manual (Application)

-R00CPU
-R01CPU
-R02CPU
-R04CPU
-R04ENCPU
-R08CPU
-R08ENCPU
-R08PCPU
-R08PSFCPU
-R08SFCPU
-R16CPU
-R16ENCPU
-R16PCPU
-R16PSFCPU
-R16SFCPU
-R32CPU
-R32ENCPU
-R32PCPU
-R32PSFCPU
-R32SFCPU
-R120CPU
-R120ENCPU
-R120PCPU
-R120PSFCPU
-R120SFCPU
-R6RFM
-R6PSFM
-R6SFM

'TRANSLATION OF THE ORIGINAL INSTRUCTIONS'

SAFETY PRECAUTIONS

(Read these precautions before using this product.)
Before using MELSEC iQ-R series programmable controllers, please read the manuals for the product and the relevant manuals introduced in those manuals carefully, and pay full attention to safety to handle the product correctly. If products are used in a different way from that specified by manufacturers, the protection function of the products may not work properly.
In this manual, the safety precautions are classified into two levels: "⚠WARNING" and "⚠CAUTION".

⚠ **WARNING**	Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.

⚠ **CAUTION**	Indicates that incorrect handling may cause hazardous conditions, resulting in minor or moderate injury or property damage.

Under some circumstances, failure to observe the precautions given under "⚠CAUTION" may lead to serious consequences.
Observe the precautions of both levels because they are important for personal and system safety.
Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

[Design Precautions]

⚠WARNING

- Configure safety circuits external to the programmable controller to ensure that the entire system operates safely even when a fault occurs in the external power supply or the programmable controller. Failure to do so may result in an accident due to an incorrect output or malfunction.
  (1) Emergency stop circuits, protection circuits, and protective interlock circuits for conflicting operations (such as forward/reverse rotations or upper/lower limit positioning) must be configured external to the programmable controller.
  (2) When the programmable controller detects an abnormal condition, it stops the operation and all outputs are:
    - Turned off if the overcurrent or overvoltage protection of the power supply module is activated.
    - Held or turned off according to the parameter setting if the self-diagnostic function of the CPU module detects an error such as a watchdog timer error.
  (3) All outputs may be turned on if an error occurs in a part, such as an I/O control part, where the CPU module cannot detect any error. To ensure safety operation in such a case, provide a safety mechanism or a fail-safe circuit external to the programmable controller. For a fail-safe circuit example, refer to "General Safety Requirements" in the MELSEC iQ-R Module Configuration Manual.
  (4) Outputs may remain on or off due to a failure of a component such as a relay and transistor in an output circuit. Configure an external circuit for monitoring output signals that could cause a serious accident.
- In an output circuit, when a load current exceeding the rated current or an overcurrent caused by a load short-circuit flows for a long time, it may cause smoke and fire. To prevent this, configure an external safety circuit, such as a fuse.
- Configure a circuit so that the programmable controller is turned on first and then the external power supply. If the external power supply is turned on first, an accident may occur due to an incorrect output or malfunction.
- Configure a circuit so that the external power supply is turned off first and then the programmable controller. If the programmable controller is turned off first, an accident may occur due to an incorrect output or malfunction.
- For the operating status of each station after a communication failure, refer to manuals for the network used. For the manuals, please consult your local Mitsubishi representative. Failure to do so may result in an accident due to an incorrect output or malfunction.
- When connecting an external device with a CPU module or intelligent function module to modify data of a running programmable controller, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification, parameter change, forced output, or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Improper operation may damage machines or cause accidents.

[Design Precautions]

⚠WARNING

- Especially, when a remote programmable controller is controlled by an external device, immediate action cannot be taken if a problem occurs in the programmable controller due to a communication failure. To prevent this, configure an interlock circuit in the program, and determine corrective actions to be taken between the external device and CPU module in case of a communication failure.
- Do not write any data to the "system area" and "write-protect area" of the buffer memory in the module. Also, do not use any "use prohibited" signals as an output signal from the CPU module to each module. Doing so may cause malfunction of the programmable controller system. For the "system area", "write-protect area", and the "use prohibited" signals, refer to the user's manual for the module used. For areas used for safety communications, they are protected from being written by users, and thus safety communications failure caused by data writing does not occur.
- If a communication cable is disconnected, the network may be unstable, resulting in a communication failure of multiple stations. Configure an interlock circuit in the program to ensure that the entire system will always operate safely even if communications fail. Incorrect output or malfunction due to a communication failure may result in an accident. When safety communications are used, an interlock by the safety station interlock function protects the system from an incorrect output or malfunction.

[Precautions for using Process CPUs]

- If the redundant system fails, control of the entire system may not be maintained depending on the failure mode. The control may not be maintained in the following case either: An error in an extension base unit or in a module on an extension base unit is detected and causes a stop error of the control system, system switching occurs, and a similar error is detected and causes a stop error of the standby system (new control system). To ensure that the entire system operates safely even in these cases, configure safety circuits external to the programmable controller.

[Precautions for using SIL2 Process CPUs]

- When the programmable controller compliant with SIL2 (IEC 61508) detects a fault in the external power supply or itself, it turns off all outputs in the safety system. Configure an external circuit to ensure that the power source of a hazard is shut off by turning off the outputs. Failure to do so may result in an accident.
- Configure short current protection circuits for safety relays and protection circuits, such as a fuse and breaker, external to the programmable controller.
- When a load current exceeding the rated current or an overcurrent caused by a load short-circuit flows, modules operating in SIL2 mode detect an error and turn off all outputs. Note that if the overcurrent state continues for a long time, it may cause smoke and fire. To prevent this, configure an external safety circuit, such as a fuse.
- When changing data and operating status, and modifying program of the running safety programmable controller from an external device such as a personal computer connected to the SIL2 Process CPU, configure an interlock circuit in the program or external to the programmable controller to ensure that the entire system always operates safely. In addition, before performing online operations, determine corrective actions to be taken between the external device and SIL2 Process CPU in case of a communication failure due to poor contact of cables.
- Do not use any "use prohibited" signals of modules as an I/O signal since they are used by the system. Do not write any data to the "use prohibited" areas in the buffer memory of modules. For the "use prohibited" signals, refer to the user's manual for each module. Do not turn on or off these signals on a program since normal operations cannot be guaranteed. Doing so may cause malfunction of the programmable controller system.

[Design Precautions]

⚠WARNING

- When a module operating in SIL2 mode detects an error in a safety communication path, it turns off outputs. However, the program does not automatically turn off outputs. Create a program that turns off outputs when an error is detected in a safety communication path. If safety communications are restored with outputs on, connected devices may suddenly operate, resulting in an accident.
- Create an interlock circuit which uses reset buttons so that the system does not restart automatically after executing safety functions and turning off outputs.
- In the case of a communication failure in the network, the status of the error station will be as follows:
  (1) Inputs from remote stations are not refreshed.
  (2) All outputs from remote stations are turned off.
  Check the communication status information and configure an interlock circuit in the program to ensure that the entire system will operate safely. Failure to do so may result in an accident due to an incorrect output or malfunction.
- Outputs may remain on or off due to a failure of an output module operating in SIL2 mode. Configure an external circuit for monitoring output signals that could cause a serious accident.

[Precautions for using Safety CPUs]

- When the safety programmable controller detects a fault in the external power supply or itself, it turns off all outputs in the safety system. Configure an external circuit to ensure that the power source of a hazard is shut off by turning off the outputs. Failure to do so may result in an accident.
- Configure short current protection circuits for safety relays and protection circuits, such as a fuse and breaker, external to the safety programmable controller.
- When a load current exceeding the rated current or an overcurrent caused by a load short-circuit flows, the CC-Link IE Field Network remote I/O module (with safety functions) detects an error and turns off all outputs. Note that if the overcurrent state continues for a long time, it may cause smoke and fire. To prevent this, configure an external safety circuit, such as a fuse.
- When changing data and operating status, and modifying program of the running safety programmable controller from an external device such as a personal computer connected to the Safety CPU, configure an interlock circuit in the program or external to the safety programmable controller to ensure that the entire system always operates safely. In addition, before performing online operations, determine corrective actions to be taken between the external device and Safety CPU in case of a communication failure due to poor contact of cables.
- Do not use any "use prohibited" signals as a remote I/O signal since they are used by the system. Do not write any data to the "use prohibited" areas in the remote register. For the "use prohibited" signals, refer to the MELSEC iQ-R CC-Link IE Field Network User's Manual (Application). Do not turn on or off these signals on a program since normal operations cannot be guaranteed. Doing so may cause malfunction of the programmable controller system.
- When the CC-Link IE Field Network remote I/O module (with safety functions) detects a CC-Link IE Field Network error, it turns off outputs. However, the program does not automatically turn off outputs. Create a program that turns off outputs when a CC-Link IE Field Network error is detected. If CC-Link IE Field Network is restored with outputs on, connected machines may suddenly operate, resulting in an accident.

[Design Precautions]

⚠ WARNING

- Ensure that the system does not restart automatically after executing safety functions and turning off outputs. Create a circuit that does not allow the system to restart until an intentional start has been manually issued by an operator.
- In the case of a communication failure in the network, the CPU module recognizes the communication destination station as an error station. The error station will be as follows:
  (1) All safety inputs from the error station to the CPU module are not refreshed.
  (2) All safety outputs from the error station to external devices are turned off because the CPU module does not communicate with the error station.
  Check the communication status information and configure an interlock circuit in the program to ensure that the entire system will operate safely. Failure to do so may result in an accident due to an incorrect output or malfunction.
- Outputs may remain on or off due to a failure of the CC-Link IE Field Network remote I/O module (with safety functions). Configure an external circuit for monitoring output signals that could cause a serious accident.

[Precautions for using redundant function modules]

- The optical transmitter and receiver of the redundant function module use laser diodes (class 1 in accordance with IEC 60825-1/JIS C6802). Do not look directly at a laser beam. Doing so may harm your eyes.

[Design Precautions]

⚠CAUTION

- Do not install the control lines or communication cables together with the main circuit lines or power cables. Doing so may result in malfunction due to electromagnetic interference. Keep a distance of 100mm or more between those cables.
- During control of an inductive load such as a lamp, heater, or solenoid valve, a large current (approximately ten times greater than normal) may flow when the output is turned from off to on. Therefore, use a module that has a sufficient current rating.
- After the CPU module is powered on or is reset, the time taken to enter the RUN status varies depending on the system configuration, parameter settings, and/or program size. Design circuits so that the entire system will always operate safely, regardless of the time.
- Do not power off the programmable controller or reset the CPU module while the settings are being written. Doing so will make the data in the flash ROM and SD memory card undefined. The values need to be set in the buffer memory and written to the flash ROM and SD memory card again. Doing so also may cause malfunction or failure of the module.
- When changing the operating status of the CPU module from external devices (such as the remote RUN/STOP functions), select "Do Not Open by Program" for "Opening Method" of "Module Parameter". If "Open by Program" is selected, an execution of the remote STOP function causes the communication line to close. Consequently, the CPU module cannot reopen the line, and external devices cannot execute the remote RUN function.

[Precautions for using SIL2 Process CPUs]

- Do not install the control lines or communication cables together with the main circuit lines or power cables. Doing so may result in malfunction due to noise. Keep a distance of 100mm or more between those cables.
- When selecting external devices to be connected to modules that operate in SIL2 mode, consider the maximum inrush current described in the user's manual for each module. Exceeding the maximum inrush current may cause malfunction or failure of the module.

[Precautions for using Safety CPUs]

- When selecting external devices to be connected to the CC-Link IE Field Network remote I/O module (with safety functions), consider the maximum inrush current described in the CC-Link IE Field Network Remote I/O Module (With Safety Functions) User's Manual. Exceeding the maximum inrush current may cause malfunction or failure of the module.

[Security Precautions]

⚠WARNING

- To maintain the security (confidentiality, integrity, and availability) of the programmable controller and the system against unauthorized access, denial-of-service (DoS) attacks, computer viruses, and other cyberattacks from external devices via the network, take appropriate measures such as firewalls, virtual private networks (VPNs), and antivirus solutions.

[Installation Precautions]

⚠WARNING

- Shut off the external power supply (all phases) used in the system before mounting or removing the module. Failure to do so may result in electric shock or cause the module to fail or malfunction.

[Installation Precautions]

⚠CAUTION

- Use the programmable controller in an environment that meets the general specifications in the Safety Guidelines (IB-0800525). Failure to do so may result in electric shock, fire, malfunction, or damage to or deterioration of the product.
- To mount a module, place the concave part(s) located at the bottom onto the guide(s) of the base unit, and push in the module until the hook(s) located at the top snaps into place. Incorrect interconnection may cause malfunction, failure, or drop of the module.
- To mount a module with no module fixing hook, place the concave part(s) located at the bottom onto the guide(s) of the base unit, push in the module, and fix it with screw(s). Incorrect interconnection may cause malfunction, failure, or drop of the module.
- When using the programmable controller in an environment of frequent vibrations, fix the module with a screw.
- Tighten the screws within the specified torque range. Undertightening can cause drop of the component or wire, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction. For the specified torque range, refer to the MELSEC iQ-R Module Configuration Manual.
- When using an extension cable, connect it to the extension cable connector of the base unit securely. Check the connection for looseness. Poor contact may cause malfunction.
- When using an SD memory card, fully insert it into the SD memory card slot. Check that it is inserted completely. Poor contact may cause malfunction.
- Securely insert an extended SRAM cassette or a battery-less option cassette into the cassette connector of the CPU module. After insertion, close the cassette cover and check that the cassette is inserted completely. Poor contact may cause malfunction.
- Beware that the module could be very hot while power is on and immediately after power-off.
- Do not directly touch any conductive parts and electronic components of the module, SD memory card, extended SRAM cassette, battery-less option cassette, or connector. Doing so can cause malfunction or failure of the module.

[Precautions for using Safety CPUs]

- Use the CC-Link IE Field Network remote I/O module (with safety functions) and CC-Link IE Field Network remote I/O module in an environment that meets the general specifications in the corresponding manuals (CC-Link IE Field Network Remote I/O Module (With Safety Functions) User's Manual and CC-Link IE Field Network Remote I/O Module User's Manual). Failure to do so may result in electric shock, fire, malfunction, or damage to or deterioration of the product.
- Securely fix the CC-Link IE Field Network remote I/O module (with safety functions) and CC-Link IE Field Network remote I/O module with a DIN rail or module fixing screws. Tighten the screws within the specified torque range. Undertightening can cause drop of the screw, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.

[Wiring Precautions]

⚠ WARNING

- Shut off the external power supply (all phases) used in the system before installation and wiring. Failure to do so may result in electric shock or cause the module to fail or malfunction.
- After installation and wiring, attach a blank cover module (RG60) to each empty slot and an included extension connector protective cover to the unused extension cable connector before powering on the system for operation. Failure to do so may result in electric shock.

[Wiring Precautions]

⚠CAUTION

- Individually ground the FG and LG terminals of the programmable controller with a ground resistance of 100 ohms or less. Failure to do so may result in electric shock or malfunction.
- Use applicable solderless terminals and tighten them within the specified torque range. If any spade solderless terminal is used, it may be disconnected when the terminal screw comes loose, resulting in failure.
- Check the rated voltage and signal layout before wiring to the module, and connect the cables correctly. Connecting a power supply with a different voltage rating or incorrect wiring may cause fire or failure.
- Connectors for external devices must be crimped or pressed with the tool specified by the manufacturer, or must be correctly soldered.
- Securely connect the connector to the module. Poor contact may cause malfunction.
- Do not install the control lines or communication cables together with the main circuit lines or power cables. Doing so may result in malfunction due to noise. Keep a distance of 100mm or more between those cables.
- Place the cables in a duct or clamp them. If not, dangling cables may swing or inadvertently be pulled, resulting in malfunction or damage to modules or cables.
  In addition, the weight of the cables may put stress on modules in an environment of strong vibrations and shocks.
  Do not clamp the extension cables with the jacket stripped. Doing so may change the characteristics of the cables, resulting in malfunction.
- Check the interface type and correctly connect the cable. Incorrect wiring (connecting the cable to an incorrect interface) may cause failure of the module and external device.
- Tighten the terminal screws or connector screws within the specified torque range. Undertightening can cause drop of the screw, short circuit, fire, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
- When disconnecting the cable from the module, do not pull the cable by the cable part. For the cable with connector, hold the connector part of the cable. For the cable connected to the terminal block, loosen the terminal screw. Pulling the cable connected to the module may result in malfunction or damage to the module or cable.

[Wiring Precautions]

⚠ CAUTION

- Prevent foreign matter such as dust or wire chips from entering the module. Such foreign matter can cause a fire, failure, or malfunction.
- When a protective film is attached to the top of the module, remove it before system operation. If not, inadequate heat dissipation of the module may cause a fire, failure, or malfunction.
- Programmable controllers must be installed in control panels. Connect the main power supply to the power supply module in the control panel through a relay terminal block. Wiring and replacement of a power supply module must be performed by qualified maintenance personnel with knowledge of protection against electric shock. For wiring, refer to the MELSEC iQ-R Module Configuration Manual.
- For Ethernet cables to be used in the system, select the ones that meet the specifications in the user's manual for the module used. If not, normal data transmission is not guaranteed.

[Precautions for using redundant function modules]

- For tracking cables to be used in the system, select the ones that meet the specifications in the MELSEC iQ-R CPU Module User's Manual (Startup). If not, normal data transmission is not guaranteed.

[Startup and Maintenance Precautions]

⚠ WARNING

- Do not touch any terminal while power is on. Doing so will cause electric shock or malfunction.
- Correctly connect the battery connector. Do not charge, disassemble, heat, short-circuit, solder, or throw the battery into the fire. Also, do not expose it to liquid or strong shock. Doing so will cause the battery to produce heat, explode, ignite, or leak, resulting in injury and fire.
- Shut off the external power supply (all phases) used in the system before cleaning the module or retightening the terminal screws, connector screws, or module fixing screws. Failure to do so may result in electric shock.

[Startup and Maintenance Precautions]

⚠ CAUTION

- When connecting an external device with a CPU module or intelligent function module to modify data of a running programmable controller, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification, parameter change, forced output, or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Improper operation may damage machines or cause accidents.
- Especially, when a remote programmable controller is controlled by an external device, immediate action cannot be taken if a problem occurs in the programmable controller due to a communication failure. To prevent this, configure an interlock circuit in the program, and determine corrective actions to be taken between the external device and CPU module in case of a communication failure.
- Do not disassemble or modify the modules. Doing so may cause failure, malfunction, injury, or a fire.
- Use any radio communication device such as a cellular phone or PHS (Personal Handy-phone System) 25cm or more away in all directions from the programmable controller.
- Shut off the external power supply (all phases) used in the system before mounting or removing the module. Failure to do so may cause the module to fail or malfunction.
- Tighten the screws within the specified torque range. Undertightening can cause drop of the component or wire, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
- After the first use of the product, do not perform each of the following operations more than 50 times (IEC 61131-2/JIS B 3502 compliant).
  Exceeding the limit may cause malfunction.
  - Mounting/removing the module to/from the base unit
  - Inserting/removing the extended SRAM cassette or battery-less option cassette to/from the CPU module
  - Mounting/removing the terminal block to/from the module
  - Connecting/disconnecting the extension cable to/from the base unit
- After the first use of the product, do not insert/remove the SD memory card to/from the CPU module more than 500 times. Exceeding the limit may cause malfunction.
- Do not touch the metal terminals on the back side of the SD memory card. Doing so may cause malfunction or failure of the module.
- Do not touch the integrated circuits on the circuit board of an extended SRAM cassette or a battery-less option cassette. Doing so may cause malfunction or failure of the module.
- Do not drop or apply shock to the battery to be installed in the module. Doing so may damage the battery, causing the battery fluid to leak inside the battery. If the battery is dropped or any shock is applied to it, dispose of it without using.
- Use a clean and dry cloth to wipe off dirt on the module.

[Startup and Maintenance Precautions]

⚠ CAUTION

- Startup and maintenance of a control panel must be performed by qualified maintenance personnel with knowledge of protection against electric shock. Lock the control panel so that only qualified maintenance personnel can operate it.
- Before handling the module, touch a conducting object such as a grounded metal to discharge the static electricity from the human body. Wearing a grounded antistatic wrist strap is recommended. Failure to discharge the static electricity may cause the module to fail or malfunction.

[Precautions for using SIL2 Process CPUs]

- When performing online operations to the running SIL2 Process CPU such as program modification, device test, and operating status change (for example, from RUN to STOP) of the running programmable controller from an external device such as a personal computer connected, read relevant manuals carefully and ensure the safety before operation. The operations must be performed by qualified operators following the operating procedure that is predetermined at the design stage. Modifying a program while the SIL2 Process CPU is running (the online change) may cause corruption of the program depending on operating conditions. Fully understand the precautions described in the GX Works3 Operating Manual before operation.

[Operating Precautions]

⚠ CAUTION

- When changing data and operating status, and modifying program of the running programmable controller from an external device such as a personal computer connected to an intelligent function module, read relevant manuals carefully and ensure the safety before operation. Incorrect change or modification may cause system malfunction, damage to the machines, or accidents.
- Do not power off the programmable controller or reset the CPU module while the setting values in the buffer memory are being written to the flash ROM in the module. Doing so will make the data in the flash ROM and SD memory card undefined. The values need to be set in the buffer memory and written to the flash ROM and SD memory card again. Doing so also may cause malfunction or failure of the module.

[Computer Connection Precautions]

⚠ CAUTION

- When connecting a personal computer to a module having a USB interface, observe the following precautions as well as the instructions described in the manual for the personal computer used. Failure to do so may cause the module to fail.
  (1) When the personal computer is AC-powered
  When the personal computer has a 3-pin AC plug or an AC plug with a grounding wire, connect the plug to a grounding receptacle or ground the grounding wire. Ground the personal computer and the module with a ground resistance of 100 ohms or less.
  When the personal computer has a 2-pin AC plug without a grounding wire, connect the computer to the module by following the procedure below. For power supplied to the personal computer and the module, using the same power source is recommended.
  1. Unplug the personal computer from the AC receptacle.
  2. Check that the personal computer is unplugged. Then, connect the personal computer to the module with a USB cable.
  3. Plug the personal computer into the AC receptacle.
  (2) When the personal computer is battery-powered
  The personal computer can be connected to the module without taking specific measures.
  For details, refer to the following.
  Cautions When Using Mitsubishi Programmable Controllers or GOTs Connected to a Personal Computer With the RS-232/USB Interface (FA-A-0298)
  When the USB cable used is the GT09-C30USB-5P manufactured by Mitsubishi Electric, specific measures are not required to connect the AC-powered personal computer to the module. However, note that the signal ground (SG) is common for the module and its USB interface. Therefore, if an SG potential difference occurs between the module and the connected devices, it causes failures of the module and the connected devices.

[Disposal Precautions]

⚠ CAUTION

- When disposing of this product, treat it as industrial waste.
- When disposing of batteries, separate them from other wastes according to the local regulations. For details on battery regulations in EU member states, refer to the MELSEC iQ-R Module Configuration Manual.

[Transportation Precautions]

⚠ CAUTION

- When transporting lithium batteries, follow the transportation regulations. For details on the regulated models, refer to the MELSEC iQ-R Module Configuration Manual.
- The halogens (such as fluorine, chlorine, bromine, and iodine), which are contained in a fumigant used for disinfection and pest control of wood packaging materials, may cause failure of the product. Prevent the entry of fumigant residues into the product or consider other methods (such as heat treatment) instead of fumigation. The disinfection and pest control measures must be applied to unprocessed raw wood.

CONDITIONS OF USE FOR THE PRODUCT

(1) MELSEC programmable controller ("the PRODUCT") shall be used in conditions;
i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and
ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.

(2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries.
MITSUBISHI ELECTRIC SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI ELECTRIC USER'S, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.
("Prohibited Application")
Prohibited Applications include, but not limited to, the use of the PRODUCT in;
- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
- Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
- Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.

Notwithstanding the above restrictions, Mitsubishi Electric may in its sole discretion, authorize use of the PRODUCT in one or more of the Prohibited Applications, provided that the usage of the PRODUCT is limited only for the specific applications agreed to by Mitsubishi Electric and provided further that no special quality assurance or fail-safe, redundant or other safety features which exceed the general specifications of the PRODUCTs are required. For details, please contact the Mitsubishi Electric representative in your region.

(3) Mitsubishi Electric shall have no responsibility or liability for any problems involving programmable controller trouble and system trouble caused by DoS attacks, unauthorized access, computer viruses, and other cyberattacks.

- For SIL2 Process CPUs

(1) Although Mitsubishi Electric has declared Product's compliance with the international safety standards IEC61508, IEC61511, this fact does not guarantee that Product will be free from any malfunction or failure. The user of this Product shall comply with any and all applicable safety standard, regulation or law and take appropriate safety measures for the system in which the Product is installed or used and shall take the second or third safety measures other than the Product. Mitsubishi Electric is not liable for damages that could have been prevented by compliance with any applicable safety standard, regulation or law.

(2) Mitsubishi Electric prohibits the use of Products with or in any application involving, and Mitsubishi Electric shall not be liable for a default, a liability for defect warranty, a quality assurance, negligence or other tort and a product liability in these applications.
(a) power plants,
(b) trains, railway systems, airplanes, airline operations, other transportation systems,
(c) hospitals, medical care, dialysis and life support facilities or equipment,
(d) amusement equipments,
(e) incineration and fuel devices,
(f) handling of nuclear or hazardous materials or chemicals,
(g) mining and drilling,
(h) and other applications where the level of risk to human life, health or property are elevated.

(3) Mitsubishi Electric shall have no responsibility or liability for any problems involving programmable controller trouble and system trouble caused by DoS attacks, unauthorized access, computer viruses, and other cyberattacks.

- For Safety CPUs

(1) Although Mitsubishi Electric has obtained the certification for Product's compliance to the international safety standards IEC61508, ISO13849-1 from TUV Rheinland, this fact does not guarantee that Product will be free from any malfunction or failure. The user of this Product shall comply with any and all applicable safety standard, regulation or law and take appropriate safety measures for the system in which the Product is installed or used and shall take the second or third safety measures other than the Product. Mitsubishi Electric is not liable for damages that could have been prevented by compliance with any applicable safety standard, regulation or law.

(2) Mitsubishi Electric prohibits the use of Products with or in any application involving, and Mitsubishi Electric shall not be liable for a default, a liability for defect warranty, a quality assurance, negligence or other tort and a product liability in these applications.

(a) power plants,

(b) trains, railway systems, airplanes, airline operations, other transportation systems,

(c) hospitals, medical care, dialysis and life support facilities or equipment,

(d) amusement equipments,

(e) incineration and fuel devices,

(f) handling of nuclear or hazardous materials or chemicals,

(g) mining and drilling,

(h) and other applications where the level of risk to human life, health or property are elevated.

(3) Mitsubishi Electric shall have no responsibility or liability for any problems involving programmable controller trouble and system trouble caused by DoS attacks, unauthorized access, computer viruses, and other cyberattacks.

INTRODUCTION

Thank you for purchasing the Mitsubishi Electric MELSEC iQ-R series programmable controllers.
This manual describes the memory, functions, devices, and parameters of the relevant products listed below.
Before using this product, please read this manual and the relevant manuals carefully and develop familiarity with the functions and performance of the MELSEC iQ-R series programmable controller to handle the product correctly.
When applying the program examples provided in this manual to an actual system, ensure the applicability and confirm that it will not cause system control problems.
Note that the menu names and operating procedures may differ depending on an operating system in use and its version. When reading this manual, replace the names and procedures with the applicable ones as necessary.
Please make sure that the end users read this manual.

Relevant products

Item	Model
CPU module	R00CPU, R01CPU, R02CPU, R04CPU, R04ENCPU, R08CPU, R08ENCPU, R08PCPU, R08PSFCPU, R08SFCPU, R16CPU, R16ENCPU, R16PCPU, R16PSFCPU, R16SFCPU, R32CPU, R32ENCPU, R32PCPU, R32PSFCPU, R32SFCPU, R120CPU, R120ENCPU, R120PCPU, R120PSFCPU, R120SFCPU
Redundant function module	R6RFM
SIL2 function module	R6PSFM
Safety function module	R6SFM

Point

When using the R00CPU, R01CPU, and R02CPU, replace terms of some LEDs as follows to read this manual.

- PROGRAM RUN LED → P RUN LED
- CARD READY LED → C RDY LED
- CARD ACCESS LED → C ACS LED
- FUNCTION LED → FUNC LED

To users who use this manual

Since the user's manuals "Startup" and "Application" for the CPU module are unified, we offer the user's manual for each CPU module type.

Type	Manual name	Manual number	Replacement timing
Programmable controller CPU	MELSEC iQ-R Programmable Controller CPU Module User's Manual	SH-082488ENG	July 2022
Process CPU	MELSEC iQ-R Process CPU Module User's Manual	SH-082493ENG	July 2022
Safety CPU	MELSEC iQ-R Safety CPU Module User's Manual	—	In the future
SIL2 Process CPU	MELSEC iQ-R SIL2 Process CPU Module User's Manual	—	In the future

For the operation method of CPU Module Logging Configuration Tool, which is included in the user's manual (Application), we offer the operating manual separately.

📖CPU Module Logging Configuration Tool Version 1 Operating Manual

For details on the manual unification, refer to the following technical bulletin.

📖Change of the MELSEC iQ-R Series User's Manual Configuration (FA-A-0378)

CONTENTS

PART 1 CPU MODULE OPERATION

CONTENTS

CHAPTER 30 PRECAUTIONS ON PROGRAMMING 580

CHAPTER 31 MAINTENANCE AND INSPECTION FOR A REDUNDANT SYSTEM 596

PART 5 WHEN USING THE SAFETY CPU

APPENDICES 803

CONTENTS

RELEVANT MANUALS

Manual name [manual number]	Description
MELSEC iQ-R CPU Module User's Manual (Application) [SH-081264ENG] (this manual)	Memory, functions, devices, and parameters of the CPU module
MELSEC iQ-R CPU Module User's Manual (Startup) [SH-081263ENG]	Specifications, procedures before operation, and troubleshooting of the CPU module
MELSEC iQ-R Programmable Controller CPU Module User's Manual [SH-082488ENG]	Procedures before operation, specifications, devices, memory, functions, parameters, and troubleshooting of the programmable controller CPU module
MELSEC iQ-R Process CPU Module User's Manual [SH-082493ENG]	Procedures before operation, specifications, devices, memory, functions, parameters, and troubleshooting of the process CPU module
MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup) [SH-081256ENG]	Specifications, procedures before operation, system configuration, wiring, and communication examples of Ethernet, CC-Link IE Controller Network, and CC-Link IE Field Network
MELSEC iQ-R Ethernet User's Manual (Application) [SH-081257ENG]	Functions, parameter settings, programming, troubleshooting, I/O signals, and buffer memory of Ethernet
MELSEC iQ-R CPU Module Function Block Reference [BCN-P5999-0374]	Specifications of the MELSEC iQ-R series CPU module FBs
MELSEC iQ-R Ethernet, CC-Link IE, and MELSECNET/H Function Block Reference [BCN-P5999-0381]	Specifications of the following MELSEC iQ-R series module FBs: Ethernet-equipped module FBs, CC-Link IE TSN module FBs, CC-Link IE Controller Network module FBs, CC-Link IE Field Network module FBs, and MELSECNET/H network module FBs
MELSEC iQ-R Safety Function Block Reference [BCN-P5999-0815]	Specifications of the safety FBs
MELSEC iQ-R Programming Manual (Program Design) [SH-081265ENG]	Program specifications (ladder, ST, FBD/LD, and SFC programs)
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks) [SH-081266ENG]	Instructions for the CPU module and standard functions/function blocks
MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions) [SH-081749ENG]	Standard process function blocks, tag access function blocks, tag function blocks, and process control instruction designed for the process control
MELSEC iQ-R Programming Manual (Module Dedicated Instructions) [SH-081976ENG]	Dedicated instructions for the intelligent function modules
GX Works3 Operating Manual [SH-081215ENG]	System configuration, parameter settings, and online operations of GX Works3
CPU Module Logging Configuration Tool Version 1 Operating Manual [SH-082478ENG]	System configuration and operation/setting procedures when using CPU Module Logging Configuration Tool

TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
Backup mode	A mode to continue operation in a redundant system by switching the standby system to the control system when an error occurs in the control system.
Buffer memory	Memory in an intelligent function module for storing data such as setting values and monitored values. When integrated into the CPU module, this memory refers to a memory for storing data such as setting values and monitored values of the Ethernet function, and data used for data communication of the multiple CPU system function.
Control CPU	A CPU module that controls connected I/O modules and intelligent function modules. The multiple CPU system allows the user to assign this control to any CPU module on a module-by-module basis.
Control system	A system that controls a redundant system and performs network communications in a redundant system
Control system execution program	A program that is executed only in the CPU module of the control system.
CPU Module Logging Configuration Tool	Software to configure data logging settings and to manage collected data
Dedicated instruction	An instruction that simplifies programming for using functions of intelligent function modules
Device	A memory of a CPU module to store data. Devices such as X, Y, M, D, and others are provided depending on the intended use.
Engineering tool	A tool used for setting up programmable controllers, programming, debugging, and maintenance.
Global label	A label that is valid for all the program data when multiple program data are created in the project. There are two types of global label: a module specific label (module label), which is generated automatically by GX Works3, and an optional label, which can be created for any specified device.
GX LogViewer	Software to display data collected by data logging
Intelligent function module	A module that has functions other than input and output, such as an A/D converter module and D/A converter module
Label	A variable consisting of a specified string used in I/O data or internal processing
Module label	A label that represents one of memory areas (I/O signals and buffer memory areas) specific to each module in a given character string. For the module used, GX Works3 automatically generates this label, which can be used as a global label.
New control system	A system that has switched to control system from standby system after system switching
New standby system	A system that has switched to standby system from control system after system switching
Own system	The system that contains the CPU module that is used for descriptions.
POU	A unit that configures a program. Units are categorized and provided in accordance with functions. A program that is composed of POUs (program organization units) allows the lower-layer processing, when the program is multi-layered, to be divided into several units by processing and function, enabling the creation of programs based on each unit.
Program block	A group of POUs that configure a program
Program executed in both systems	A program that is executed in both CPU modules of the control system and the standby system
Redundant function module	A module that configures a redundant system and is used with a Process CPU (redundant mode) or a SIL2 Process CPU. The redundant function module model name is R6RFM.
Redundant system	A system consisting of two systems that have the same configuration (CPU module, power supply module, network module, and other modules). Even after an error occurs in one of the two systems, the other system takes over the control of the entire system. For details, refer to the descriptions of the redundant system in the following manual. MELSEC iQ-R Module Configuration Manual
Redundant system with redundant extension base unit	A redundant system that is configured using extension base unit(s)
Safety CPU	A module that performs both standard control and safety control and is used with a safety function module. The Safety CPU models include the R08SFCPU, R16SFCPU, R32SFCPU, and R120SFCPU.
Safety function module	A module that performs safety control and must be used with a Safety CPU. This module can only be used with the Safety CPU. The safety function module model name is R6SFM.
Separate mode	A mode for system maintenance in a redundant system. This mode can maintain a redundant system without stopping control while the system is running.
Signal flow	The execution status that the last time an operation of a program or an FB is executed in each step.
SIL2 function module	A module that performs safety control and must be used with a SIL2 Process CPU. This module can only be used with the SIL2 Process CPU. The SIL2 function module model name is R6PSFM.
SIL2 mode	An operation mode of the I/O module and the intelligent function module to perform safety I/O at the SIL2 level. For details on the SIL2 mode, refer to the following. Manual for the I/O module or intelligent function module used

Term	Description
SIL2 Process CPU	A module that performs both standard control and safety control and is used with a SIL2 function module. This module is also used with a redundant function module and configures a redundant system. The SIL2 Process CPU models include the R08PSFCPU, R16PSFCPU, R32PSFCPU, and R120PSFCPU.
Standard/safety shared label	A label that can be used in both standard programs and safety programs. This label is used at data communications between a safety program and standard program.
Standby system	A backup system in a redundant system
System A	A system that is set as system A to distinguish two systems, which are connected with two tracking cables. When the two systems start up at the same time, this system will be a control system. System switching does not affect the system A/B setting.
System B	A system that is set as system B to distinguish two systems, which are connected with two tracking cables. When the two systems start up at the same time, this system will be a standby system. System switching does not affect the system A/B setting.
The other system	The other system that has been connected to own system with tracking cables. When the system A is the own system, the system B is the other system, and vice versa.
Tracking cable	An optical fiber cable used to connect two redundant function modules in a redundant system

The following terms are used to explain systems using the SIL2 Process CPU and the Safety CPU.

Term	Description
Pair version	Version information that determines combination of the SIL2 Process CPU and SIL2 function module, and the Safety CPU and safety function module
Safety communications	Communication service that processes the send/receive of network layers for the safety defined in the safety predefined protocol
Safety control	Machine control by safety programs and safety data communications. When an error occurs, the machine in operation is securely stopped.
Safety cycle processing	A safety cycle processing is a execution processing for safety programs and safety input/output.
Safety cycle time	A interrupt time for executing a safety I/O and safety program
Safety device	A device that can be used in safety programs
Safety program	A program for performing safety control
Standard communications	Communications other than safety communications, such as cyclic transmission and transient transmission of CC-Link IE Field Network
Standard control	Machine control by standard programs and standard data communications. Modules other than the safety programmable controller hold only standard control. (This term is used to distinguish from safety control.)
Standard device	A device (X, Y, M, D, or others) in a CPU module. (Safety devices are excluded.) This device can be used only in standard programs. (This term is used to distinguish from a safety device.)
Standard program	A program that performs sequence control. (Safety programs are excluded.) (This term is used to distinguish from a safety program.)

GENERIC TERMS AND ABBREVIATIONS

Unless otherwise specified, this manual uses the following generic terms and abbreviations.

Generic term/abbreviation	Description
CC-Link IE Controller Network-equipped module	An RJ71GP21-SX CC-Link IE Controller Network module, an RJ71GP21S-SX CC-Link IE Controller Network module, and the following modules when the CC-Link IE Controller Network function is used: • RJ71EN71 • RnENCPU
CC-Link IE Field Network-equipped master/local module	An RJ71GF11-T2 CC-Link IE Field Network master/local module and the following modules when the CC-Link IE Field Network function is used: • RJ71EN71 • RnENCPU
CC-Link IE TSN master/local module	RJ71GN11-T2, RJ71GN11-EIP (CC-Link IE TSN part)
Device supporting iQSS	A device which supports iQ Sensor Solution. For iQ Sensor Solution, refer to the following. iQ Sensor Solution Reference Manual
Ethernet interface module with built-in CC-Link IE	RJ71EN71
Ethernet-equipped module	The following modules when the Ethernet function is used: • RJ71EN71 • CPU module
I/O module	An input module, an output module, an I/O combined module, and an interrupt module
Motion module	RD78G4, RD78G8, RD78G16, RD78G32, RD78G64, RD78GHV, RD78GHW
Network module	Includes the following: • Ethernet interface module • CC-Link IE TSN master/local module • CC-Link IE Controller Network module • CC-Link IE Field Network master/local module • MELSECNET/H network module • MELSECNET/10 network module • RnENCPU (network part)
Process CPU	R08PCPU, R16PCPU, R32PCPU, R120PCPU
Process CPU (process mode)	A Process CPU operating in process mode. Process control function blocks and the online module change function can be executed.
Process CPU (redundant mode)	A Process CPU operating in redundant mode. A redundant system is configured with this CPU module. Process control function blocks and the online module change function can be used even in this mode.
Programmable controller CPU	R00CPU, R01CPU, R02CPU, R04CPU, R04ENCPU, R08CPU, R08ENCPU, R16CPU, R16ENCPU, R32CPU, R32ENCPU, R120CPU, R120ENCPU
RAS	Reliability, Availability, Serviceability. This term refers to the overall usability of automated equipment.
Redundant extension base unit	An extension base unit which is essential for configuring a redundant system with redundant extension base unit
Remote head module	An RJ72GF15-T2 CC-Link IE Field Network remote head module
RJ71GN11-EIP (CC-Link IE TSN part)	An RJ71GN11-EIP when it performs communications on CC-Link IE TSN
RJ71GN11-EIP (EtherNet/IP part)	An RJ71GN11-EIP when it performs communications on EtherNet/IP
RnCPU	R00CPU, R01CPU, R02CPU, R04CPU, R08CPU, R16CPU, R32CPU, R120CPU
RnENCPU	R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, R120ENCPU
RnENCPU (CPU part)	A module on the left-hand side of the RnENCPU (MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup))
RnENCPU (network part)	A module on the right-hand side of the RnENCPU (MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup))
Station sub ID number	An ID number of a sensor connected to a CC-Link-compatible communication module
Windows® 7 or later	Windows®7, Windows®8, Windows®8.1, Windows®10

The following generic terms and abbreviations are used to explain systems using the SIL2 Process CPU and the Safety CPU.

Generic term and abbreviation	Description
Safety programmable controller	A MELSEC iQ-R series module that performs safety control: a Safety CPU, a safety function module, a CC-Link IE Field Network remote I/O module (with safety functions)
Standard CPU	A MELSEC iQ-R series CPU module that performs standard control (This term is used to distinguish from CPU modules that perform safety control.)
Standard programmable controller	A MELSEC iQ-R series module that performs standard control (This term is used to distinguish a programmable controller from a safety programmable controller.)

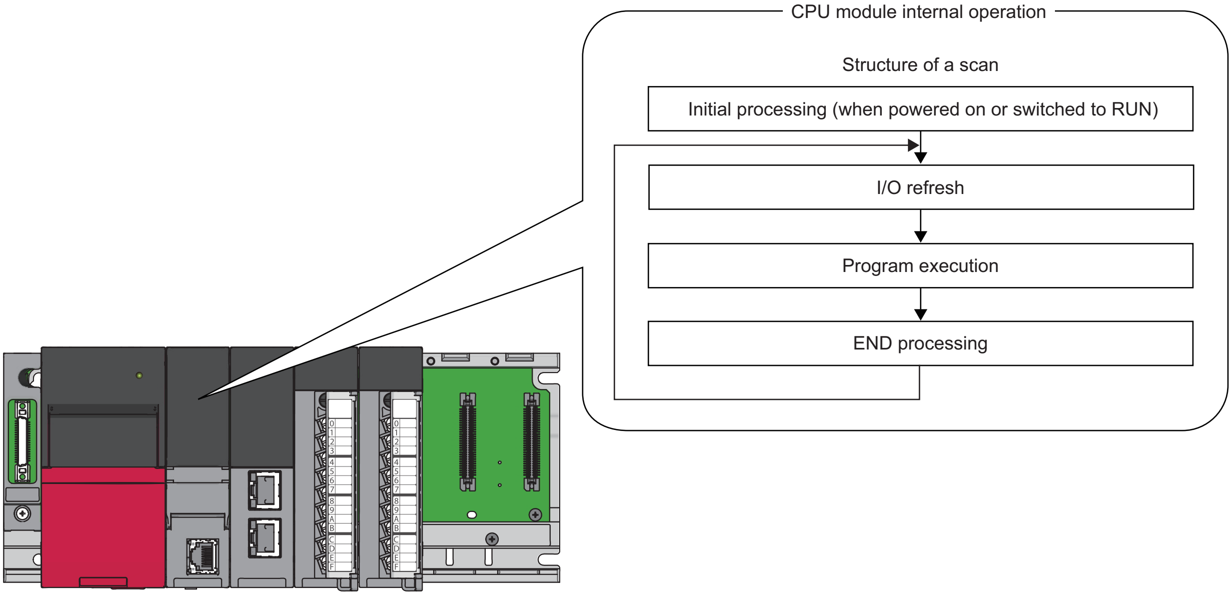
PART 1 CPU MODULE OPERATION

This part consists of the following chapters.

1 RUNNING A PROGRAM

1.1 Scan Configuration

The following shows the scan configuration of the CPU module.



Initial processing (when powered on or switched to RUN)

For the initial processing (when powered on or switched to RUN), the following processes are performed:

○: Performed, ×: Not performed

Item	Initial processing	Initial processing (when switched to RUN)
Booting from an SD memory card	○	×
Checking each parameter and program*1	○	○
Checking the consistency of parameters for the multiple CPU system configuration	○	○
Assigning the I/O number to the mounted module	○	○
Initializing and setting the information of each module	○	×
Initializing a device/label outside the latch range (Bit device: Off, Others: 0)	○	×
Setting the device/initial label value*2	○	○

*1 Checking each parameter and program takes time depending on the parameter setting and the number of programs, and thus the CPU module may take time to respond to the peripheral.

*2 For details on the initial label value setting for the Process CPU and SIL2 Process CPU, refer to the following.

☞ Page 364 LABEL INITIALIZATION FUNCTION

When functions that require an SD memory card are enabled and the status of the SD memory card is as follows, the initial processing (when powered on or switched to RUN) takes time, and thus the CPU module may take time to respond to the peripheral.

- A large number of files is stored in the SD memory card.
- The SD memory card is close to the end of life or it has reached the end of life.

I/O refresh

The module performs the following before starting program operation.

- ON/OFF data input from the input module/intelligent function module to the CPU module.
- ON/OFF data output from the CPU module to the output module/intelligent function module.

Point

While constant scan is in progress, I/O refresh is performed after the waiting time for constant scan expires.

Program operation

According to the program settings, the module executes from step 0 through the END/FEND instruction for each program. This program is referred to as a main routine program. A main routine program can be divided into subroutine programs. (☞ Page 73 Subroutine Program)

END processing

The CPU module performs the following processing.

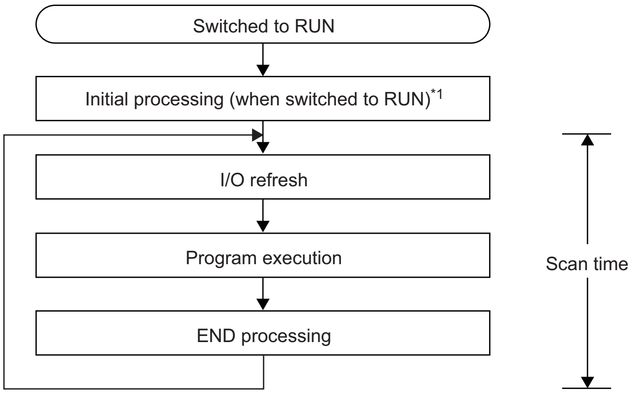
- Network module link refresh
- Link refresh of CC-Link IE Field Network Basic
- Intelligent function module refresh
- Instruction end processing (including dedicated instruction for the module)
- Device latch processing
- The device latch processing is a kind of service processing such as read and write of devices, labels, and program access files (☞ Page 1095 Target List and Operation Details of the Device/Label Access Service Processing Setting).
- Watchdog timer reset (☞ Page 136 Watchdog timer reset)
- Refresh between CPU modules (for the multiple CPU system configuration)
- Data collection by the data logging function (when the collection interval is set to "Each scanning cycle" or "Sample data at the next END processing after the specified time has elapsed")
- Sequence scan synchronization sampling function
- Recording function (when the sampling method is set to "Each scanning cycle" or "Time specification")
- Self-diagnostics processing
- Sets a value to the special relay/special register (for those with the set timing specified as the END processing timing)

Point

Even during the END processing, an interrupt program, fixed scan execution type program, or event execution type program (when the trigger type is set to at interrupt occurrence) is executed. To prevent the execution of an interrupt program during the END processing, disable an interrupt by the DI instruction immediately before the END processing, and enable an interrupt by the EI instruction at the head of the interrupt program.

1.2 Scan Time

The CPU module repeats the following processing. The scan time is the sum of the following processing and execution time.



*1 The initial scan time includes this processing.

Initial scan time

The first scan time after the CPU module becomes in the RUN state.

How to check the initial scan time

See below.

- Values stored in SD518 (Initial Scan Time) and SD519 (Initial Scan Time)
- Program List Monitor (GX Works3 Operating Manual)

Initial scan time monitoring

Monitoring is performed with the initial scan time monitoring time. (Page 136 Scan time monitoring time setting)

■Precautions on the initial scan time monitoring time

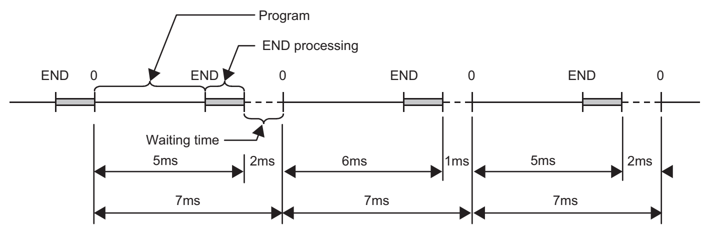
- Set the initial scan time monitoring time longer than the execution time of the initial scan time.

Constant scan

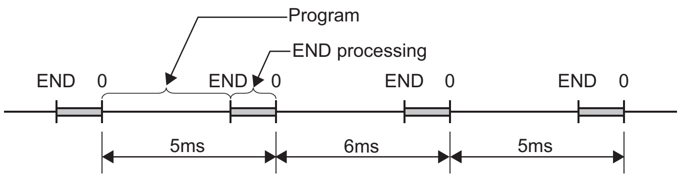
RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

Scan time is different for each scan because its processing time varies depending on whether instructions used in a program are executed or not. By setting constant scan, the I/O refresh interval can be kept constant even when the program execution time varies because the program can be executed repeatedly by keeping the scan time constant.

- When constant scan is set (Setting value = 7ms)



- When constant scan is not set

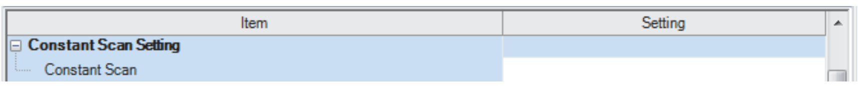


Setting constant scan

Constant scan can be set.

[CPU Parameter] ⇨ [RAS Setting] ⇨ [Constant Scan Setting]

Window



Displayed items

Item	Description	Setting range	Default
Constant Scan	Sets the constant scan time.	• R00CPU, R01CPU, R02CPU: 0.5 to 2000ms (in increments of 0.1ms) • Other CPU modules: 0.2 to 2000ms (in increments of 0.1ms)	—

Point

The setting time for the constant scan must be a value that satisfies the relational expression below.

"Watchdog timer setting time" > "Constant scan setting time" > "Maximum scan time of the program"

If the maximum scan time of the program is longer than the constant scan setting time, it is assumed as an error to ignore the constant scan, and the scan time of the program is applied.

Accuracy of constant scan

The accuracy of the constant scan is 0.01ms. However, if processing, which should be executed during the waiting time from the completion of the END processing to the start of the next scan, is being executed, the constant scan cannot finish even if the constant scan time is reached. The constant scan may increase by the program execution time of the interrupt factor.
(☞ Page 44 Precautions)

Precautions

Processing of the program is stopped during the waiting time from the time the END processing is completed for the program until the next scan starts. If any of the following processing requests is received during the waiting time, the corresponding processing is executed.

- Interrupt program
- Fixed scan execution type program
- Event execution type program which uses occurrence of an interrupt as a trigger
- Device/label access service processing

Device/label access service processing setting

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using a SIL2 Process CPU, refer to the following as well.

☞ Page 704 FUNCTIONS

- When using a Safety CPU, refer to the following as well.

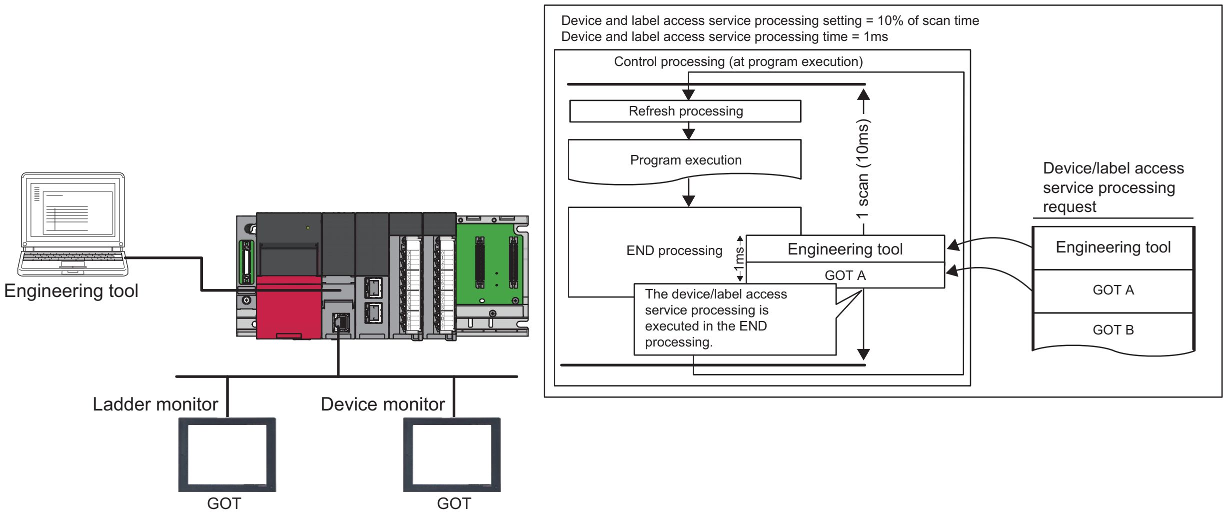
☞ Page 621 FUNCTIONS

The user can specify the time or the execution timing of the device/label access service processing which is performed during the END processing.

A request to the CPU module from a peripheral is processed by the device/label access service processing. A communication response to a request from a peripheral varies depending on the scan time and the state of communication load. To create the service processing environment suitable for the system, change the device/label access service processing setting as shown below.

- Set a longer time for the device/label access service processing to improve a communication response to a peripheral.
- Set a shorter time for the device/label access service processing to reduce a scan time extension due to service processing.

Note that communications to the CPU module from multiple peripherals may lower a communication response to each peripheral. Thus, consider the performance of communication responses and the extension of scan time, and adjust the service processing environment according to the system by methods such as setting a longer time for the device/label access service processing or changing the parameter settings of peripherals so that the timeout time becomes longer.



Point

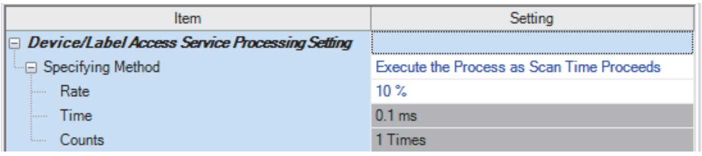
For functions to which the device/label access service processing setting can be applied, refer to the list targeted for device/label access service processing setting. (☞ Page 1095 Target list)

Setting method

The device/label access service processing can be configured as follows.

[CPU Parameter] ⇨ [Service Processing Setting] ⇨ [Device/Label Access Service Processing Setting]

Window



Displayed items

Item	Description	Setting range	Default
Specifying Method	Selects a method for specifying the service processing for access to devices/labels.	• Execute the Process as Scan Time Proceeds • Set Processing Time • Set Processing Counts • Execute END Processing between Programs	Execute the Process as Scan Time Proceeds
Rate	Sets the ratio of the device/label access service processing to execute during a single scan when "Execute the Process as Scan Time Proceeds" is enabled. The device/label access service processing time is determined based on the scan time.	1 to 99% (unit: 1%)	10%
Time	Sets the device/label access service processing time per scan when "Set Processing Time" is enabled.	0.1 to 1000ms (unit: 0.1ms)	0.1ms
Counts	Sets the number of executions of the device/label access service processing per scan when "Set Processing Counts" is enabled.	1 to 10 times (unit: 1 time)	1 time

Operations enabled by setting details

Operations enabled by setting details of the device/label access service processing setting are as follows.

Item	Scan performance		Device/label access service process performance		Inter-program monitoring*5	Application
	Increase*1	Stability*2	Response time*3	Stability*4		
Execute the Process as Scan Time Proceeds	Medium	Medium	Medium	Medium	None	This setting is useful to execute the device/label access service processing in a way commensurate with the system size. It allows the system to be designed without considering the device/label access service processing time because it is determined as a function of the scan time. Multiple requests are processed until the specified ratio is exceeded. If exceeded, the remaining requests are processed in the END processing of the next scan. When no request data is received in the current scan, the scan time is shortened by the specified ratio as the CPU module proceeds to the next scan without waiting for requests.
Set Processing Time	Medium	High	Medium	High	None	This setting is useful to give priority to the device/label access service processing. It allows for stable communication because the CPU module can always process a constant amount of the device/label access service processing without affecting the scan time. Multiple requests are processed until the specified processing time is exceeded. If exceeded, the remaining requests are processed in the END processing of the next scan. When no request data is received in the current scan, the scan time is shortened by the specified processing time as the CPU module proceeds to the next scan without waiting for requests.
Set Processing Counts	Large	Low	Medium	High	None	This setting is useful to stably execute the device/label access service processing in a system where requests come from multiple peripherals. It provides stable communication in a system where multiple peripherals exist because the CPU module can execute the device/label access service processing based on the number of request sources. Multiple requests are processed until the specified number of executions of the service processing is reached. When no request data is received in the current scan, the CPU module proceeds to the next scan without waiting for requests.
Execute END Processing between Programs	Large	Medium	Quick	High	Yes	This setting is useful to give priority to the device/label access service processing in a system with a large number of programs. It improves the communication response as it enables the device/label access service processing to be executed multiple times during a single scan. When no request data is received between program executions and/or during the END processing, the CPU module proceeds to the next program or scan without waiting for requests.

*1 This term indicates how long the scan time will be extended at its maximum due to the device/label access service processing.
*2 This term indicates how much the scan time will be varied due to the device/label access service processing.
*3 This term indicates how long the response time will take after a request for the device/label access service processing is received from an engineering tool or the like.
*4 This term indicates how much the response time will be varied depending on the contents of requests for the device/label access service processing from engineering tools or the like.
*5 When this function is enabled, monitoring is performed between execution of programs, and therefore values of ongoing operation may be read out. (☞ Page 49 When "Execute END Processing between Programs" is enabled)

For what operation is enabled by setting each item, refer to the following.

☞ Page 1097 Operation details

Precautions

This section describes the precautions on the device/label access service processing setting.

■Functions that may prolong the scan time

For the following functions, the scan time may become longer than the specified time during processing even when this setting is applied.

- Online change
- Device/buffer memory batch monitor
- Real-time monitor function
- Data logging function
- Sequence scan synchronization sampling function
- Recording function
- File register write or read*1

*1 The scan time becomes longer when the size of data to write or read is large.

■When constant scan is set

The CPU module always processes one request during the END processing, and processes the remaining requests during the wait time for the next constant scan. Set constant scan setting time considering the time required for the device/label access service processing to execute.

■When "Execute END Processing between Programs" is enabled

When "Execute END Processing between Programs" is enabled, the device/label access service processing, such as device access, is performed between program executions and during the END processing. Therefore, when monitoring and current value change are performed in the situation where a device value is processed across programs, values of ongoing operation may be read or written.[*1]

Also, when "Execute END Processing between Programs" is enabled, if monitoring and current value change are performed for the file register where "Use File Register of Each Program" is enabled, unintended data may be obtained or written. In this case, the value of the file register may not fall into the range depending on whether the file register is used for each program and the size of the file register.[*2]

The target timing of monitoring or current value change can be specified by using the specified program monitor. (☞ Page 176 Specified Program Monitor)

*1 Because the timing of monitoring or current value change (during execution of a program or the END processing) cannot be specified, values of ongoing operation may be read or written.

*2 Because monitoring is performed between programs and the operation is executed during the END processing, a file register to be registered and whose current value is to be changed cannot be specified.

Device/label access service processing constant wait function

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 492 FUNCTIONS

This function improves the communication response of device/label access service processing requests.
Based on SM315 (Service processing constant wait setting flag) and SD315 (Service processing constant wait status setting), device/label access service processing requests are accepted until the time or ratio set for the device/label access service processing setting of the CPU parameters is reached.
Even if there is no request during device/label access service processing, the function waits until the set time is reached to allow any request accepted while waiting to be processed in the same scan and its response to be returned, thereby improving the communication response for device/label access service processing requests.

Point

The response becomes faster because service processing requests are accepted until the time set for the device/label access service processing setting is reached. Regardless of whether there is any service processing request from a peripheral, the scan time is extended by the period of time set for the device/label access service processing setting. Therefore, check that the extension of the scan time does not affect control before using this function.

Restriction

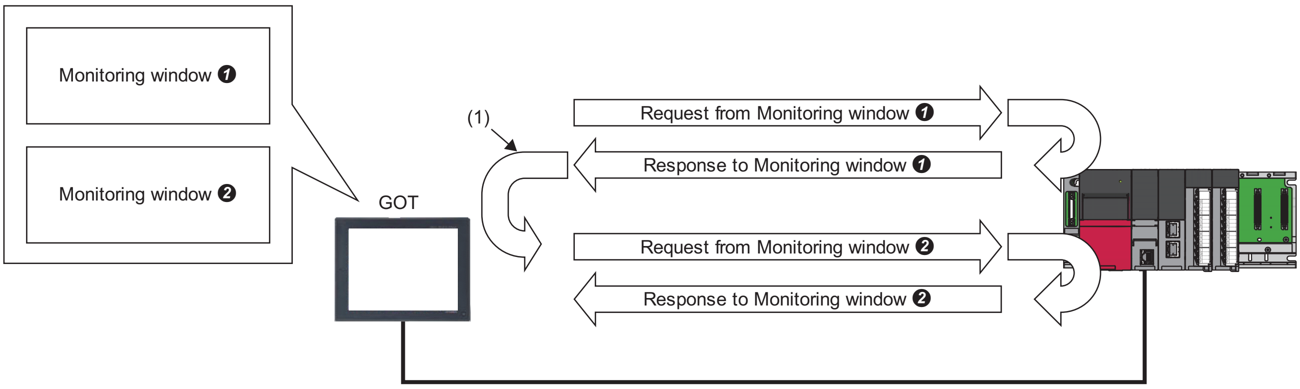
Before using the device/label access service processing constant wait function, check the versions of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

Operation of the device/label access service processing

This section describes the operation of the device/label access service processing.

Ex.

When updating multiple monitor windows on the GOT



(1) From the time the CPU module sends a response until the GOT sends the next request, the CPU module will not receive requests from the GOT.

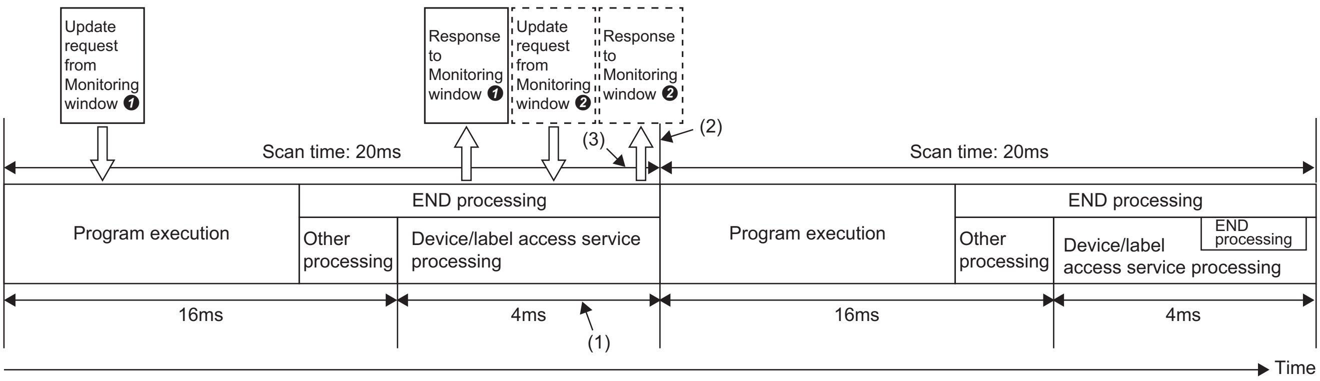
Depending on the GOT, when updating multiple monitor windows, the GOT repeats the action of sending a request (a device/label read request) to the CPU module for each window and waiting for a request (a device/label current value).

As a result, when the windows are updated in the order of monitor window ❶→monitor window❷, from the time the CPU module sends a response for monitor window ❶ until it receives a request for monitor window ❷ that follows, the CPU module has no request to be processed.

When in this state, use this setting to specify whether to process a request for monitor window ❷ in the next scan or to wait until the set time is reached before processing in the same scan.

■Operation when this function is enabled

The following figure shows the operation when SM315 (Service processing constant wait setting flag) and SD315 (Service processing constant wait status setting) are enabled. (Device/label access service processing setting: "Set Processing Time": 4ms)



(1) Update requests from multiple monitor windows can be processed in one scan.

(2) Requests are accepted until the time set for the device/label access service processing setting is reached.

(3) The communication response is improved because requests are accepted within the device/label access service processing time.

Even after the CPU module returns a response to the GOT, subsequent requests are accepted until the time or ratio specified for the device/label access service processing setting is reached, thereby enabling update requests from multiple monitor windows to be processed in one scan. This improves the communication response.

Setting method

To enable this setting, set the special relay and special register as follows.

1. Check that the device/label access service processing setting is set to "Execute the Process as Scan Time Proceeds" or "Set Processing Time". (☞ Page 46 Setting method)
2. Set "AFFFH" for SD315 (Service processing constant wait status setting).
3. Turn off and on SM315 (Service processing constant wait setting flag).

Point

Because SM315 or SD315 is not latched, set it again after turning off and on the power or resetting.

Precautions

- This function is enabled only when "Execute the Process as Scan Time Proceeds" or "Set Processing Time" for the device/label access service processing setting is set.
- When the constant scan is set, the operation will be that of when the constant scan is set regardless of the device/label access service processing setting and the setting for the device/label access service processing constant wait function. (The CPU module always processes one request during the END processing, and processes the remaining requests during the waiting time for the next constant scan.)

1.3 Data Communication and I/O Processing

Data communication

In data communication, data such as I/O signals, buffer memory, and link device of the CPU module and intelligent function module are communicated.

There are two modes for data communication: refresh mode which automatically sends/receives the module data into the device or label of the CPU module at END processing and direct mode which accesses when an instruction is executed in a program.

When data communication is conducted for the module where a major error occurs, the following operation is performed:

- When access by the contact instruction is executed for the controlled module, non-running, rather than an error, is returned. When direct access by other instructions is executed, "Major error in module" (error code: 2450) occurs.
- No error occurs when direct access for the non-controlled module is executed.

I/O processing and response delay

The CPU module performs the I/O processing in the refresh mode. Using the direct access input/output in a program, however, allows the CPU module to perform I/O processing in the direct mode at the time of each instruction execution. This section describes these I/O processing modes of the CPU module and response delays.

Refresh mode

The CPU module performs I/O processing collectively at a specified timing. The following table lists the refresh timing which can be specified. (☞ Page 54 Refresh mode)

Refresh timing	Description
When END processing is performed	The refresh processing is performed in every END processing[*1].
At the execution time of specified program	The refresh processing is performed before and after specified program execution. (☞ Page 72 Group setting for refresh)

*1 There are modules which performs the refresh processing at a specified END processing rather than at every END processing. (📖 User's manual for the module used)

Direct mode

The CPU module performs I/O processing when each instruction is executed in a program. To access input/output modules in the direct mode, use the direct access input or direct access output in a program. (☞ Page 56 Direct mode)

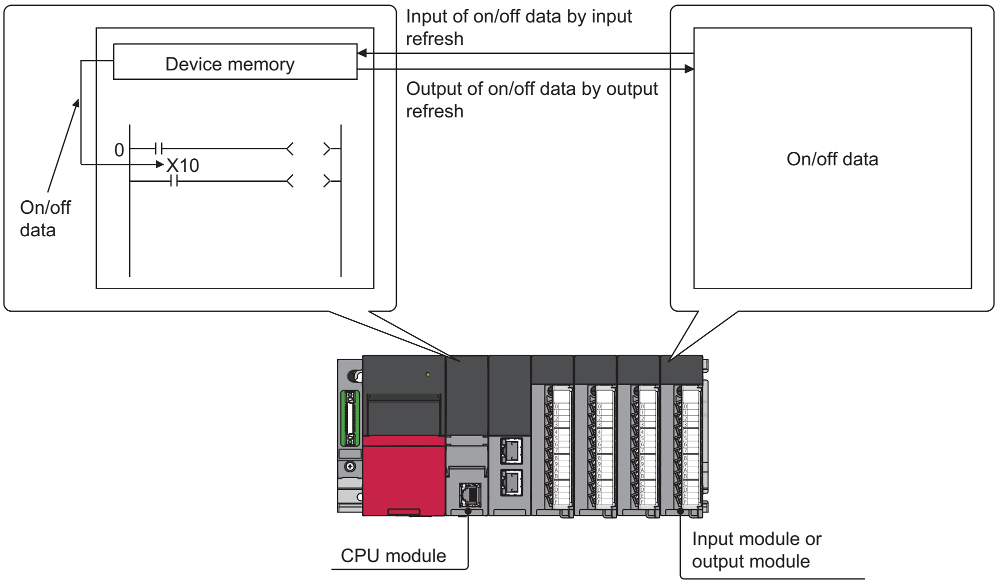
Differences between refresh mode and direct mode

The direct mode directly accesses I/O modules at execution of an instruction. Therefore, data is input faster than when it is input in refresh mode. Processing time required for each instruction, however, takes longer. The following table lists the availability of the refresh mode and the direct mode for each input and output.

Item	Refresh mode	Direct mode
Input/output modules	Enabled	Enabled
Input/output of intelligent function modules		
Remote input/output in CC-Link IE Controller Network, CC-Link IE Field Network, CC-Link IE Field Network Basic, MELSECNET/H, or CC-Link		Disabled

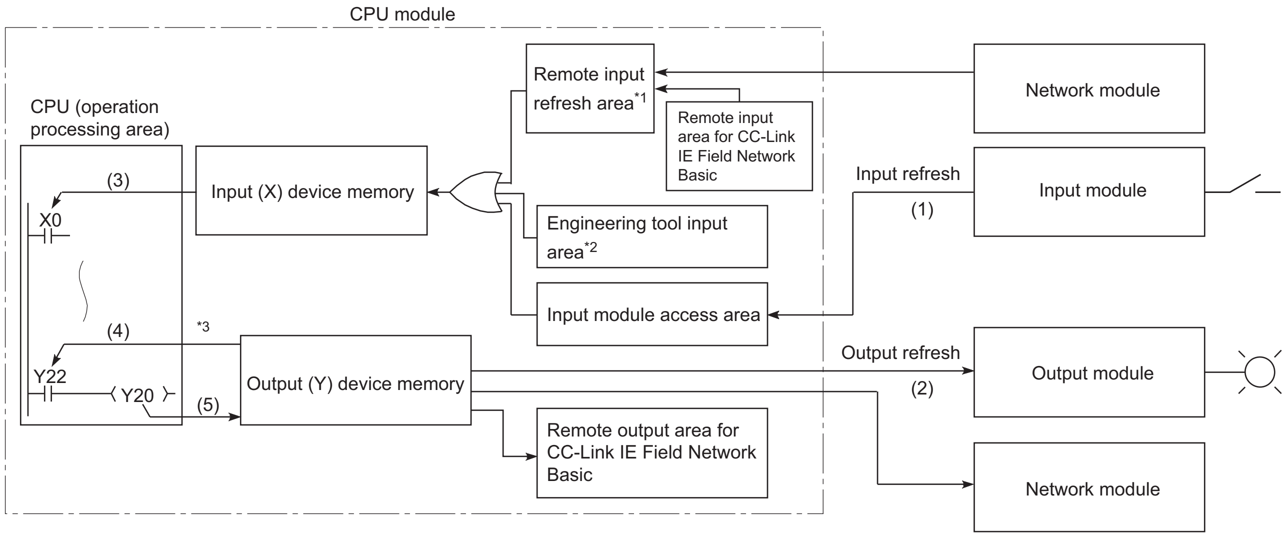
Refresh mode

The CPU module performs I/O processing collectively at a specified timing.
The timing of the input refresh and output refresh follows the specified refresh timing setting.



Outline of the processing

The following describes the details of the refresh mode.



*1 The remote input refresh area indicates the area to be used when refresh is set to the input (X) in CC-Link IE Controller Network, CC-Link IE Field Network, CC-Link IE Field Network Basic, MELSECNET/H, or CC-Link.
*2 Data in the engineering tool input area can be turned on or off by the following:
· Test operation of an engineering tool
· Writing data from the network module
· Writing data from an external device using SLMP
*3 Data in the output (Y) device memory can be turned on or off by the following:
· Test operation of an engineering tool
· Writing data from the network module
· Writing data from an external device using SLMP

Item	Description
Input refresh	At a specified timing (at END instruction execution or before specified program execution), input data are collectively read out from the input modules (1), the OR processing with the engineering tool input area and remote input refresh area is executed, and then the data are stored in the input (X) device memory.
Output refresh	At a specified timing (at END instruction execution or before specified program execution), data in the output (Y) device memory (2) are collectively output to the output module.

Item	Description
Execution of an input contact instruction	Input data in the input (X) device memory (3) are read out and the program is executed.
Execution of an output contact instruction	Output data in the output (Y) device memory (4) are read out and the program is executed.
Execution of the OUT instruction	The operation result of the program (5) are stored to the output (Y) device memory.

■Input

On/off data of an input module are batch-input to the area for communication with the input module in the CPU module at a specified timing (at END instruction execution or before specified program execution). The CPU module performs program operations using the on/off data stored in the input (X) device memory.

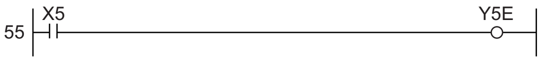
■Output

The operation results of the program is output to the output (Y) device memory in the CPU module every time program operation is performed. Then, the CPU module batch-outputs the on/off data in the output (Y) device memory to an output module at a specified timing (at END instruction execution or before specified program execution).

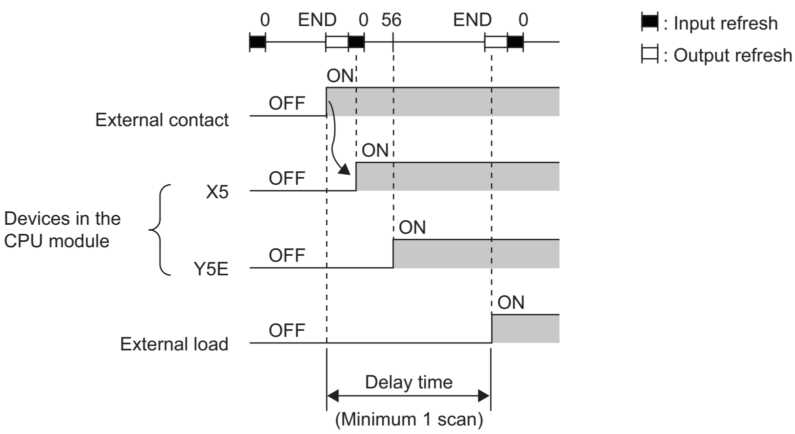
Response delay

An output response which corresponds to the status change in the input module delays for two scans (maximum) depending on the on timing of an external contact.

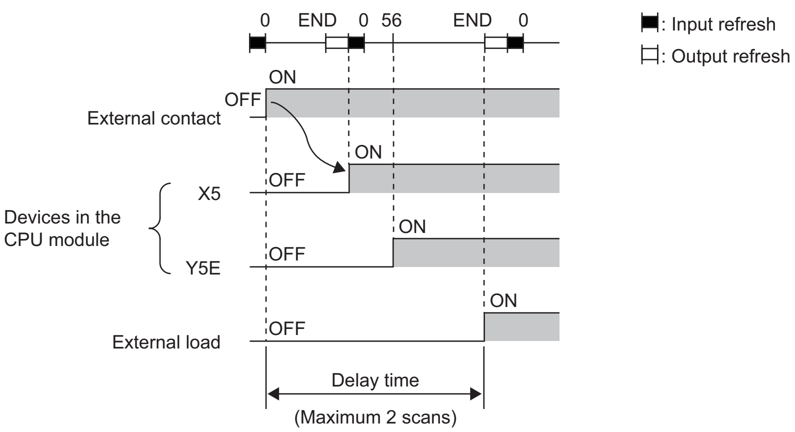
[Example] A program that turns on the output Y5E when the input X5 turns on



- Y5E turns on the earliest

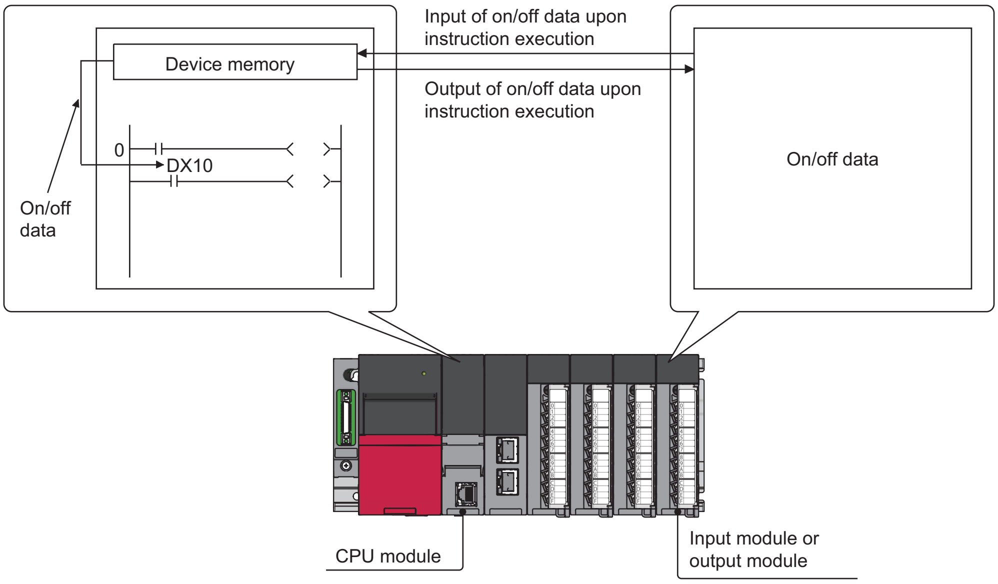


- Y5E turns on the latest



Direct mode

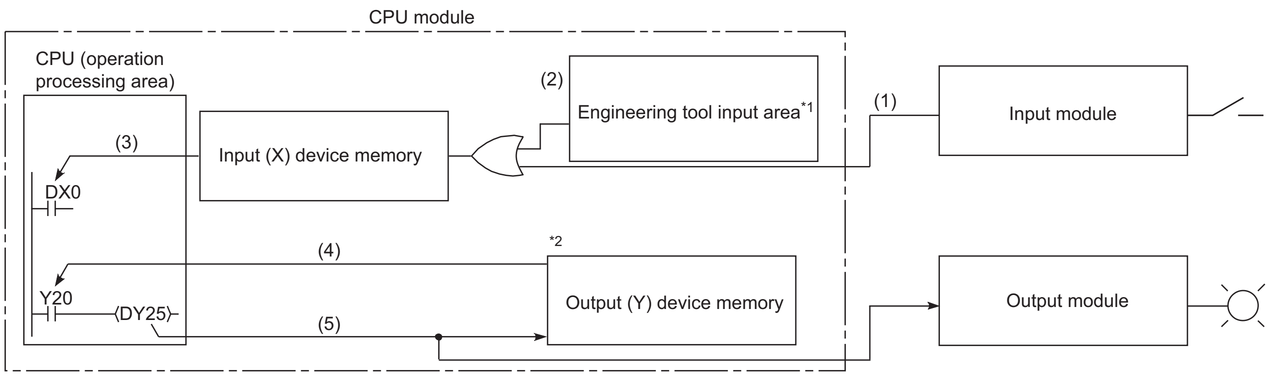
The CPU module performs I/O processing when each instruction is executed in a program.



With this mode, the CPU module uses the direct access input (DX) and direct access output (DY) to perform I/O processing.

Outline of the processing

The following describes the details of the direct mode.



*1 Data in the engineering tool input area can be turned on or off by the following:
- · Test operation of an engineering tool
- · Writing data from the network module
- · Writing data from an external device using SLMP

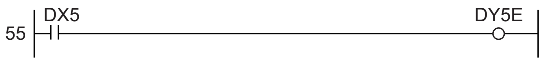
*2 Data in the output (Y) device memory can be turned on or off by the following:
- · Test operation of an engineering tool
- · Writing data from the network module
- · Writing data from an external device using SLMP

Item	Description
Execution of an input contact instruction	The OR processing is executed with the input information of the input module (1) and the input data of the engineering tool input area (2) or remote input refresh area. The result is stored in the input (X) device memory and is used as input data (3) to execute the program.
Execution of an output contact instruction	Output data in the output (Y) device memory (4) are read out and the program is executed.
Execution of the OUT instruction	The operation result of the program (5) are output to the output module, and stored in the output (Y) device memory.

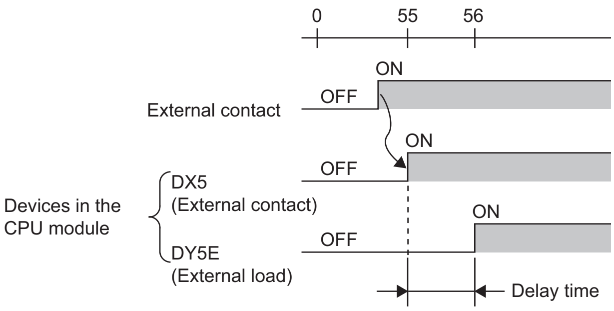
Response delay

An output response which corresponds to the status change in the input module delays for one scan (maximum) depending on the on timing of an external contact.

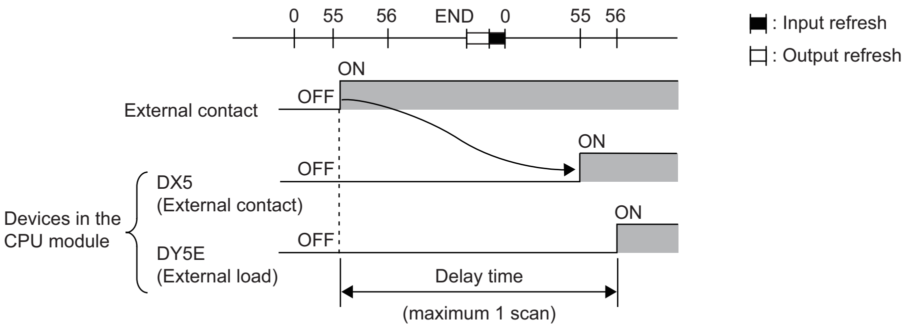
[Example] A program that turns on the output DY5E when the input DX5 turns on



- Y5E turns on the earliest

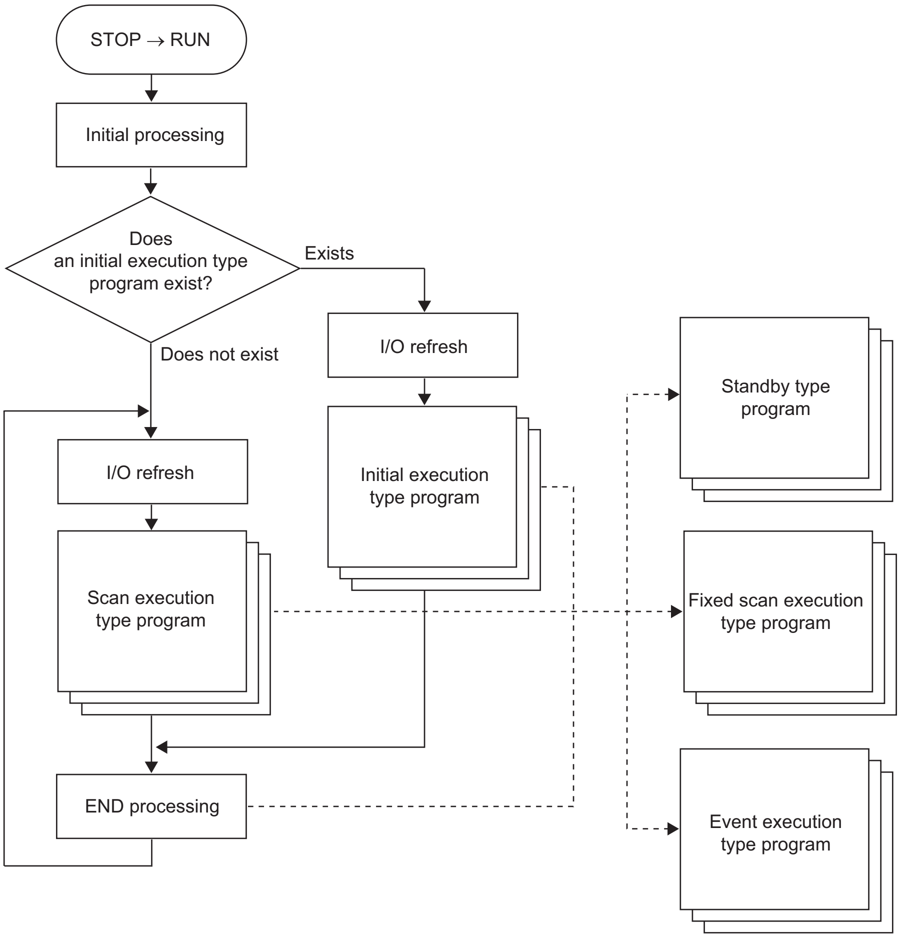


- Y5E turns on the latest



1.4 Program Flow

Programs are executed in order when the CPU module is switched to the RUN state according to the program execution type and execution sequence settings (☞ Page 59 Program Execution Type, ☞ Page 70 Execution type change).



Point

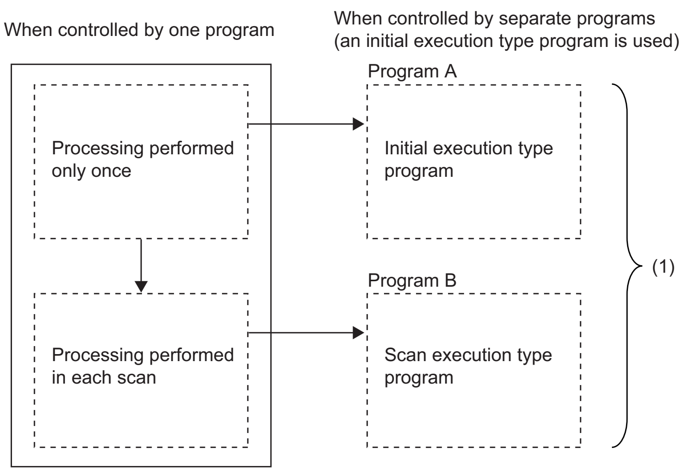
Programs with the same execution type are executed according to the execution sequence settings.

1.5 Program Execution Type

Set the execution condition of the program. (☞ Page 70 Execution type change)

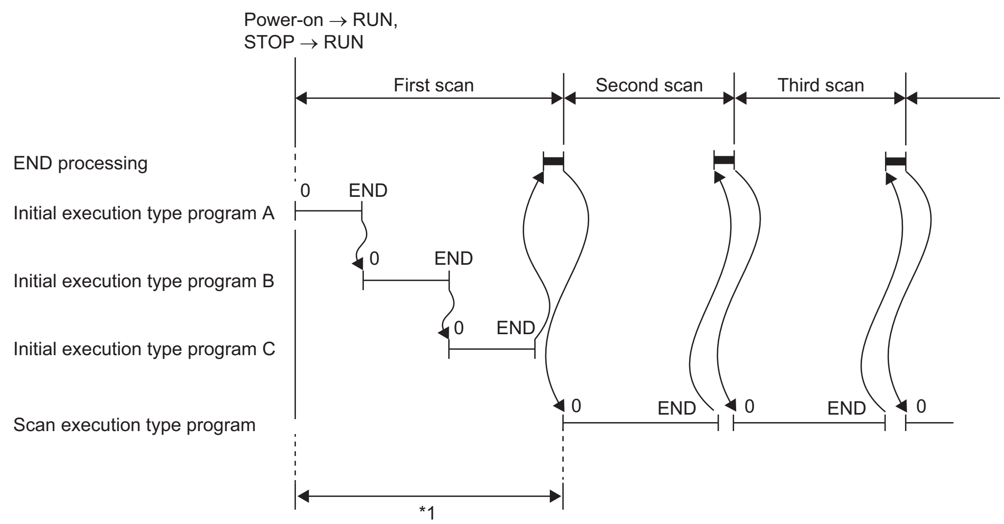
Initial execution type program

Initial execution type program is executed only once when the CPU module has been powered off and on, or switched from the STOP state to the RUN state. Use this execution type for a program, for example, initial processing for the intelligent function module, which does not need execution after executed once in the initial scan.



(1) Divides programs into the initial execution type programs and scan execution type programs.

Note that the execution time of the initial execution type program is the same as the initial scan time. (☞ Page 42 Initial scan time) When multiple initial execution type programs are executed, the execution time of the initial execution type program will be the total time until all the initial execution type programs finish execution.



*1 The sum of the initial execution type program execution time and END processing time is the initial scan time.

Precautions

The following lists the precautions for the initial execution type programs.

■Restrictions on program creation

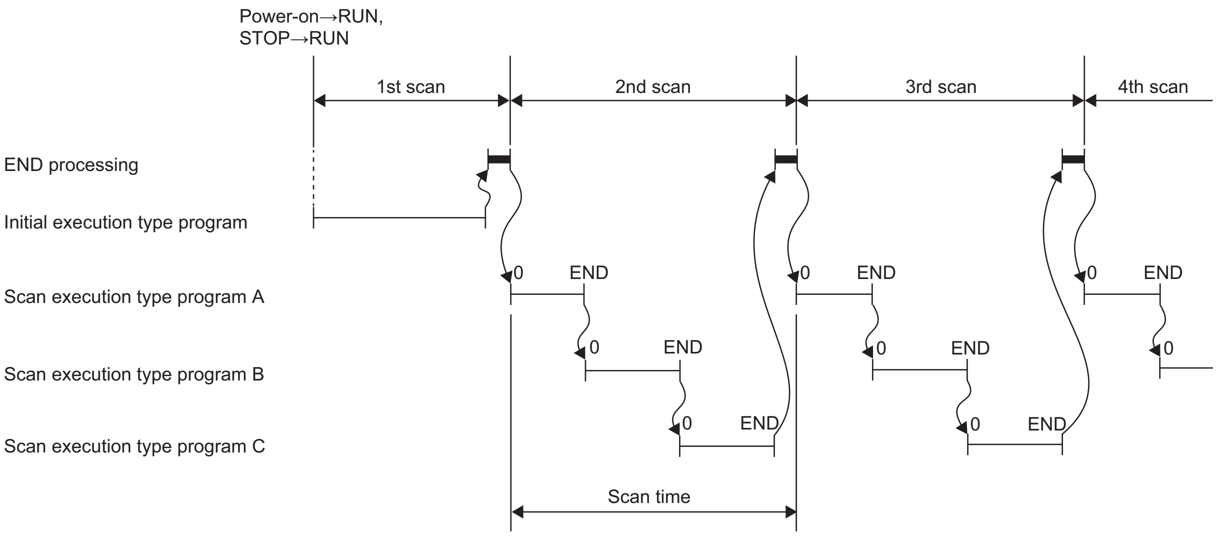
In the initial execution type program, do not use an instruction which requires several scans to complete (i.e., an instruction with a completion device).

Ex.

e.g. SEND and RECV instructions

Scan execution type program

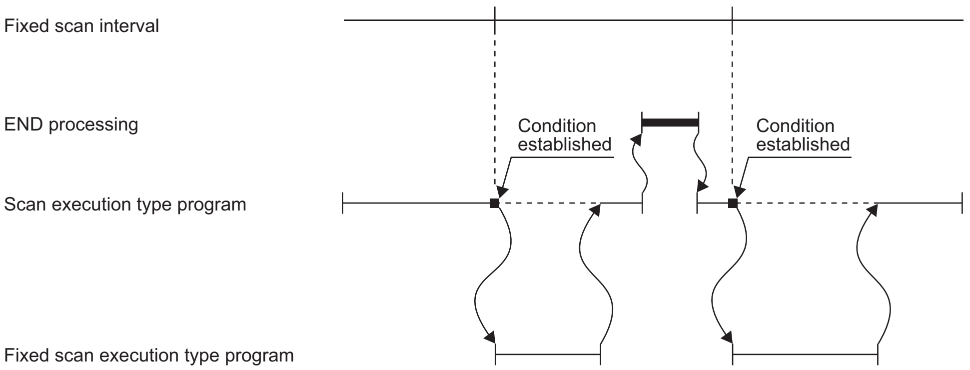
Scan execution type program is executed only once per every scan starting from the scan following the scan in which the initial execution type program was executed.



When multiple scan execution type programs are executed, the execution time of the scan execution type program will be the total time until all the scan execution type programs finish execution. Note that if interrupt program, fixed scan execution type program, and/or event execution type program are executed before the scan execution type program finishes execution, the execution time for these programs is also included.

Fixed scan execution type program

An interrupt program which is executed at a specified time interval. Differently from the normal interrupt program, this type of program does not require the interrupt pointer (I) and the IRET instruction and is executed for each program file unit.



Point

To execute a fixed scan execution type program, the EI instruction must be used to enable interrupts.

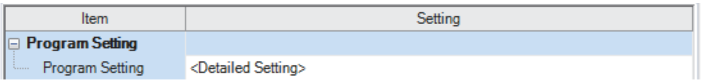
Fixed scan interval setting

Set the execution condition of the fixed scan execution type program.

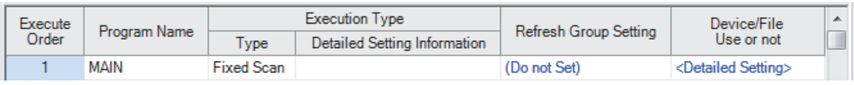
[CPU Parameter] ⇨ [Program Setting]

Operating procedure

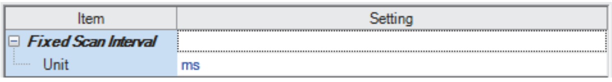
"Program Setting" window



"Detailed Setting" window



"Fixed Scan Execution Type Detailed Setting" window



1. Click "Detailed Setting" on the "Program Setting" window.
2. Select the program name and set the execution type to "Fixed Scan".
3. Click "Detailed Setting Information".
4. Set the fixed scan interval.

Displayed items

Item	Description	Setting range	Default
Specified Time Intervals	Sets the fixed scan interval to execute the fixed scan execution type program.	• When "ms" is selected: 0.5 to 60000ms (in increments of 0.5ms) • When "s" is selected: 1 to 60s (in increments of 1s)	—
Unit	Selects the unit for the fixed scan interval.	• ms • s	ms

Operation when the execution condition is satisfied

The following describes operation of the program.

■If the execution condition is satisfied before the interrupt is enabled by the EI instruction

The program enters the waiting status and is executed when the interrupt is enabled. Note that if the execution condition for this fixed scan execution type program is satisfied more than once during the waiting status, the program is executed only once when the interrupt is enabled.

■If there are multiple fixed scan execution type programs

If the specified time comes simultaneously for these programs, they are executed in order according to the specified execution sequence.

■If another or the same execution condition is satisfied while the fixed scan execution type program is being executed

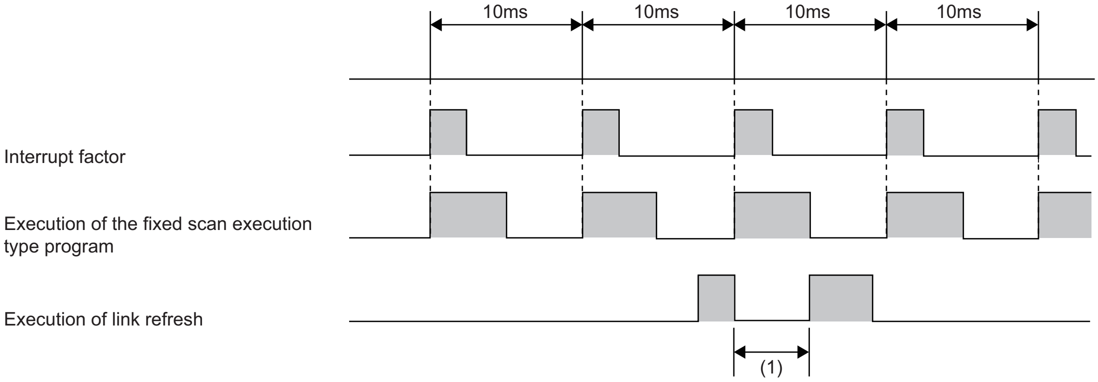
The program is executed according to the fixed scan execution mode setting.

■If the execution condition is satisfied while the interrupt is disabled by the system

The program is executed according to the fixed scan execution mode setting.

■If an interrupt factor occurs during link refresh

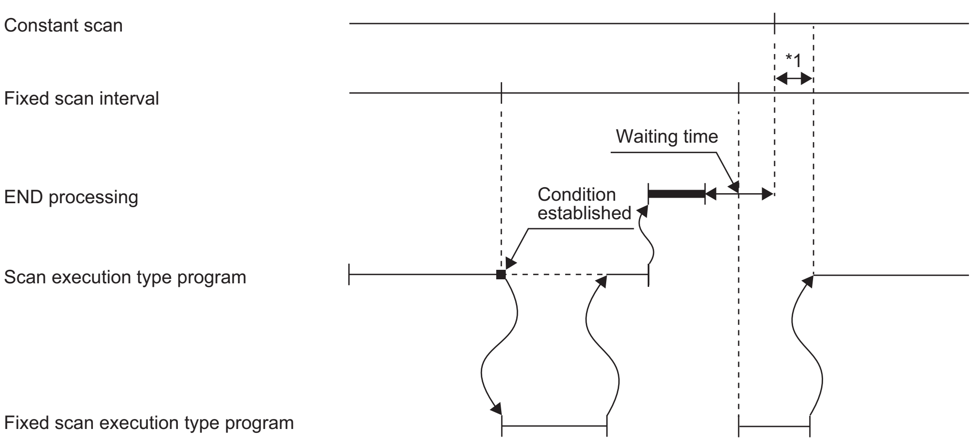
The link refresh is suspended and the fixed scan execution type program is executed. Even while the station-based block data assurance is enabled for cyclic data during refresh of such links as CC-Link IE Field Network, if the fixed scan execution type program uses a device specified as the refresh target, the station-based block data assurance for cyclic data is not available.



(1) The link refresh is suspended and the fixed scan execution type program is executed.

■If an interrupt factor occurs during the waiting time after END processing in constant scan execution

The fixed scan execution type program is executed.



*1 If processing does not finish during the waiting time, the scan time is extended.

■If another interrupt occurs while the fixed scan execution type program is being executed

If an interrupt program (including an interrupt which triggers the event execution type program) is triggered while the fixed scan execution type program is being executed, the program operates in accordance with the interrupt priority. (☞ Page 89 Multiple interrupt function)

Processing when the fixed scan execution type program starts

The same processing as when the interrupt program starts. (☞ Page 86 Processing at interrupt program startup)

Fixed scan execution mode

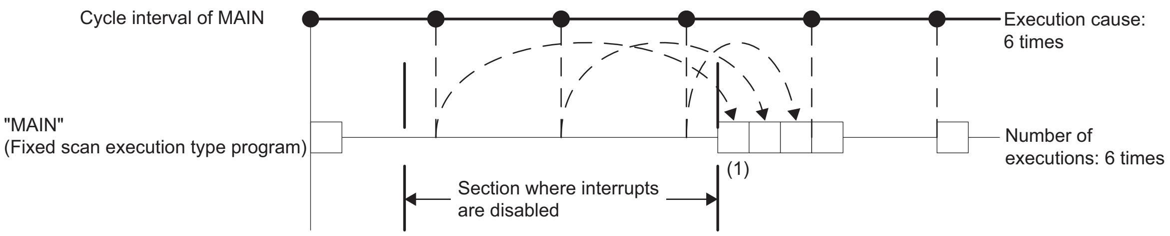
For fixed scan interrupts (I28 to I31, I48, I49) triggered by the fixed scan execution type program or the internal timer of the CPU module, this mode specifies the program execution operation that is performed when more than one interrupt occurs. (☞ Page 75 Operation upon occurrence of an interrupt factor) However, an execution cause which occurs while the interrupt is disabled by the DI instruction is excluded from the fixed scan execution mode target.

■Operation in the fixed scan execution mode

This section describes the operation which can be performed in the fixed scan execution mode.

- Execution Count Takes Priority

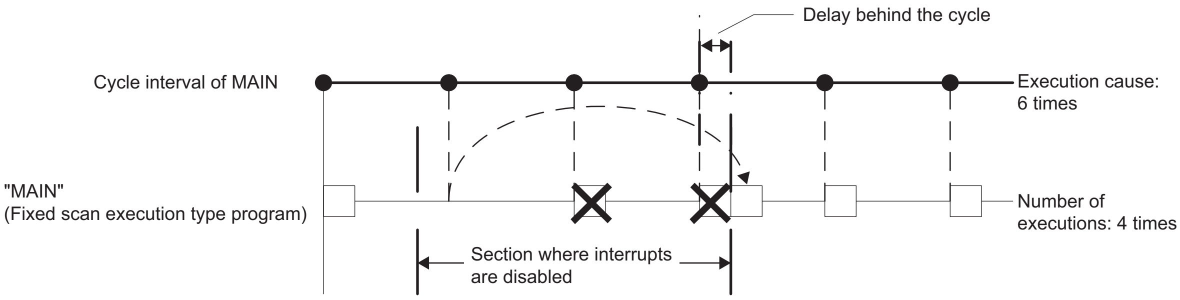
The program is executed for all the pending number of executions so that it can be executed the same number of times as execution causes.



(1) The program is executed just three times, which is the pending number of executions, as soon as interrupt becomes available.

- Precede Fixed Scan

When a pending execution exists, it is performed within the delay limit value behind a cycle. (☞ Page 64 Delay limit value behind a cycle) Note that, even if more than one pending execution exists, only one execution is performed.



■Fixed scan execution mode setting

Use the fixed scan execution mode setting.

[CPU Parameter] ⇨ [Interrupt Settings] ⇨ [Fixed Scan Execution Mode Setting]

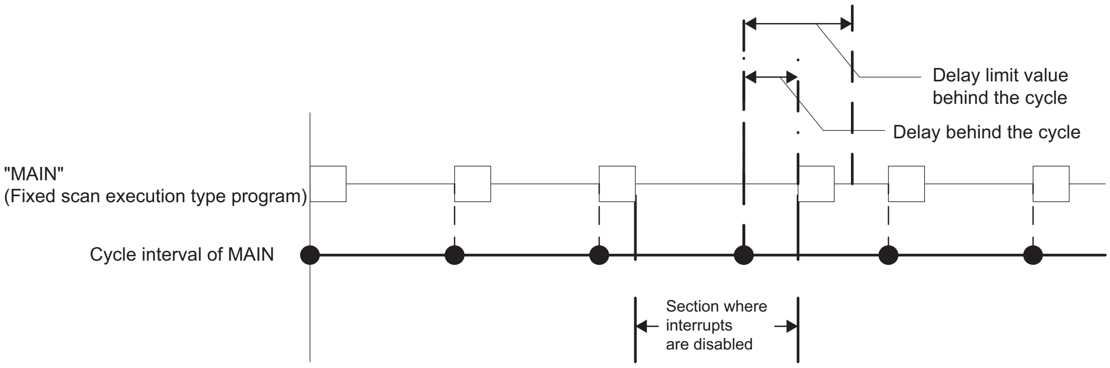
Window

Displayed items

Item	Description	Setting range	Default
Fixed Scan Execution Mode	When fixed scan characteristics are prioritized, an execution is performed within the delay limit value behind a cycle. When the number of executions is prioritized, all the pending executions are performed.	• Precede Fixed Scan • Execution Count Takes Priority	Precede Fixed Scan

■Delay limit value behind a cycle

This value indicates the allowable period of time for a delay (a time lag) behind a cycle and a waiting program is executed if an interrupt is enabled within the period. If an interrupt is enabled outside the period, the program is not executed.



The delay limit value behind a cycle differs as follows.

- For the fixed scan execution type program

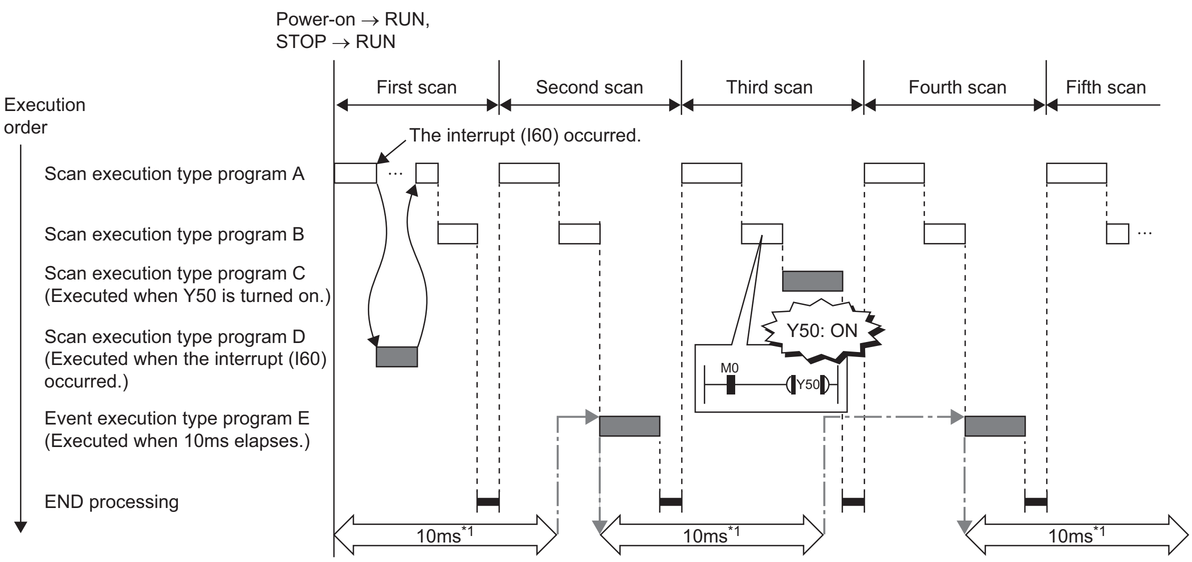
The value is the greatest common divisor of the fixed scan intervals of all the fixed scan execution type programs. For example, if there are fixed scan execution type programs with 2ms, 4ms, and 12ms scan periods, the delay limit value behind a cycle is 2ms.

- For fixed scan interrupts (I28 to I31, I48, I49) using the internal timer of the CPU module

The value is fixed to the same value as the fixed scan interval. For example, if the periods are I28: 100ms and I29: 20ms, the delay limit value behind a cycle is 100ms for I28, and 20ms for I29.

Event execution type program

This type of program starts execution when triggered by a specified event. (☞ Page 65 Trigger type) The program is executed at the execution turn specified in the program settings of the CPU parameters, and if execution conditions of the specified trigger are met when the execution turn of the event execution type program comes, the program is executed.



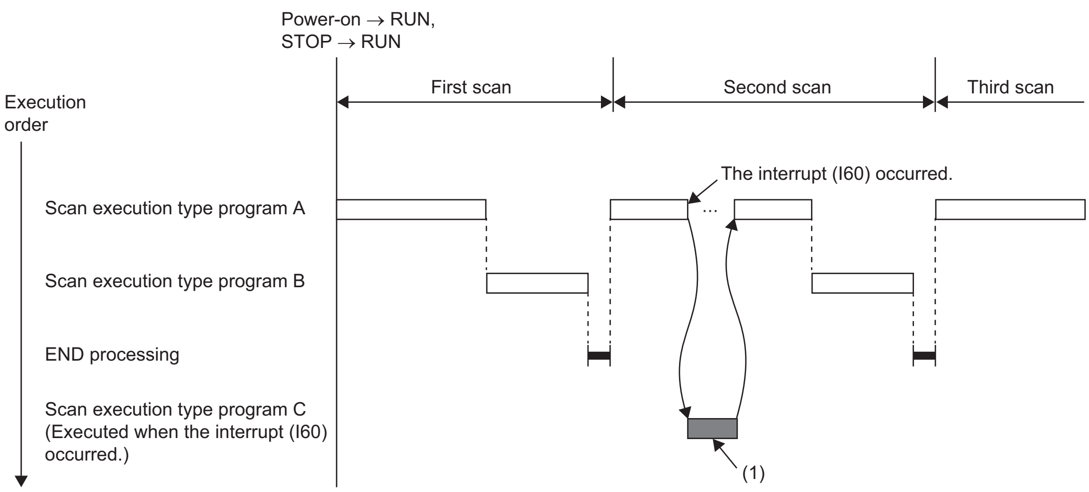
*1 Measurement of elapsed time is 10ms or more because it is determined depending on the scan time.

Trigger type

The following lists the triggers for the event execution type programs. (☞ Page 68 Trigger setting)

■Interrupt occurrence by the interrupt pointer (I)

When the specified interrupt factor occurs, the program is immediately executed once. However, before that, interrupt must be enabled by executing the EI instruction. Enabling programs to be independently executed as interrupt programs, this method eliminates the need to write the FEND instruction, interrupt pointers, and the IRET instruction within scan execution type programs.



(1) When an interrupt occurs, the event execution type program C is executed immediately.

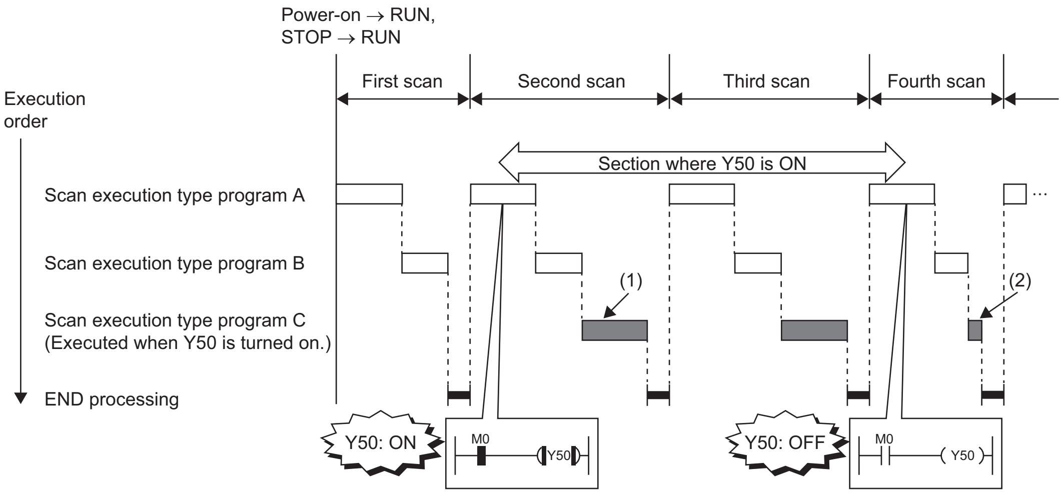
Interrupt pointers (I) which can be specified are I0 to I15, I28 to I31, I44, I45, I48, I49, I50 to I1023.

Point

Execution conditions for the event execution type program which is triggered by interrupt occurred by the interrupt pointer (I) are the same as those for general interrupt programs. (☞ Page 75 Operation upon occurrence of an interrupt factor)

■Bit data ON (TRUE)

The program is executed at the execution turn specified in program setting of the CPU parameters, and if the specified bit data is ON (TRUE) when the execution turn of the event execution type program comes, the program is executed. The current values of the output (Y), timer (T), and long timer (LT) used in this program can be cleared at the execution turn that comes after the specified bit data is changed from ON (TRUE) to OFF (FALSE).



(1) If Y50 is on when the execution turn of the event execution type program C comes, the program is executed. If Y50 is OFF, the program is not executed.
(2) When "Clear Output and Current Value of Timer" is set, the current values of the output (Y), timer (T), and long timer (LT) used in this program are cleared at the execution turn of the event execution type program that comes after Y50 is off.

Applicable devices are as follows.

Item		Description
Device*1	Bit device	X(DX), Y, M, L, F, SM, B, SB, Jn\X, Jn\Y, Jn\B, Jn\SB
	Bit specification of word device	D, SD, W, SW, R*2, ZR*2, RD, Un\G, Jn\W, Jn\SW

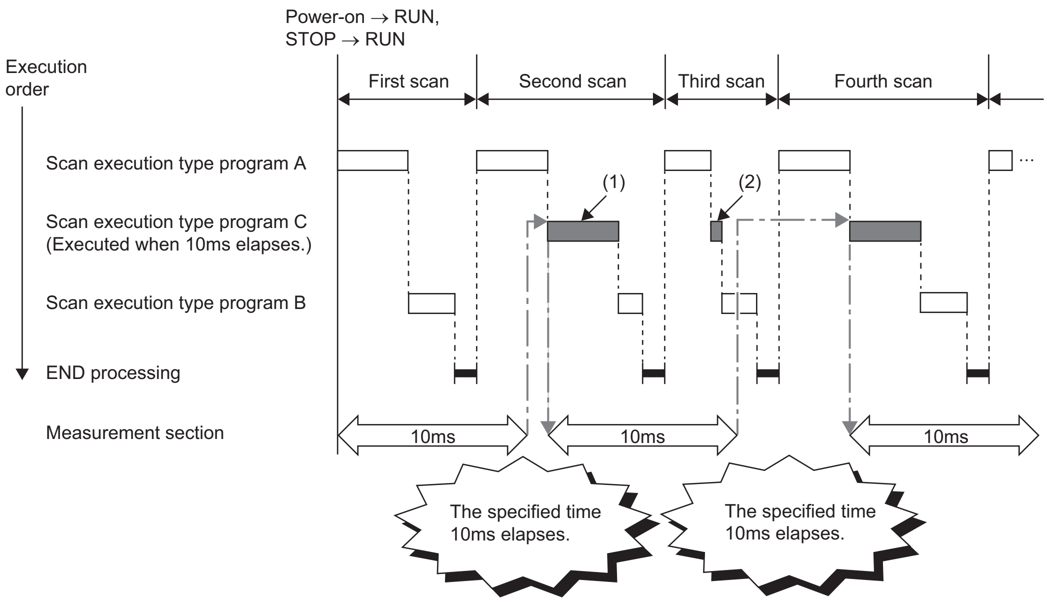
*1 A local device or index-modified device cannot be specified.
*2 This is not available when "Use File Register of Each Program" is enabled. However, if no file register is assigned or if specification is out of the range, interrupt is not be turned on (TRUE).

Point

To measure the timer and the retentive timer continuously even when the trigger execution condition of the event execution type program is not met, use the long timer (LT) and the long retentive timer (LST). The timer (T) and the retentive timer (ST) can be used in the event execution type program. However, they do not measure time when the trigger execution condition of the event execution type program is not met. Set "Clear Output and Current Value of Timer" to "Clear" when using the timer (T) and the retentive timer (ST). (☞ Page 68 Trigger setting)
Note that if the above parameter is set, the use of the long timer (LT) and the long retentive timer (LST) is limited to the same as that of the timer (T) and the retentive timer (ST). (They do not measure time continuously when the trigger execution condition of the event execution type program is not met.)

■Passing time

After the status of the CPU module is changed into the RUN state, programs are executed in execution turn specified in "Program Setting" of "CPU Parameter". If the specified time passes, the event execution type program is executed once when the execution turn of the program comes. The time is measured again starting from the program execution and the above operation is repeated. This method can be used when the processing is executed regularly after the specified time passes. This method is not for the interrupt execution which another program execution is interrupted and processed to secure the punctuality. The current values of the output (Y), timer (T), and long timer (LT) used in this program can be cleared at the execution turn that comes after the specified time passes.



(1) After the specified time elapses, the event execution type program C is executed when the execution turn of the program comes.

(2) When "Clear Output and Current Value of Timer" is set, the current values of the output (Y), timer (T), and long timer (LT) used in this program are cleared at the execution turn of the event execution type program that comes after the specified time elapses.

Point

- Even though "Clear Output and Current Value of Timer" is set to "Clear" in the CPU parameters, the output value and the current value of the timer are not cleared if the scan time is longer than the value set to "Passing Time".
- To measure time when the trigger type is set to "Passing Time", use the long timer (LT) and the long retentive timer (LST).

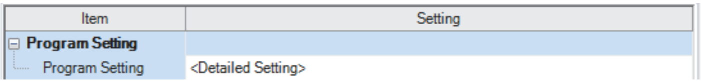
Trigger setting

Use the event execution type detail setting.

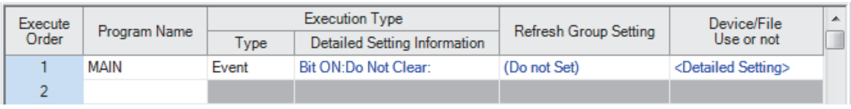
[CPU Parameter] ⇨ [Program Setting]

Operating procedure

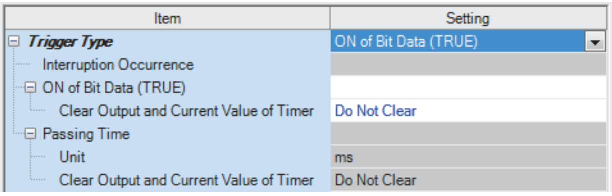
"Program Setting" window



"Detailed Setting" window



"Event Execution Type Detailed Setting" window



1. Click "Detailed Setting" on the "Program Setting" window.

2. Select the program name and set the execution type to "Event".

3. Click "Detailed Setting Information".

4. Sets the trigger type to execute the event execution type program.

Displayed items

Item		Description	Setting range	Default
Interruption Occurrence		Sets the interrupt pointer used as the trigger.	I0 to I15, I28 to I31, I44, I45, I48*1, I49*1, I50 to I1023	—
ON of Bit Data (TRUE)		Sets the device used as the trigger.	☞ Page 66 Bit data ON (TRUE)	—
Clear Output and Current Value of Timer		Sets that the current values of the output (Y), timer (T), and long timer (LT) used in this program are cleared at the execution turn of the event execution type program that comes after the specified bit data is OFF.	• Do Not Clear • Clear	Do Not Clear
Passing Time	Unit	Sets the time passed.	• When "ms" is selected: 1 to 65535ms (in units of 1ms) • When "s" is selected: 1 to 65535s (in units of 1s)	ms
	Clear Output and Current Value of Timer	Sets that the current values of the output (Y), timer (T), and long timer (LT) used in this program are cleared at the execution turn of the event execution type program that comes after the specified time passes.	• Do Not Clear • Clear	Do Not Clear

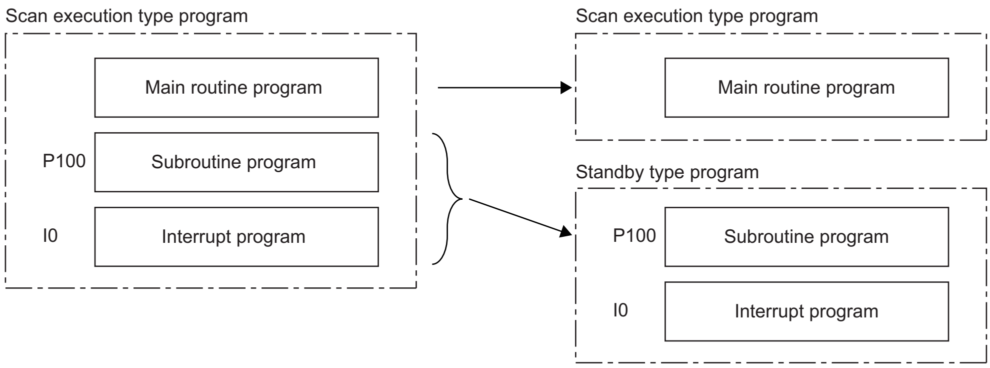
*1 It cannot be set in the R00CPU, R01CPU, and R02CPU.

Standby type program

This type of program is executed only when its execution is requested.

Librarization of programs

Set a subroutine program and/or an interrupt program as a standby type program to manage them separately from the main routine program. In a single standby type program, multiple subroutine programs and interrupt programs can be created.



Program arrangement change

Prepare programs available in all systems to use them only when necessary. For example, a program set in advance as the standby type with a parameter can be changed to scan execution type and executed. (☞ Page 70 Using an instruction)

Execution method

The standby type program is executed as follows.

- Create a subroutine program and interrupt program within the standby type program, and then call them when an interrupt occurs or by specifying with a pointer.
- Switch to another execution type program.

Execution type change

This section describes how to change the execution type of programs.

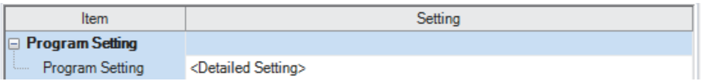
Using parameter settings

"Program Setting" can be used to specify the execution type of programs.

[CPU Parameter] ⇨ [Program Setting] ⇨ [Detailed Setting]

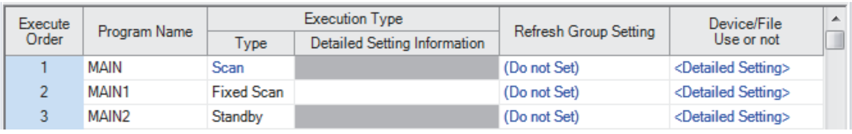
Operating procedure

"Program Setting" window



Item	Setting
⊟ **Program Setting**	
Program Setting	<Detailed Setting>

"Detailed Setting" window



Execute Order	Program Name	Execution Type: Type	Execution Type: Detailed Setting Information	Refresh Group Setting	Device/File Use or not
1	MAIN	Scan		(Do not Set)	<Detailed Setting>
2	MAIN1	Fixed Scan		(Do not Set)	<Detailed Setting>
3	MAIN2	Standby		(Do not Set)	<Detailed Setting>

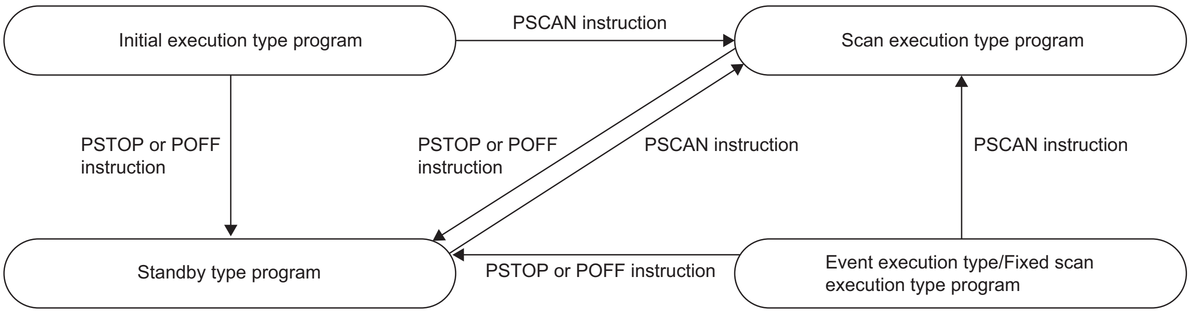
1. Click "Detailed Setting" on the "Program Setting" window.

2. For each program, select the execution type in "Type" of "Execution Type".

Using an instruction

Use the following instructions.

- PSCAN
- PSTOP
- POFF



■Change timing

The execution type change timing is shown below.

<table>
<tr><th rowspan="2">Execution type before change</th><th colspan="3">Execution instruction</th></tr>
<tr><th>PSCAN</th><th>PSTOP</th><th>POFF</th></tr>
<tr><td>Initial</td><td>Switches to "Scan" at the next scan.</td><td rowspan="3">Switches to "Standby" at the next scan.</td><td rowspan="2">Turns off the output at the next scan and switches to "Standby" at the scan after that.</td></tr>
<tr><td>Scan</td><td>Non-processing (does not change from "Scan")</td></tr>
<tr><td>Fixed scan/event</td><td rowspan="2">Switches to "Scan" at the next scan.</td><td>Turns off the output at the next scan and switches to "Standby" at the scan after that.*1</td></tr>
<tr><td>Standby</td><td>Non-processing (does not change from "Standby")</td><td>Non-processing (does not change from "Standby")</td></tr>
</table>

*1 This instruction stops the execution of the program at END processing after instruction execution.

■Precautions

The following lists the precautions when changing the execution type.

- If a fixed scan execution type program or event execution type program is changed to another execution type, it cannot be restored to the original execution type.
- If an instruction is executed multiple times in a single scan for a single program, the program operates for the instruction executed last.
- For details on operation if an SFC program is specified, refer to the following.

📖 MELSEC iQ-R Programming Manual (Program Design)

Group setting for refresh

Refresh can be performed when a specified program is executed*1 by setting a group number to each program and specifying the number for each module.

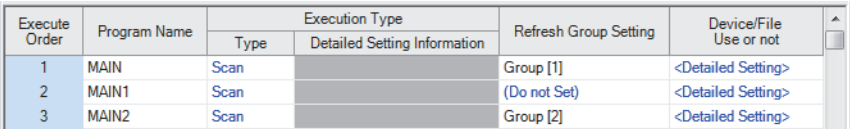
*1 Input refresh (load of analog input, Input (X)) is performed before execution of a program, and output refresh (analog output, Output (Y)) is performed after execution of a program.

Refresh group setting

A group number is set to each program.

[CPU Parameter] ⇨ [Program Setting] ⇨ [Program Setting] ⇨ [Detailed Setting]

Window



Execute Order	Program Name	Execution Type		Refresh Group Setting	Device/File Use or not
		Type	Detailed Setting Information		
1	MAIN	Scan		Group [1]	<Detailed Setting>
2	MAIN1	Scan		(Do not Set)	<Detailed Setting>
3	MAIN2	Scan		Group [2]	<Detailed Setting>

Displayed items

Item	Description	Setting range	Default
Refresh Group Setting	Set the group number of each program which is specified for each module.	• (Do not Set) • Groups [1] to [64]	(Do not Set)

Point

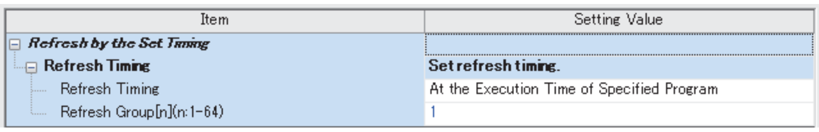
When the specified group number is not set in the refresh settings of each module, the program is executed, but the refresh of the relevant group number is not executed.

Refresh setting of each module

Select "At the execution time of specified program" in "Refresh Setting" of each module, and type the group number of the program to be refreshed.

[Module Parameter] ⇨ [Refresh Setting]

Window



Item	Setting Value
Refresh by the Set Timing	
Refresh Timing	Set refresh timing.
Refresh Timing	At the Execution Time of Specified Program
Refresh Group[n](n:1-64)	1

(Example: The "Refresh Setting" window for an I/O module)

Point

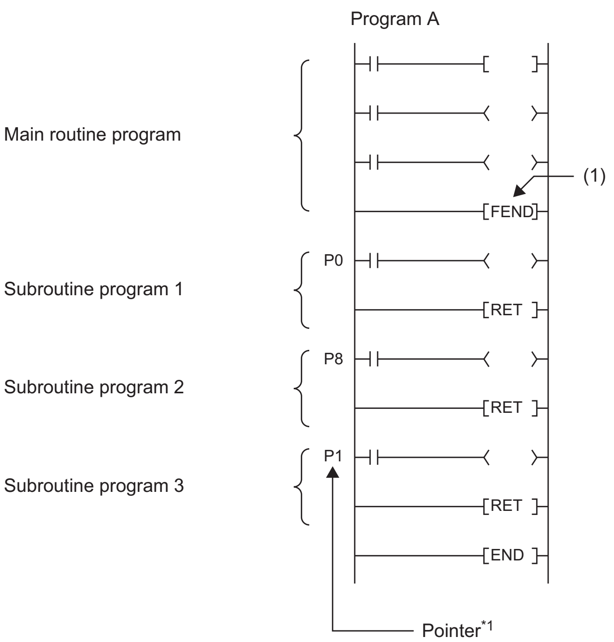
For refresh of each module, refer to the manual for the module used.

1.6 Subroutine Program

Subroutine program is a program that is executed from a pointer (P) through the RET instruction. It is executed only when called by a subroutine call instruction (such as the CALL instruction or the ECALL instruction). A pointer type label can also be used instead of a pointer (P). The subroutine program is used for the following purposes.

- By grouping programs executed multiple times in a single scan into a single subroutine program, the total number of steps in a single program can be decreased.
- By making a program which is only executed in a certain condition a subroutine program, the scan time can be shortened for that program execution.

(1) The end of the main routine program



*1 The pointers are not required to be defined in an ascending order.

Point

- Making a program a standby type allows it to be managed as a separate program. (☞ Page 69 Standby type program) Use the ECALL instruction and other similar instructions to call program files.
- For details on nesting (nesting structure) of subroutine programs, refer to the nesting (N). (☞ Page 408 Nesting (N))

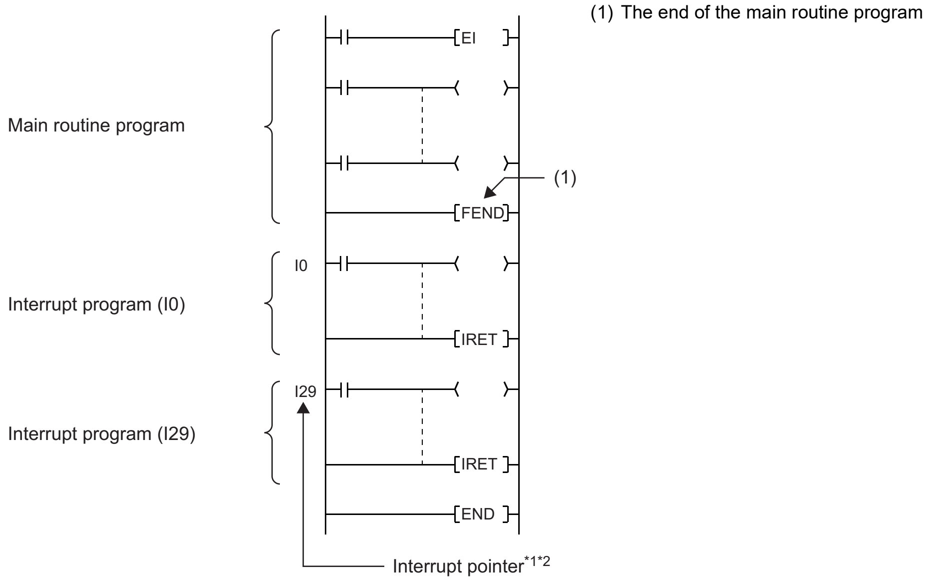
Precautions

The following lists the precautions when using the subroutine program.

- When using a local device, setting of SM776 (Local device setting at CALL) allows for using a local device of a program file at the storage location of the subroutine program.
- Do not use the timer (T, ST). However, the timer can be used if the timer coil (the OUT T□ instruction) is executed only once per scan.
- If the RET instruction is not used to return to the calling program and then the program is terminated, an error will occur.
- If a pointer (P) or pointer type global label exists within FB or FUN, an error will occur.

1.7 Interrupt Program

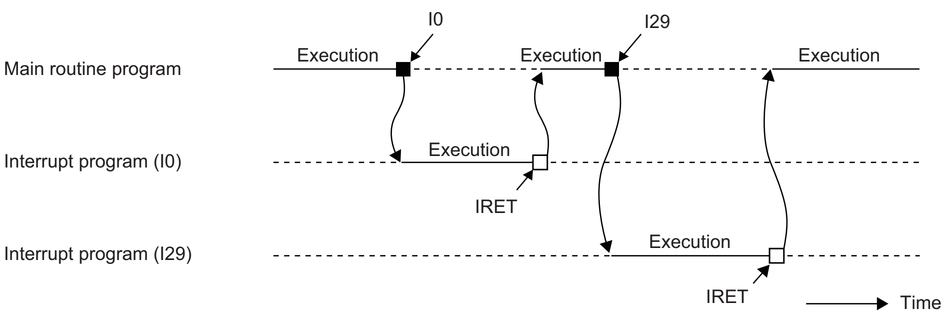
A program from an interrupt pointer (I) through the IRET instruction.



*1 Only one interrupt program can be created with a single interrupt pointer number.

*2 The interrupt pointers are not required to be defined in an ascending order.

When an interrupt factor occurs, the interrupt program corresponding to its interrupt pointer number is executed. (☞ Page 412 The priority for the interrupt pointer numbers and interrupt factors) However, before that, interrupt must be enabled by using the EI instruction.



I0: Interrupt factor for I0

I29: Interrupt factor for I29

Point

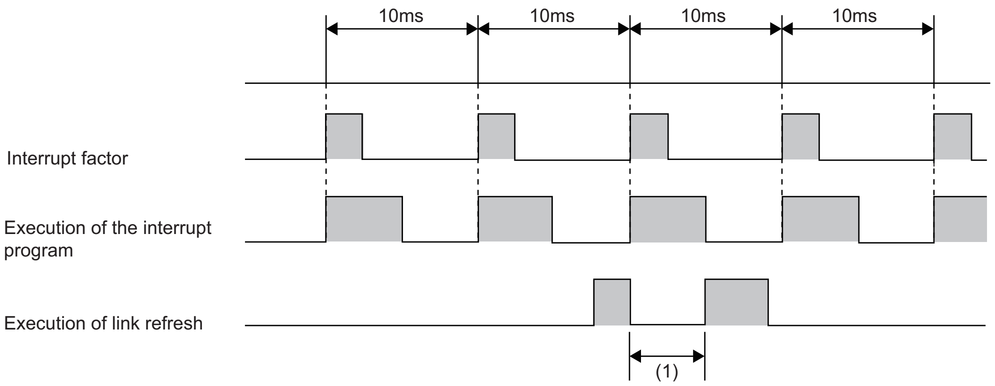
Making a program a standby type allows it to be managed as a separate program. (☞ Page 69 Standby type program)

Operation upon occurrence of an interrupt factor

The following shows the operation when an interrupt factor occurs.

■If an interrupt factor occurs during link refresh

The link refresh is suspended and the interrupt program is executed. Even though the station-based block data assurance is enabled for cyclic data during refresh of such links as CC-Link IE Field Network, if the interrupt program uses a device specified as the refresh target, the station-based block data assurance for cyclic data is not available.



(1) The link refresh is suspended and the interrupt program is executed.

■If an interrupt factor occurs during the waiting time after END processing in constant scan execution

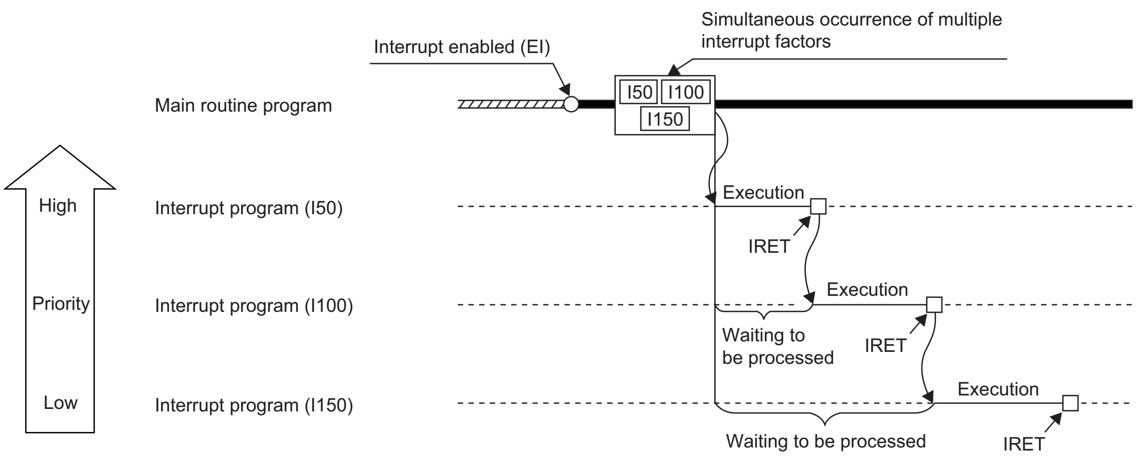
The interrupt program for the interrupt factor is executed.

■If another interrupt occurs while the interrupt program is being executed

If an interrupt such as a fixed scan execution type program (including an interrupt which triggers the event execution type program) is triggered while an interrupt program is being executed, the program operates in accordance with the interrupt priority. (☞ Page 89 Interrupt priority)

■If multiple interrupt factors occur simultaneously while the interrupt is enabled

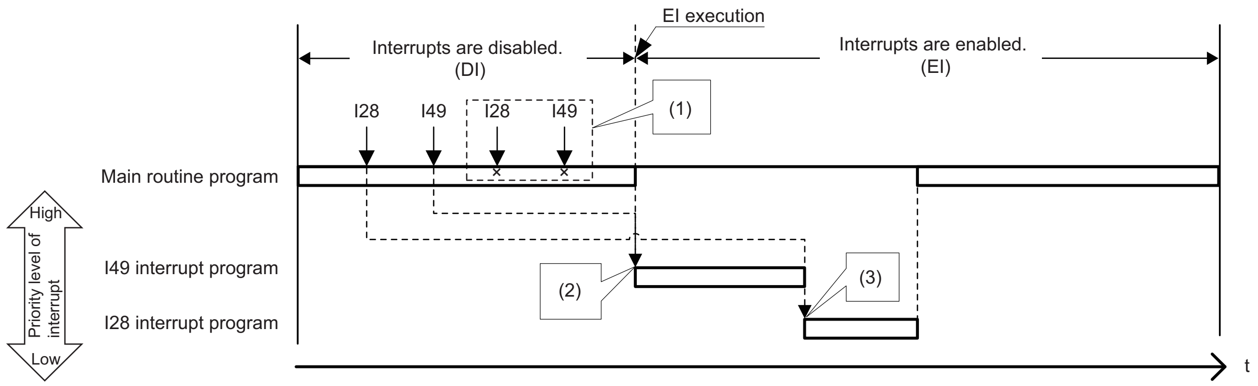
The interrupt programs are executed in the order of priority. If multiple interrupts with the same interrupt priority occur simultaneously, the interrupt programs are executed in the order of interrupt priority. (☞ Page 92 Multiple interrupt execution sequence)



■If an interrupt factor occurs while interrupt is disabled (DI)

- For I0 to I15, I28 to I31, I48, I49, and I50 to I1023

The interrupt factor that has occurred is memorized, and the interrupt program corresponding to the factor will be executed when the interrupt is enabled. Even if the same interrupt factor occurs multiple times, it will be memorized only once. If the IMASK instruction and SIMASK instruction are used to disable the interrupt, all the memorized factors will be discarded.



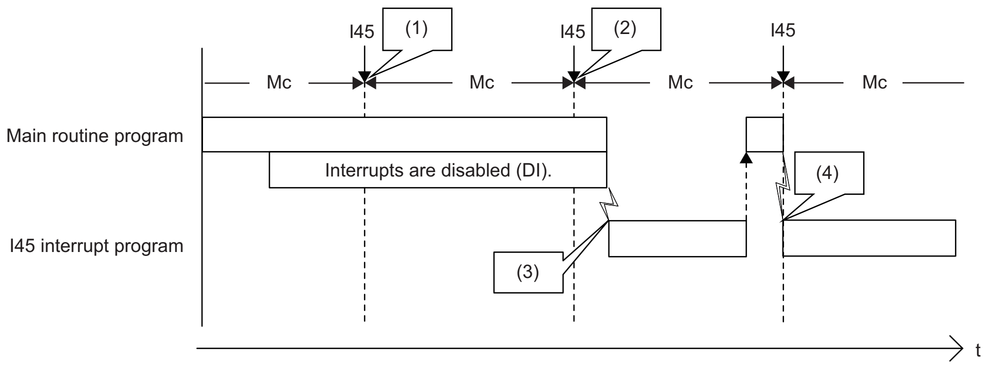
(1) The second and following interrupt factors that occur while interrupts have been disabled (DI) are not memorized.

(2) When interrupts are enabled, interrupts are executed in order from I49 because it has a higher priority.

(3) I28 is executed. (I49 is not executed for the second time.)

- For I45

The interrupt factor that has occurred is memorized, and the I45 interrupt program will be executed when the interrupt is enabled. Even if the I45 interrupt occurs multiple times, its interrupt factor will be memorized only once. If the IMASK instruction and SIMASK instruction are used to disable the interrupt, all the memorized factors will be discarded.



Mc: Multiple CPU synchronization cycle

(1) The interrupt factor is memorized.

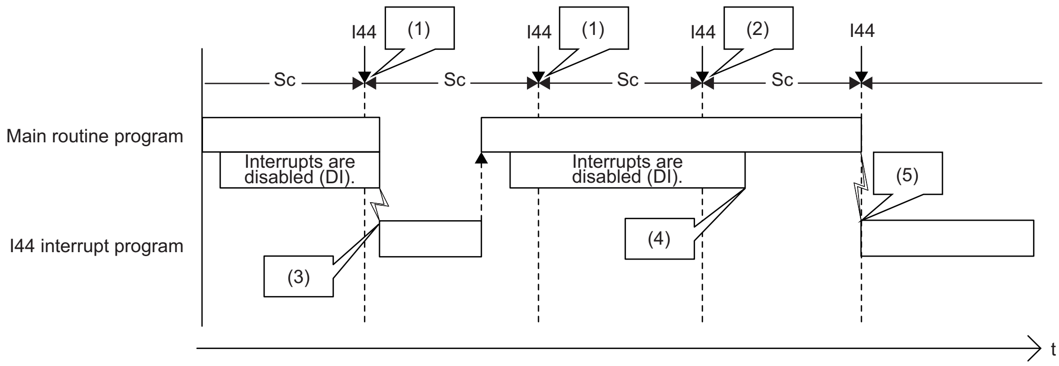
(2) The second and following interrupt factors that occur while interrupts have been disabled (DI) are not memorized.

(3) The interrupt is executed.

(4) The interrupt is executed because interrupts are enabled.

• For I44

If interrupt is enabled before the next cycle, the I44 interrupt program will be executed when the interrupt is enabled. If interrupt continues to be disabled beyond the start of the next cycle (the second cycle), the memorized information will be discarded (even when the interrupt is enabled, the I44 interrupt program will not be executed). Also, if the I44 interrupt program for this cause cannot be executed, SM480 (Cycle overrun flag for inter-module synchronization program (I44)) is turned on, and SD480 (Number of cycle overrun events for inter-module synchronization cycle program (I44)) reaches its upper limit. If the IMASK instruction and SIMASK instruction are used to disable the interrupt, all the memorized factors will be discarded.



Sc: Inter-module synchronization cycle

(1) The interrupt factor is memorized.

(2) Because an interrupt does not occur, the memorized information is discarded in the second cycle.

(3) The interrupt is executed when interrupts are enabled.

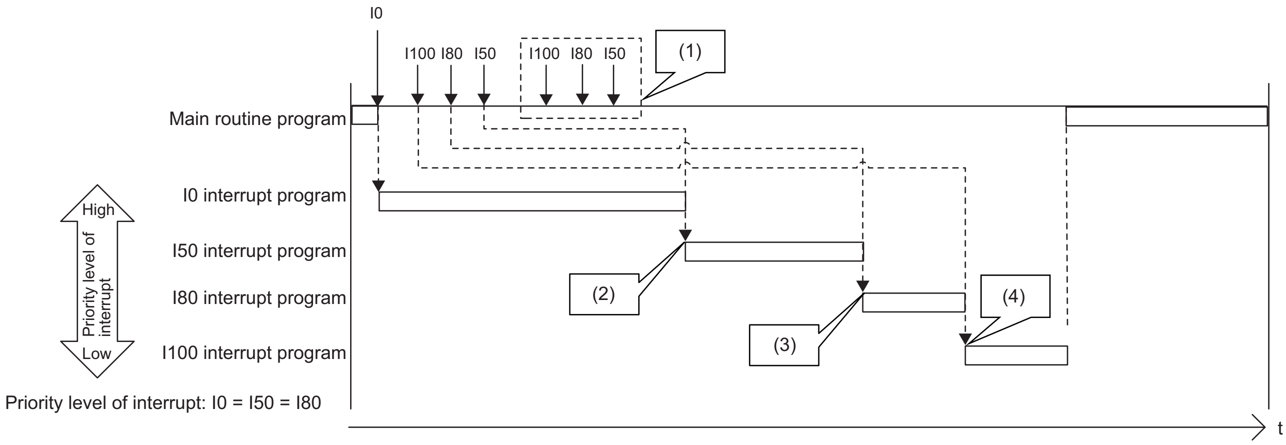
(4) I44 is not executed because interrupts continue to be disabled (DI) beyond the second cycle of I44.

(5) The interrupt is executed because interrupts are enabled.

■If an interrupt factor with the same or a lower priority occurs while the interrupt program is being executed

- For I0 to I15 and I50 to I1023

The interrupt factor that has occurred is memorized. After the running interrupt program finishes, the interrupt program corresponding to the factor will be executed. Even if the same interrupt factor occurs multiple times, it will be memorized only once.



(1) The second and following interrupt factors that occur while interrupts have been disabled (DI) are not memorized.

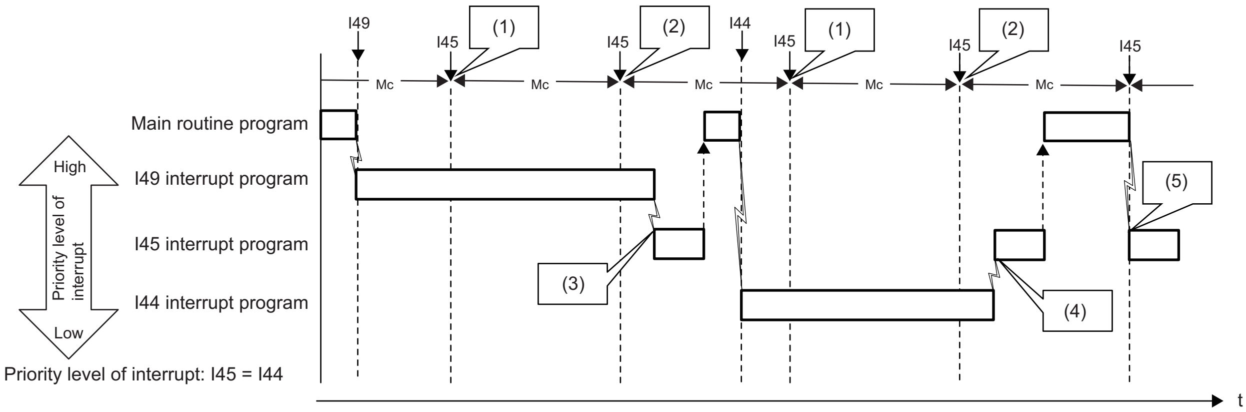
(2) After the interrupt in execution is completed, interrupts are executed in order from I50 because it has a higher priority. Although both I50 and I80 have the same priority level, I50 is executed ahead of I80 because I50 has a higher priority order.

(3) I80 is executed. (I50 is not executed for the second time.)

(4) I100 is executed. (I80 is not executed for the second time.)

- For I45

The interrupt factor that has occurred is memorized. After the running interrupt program finishes, the I45 interrupt program will be executed. Even if the I45 interrupt occurs multiple times, its interrupt factor will be memorized only once.



Mc: Multiple CPU synchronization cycle

(1) The interrupt factor is memorized.

(2) The second and following interrupt factors that occur while interrupts have been disabled (DI) are not memorized.

(3) After I49 that has a higher priority level is completed, I45 is executed.

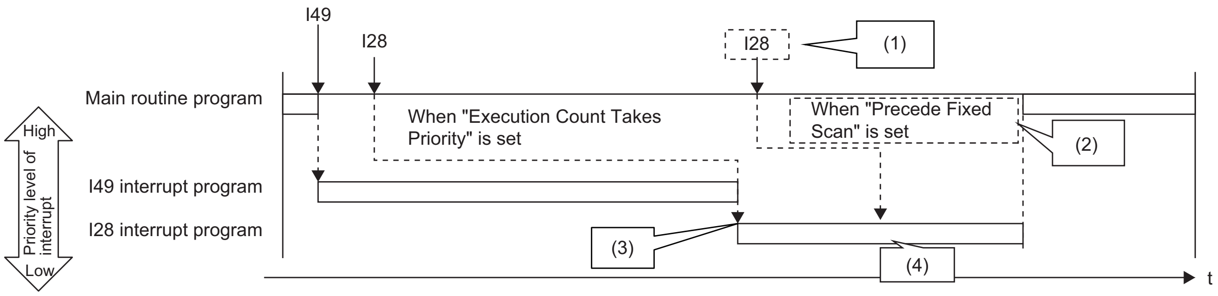
(4) Because both I44 and I45 have the same priority level, the interrupt is executed after I44 is completed.

(5) The interrupt is executed because I49 and I44 have been completed.

• For I28 to I31, I48, and I49

The interrupt factor that has occurred is memorized. After the running interrupt program finishes, the interrupt program corresponding to the factor will be executed. If the same interrupt factor occurs multiple times, it will be memorized once but operation at the second and later occurrences depends on setting of the fixed scan execution mode (☞ Page 63 Fixed scan execution mode). When "Execution Count Takes Priority" is enabled, the interrupt program corresponding to the memorized interrupt factor will be executed after the running interrupt program finishes. When "Precede Fixed Scan" is enabled, the second and later occurrences will not be memorized.

When interrupts with lower priority level than the one being executed occur



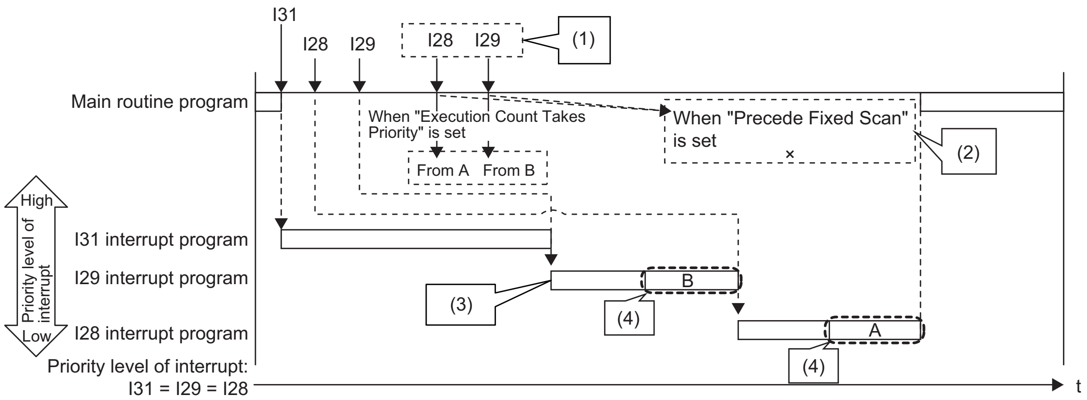
(1) The second and following interrupt factors that occur while interrupts are being executed operate according to the setting of the fixed scan execution mode.

(2) The program is not executed for the second time.

(3) After the interrupt in execution is completed, interrupts are executed in order from I29 because it has a higher priority.

(4) The second interrupt is executed.

When interrupts with the same priority level as the one being executed occur



(1) The second and following interrupt factors that occur while interrupts are being executed operate according to the setting of the fixed scan execution mode.

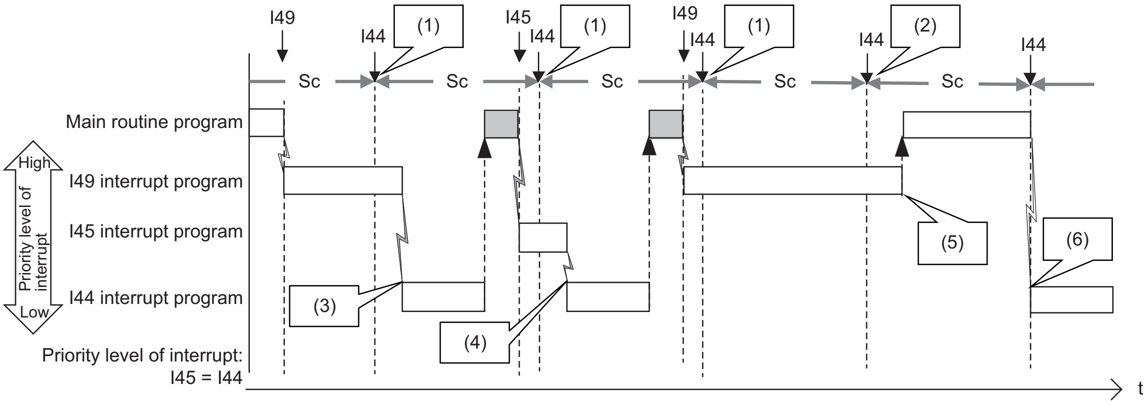
(2) The program is not executed for the second time.

(3) After the interrupt in execution is completed, interrupts are executed in order from I29 because it has a higher priority.

(4) The second interrupt is executed.

- For I44

If the running interrupt program finishes before the next cycle, the I44 interrupt program will be executed when the running interrupt program finishes. If the running interrupt program continues beyond the start of the next cycle (the second cycle), the memorized information will be discarded (even when the running interrupt program finishes, the I44 interrupt program will not be executed). Also, if the I44 interrupt program for this cause cannot be executed, SM480 (Cycle overrun flag for inter-module synchronization program (I44)) is turned on, and SD480 (Number of cycle overrun events for inter-module synchronization cycle program (I44)) reaches its upper limit.



Sc: Inter-module synchronization cycle

(1) The interrupt factor is memorized.

(2) Because an interrupt does not occur, the memorized information is discarded in the second cycle.

(3) After I49 that has a higher priority level is completed, I44 is executed.

(4) After I45 that has the same priority level and a higher priority order is completed, the interrupt is executed.

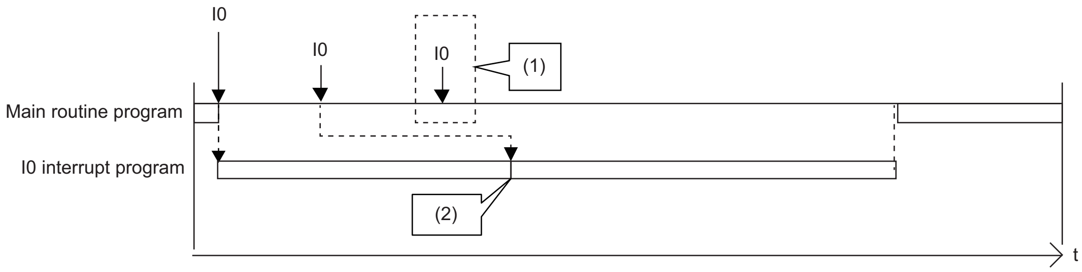
(5) I44 is not executed because the termination of I49 or I45 is in the second cycle for I44.

(6) The interrupt is executed because I49 and I45 have been completed.

■If the same interrupt factor occurs while the interrupt program is being executed

- For I0 to I15 and I50 to I1023

The interrupt factor that has occurred is memorized, and the interrupt program corresponding to the factor will be executed when the interrupt is enabled. Even if the same interrupt factor occurs multiple times, it will be memorized only once.

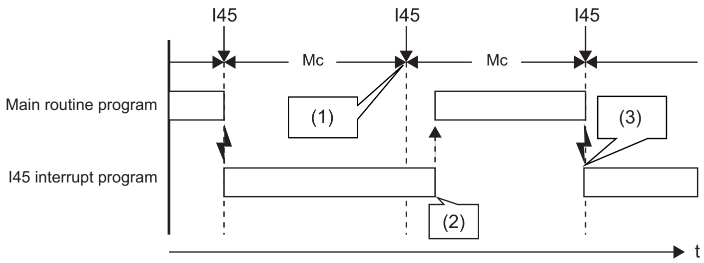


(1) The second and following interrupt factors that occur while interrupts are executed are not memorized.

(2) After the interrupt in execution is completed, the first interrupt program is executed.

- For I45

If an interrupt factor which is the same as that for the running interrupt program occurs, the factor is not memorized. Therefore, the corresponding interrupt program will not be executed after the running interrupt program finishes. Also, if the I45 interrupt program for this cause cannot be executed, SM481 (Cycle overrun flag for multiple CPU synchronization program (I45)) is turned on, and SD481 (Number of cycle overrun events for multiple CPU synchronization program (I45)) reaches its upper limit.



Mc: Multiple CPU synchronization cycle

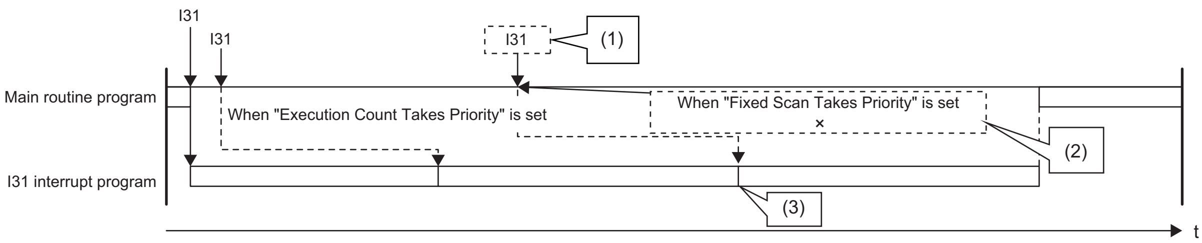
(1) The interrupt factor is discarded because the same interrupt (I45) is being executed.

(2) The interrupt is not executed.

(3) The interrupt is executed.

- For I28 to I31, I48, and I49

The interrupt factor that has occurred is memorized. After the running interrupt program finishes, the interrupt program corresponding to the factor will be executed. If the same interrupt factor occurs multiple times, it will be memorized once but operation at the second and later occurrences depends on setting of the fixed scan execution mode (☞ Page 63 Fixed scan execution mode). When "Execution Count Takes Priority" is enabled, the interrupt program corresponding to the memorized interrupt factor will be executed after the running interrupt program finishes. When "Precede Fixed Scan" is enabled, the second and later occurrences will not be memorized.



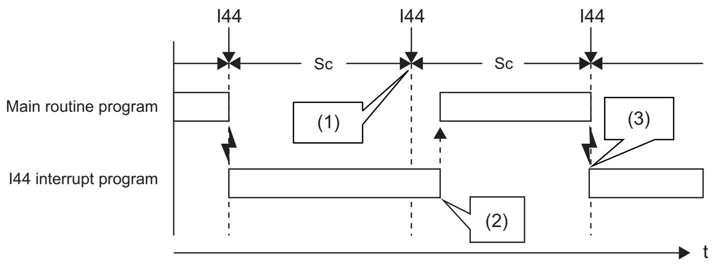
(1) The second and following interrupt factors that occur while interrupts are being executed operate according to the setting of the fixed scan execution mode.

(2) The program is not executed for the second time.

(3) The second interrupt is executed.

• For I44

If an interrupt factor which is the same as that for the running interrupt program occurs, the factor is not memorized. Therefore, the corresponding interrupt program will not be executed after the running interrupt program finishes. Also, if the I44 interrupt program for this cause cannot be executed, SM480 (Cycle overrun flag for inter-module synchronization program (I44)) is turned on, and SD480 (Number of cycle overrun events for inter-module synchronization cycle program (I44)) reaches its upper limit.



Sc: Inter-module synchronization cycle

(1) The interrupt factor is discarded because the same interrupt (I44) is being executed.

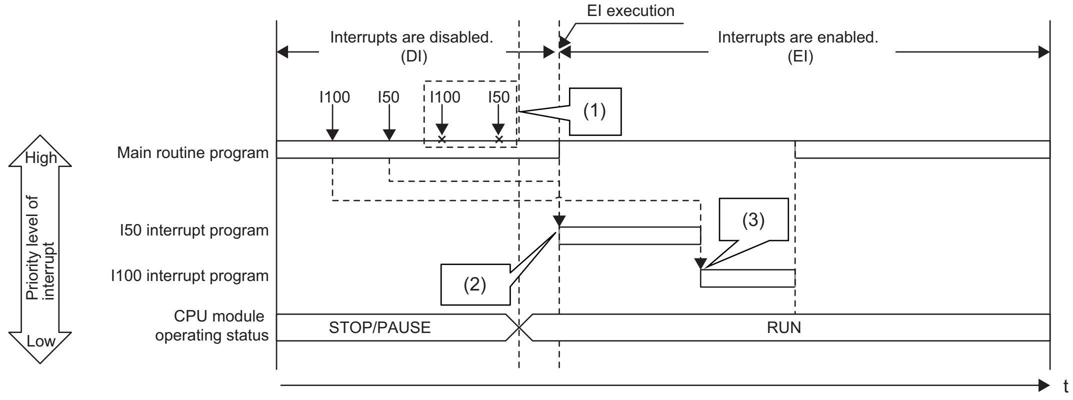
(2) The interrupt is not executed.

(3) The interrupt is executed.

■If an interrupt factor occurs in the STOP/PAUSE status

• For I0 to I15, I28 to I31, I48, I49, and I50 to I1023

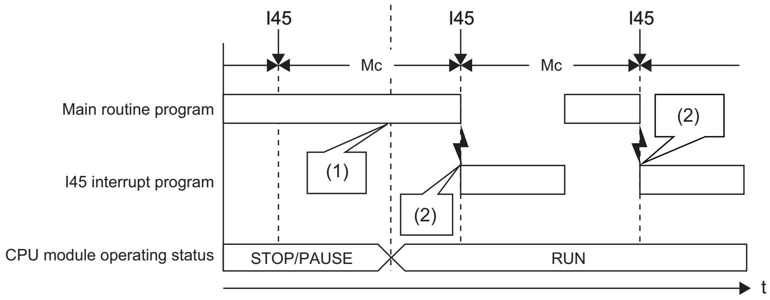
The interrupt factor that has occurred is memorized, and the corresponding interrupt program will be executed when the CPU module switches to the RUN state and the interrupt is enabled. Even if the same interrupt factor occurs multiple times before switching to the RUN state, it will be memorized only once.



(1) The second and following interrupt factors that occur while the CPU module is in the STOP state are not memorized.
(2) When interrupts are enabled by changing the operating status of the CPU module from STOP to RUN, interrupts are executed in order from I50 that has a higher priority level.
(3) I100 is executed. (I50 is not executed for the second time.)

• For I45

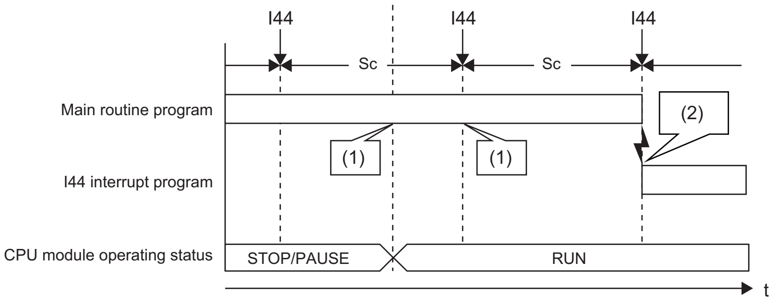
The interrupt factor that has occurred is not memorized, and therefore the corresponding interrupt program will not be executed even when the CPU module switches to the RUN state and the interrupt is enabled. The interrupt program will be executed when the CPU module switches to the RUN state and then the first interrupt factor occurs.



Mc: Multiple CPU synchronization cycle
(1) The interrupt is not executed.
(2) The interrupt is executed.

• For I44

The interrupt factor that has occurred is not memorized, and therefore the corresponding interrupt program will not be executed even when the CPU module switches to the RUN state and the interrupt is enabled. Instead, startup of interrupt is prepared when the CPU module switches to the RUN state and the interrupt is enabled (the interrupt program will not be executed upon occurrence of the first interrupt factor). Then, the interrupt program will be executed during the second cycle after the switch to the RUN state.



Sc: Inter-module synchronization cycle

(1) The interrupt is not executed.

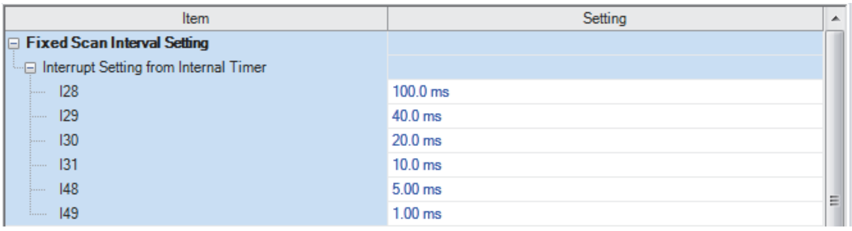
(2) The interrupt is executed.

Interrupt period setting

The interrupt cycle based on the internal timer can be specified.

[CPU Parameter] ⇨ [Interrupt Settings] ⇨ [Fixed Scan Interval Setting]

Window



Displayed items

Item	Item	Description	Setting range	Default
Interrupt Setting from Internal Timer	I28	Sets the execution interval of I28.	0.5 to 1000ms (in units of 0.5ms)	100.0ms
	I29	Sets the execution interval of I29.		40.0ms
	I30	Sets the execution interval of I30.		20.0ms
	I31	Sets the execution interval of I31.		10.0ms
	I48*1	Sets the execution interval of I48.	0.05 to 1000ms (in units of 0.05ms)	5.00ms
	I49*1	Sets the execution interval of I49.		1.00ms

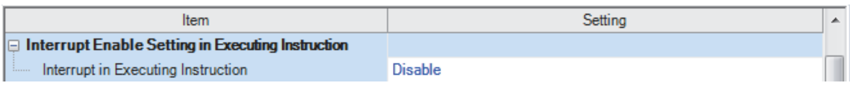
*1 This item is not displayed on the R00CPU, R01CPU, and R02CPU.

Interrupt enabled during instruction execution

Sets whether or not to enable or disable execution of an interrupt program during execution of an instruction. By enabling the interrupt during instruction execution, an interrupt can occur even while an instruction with a long processing time is being executed, resulting in higher interrupt accuracy.

[CPU Parameter] ⇨ [Interrupt Settings] ⇨ [Interrupt Enable Setting in Executing Instruction]

Window



Displayed items

Item	Description	Setting range	Default
Interrupt in Executing Instruction	Sets whether or not to enable or disable execution of an interrupt program during execution of an instruction. For the precautions with "Enable" set, refer to the precautions for an interrupt program. (☞ Page 93 Precautions)	• Disable • Enable	Disable

Processing at interrupt program startup

The processing shown below is performed when the interrupt program starts.

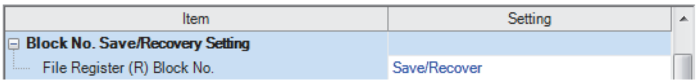
- Saving/restoring of the file register (R) block number
- Saving/restoring of the index register (Z, LZ)

Saving/restoring of the file register (R) block number

When an interrupt program starts, the block number of the file register (R) of the running program is saved and passed to the interrupt program. Also, when the interrupt program finishes, the saved block number is restored to the running program.

[CPU Parameter] ⇨ [Interrupt Settings] ⇨ [Block No. Save/Recovery Setting]

Window



Displayed items

Item	Description	Setting range	Default
File Register (R) Block No.	Sets whether or not the block number of the file register (R) is saved/restored when an interrupt program is being executed.	• Not Saved/Recovered • Save/Recover	Save/Recover

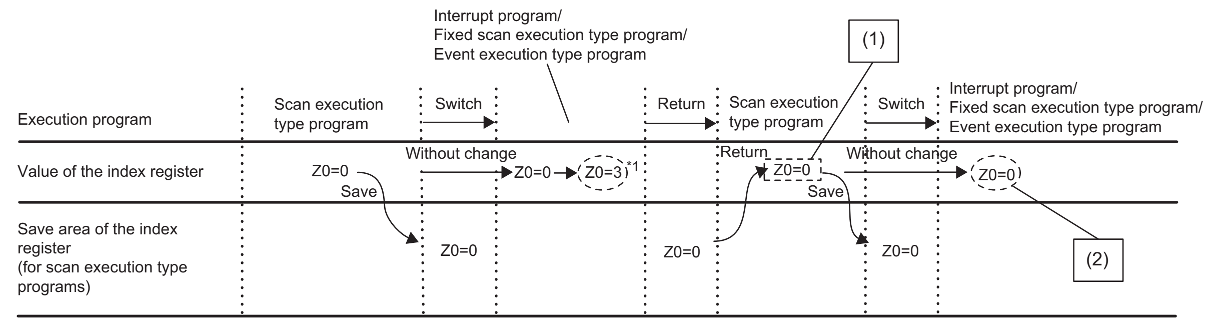
Point

When the block number of the file register (R) is not changed during execution of an interrupt program, enabling "Not Saved/Recovered" can reduce the startup time and the termination time of the interrupt program (☞ Page 1041 Overhead time when executing the interrupt program). To change the block number of the file register (R) while "Not Saved/Recovered" is enabled, the block number of that file register (R) must be saved/restored by the program.

Saving/restoring of the index register (Z, LZ)

When an interrupt program starts, the value of the index register (Z, LZ) of the running program is saved. When the interrupt program finishes, and the saved value is restored to the running program. Note that when an interrupt program starts, the local index register (Z, LZ) is not switched to the different one. When the local index register (Z, LZ) is used for the interrupt program/fixed scan execution type program/event execution type program which uses occurrence of an interrupt as a trigger, the register which has been used for the previous program is continuously used. Thus, the local index register (Z, LZ) cannot be used independently.

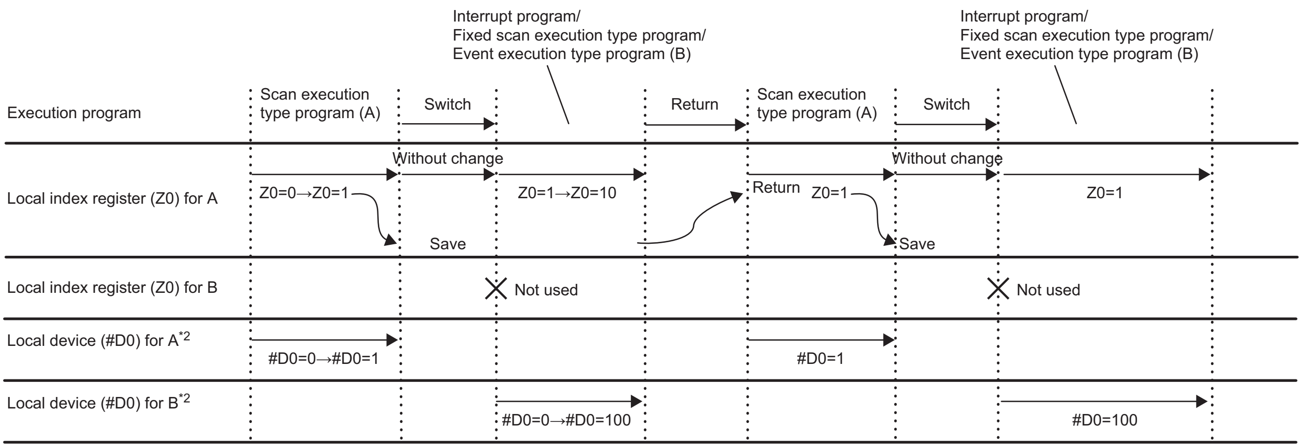
- Saving/restoring of the index register



(1) This value can be used as a value specific to the scan execution type program. (Using the value in an interrupt program is not needed to be concerned.)

(2) This value can be used only at the moment an interrupt program, fixed scan execution type program, or event execution type program is executed. (The value cannot be used continuously.)

- Operations of the local index register and local device

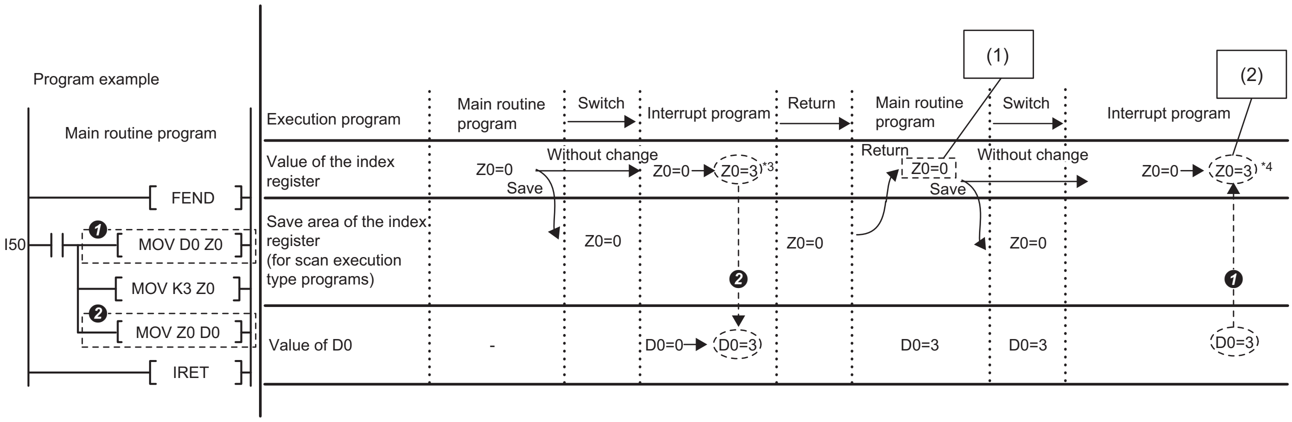


*1 The value of Z0 is changed by the interrupt program, fixed scan execution type program, or event execution type program.

*2 For when SM777 is on (when using the local device of the program file of storage location)

Point

If the value of the index register used for the interrupt program is continuously used for the next interrupt program, the value of the index register for the interrupt program must be saved or restored. Create a program to add the MOV instruction and the ZPUSH/ZPOP instruction.

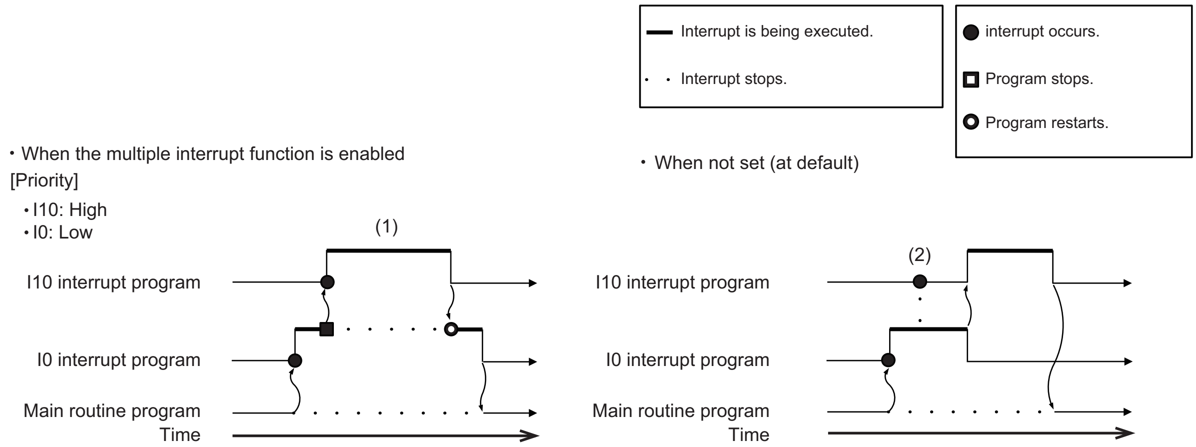


*3 A value in Z0 changes at MOV D0 Z0.
*4 Z is zero when the program ❶ and ❷ are not added.

(1)This value can be used as a value specific to the main routine program.
(2)This value can be used as a value specific to the interrupt program. (This value can be used continuously.)

Multiple interrupt function

When a new interrupt triggered by another factor occurs during execution of an interrupt program, the running program will be suspended if its priority is lower than that of the new interrupt. A program with higher priority is executed based on the set priority whenever its execution condition is satisfied.



(1) A high-priority interrupt is executed by interrupting a low-priority interrupt.
(2) Even if a high-priority interrupt occurs, it enters the waiting status until the interrupt in execution is completed.

Interrupt priority

If the interrupt priority of a program for which its execution condition has been satisfied is higher than that of the running program, the programs are executed in accordance with their interrupt priority. If the interrupt priority of the new program is the same or lower, it enters the waiting status until the running program finishes. The interrupt priorities 1 to 4 listed below cannot be changed, whereas the interrupt priorities 5 to 8 can. (☞ Page 90 Interrupt priority setting)

Interrupt priority		Interrupt factor (Interrupt pointer)	Execution sequence at simultaneous occurrence	Changeability
High ↑ ↓ Low	1	High-speed interrupt by internal timer 1 (I49)	—	Unchangeable (Fixed)
	2	High-speed interrupt by internal timer 2 (I48)	—	
	3	Inter-module synchronous interrupt (I44), Multiple CPU synchronous interrupt (I45)	I45 → I44	
	4	Interrupt by internal timer (I28 to I31), Fixed scan execution type program	I31 → I30 → I29 → I28 → Fixed scan execution type program	
	5 to 8	Interrupt from module (I0 to I15, I50 to I1023)	I0 ··· → ··· I1023	Changeable

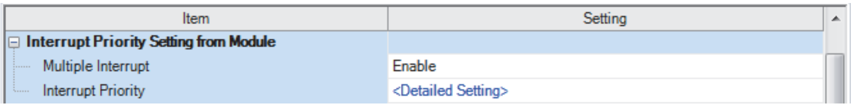
Interrupt priority setting

The interrupt priority (5 to 8) of interrupts from modules can be changed.

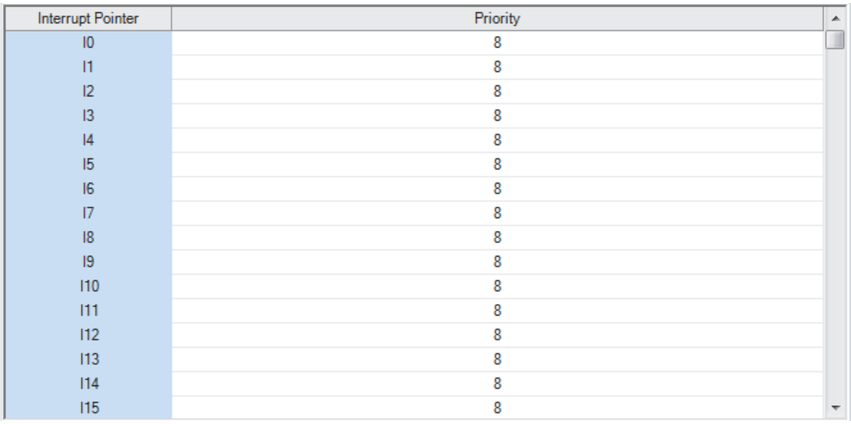
[CPU Parameter] ⇨ [Interrupt Settings] ⇨ [Interrupt Priority Setting from Module]

Operating procedure

"Interrupt Settings" window



"Detailed Setting" window



1. Set "Multiple Interrupt" to "Enable" on the "Interrupt Settings" window, and click "Detailed Setting".

2. Change the priority of each interrupt pointer.

Displayed items

Item		Description	Setting range	Default
Multiple Interrupt		Sets whether or not to enable multiple interrupt.	• Disable • Enable	Disable
Interrupt Priority	Detailed Setting	Sets the priority of the interrupt pointers I0 to I15 and I50 to I1023.	5 to 8	8

Disabling/enabling interrupts with a specified or lower priority

Interrupts with a priority equal or lower than that specified by the DI or EI instruction can be disabled or enabled even when multiple interrupts are present.

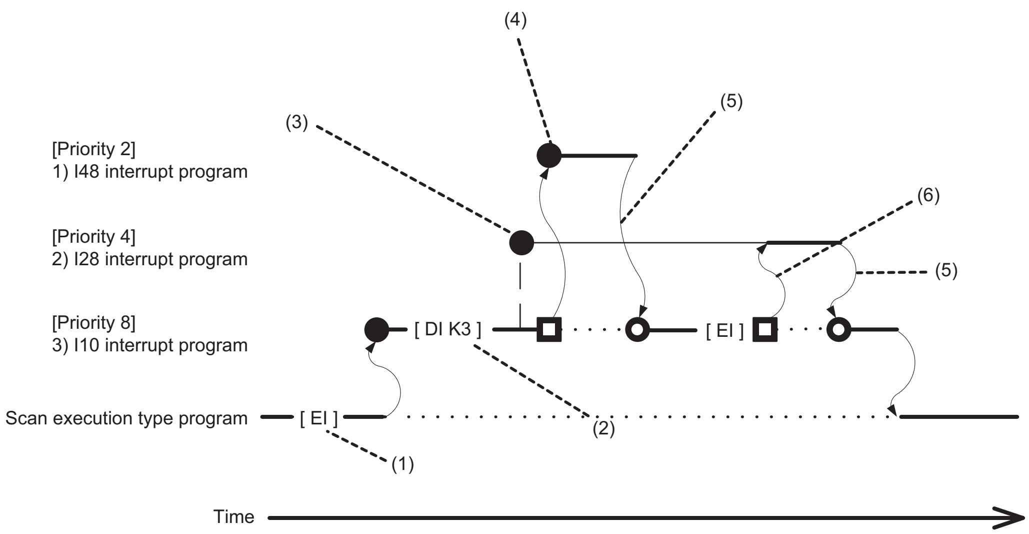
Ex.

Order of interrupt occurrence: ❸→❷→❶

Order of interrupt execution: ❸→❶→❷

Order of interrupt completion: ❸→❶→❷

Priority		Interrupt pointer
High ↑ ↓ Low	1	I49
	2	I48
	3	I44, I45
	4	I28, I29, I30, I31
	5	I101
	6	I0, I50, I100
	7	I1020
	8	An interrupt pointer among I50 to I1023 with the priority other than 5 to 7



(1) Interrupt is enabled.
(2) Interrupts with priority 3 to 8 are disabled.
(3) I28 is not executed because interrupts with priority 3 to 8 are disabled.
(4) I48 is executed because its interrupt priority is higher.
(5) Return from interrupt. Execution of the interrupted I10 resumes.
(6) High-priority interrupt I28 is executed because interrupts with priority 3 to 8 have been enabled.

Point

Disabled interrupt priorities and the current interrupt priority can be checked in SD758 (Interrupt disabling for each priority setting value) and SD757 (Current interrupt priority) respectively.

Multiple interrupt execution sequence

When multiple interrupts occur, the interrupt program with the highest priority is executed. Then, the interrupt program with the highest priority among those interrupted and in waiting status as a result of interrupts will be executed next when the previous is finished.

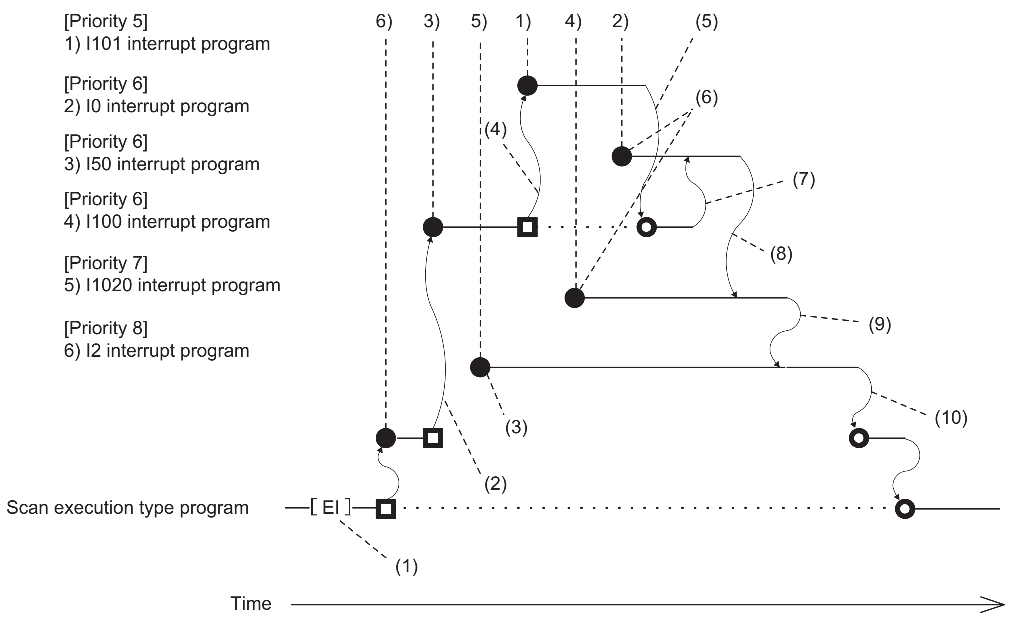
Ex.

Order of interrupt occurrence: ❻→❸→❺→❶→❹→❷

Order of interrupt execution: ❻→❸→❶→❷→❹→❺

Order of interrupt completion: ❶→❸→❷→❹→❺→❻

Priority		Interrupt pointer
High ↑ ↓ Low	1	I49
	2	I48
	3	I44, I45
	4	I28, I29, I30, I31
	5	I101
	6	I0, I50, I100
	7	I1020
	8	An interrupt pointer among I50 to I1023 with the priority other than 5 to 7



(1) Interrupt is enabled.
(2) I50 is executed because its interrupt priority is higher.
(3) Enters waiting status because its interrupt priority is lower.
(4) I101 is executed because its interrupt priority is higher.
(5) Return from interrupt. Execution of the interrupted I50 resumes.
(6) Enters the waiting status until the execution of I50 completes because its interrupt priority is the same as that of I50 by setting.
(7) I0 is executed before I100 because its interrupt pointer is smaller.
(8) Return from interrupt. I1020 is executed because its interrupt priority is higher than those of I1020 and I0.
(9) Return from interrupt. I1020 is executed because its interrupt priority is higher than that of I2.
(10)Return from interrupt. Execution of the interrupted I2 resumes.

Precautions

The precautions for the interrupt program are mentioned below.

Restrictions on program creation

- The PLS/PLF instruction performs OFF processing in the scan after the instruction execution. The device turned on remains on until the interrupt program starts again and the instruction is executed.
- The timer (T, ST) and the counter (C) cannot be used for interrupt programs.

Data inconsistency

When the interrupt during instruction execution is set to "Enable", processing of an instruction may be suspended and the interrupt program may be executed. Accordingly, if the interrupted program and the interrupt program both use the same device, data may become inconsistent. Take the following preventive measures.

- Use the DI instruction to disable the interrupt for an instruction that causes inconsistency when interrupted.
- In an interrupt program, before accessing the device shared by overlapping programs, transfer data to another device in batch to use, and then write back to the shared device in batch.
- When using the bit data, be careful so that the interrupted program and the interrupt program shall not use the same bit data.

If the interrupt accuracy does not increase

If the interrupt accuracy does not increase even when the interrupt during instruction execution is set to "Enable", it may increase by taking the following measures:

- Limit each character string used in a program to 32 characters or less in length.
- Decrease the multiplex interrupts.
- Reduce the number of times to access the Q series module or the number of points.
- Do not access the Q series module from multiple CPU modules when the multiple CPU system is configured.

Monitoring the interrupt program execution time

The execution time of the interrupt program can be monitored with the interrupt program in which the inter-module synchronous interrupt (I44) and multiple CPU synchronous interrupt (I45) are used. (☞ Page 139 Error detection setting)

When the start-up of an interrupt program is delayed

If the execution interval of the interrupt program is short, the start-up of the interrupt program may be delayed when the instruction with a long processing time is executed, the online ladder block change is performed, or a Q series module is accessed simultaneously from multiple CPU modules using instructions in the multiple CPU system configuration. Therefore, when the execution time is monitored for the interrupt program that uses the inter-module synchronous interrupt (I44) and multiple CPU synchronous interrupt (I45), an error may be detected in the CPU module. (☞ Page 139 Error detection setting)

Consideration when using the data logging function and the recording function

When occurrence of an interrupt is specified as the data collection condition of data logging or when the sampling method of the recording function is set to use the trigger instruction in an interrupt program, set the fixed scan interval of the interrupt program, taking into consideration the processing time for these functions. (☞ Page 251 Using together with interrupt programs)

Interrupt processing with FB/FUN

FB/FUN consists of multiple instructions. When an interrupt occurs during execution of the FB/FUN, the execution will be suspended and an interrupt program will be executed even though "Interrupt Enable Setting in Executing Instruction" of the CPU parameter has been set to "Disable".

To disable the execution of the interrupt program during execution of FB/FUN, use the DI instruction to disable the interrupt before the FB/FUN is called and use the EI instruction to enable the interrupt after execution of the FB/FUN.

In ST programs and FBD/LD programs, each of the following elements and program elements consists of multiple instructions.

- Instruction (manufacturer-defined FB/FUN)
- Arithmetic operations in ST programs (Example: D0 := D1 + D2 - D3;)
- Logical operations in ST programs (Example: M0 := M1 AND M2 OR M3;)

To disable the interrupt during execution of an ST program or an FBD/LD program, use the DI instruction and the EI instruction before and after each element or program element regardless of the setting of "Interrupt Enable Setting in Executing Instruction".

Extended interrupt program processing time

If interrupts during instruction execution are set to "Enable" in the interrupt enabled during instruction execution setting, and the interrupt program is executed when the following conditions are met, the interrupt program processing time will take longer. (☞ Page 85 Interrupt enabled during instruction execution)

- If accessing the refresh data register (RD), a module label, or labels, or using FB/ST/FBD in the interrupt program while executing an instruction involving the transfer of a significant amount of data to the device/label memory, or during refresh processing.
- If accessing the device/label memory in the interrupt program when using FB/ST/FBD while executing an instruction involving the transfer of a significant amount of data to the refresh data register (RD), a module label, or a label, or during refresh processing to the refresh data register (RD) or a module label.

Delay in interrupt programs

Completing low-priority interrupt programs may take time if high-priority interrupt programs are executed frequently when the multiple interrupt function is used. Adjust the processing of high-priority interrupt programs to allow execution of low-priority interrupts.

Devices and labels used by interrupt programs

Do not use the same global devices and labels in multiple interrupt programs because it may cause data inconsistency when the multiple interrupt function is used.

2 CPU MODULE OPERATION PROCESSING

Here is a list of the operating status of the CPU module:

- RUN state
- STOP state
- PAUSE state

2.1 Operation Processing by Operating Status

This displays operation processing according to the operating status of the CPU module.

Operation processing in RUN state

In RUN state, the program operation is repeatedly performed in the following order: Step 0 → END/FEND instruction → Step 0.

■Output at the time of entering RUN state

Depending on the setting of output (Y) at the time of change from STOP state to RUN state, either of the following is output. (☞ Page 97 Output mode at operating status change (STOP to RUN))

- Status of the output (Y) that was saved during the STOP state
- Operation result after the completion of one scan

The device memory other than the output (Y) holds the status immediately before the RUN state. However, if a device initial value is set up, this initial value is set. In addition, the local devices are cleared.

Operation processing in STOP state

In STOP state, the operation is stopped (state with a stop error included).

■Output at the time of entering STOP state

All points are turned off with the output status immediately before the STOP state saved. The device memory other than the output (Y) holds the status immediately before the STOP state.

Operation processing in PAUSE state

In PAUSE state, the program operation is suspended holding the status of the output and the device memory after the completion of one scan.

2.2 Operation Processing When Operating Status Is Changed

This displays operation processing when the operating status of the CPU module is changed.

CPU module operating status	CPU module processing			
	Program	External output	Device memory	
			Other than Y	Y
STOP → RUN	Executes the program from the start.	Determines the status depending on the setting of "Output Mode Setting at STOP to RUN" in the CPU parameter. (☞ Page 97 Output mode at operating status change (STOP to RUN))	Retains the device memory status immediately before the RUN state. However, if a device/label initial value has been set, this initial value is set.[*1] Clears local devices.	Determines the status depending on the setting of "Output Mode Setting at STOP to RUN" in the CPU parameter. (☞ Page 97 Output mode at operating status change (STOP to RUN))
RUN → STOP	Executes the program up to the END instruction and then stops.	Saves the output (Y) status immediately before the STOP state, and all points turn off.	Retains the device memory status immediately before the STOP state.	Saves the output (Y) status immediately before the STOP state, and all points turn off.
RUN → PAUSE	Stops the operation after the execution of one scan.	Retains the output (Y) status immediately before the PAUSE state.	Retains the device memory status immediately before the PAUSE state.	Retains the output (Y) status immediately before the PAUSE state.
PAUSE → RUN	Executes the program from the start.	Retains the output (Y) status immediately before the RUN state.	Retains the device memory status immediately before the RUN state. Clears local devices.	Retains the output (Y) status immediately before the RUN state.
PAUSE → STOP	Operation remains stopped.	Saves the output (Y) status immediately before the STOP state, and all points turn off.	Retains the device memory status immediately before the STOP state.	Saves the output (Y) status immediately before the STOP state, and all points turn off.
STOP → PAUSE	Operation remains stopped.	Retains the output (Y) status immediately before the PAUSE state.	Retains the device memory status immediately before the PAUSE state.	Retains the output (Y) status immediately before the PAUSE state.

*1 For details on the initial label value setting for the Process CPU and SIL2 Process CPU, refer to the following.
☞ Page 364 LABEL INITIALIZATION FUNCTION

Output mode at operating status change (STOP to RUN)

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using a SIL2 Process CPU, refer to the following as well.

☞ Page 704 FUNCTIONS

- When using a Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

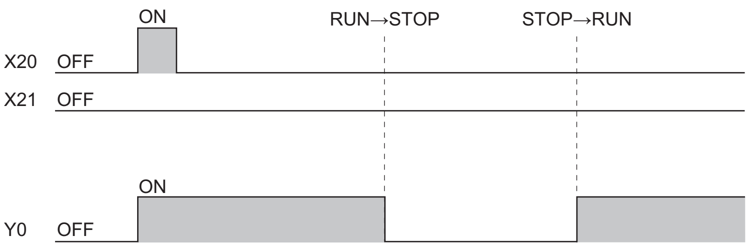
When the operating status changes from RUN to STOP, for example, the CPU module internally stores the status of the outputs (Y) to turn them all off.

Operation when the operating status is changed from STOP to RUN

Select whether or not to resume from the previous output status when the operation status of the CPU module is changed from STOP to RUN by using a holding circuit.

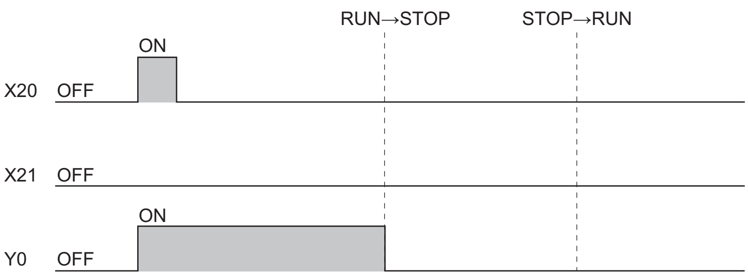
■Outputting the output (Y) status before STOP

The program operation is resumed after outputting the output (Y) status immediately before the operating status of the CPU module is changed to the STOP state. If the output (Y) is forcibly turned on in the STOP state, the status prior to the STOP state is output. If the output (Y) was off before entering the STOP state, the status when it was on is not retained. (In the following figure, X20 represents the start button, X21 represents the stop button, and Y represents an output.)



■Clearing the output (Y) (output after one scan)

All outputs are turned off, and the output (Y) is output after the program operations are executed. If the output (Y) is forcibly turned on when the operating state is STOP, the ON state is retained. (In the following figure, X20 represents the start button, X21 represents the stop button, and Y represents an output.)

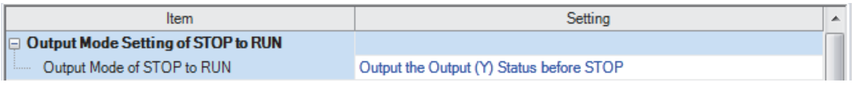


Setting the output mode

Set the mode in "Output Mode Setting of STOP to RUN".

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Output Mode Setting of STOP to RUN]

Window



Displayed items

Item	Description	Setting range	Default
Output Mode of STOP to RUN	Set the operation of the output (Y) when the operating status is changed from STOP to RUN.	• Output the Output (Y) Status before STOP • Clear (Output is 1 scan later) the Output (Y)	Output the Output (Y) Status before STOP

2.3 Operation Processing at Momentary Power Failure

When an input power supply voltage supplied to the power supply module falls below the specified range, a momentary power failure is detected and the following operation processing is performed.*1

*1 In the redundant extension base unit configuration, refer to the following as well.
☞ Page 460 Operation Processing at Momentary Power Failure

Momentary power failure not exceeding the allowable momentary power failure time

If a momentary power failure occurs, the event history is registered to suspend the operation processing. Note however that measurement of the timer device continues. In addition, the output status is held.

■When the momentary power failure is cleared

Once the momentary power failure is cleared, the operation processing continues.

■Measurement of the watchdog timer (WDT) at a momentary power failure

Even if the operation is suspended due to a momentary power failure, measurement of the watchdog timer continues. For example, assuming that the monitoring time of scan time is set to 200ms and the scan time is 190ms, a momentary power failure of 15ms causes a WDT error. (☞ Page 136 Scan time monitoring time setting)

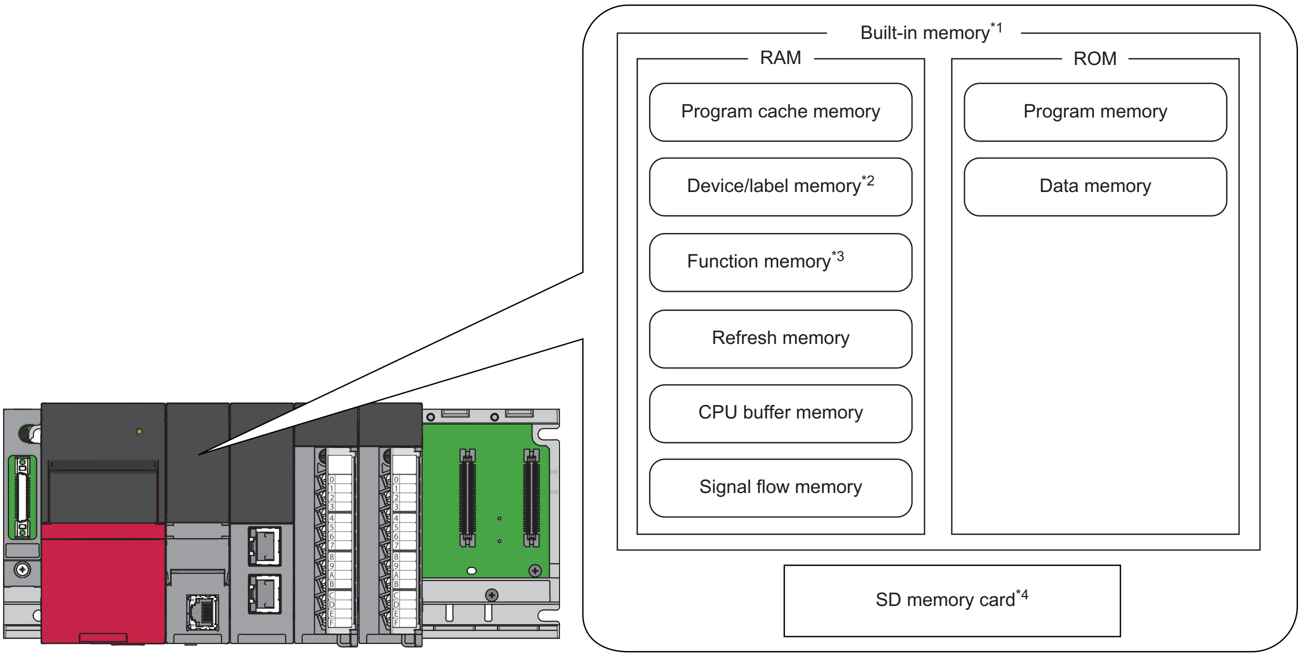
Momentary power failure exceeding the allowable momentary power failure time

This case results in initial start, the same operation processing as when the CPU module is powered on or is reset.

3 MEMORY CONFIGURATION OF THE CPU MODULE

3.1 Memory Configuration

The following shows the memory configuration of the CPU module.



*1 The built-in memory is a generic term of the memory built in the CPU module.

*2 The R00CPU, R01CPU, and R02CPU retain the data in each device/label with latch setting without a battery during power failure. (☞ Page 436 Latch with Battery)
The other programmable controller CPUs retain the data in each device/label with latch setting without a battery during power failure by inserting a battery-less option cassette to the CPU module. (☞ Page 443 Latch with Battery-less Option Cassette)

*3 This memory can be used in the R04CPU, R08CPU, R16CPU, R32CPU, R120CPU, R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, and R120ENCPU.

*4 An SD memory card cannot be used in the R00CPU.

Point

- The usage of the memory can be checked from the engineering tool. (📖 GX Works3 Operating Manual)
- The number of writes for the program memory and data memory is limited to 100000 times. When using the function that the target file is written, pay attention to the number of writes. (☞ Page 112 File types and storage memory) The current number of writes for the program memory and data memory can be checked in the special register (SD630/SD631 and SD634/SD635). (☞ Page 988 Drive information)

Program memory/program cache memory

The program memory and program cache memory store necessary programs for the CPU module to perform operations. At the following timing, data in the program memory is transferred to the program cache memory*1 and an operation is performed.

*1 This memory is used for program operations.

- Powered-on
- Reset operation

Application

The program memory and program cache memory store execution programs of program files and FB (function block) files. Execution programs include instruction codes, statements, and notes.

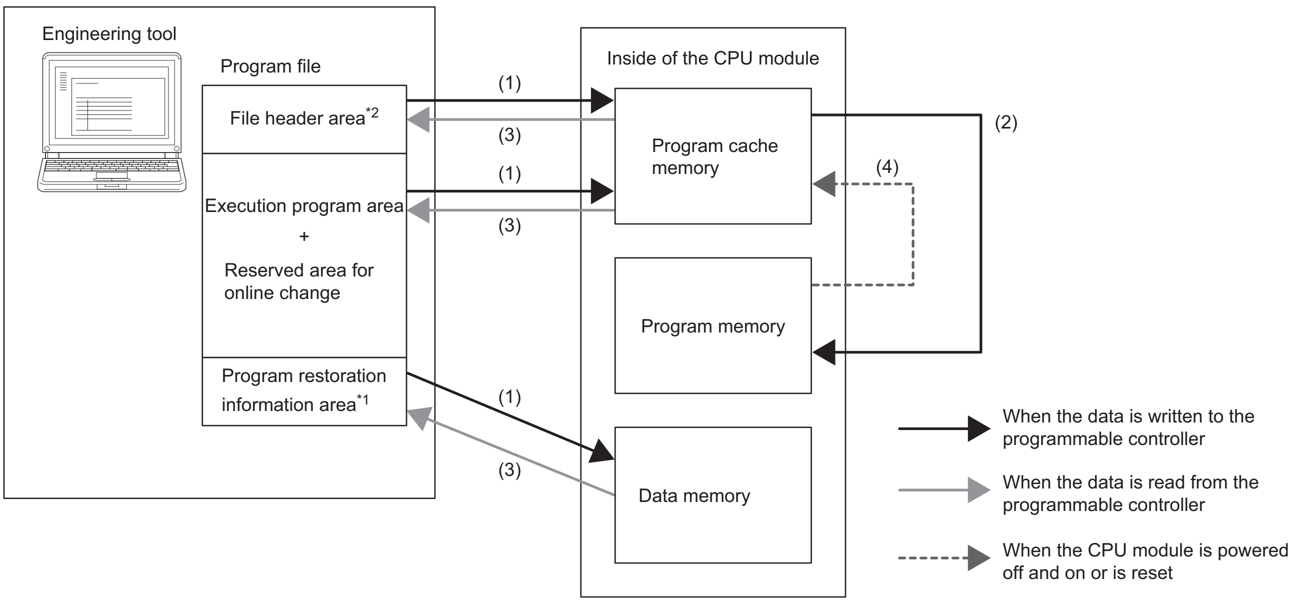
Data to be allocated

The file header area*1 and the execution program area (including reserved area for online change) that exist in program files and FB files are allocated to the program memory.

*1 Depending on the model and firmware version of the CPU module, the file header area is allocated to the data memory. (☞ Page 102 Destination of the file header area)

Data allocation and procedure of read/write operations

The following figure shows the data allocation of the program memory and program cache memory and the procedure of read/write operations from/to the programmable controller.



*1 Program restoration information includes the information required to read a program from the programmable controller with the engineering tool.

(1) When data is written to the programmable controller, the file header, execution program, and reserved area for online change are written to the program cache memory, and program restoration information is written to the data memory.

(2) After being written to the program cache memory, the data is automatically transferred to the program memory.

(3) When data is read from the programmable controller, the file header, execution program, and reserved area for online change are read from the program memory, and the program restoration information is read from the data memory.

(4) After the CPU module is powered off and on or is reset, the data in the program memory is transferred to the program cache memory and operations are executed.

*2 Depending on the model and firmware version of the CPU module, the destination of the file header area is the data memory. (☞ Page 102 Destination of the file header area)

Point

For read/write operations from/to the programmable controller, refer to the following.

📖 GX Works3 Operating Manual

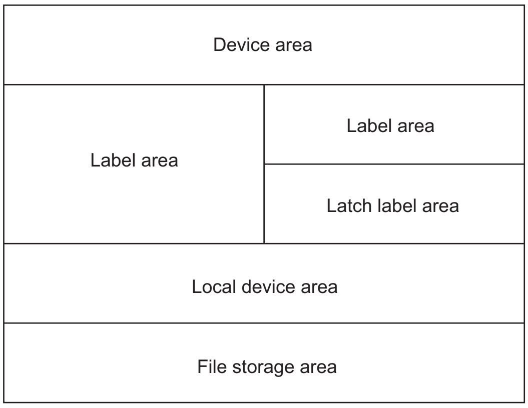
■Destination of the file header area

For the following CPU modules, the destination of the file header area is the data memory.

CPU module		Firmware version
R04CPU, R08CPU, R16CPU, R32CPU, R120CPU R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, R120ENCPU		"30" or earlier
Process CPU		"12" or earlier
Safety CPU	Standard program	"12" or earlier
	Safety program	"16" or earlier

Device/label memory

The device/label memory has the following areas.



The capacity of each area can be changed with the setting of the CPU parameter. (☞ Page 103 Device/label memory area setting)

Data to be allocated

The following table lists the data allocated to each area.

Area		Application
Device area		User device
Label area	Label area	Global label and local label
	Latch label area	Global label and local label with latch specified
Local device area		Local device (excluding index register)
File storage area		File register file and other data*1

*1 File register files which are stored in the area for storing file register files can be written or read in file unit.

Point

Free space of areas can be checked in "Device/Label Memory Area Capacity Setting". (☞ Page 103 Device/label memory area setting)

Device/label memory area setting

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using a SIL2 Process CPU, refer to the following as well.

☞ Page 704 FUNCTIONS

- When using a Safety CPU, refer to the following as well.

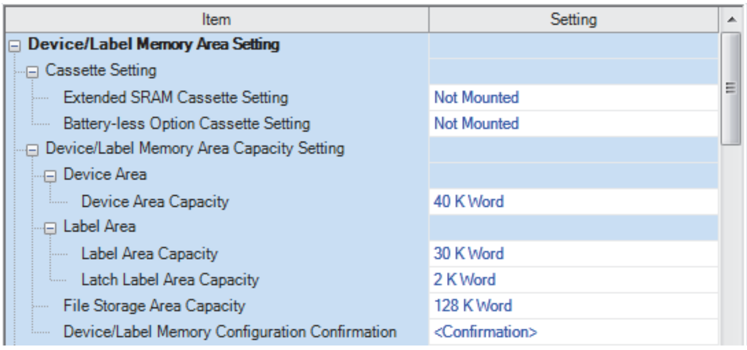
☞ Page 621 FUNCTIONS

The capacity of each data area allocated within the device/label memory can be changed. (☞ Page 102 Device/label memory)

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Setting]

Operating procedure

"Device/Label Memory Area Setting" window



1. Set whether to use an extended SRAM cassette or battery-less option cassette in "Cassette Setting".
2. Set the capacity of each area in "Device/Label Memory Area Capacity Setting".

Displayed items

Item			Description	Setting range	Default
Cassette Setting*3	Extended SRAM Cassette Setting		When using an extended SRAM cassette, select the capacity of the cassette.	• Not Mounted • 1MB • 2MB • 4MB • 8MB • 16MB*1	Not Mounted
	Battery-less Option Cassette Setting*1*4		Set whether to insert a battery-less option cassette.	• Not Mounted • Mounted	Not Mounted
Device/Label Memory Area Capacity Setting	Device Area	Device Area Capacity	Sets the capacity of the device area to be used for global devices.	☞ Page 104 Default capacity of each area	☞ Page 104 Default capacity of each area
	Label Area	Label Area Capacity	Sets the capacity of the label area to be used for global labels and local labels.		
		Latch Label Area Capacity	Sets the capacity of the latch label area to be used for latch type labels.		
	File Storage Area Capacity*2		Sets the capacity of the file storage area to be used for storing register files and others.		
	Device/Label Memory Area Capacity Setting		Shows the device/label memory configuration.		

*1 If using the items with the programmable controller CPU, check the versions of the CPU module and engineering tool. (☞ Page 1139 Added and Enhanced Functions)

*2 When using 16MB extended SRAM cassettes (NZ2MC-16MBS), precautions apply when setting the file register capacity. (☞ Page 406 Setting file registers)

*3 This item is not displayed on the R00CPU, R01CPU, and R02CPU.

*4 This item is not displayed for CPU modules for which the latch with the battery-less option cassette cannot be used.

Point

- Please note that the total of the capacity of each area (including the capacity of the local device area) should not exceed the capacity of the device/label memory (MELSEC iQ-R CPU Module User's Manual (Startup)). The total of the capacity of each area can be checked in "Device/Label Memory Area Capacity Setting".
- When the capacity of the file storage area is changed, files (file register files) stored in the area are deleted, and therefore it is required to write file register files. Also, when files are stored in the CPU module, they must be read to other place before the capacity of the file storage area is changed and written back to the CPU module after the change.

Default capacity of each area

The default capacity of each area is as follows.

Item	R00CPU, R01CPU, R02CPU	R04CPU, R04ENCPU	R08CPU, R08ENCPU, R08PCPU	R16CPU, R16ENCPU, R16PCPU	R32CPU, R32ENCPU, R32PCPU	R120CPU, R120ENCPU, R120PCPU
Device area	30K words	40K words	40K words	40K words	40K words	40K words
Label area	30K words	30K words	40K words	50K words	90K words	110K words
Latch label area	2K words	2K words	2K words	2K words	4K words	4K words
Local device area	0K word	0K word	0K word	0K word	0K word	0K word
File storage area	64K words	128K words	512K words	768K words	1024K words	1536K words

Point

The capacity of the local device area to be set is obtained by deducting the total capacity of the device area, label area, latch label area, and file storage area from the total capacity of the device/label memory of each model. However, even if the total capacity of the device area and label area is smaller than the following, the capacity cannot be assigned to the local device area. (The area which is smaller than the following is the area not used.)

- R00CPU, R01CPU, R02CPU: 30K words
- R04CPU, R04ENCPU: 40K words
- R08CPU, R08ENCPU, R08PCPU: 50K words
- R16CPU, R16ENCPU, R16PCPU: 60K words
- R32CPU, R32ENCPU, R32PCPU: 70K words
- R120CPU, R120ENCPU, R120PCPU: 90K words

The setting range of the capacity of each area

The following tables list the setting range of the capacity of each area on the device/label memory.*1

The availability of an extended SRAM cassette varies depending on the CPU module used. For the availability, check the performance specifications of the extended SRAM cassette. (MELSEC iQ-R CPU Module User's Manual (Startup))

*1 The rest of other areas are automatically set as the capacity of the local device area.

■R00CPU, R01CPU, R02CPU

Area	Setting range of capacity of each area
Device area	1 to 126K words
Label area	0 to 125K words
Latch label area	0 to 96K words
File storage area	0 to 96K words

■R04CPU, R04ENCPU

Area	Setting range of capacity of each area					
	Without an extended SRAM cassette	With an extended SRAM cassette (1MB)	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (4MB)	With an extended SRAM cassette (8MB)	With an extended SRAM cassette (16MB)
Device area	2 to 200K words	2 to 712K words	2 to 1224K words	2 to 2248K words	2 to 4296K words	2 to 8392K words
Label area	0 to 198K words	0 to 710K words	0 to 1222K words	0 to 2246K words	0 to 4294K words	0 to 8390K words
Latch label area	0 to 160K words	0 to 672K words	0 to 1184K words	0 to 2208K words	0 to 4256K words	0 to 8352K words
File storage area	0 to 160K words	0 to 672K words	0 to 1184K words	0 to 2208K words	0 to 4256K words	0 to 8352K words

3

■R08CPU, R08ENCPU, R08PCPU

Area	Setting range of capacity of each area					
	Without an extended SRAM cassette	With an extended SRAM cassette (1MB)	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (4MB)	With an extended SRAM cassette (8MB)	With an extended SRAM cassette (16MB)
Device area	2 to 594K words	2 to 1106K words	2 to 1618K words	2 to 2642K words	2 to 4690K words	2 to 8786K words
Label area	0 to 592K words	0 to 1104K words	0 to 1616K words	0 to 2640K words	0 to 4688K words	0 to 8784K words
Latch label area	0 to 544K words	0 to 1056K words	0 to 1568K words	0 to 2592K words	0 to 4640K words	0 to 8736K words
File storage area	0 to 544K words	0 to 1056K words	0 to 1568K words	0 to 2592K words	0 to 4640K words	0 to 8736K words

■R16CPU, R16ENCPU, R16PCPU

Area	Setting range of capacity of each area					
	Without an extended SRAM cassette	With an extended SRAM cassette (1MB)	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (4MB)	With an extended SRAM cassette (8MB)	With an extended SRAM cassette (16MB)
Device area	2 to 860K words	2 to 1372K words	2 to 1884K words	2 to 2908K words	2 to 4956K words	2 to 9052K words
Label area	0 to 858K words	0 to 1370K words	0 to 1882K words	0 to 2906K words	0 to 4954K words	0 to 9050K words
Latch label area	0 to 800K words	0 to 1312K words	0 to 1824K words	0 to 2848K words	0 to 4896K words	0 to 8992K words
File storage area	0 to 800K words	0 to 1312K words	0 to 1824K words	0 to 2848K words	0 to 4896K words	0 to 8992K words

■R32CPU, R32ENCPU, R32PCPU

Area	Setting range of capacity of each area					
	Without an extended SRAM cassette	With an extended SRAM cassette (1MB)	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (4MB)	With an extended SRAM cassette (8MB)	With an extended SRAM cassette (16MB)
Device area	2 to 1158K words	2 to 1670K words	2 to 2182K words	2 to 3206K words	2 to 5254K words	2 to 9350K words
Label area	0 to 1156K words	0 to 1668K words	0 to 2180K words	0 to 3204K words	0 to 5252K words	0 to 9348K words
Latch label area	0 to 1088K words	0 to 1600K words	0 to 2112K words	0 to 3136K words	0 to 5184K words	0 to 9280K words
File storage area	0 to 1088K words	0 to 1600K words	0 to 2112K words	0 to 3136K words	0 to 5184K words	0 to 9280K words

■R120CPU, R120ENCPU, R120PCPU

Area	Setting range of capacity of each area					
	Without an extended SRAM cassette	With an extended SRAM cassette (1MB)	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (4MB)	With an extended SRAM cassette (8MB)	With an extended SRAM cassette (16MB)
Device area	2 to 1690K words	2 to 2202K words	2 to 2714K words	2 to 3738K words	2 to 5786K words	2 to 9882K words
Label area	0 to 1688K words	0 to 2200K words	0 to 2712K words	0 to 3736K words	0 to 5784K words	0 to 9880K words
Latch label area	0 to 1600K words	0 to 2112K words	0 to 2624K words	0 to 3648K words	0 to 5696K words	0 to 9792K words
File storage area	0 to 1600K words	0 to 2112K words	0 to 2624K words	0 to 3648K words	0 to 5696K words	0 to 9792K words

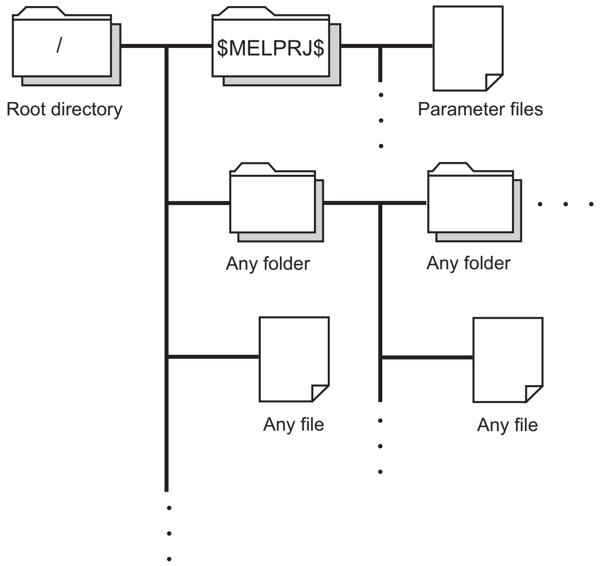
Point

To check which CPU modules can be used with extended SRAM cassettes, refer to the following.

MELSEC iQ-R CPU Module User's Manual (Startup)

Data memory

This memory is used to store the parameter file, device comment file, and/or the user's folder/file. Data such as the parameter file and the device comment files written by the engineering tool is stored in the "$MELPRJ$" folder. The "$MELPRJ$" folder is created after memory initialization. Note that the "$MELPRJ$" folder cannot be deleted. (Folders under the "$MELPRJ$" folder can be deleted.)



Point

For details on how to create and delete user folders, refer to the following.
GX Works3 Operating Manual

Precautions

Do not delete or rename the folders or files being accessed or those being executed by a function.

Function memory

This memory is temporarily used for the specific function.
The following is the function that uses the function memory.

Function	Reference
Data logging function (when "CPU built-in memory (function memory)" is selected for "Data storage destination memory")	Page 199 DATA LOGGING FUNCTION

Also, data can be deleted or transferred to the data memory by using the special relay. (Page 943 Drive information)

Precautions

Do not delete or rename the folders or files being accessed or those being executed by a function.

Refresh memory

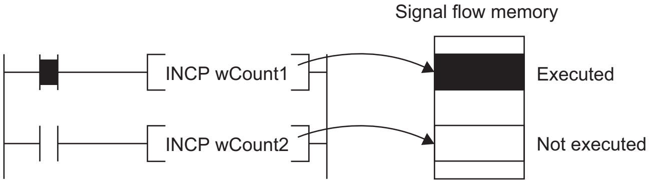
This memory is used to store data used to refresh communication with the intelligent function module. (Page 407 Refresh Data Register (RD))

CPU buffer memory

This memory is used by the Ethernet function or in data communication between multiple CPUs. (Page 1033 Buffer Memory)

Signal flow memory

This memory is used to memorize the execution status of the instruction in the last scan. The CPU module judges whether to execute a rising/falling edge execution instruction by referring to the signal flow memory.



The execution status of the last instruction is stored in the signal flow memory in two ways: executed or not executed. The instructions that refer to the signal flow memory judge whether to execute a rising/falling edge execution instruction depending on the input condition of the instruction and the execution status of the last instruction stored in the signal flow memory.

- For a program, the same number of areas as steps of the program are assigned to the signal flow memory (for program).
- For a function, the signal flow memory is not assigned since the instructions that refer to the last execution status of the signal flow memory cannot be used in the function.
- For a subroutine type function block, the same number of areas as instructions that refer to the signal flow memory in the function block are assigned to the signal flow memory (for FB). Different areas are assigned to each instance. When the macro type function block is called from the subroutine type function block, the areas including the ones used for the macro type function block are assigned.
- For a macro type function block, the same number of areas as the number of steps of the macro type function block are assigned to the signal flow memory (for program).

For instances of the function block, refer to the following.

MELSEC iQ-R Programming Manual (Program Design)

Point

The following shows the instructions that refer to the last execution status of the signal flow memory.

- Contact instruction: LDP, LDF, ANDP, ANDF, ORP, ORF, LDPI, LDFI, ANDPI, ANDFI, ORPI, and ORFI
- Association instruction: MEP, MEF
- Output instruction: SET F, RST F, PLS, PLF, FF
- Rising edge execution instruction: □P (such as INCP and MOVP), SP.□, ZP.□, GP.□, JP.□, DP.□, MP.□, UDCNT1, UDCNT2, TTMR, STMR, RAMPQ, SPD, PLSY, PWM, MTR, SORTD(_U), DSORTD(_U), LEDR, DUTY, LOGTRG, LOGTRGR, TIMCHK, HOURM, DHOURM, PID, XCALL, SCJ

SD memory card

This memory is used to store the folder/file created by a function using the SD memory card as well as the user's folder/file. The folder configuration is the same as the data memory. However, in the case of the SD memory card, the "$MELPRJ$" folder will be created when the SD memory card becomes available (when the SD memory card is mounted).

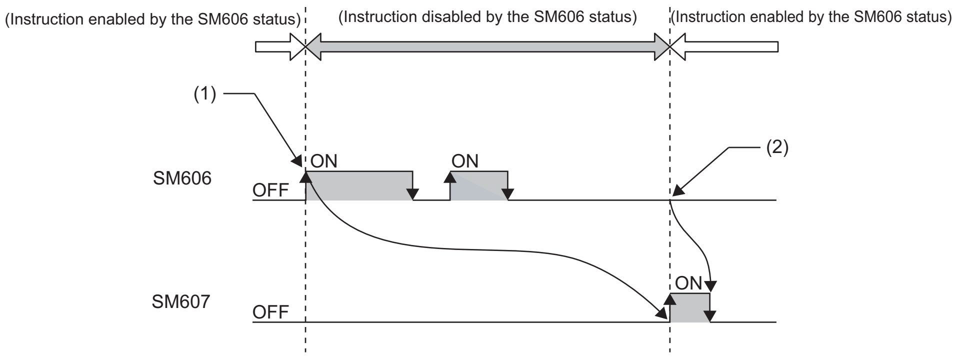
Point

For details on how to insert or remove the SD memory card, refer to the following.
(MELSEC iQ-R CPU Module User's Manual (Startup))

How to forcibly disable the SD memory card with a special relay

If the SD memory card is removed while the CPU module is powered on, data in the SD memory card may be corrupted. Use a special relay to forcibly disable access to the SD memory card without turning off the power.
The following shows how to forcibly disable the SD memory card with a special relay.

1. Turn on SM606 (SD memory card forced disable instruction). After an instruction by SM606, and until SM607 changes, an instruction by the ON/OFF state change of SM606 is disabled. Note that after SM607 changes, an instruction by the ON/OFF state change of SM606 is enabled, and operation is performed based on the ON/OFF state of SM606.



(1) Forced disable instruction by the ON state
(2) Instruction to cancel forced disable status by the OFF state enabled

2. Check that the CARD READY LED is off or SM607 (SD memory card forced disable status flag) is on.
3. Remove the SD memory card.

Point

After the SD memory card is disabled, to cancel the disabled status, insert the SD memory card again and power off and on or reset the CPU module.
When the SD memory card is inserted again, the CARD READY LED changes from flashing to on.

■Operation of each function accessing the SD memory card

Disabling the SD memory card affects the operation of each function accessing the SD memory card. For the functions shown below, the following table shows the operations when the SD memory card forced disable instruction is executed during access to the SD memory card and when access is made to the SD memory card after the SD memory card is disabled.

Function being executed	SD memory card forced disable instruction executed during access to the SD memory card	Access made to the SD memory card after the SD memory card is disabled
Boot operation	—	—
• Access to device comment or label in SD memory card • Device/label initial value operation at status change from STOP to RUN	The SD memory card becomes disabled after the executed function is completed.	An error occurs in the CPU module.*1
Access to SD memory card by engineering tool, SLMP, FTP function	The function is responded to with an error.	The function is responded to with an error.
Instruction to access SD memory card	The SD memory card becomes disabled after the instruction is completed.	The instruction is completed with an error.
CPU module data backup/restoration function	When the processing for the file being backed up/restored is completed, the SD memory card is disabled, the executed function is completed with an error, and the error cause is stored in the special register.	The error cause is stored in the special register.
iQ Sensor Solution data backup/restoration function	After the backup or restore operation is completed, the SD memory card is disabled.	The function is responded to with an error.
File transfer function (FTP client)	When the processing for the file during file transfer execution is completed, the SD memory card is disabled, the file transfer function instruction is completed with an error, and the error cause is stored in the completion status.	The file transfer function instruction is completed with an error, and the error cause is stored in the completion status.

*1 Same as the operation when the SD memory card is not inserted

Precautions

Do not delete or rename the folders or files being accessed or those being executed by a function.

■Precautions when forcibly disabling the SD memory card

- When forced disable operation with the SD CARD OFF button and forced disable operation by SM606 are executed together, the operation executed earlier is enabled, and the operation executed later is disabled. For example, after the SD memory card is forcibly disabled with the SD CARD OFF button, when SM606 is turned off without removing the SD memory card, the disabled status of the SD memory card can be canceled. However, after forcibly disabling the SD memory card with the SD CARD OFF button, even by removing the SD memory card, turning on SM606, and inserting the SD memory card, the SD memory card is not enabled. To enable the SD memory card, after it is inserted again, SM606 must be turned off.
- If this function is executed while an external device is writing a file to the SD memory card, writing of the file may fail. Cancel the SD memory card disable status, then write the file again.

3.2 File Size Unit in Memory

The minimum unit of capacity for storing a file in the memory is referred to as the file size unit (cluster size).

File size unit based on memory area

CPU module	File size unit			
	Program memory	Device/label memory	Data memory	Function memory
R00CPU, R01CPU, R02CPU	4 bytes	512 bytes	1024 bytes	—
R04CPU, R04ENCPU				8192/2048 bytes*1
R08CPU, R08ENCPU			2048 bytes	
R16CPU, R16ENCPU			4096 bytes	
R32CPU, R32ENCPU			8192 bytes	
R120CPU, R120ENCPU			16384 bytes	
R08PCPU	128 bytes		2048 bytes	—
R16PCPU			4096 bytes	
R32PCPU			8192 bytes	
R120PCPU			16384 bytes	

*1 The file size unit (cluster size) differs depending on the firmware version and production information of the CPU module. (☞ Page 1139 Added and Enhanced Functions)

Point

Data is written in the unit of the file size (cluster size). For example, when 464 bytes of CPU parameter is written to the data memory on R04CPU, it is written as 1024 bytes of data because the file size unit of the data memory is 1024 bytes.

File size unit of each SD memory card

SD memory card	File size unit
NZ1MEM-2GBSD	32K bytes
NZ1MEM-4GBSD	
NZ1MEM-8GBSD	
NZ1MEM-16GBSD	

3.3 Memory Operation

Initialization and value clear

Each memory can be initialized and cleared to zero by using the engineering tool. For details on the operation method, refer to the following.

📖 GX Works3 Operating Manual

Items to be specified in the engineering tool				Target
Initialization	Data memory			Deletes all the folders and files in the program memory and data memory.
	Device/label memory			Deletes all the files in the file storage areas in the device/label memory. The battery-less option cassette is initialized.
	SD memory card			Deletes all the folders and files in the SD memory card.
Value clear	Device, label	Zero clear		Excluding devices and labels with latch specified, clears the following to zero: X, Y, M, B, F, SB, V, S, T, ST, LT, LST, C, LC, D, W, SW, FX, FY, FD, Z, LZ, RD, and all labels (including module labels).
		Zero clear (including Latches (1) and (2))		Including devices and labels with latch specified, clears the following to zero: X, Y, M, B, F, SB, V, S, T, ST, LT, LST, C, LC, D, W, SW, FX, FY, FD, Z, LZ, RD, and all labels (including module labels).
	File register	Zero clear	All files	Clears the contents of all the file registers to zero.
			File specification	Clears only the contents of the specified file register(s) to zero.
		Zero clear excluding Latch (2)		Clears the file registers other than Latch (2) to zero.
	Device / label / file register latch clear			Clears devices, labels, and file registers other than Latch (2) to zero.

Point

If the power goes off during initialization or zero clear, the memory is left in the state of that point, and it is necessary to re-execute the memory operation.

Memory initialization during execution of another function

No memory can be initialized during execution of the following functions. Check that the following functions are not being executed and then initialize the memory.

- CPU module data backup/restoration function
- iQ Sensor Solution data backup/restoration function

Clearing values during execution of another function

■CPU module data backup/restoration function

During execution of the CPU module data backup/restoration function, devices, labels, file register areas, and latch areas cannot be cleared to zero. Check that the CPU module data backup/restoration function is not being executed and then clear devices, labels, file register areas, and latch areas to zero.

■iQ Sensor Solution data backup/restoration function

During execution of the iQ Sensor Solution data backup/restoration function, file register areas cannot be cleared to zero. Check that the iQ Sensor Solution data backup/restoration function is not being executed and then clear file register areas to zero.

3.4 Files

This section lists the files used by the CPU module.

Point

For the files used by the Safety CPU, refer to the following.

☞ Page 617 Files

File types and storage memory

This table lists the types of files, which are generated by parameter settings and functions in use, as well as their storage memory.

◎: Can be stored (Mandatory), ○: Can be stored, ×: Cannot be stored

File type		CPU built-in memory			SD memory card	File name and extension
		Program memory	Device/label memory	Data memory		
		Drive 0	Drive 3	Drive 4	Drive 2	
Program		◎*4	×	◎*4	○	ANY_STRING.PRG
FB file		○*4	×	○*4	○	ANY_STRING.PFB
CPU parameter		×	×	◎	○	CPU.PRM
System parameter		×	×	◎	○	SYSTEM.PRM
Module parameter		×	×	○	○	UNIT.PRM
Module extension parameter*8*10		×	×	○*11	○*11	• UEXmmmnn.PRM*1 • UEXmmm00.PPR*5
Module-specific backup parameter*6		×	×	○*11*13	○*11*13	UBPmmmnn.BPR*1
Memory card parameter		×	×	×	○	MEMCARD.PRM
Device comment		×	×	○	○	ANY_STRING.DCM
Initial device value		×	×	○	○	ANY_STRING.DID
Global label setting file		×	×	○	○	GLBLINF.IFG
Initial label value file	Initial global label value file	×	×	○	○	GLBLINF.LID
	Initial local label value file	×	×	○	○	PROGRAM_NAME.LID
File register		×	○	×	○*3	ANY_STRING.QDR
Event history		×	×	○	○	• EVENT.LOG • EVEN2.LOG
Device data storage file		×	×	○	○*3	DEVSTORE.QST
General-purpose data		×	×	○	○	ANY_STRING.CSV/BIN
Data logging setting file	Common setting file	×	×	×	○	LOGCOM.LCS
	Individual setting file	×	×	○	○	LOGnn.LIS*2
Memory dump setting file		×	×	○	○*3	MEMDUMP.DPS
Remote password		×	×	○	○	00000001.SYP
Firmware update file		×	×	○*3	○	mmmm_vv.SYF
Firmware update prohibited file		×	×	○	○*3	FWUPDP.SYU
Faulty database check file		×	×	×	○	ErrorDB.txt
Database path file		×	×	×	○	dbmaintainpath.txt
System file for backing up CPU module data		×	×	×	○	$BKUP_CPU_INF.BSC
Backup data file for backing up CPU module data		×	×	×	○	BKUP_CPU.BKD
Device/label data file for backing up CPU module data		×	×	×	○	BKUP_CPU_DEVLAB.BKD
System file for the iQ Sensor Solution data backup/restoration function		×	×	×	○	$BKUP_UNIT_INF.BSI

File type	CPU built-in memory			SD memory card	File name and extension
	Program memory	Device/label memory	Data memory		
	Drive 0	Drive 3	Drive 4	Drive 2	
Backup data file for the iQ Sensor Solution data backup/restoration function	×	×	×	○	Depends on a connected device.QBR*7
ODBC server setting file	×	×	○	×	netserver.cfg
Device station parameter file	×	×	○*12	○*12	SLAVEmmmnnnxxxx.NSP*9
Recording setting file	×	×	○	○	RECCFGn.RSI
User Web page file	×	×	×	○	Files that support the Web server function*14
System file for automatic restoration with the SD CARD OFF button	×	×	×	○	$BKUP_CPU_SWRSTR.BSC

*1 mmm represents the start I/O number (first three digits in four-digit hexadecimal representation) of each module. For the CPU module, it will be 3FFH. Also, nn represents the serial number (two-digit hexadecimal representation) of module extension parameter files or module-specific backup parameter files of each module.

*2 nn corresponds to the setting number and is 01 through 10.

*3 Can be stored but cannot operate as a function.

*4 When this file is stored in the built-in memory of the CPU module, it is divided into program memory and data memory and stored. (☞ Page 116 Configuration of a program file)

*5 The module extension parameter for the protocol setting is a file for storing protocol setting information in the predefined protocol support function.

*6 The module-specific backup parameter is a file for storing the save or restore data of the module to be replaced by the online module change function. For details, refer to the manual for the module used.

*7 This file name depends on the connection type of the iQ Sensor Solution data backup/restoration function. For the file name, refer to the following.
iQ Sensor Solution Reference Manual

*8 The parameter cannot be written to the CPU modules on other stations via MELSECNET/H of the Q series.

*9 mmm represents the start I/O number (first three digits in four-digit hexadecimal representation) of the master station of CC-Link IE TSN, nnn represents the number of units in network configuration setting, and xxxx represents the serial number (four-digit hexadecimal representation) of the parameter.

*10 In the redundant extension base configuration, the module extension parameter used by the module on the extension base unit must be stored in the intelligent function module. If the parameter is stored in the CPU module, an error occurs when the module is turned on or the module status is switched from STOP to RUN.
Before writing the module extension parameter to the intelligent function module, set the CPU module to redundant mode.

*11 The parameter cannot be stored in the redundant extension base unit configuration.

*12 Up to 1024 parameters can be stored.

*13 The location where the module-specific backup parameter file is stored varies depending on the set value of "Setting of File/Data Use or Not in Memory Card" of the memory card parameter.
· "Module Extended Parameter" is set to "Not Use" (default): Data memory
· "Module Extended Parameter" is set to "Use": SD memory card

*14 For files that support the Web server function, refer to the following.
MELSEC iQ-R Ethernet User's Manual (Application)

File operation available

The following lists the file operations which can be executed to each file in the CPU module by external devices.

○: Available, —: N/A

File type		Operation from engineering tool			Operation with SLMP and FTP server function			Operation via instruction*1	
		Write	Read	Delete	Write	Read	Delete	Write	Read
Program		○*2*7	○	○*4	○*3*6	○	○*3*6	—	—
FB file		○*2*7	○	○*4	○*3*6	○	○*3*6	—	—
Parameter		○*4	○	○*4	○*3*6	○	○*3	—	—
Device comment		○	○	○*4	○*3*6	○	○*3*6	—	—
Initial device value		○	○	○*4	○*3*6	○	○*3*6	—	—
Global label setting file		○*7*8*9	○*9	○*4	○*3*6	○	○*3*6	—	—
Initial label value file	Initial global label value file	○	○	○*4	○*3	○	○*3*6	—	—
	Initial local label value file	○	○	○*4	○*3	○	○*3*6	—	—
File register		○	○	○*4	○	○	○*3	○	○
Event history		—	—	—	○*10	○	○*10	—	—
Device data storage file		—	—	—	○*3*6	○	○*3*6	○	○
General-purpose data		○	○	○*4	○	○	○	○	○
Data logging setting file	Common setting file	○*5	○*5	○*5	○	○	○	—	—
	Individual setting file	○*5	○*5	○*5	○	○	○	—	—
Memory dump setting file		○	○	○*4	○	○	○	—	—
Remote password		○*4	○	○*4	○*3*6	○	○*3*6	—	—
Faulty database check file		○	○	○*4	○	○	○	○	○
Database path file		—	—	—	○	○	○	—	—
Firmware update file		○	○	○*4	○	○	○	—	—
Firmware update prohibited file		○	○	○*4	○	○	○	—	—
Module-specific backup parameter		○	○	○*3	○	○	○*3	○	○
System file for backing up CPU module data		—	—	—	○	○	○	—	—
Backup data file for backing up CPU module data		○	○	○*4	○	○	○	—	—
Device/label data file for backing up CPU module data		○	○	○*4	○	○	○	—	—
System file for the iQ Sensor Solution data backup/restoration function		—	—	—	○	○	○	—	—
Backup data file for the iQ Sensor Solution data backup/restoration function		○	○	○*4	○	○	○	—	—
ODBC server setting file		—	—	—	○	○	○	—	—
Device station parameter file		○	○	○*4	○	○	○	—	—
Recording setting file		○*11	○	○*4	○*11	○	○*11	—	—
User Web page file		—	—	—	○*12	○*12	○*12	—	—
System file for automatic restoration with the SD CARD OFF button		—	—	—	○	○	○	—	—

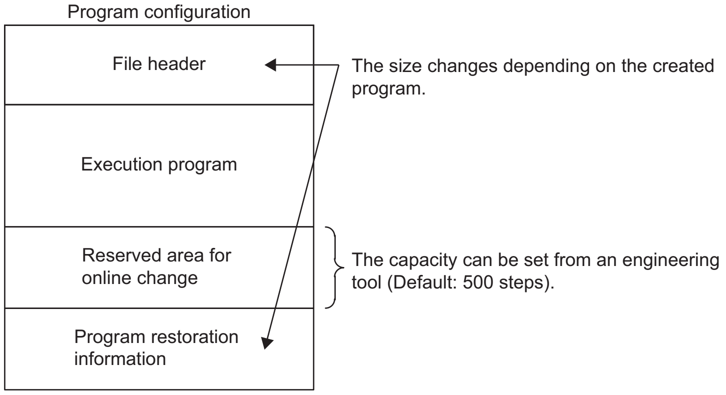
*1 Modification of data in files, such as read/write from/to file register and execution of the FWRITE/FREAD instruction
*2 When the CPU module is in the STOP state, the file operation is performed by writing the FB file to the programmable controller. In the RUN state, it is performed by online change. Note that only programs and FB files that are registered in the parameter can be written when the operating status of the CPU module is RUN.
*3 Available only when the CPU module operation status is STOP/PAUSE. A communication error occurs when operated in the RUN state.
*4 Available only when the CPU module operation status is STOP/PAUSE. When a program tries to perform memory operation while the CPU module is RUN, the operation is continued after the operation status is changed through remote STOP.
*5 Operation on CPU Module Logging Configuration Tool.
*6 When the operation target is the SD memory card, the operation can be performed even while the CPU module is RUN.
*7 The availability of the online change (the file batch online change of FB files and the global label setting file) differs depending on the model and firmware version.
For supported models and firmware versions, refer to the following.
☞ Page 1139 Added and Enhanced Functions
For executable conditions of the file batch online change, refer to the following.
📖 GX Works3 Operating Manual
For the modules that do not support the online change mentioned above, the operation continues after the operating status is changed by the remote STOP function if the operation is performed during RUN state.
*8 If the global label that can be accessed from the external device is set and the CPU module is in the RUN state, only read operation is available.
*9 When the CPU module is in the RUN state, the file operation is available only when the target file is already written to the CPU module.
*10 When the recording function is in the process of saving, the file operation is not available.
*11 When the recording function is in one of the following states, the file operation is not available.
· Preparing
· Operating
· Save trigger established
· Saving
*12 For supported models and firmware versions, refer to the following.
📖 MELSEC iQ-R Ethernet User's Manual (Application)

Configuration of a program file

The following figure shows the configuration of a program.

Configuration of a program

This file consists of a file header, execution program, reserved area for online change, and program restoration information.

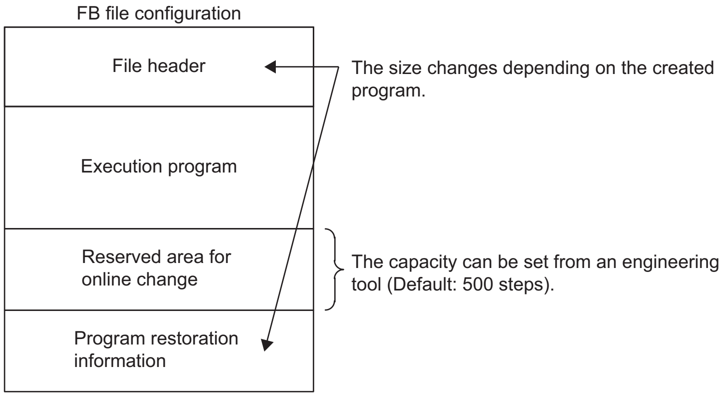


When the program is stored in the CPU module, file header, execution program, and reserved area for online change are allocated to the program memory. Program restoration information is allocated to the data memory. The following table shows the destination memory to which each area is allocated when a program is stored in the CPU module.

Area	Destination memory	Description
File header	Program memory	This area stores the configuration of the file, and the labels and information of FBs used in the program. The size differs depending on the created program.
Execution program		This area stores the execution program of the created program.
Reserved area for online change		This area is used when the online program change which increases the number of steps is executed (default: 500 steps (2000 bytes)). At writing to the programmable controller or at online ladder block change, the capacity of the area can be changed from the engineering tool.
Program restoration information	Data memory	This area stores information required for reading the program from the programmable controller.

FB file

This file consists of a file header, execution program, reserved area for online change, and program restoration information.



When an FB file is stored in the CPU module, the file header, execution program, and reserved area for online change are allocated to the program memory. Program restoration information is allocated to the data memory. The destination memory to which each area is allocated when an FB file is stored in the CPU module is the same as that for programs.

PART 2 FUNCTIONS

This part consists of the following chapters.

4 CLOCK FUNCTION

The CPU module internally maintains clock data and uses it to manage time for the system functions such as time stamp for the event history and the data logging function.

4.1 Time Setting

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 492 FUNCTIONS

- When using the SIL2 Process CPU, refer to the following as well.

☞ Page 704 FUNCTIONS

- When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

The clock continues operating with the internal battery of the CPU module while the CPU module is powered off or during power failure longer than the allowable momentary power failure time.

Point

The R00CPU, R01CPU, and R02CPU continue the clock operation for 10 days without a battery in power failure state.

Clock data

The following table lists the details of clock data in the CPU module.

Data name	Description
Year	Four digits (from 1980 to 2079)
Month	1 to 12
Day	1 to 31 (Automatic leap year detection)
Hour	0 to 23 (24 hours)
Minute	0 to 59
Second	0 to 59
Day of Week	0: Sunday, 1: Monday, 2: Tuesday, 3: Wednesday, 4: Thursday, 5: Friday, 6: Saturday
1/1000 seconds*1	0 to 999

*1 Data can be read from the S(P).DATERD instruction. (📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))

Changing the clock data

The clock data can be changed in one of the following methods:

- Using the engineering tool
- Using SM/SD
- Using the instruction
- Automatic change using the SNTP server

Point

When the clock data is changed, the following operation is performed:

- The millisecond value is reset to zero. Depending on the millisecond value immediately before the reset, the second value can be rounded up. Considering that the second value can increase by one second maximum when the clock data is changed, configure the system.
- Clock setting (Event code: 24000) is logged in the event history.

Using the engineering tool

Choose "Set Clock" from the menu. (📖 GX Works3 Operating Manual)

Using SM/SD

After SM210 (Clock data set request) is tuned on, values stored in SD210 (Clock data) to SD216 (Clock data) are written to the CPU module. Once the write operation is finished, SM210 is turned off. If values in SD210 to SD216 are out of the effective range, SM211 (Clock data set error) turns on and the values in SD210 to SD216 are not written to the CPU module.

Using the instruction

Use the DATEWR instruction to write the clock data to the CPU module. (📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))

Automatic change using the SNTP server

The time of CPU module is automatically set by collecting clock data from the time information server (SNTP server) connected to the LAN at the specified timing. (📖 MELSEC iQ-R Ethernet User's Manual (Application))

Reading the clock data

The clock data can be read in one of the following methods:

- Using SM/SD
- Using the instruction

Using SM/SD

When SM213 (Clock data read request) is turned on, the clock data is read to SD210 to SD216.

Using the instruction

Use the DATERD(P)/S(P).DATERD instructions to read the clock data from the CPU module. (📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))

Precautions on the clock data

The following lists the precautions on the clock data.

When using the product for the first time

Since the clock data is not set at the factory, be sure to set the correct data.

Modifying the clock data

Even if a portion of the clock data is changed, be sure to write all the data to the CPU module again.

Range of the clock data

The clock data must be written within the following range.

☞ Page 118 Clock data

Even within the range, the clock function does not operate normally if data outside the clock range is written to the CPU module.

Ex.

Operating status of the CPU module when impossible date is set

Date	Write operation to the CPU module	CPU module operating status
February 30	Executed	An error is not detected.
32nd of month 13	Not executed	• When the DATEWR instruction is executed, "Operation error" (error code: 3405H) is detected. • When SM210 is on, SM211 turns on.

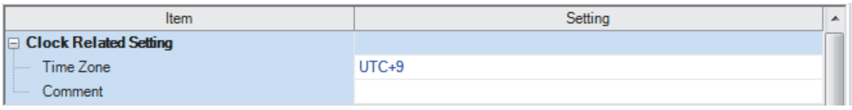
4.2 Setting Time Zone

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

The time zone used for the CPU module can be specified. Specifying the time zone enables the clock of the programmable controller to work in the local time zone.

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Clock Related Setting]

Window



Displayed items

Item	Description	Setting range	Default
Time Zone	Sets the time zone used by the CPU module.	• UTC+13 • UTC+12 • UTC+11 • UTC+10 • UTC+9:30 • UTC+9 • UTC+8 • UTC+7 • UTC+6:30 • UTC+6 • UTC+5:45 • UTC+5:30 • UTC+5 • UTC+4:30 • UTC+4 • UTC+3:30 • UTC+3 • UTC+2 • UTC+1 • UTC • UTC-1 • UTC-2 • UTC-3 • UTC-3:30 • UTC-4 • UTC-4:30 • UTC-5 • UTC-6 • UTC-7 • UTC-8 • UTC-9 • UTC-10 • UTC-11 • UTC-12	UTC+9
Comment	Enters a comment for a time zone (e.g., name of the city).	32 characters or less	—

Point

- To reflect the time zone setting on the CPU module, the module must be restarted. If no parameter is set for the CPU module, it operates with "UTC+9".
- On the multiple CPU system, the time zone setting of the CPU No.1 is used for other CPU modules. (the time zone setting of CPUs No.2 to 4 is not applied even when it is specified.)

4.3 Daylight Saving Time Function

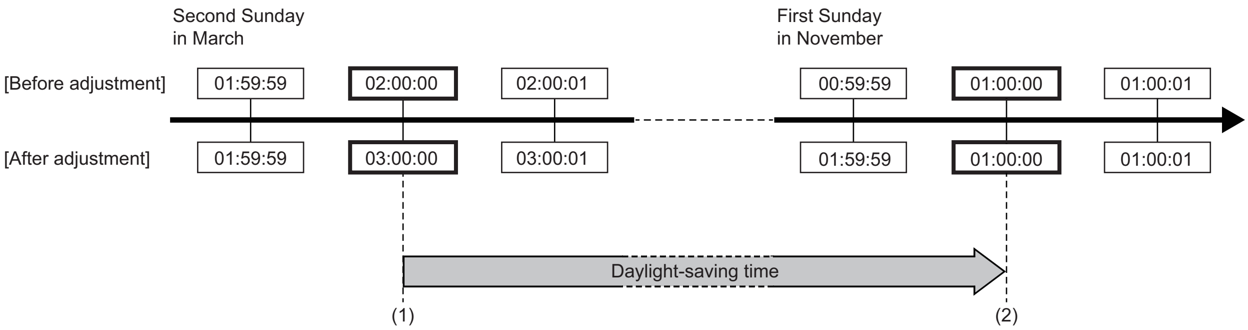
RnCPU RnENCPU ~~RnPCPU (Process)~~ ~~RnPCPU (Redundant)~~ ~~RnPSFCPU (Standard)~~ ~~RnPSFCPU (Safety)~~ RnSFCPU (Standard) RnSFCPU (Safety)

The daylight saving time function is used to adjust the CPU module time to daylight saving time.

This function advances the CPU module time by one hour on the starting date of daylight saving time, and reverses the time by 1 hour on the ending date.

Ex.

If daylight saving time starts from 2.00 on the second Sunday in March, and ends at 2.00 on the first Sunday in November



Restriction

Before executing the daylight-saving time function, check the versions of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

Daylight saving time setting

Sets the starting date and ending date for daylight saving time.

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Clock Related Setting] ⇨ [Setting to Adjust Clock for Daylight Saving Time]

Window

Item	Setting
Setting to Adjust Clock for Daylight Saving Time	
Adjust Clock for Daylight Saving Time	Disable
Start/End Time Specification Method	Week
Week	
Start	
Month	3
Week	2nd Week
Day of Week	Sunday
Time	2:00
End	
Month	11
Week	1st Week
Day of Week	Sunday
Time	2:00
Date	
Start	
Month	3
Day	1
Time	2:00
End	
Month	11
Day	1
Time	2:00

Displayed items

Item			Description	Setting range	Default
Adjust Clock for Daylight Saving Time			Sets whether to enable the daylight saving time setting.	• Enable • Disable	Disable
Start/End Time Specification Method			Sets the timing of the switch to daylight saving time to specified week or specified date.	• Specified week • Specified date	Specified week
Week	Start	Month	Sets the daylight saving time starting date.	1 to 12	3
		Week		• 1st Week • 2nd Week • 3rd Week • 4th Week • The Last Week	2nd Week
		Day of Week		• Sunday • Monday • Tuesday • Wednesday • Thursday • Friday • Saturday	Sunday
		Time		0:00 to 23:00	2:00
	End	Month	Sets the daylight saving time ending date.	1 to 12	11
		Week		• 1st Week • 2nd Week • 3rd Week • 4th Week • The Last Week	1st Week
		Day of Week		• Sunday • Monday • Tuesday • Wednesday • Thursday • Friday • Saturday	Sunday
		Time		0:00 to 23:00	2:00
Date	Start	Month	Sets the daylight saving time starting date.	1 to 12	3
		Day		1 to 31, The Last Date	1
		Time		0:00 to 23:00	2:00
	End	Month	Sets the daylight saving time ending date.	1 to 12	11
		Day		1 to 31, The Last Date	1
		Time		0:00 to 23:00	2:00

Point

With the multiple CPU system, the "Setting to Adjust Clock for Daylight Saving Time" for CPU No.1 is used for other CPU modules. (The "Setting to Adjust Clock for Daylight Saving Time" will not be valid for CPU Nos. 2 to 4.)

Restriction

- The same month cannot be specified for the start and end of the timing of the switch.
- February 29 cannot be specified directly. If specifying February 29, the date can be substituted by entering "Last Date in February".

Timing of daylight saving time adjustment

Daylight saving time is adjusted at the following times.

- ON the starting date and ending date of daylight saving time
- When the CPU module is powered off and on
- When the CPU module is reset

Daylight saving time function operation check

The daylight saving time function operation can be checked as follows.

Special relay

SM217 (Daylight saving time status flag) can be used to check whether the date lies inside or outside the daylight saving time period. (☞ Page 930 List of Special Relay Areas)

Event history

The history for the start and end of daylight saving time can be checked in the event history for the date set in "Setting to Adjust Clock for Daylight Saving Time" (☞ Page 908 Event list)

Operation of other functions using clock data during daylight saving time

Functions using CPU module clock data operate as follows during the daylight saving time period.

Item	Description
Clock data reading	Reads clock data following adjustment for daylight saving time.
Clock data writing	Writes as clock data following adjustment for daylight saving time.

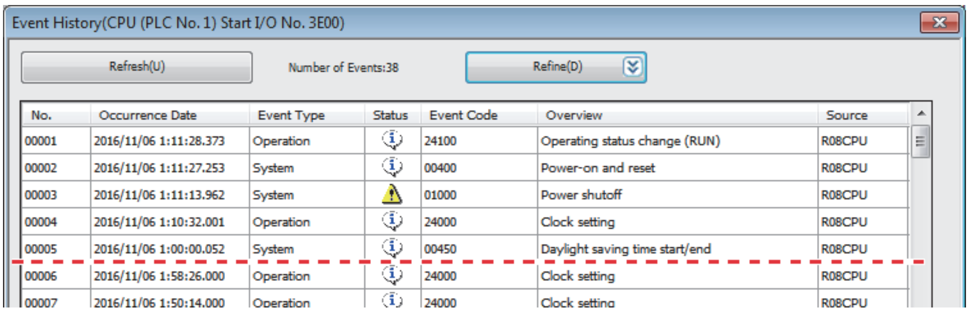
Precautions

This section describes precautions on the daylight saving time function.

- The clock data cannot be changed to data less than one hour from the starting time of the daylight saving time. Furthermore, if the "Setting to Adjust Clock for Daylight Saving Time" for CPU No.1 in a multiple CPU system is enabled, it will not be possible to change the clock data to data less than one hour from the starting time of the daylight saving time for CPU modules for CPU Nos. 2 to 4.
- For the period less than one hour from the starting time or less than one hour until the ending time of daylight saving time, the function that is triggered by time may not work or may work twice.
- When the "Setting to Adjust Clock for Daylight Saving Time" is enabled, functions operate based on date information after adjustment for daylight saving time. For this reason, dates and times output by functions that uses clock data are reversed ((before adjustment) ≥ (after adjustment)) and therefore the order (No.) in which events occur and the sorting order for the date on which events occur may not match. Consequently, when checking output results in chronological order, sort not in the order for the date on which events occur, but in the order (No.) in which events occur.

Ex.

Event history



No.	Occurrence Date	Event Type	Status	Event Code	Overview	Source
00001	2016/11/06 1:11:28.373	Operation		24100	Operating status change (RUN)	R08CPU
00002	2016/11/06 1:11:27.253	System		00400	Power-on and reset	R08CPU
00003	2016/11/06 1:11:13.962	System		01000	Power shutoff	R08CPU
00004	2016/11/06 1:10:32.001	Operation		24000	Clock setting	R08CPU
00005	2016/11/06 1:00:00.052	System		00450	Daylight saving time start/end	R08CPU
00006	2016/11/06 1:58:26.000	Operation		24000	Clock setting	R08CPU
00007	2016/11/06 1:50:14.000	Operation		24000	Clock setting	R08CPU

Since the time goes back one hour at No.00005 where the daylight saving time ends, the dates and times on which events occurred are reversed at the upper and lower side of the broken line ((time of new event history) < (time of old event history)).

4.4 System Clock

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 492 FUNCTIONS

- When using the SIL2 Process CPU, refer to the following as well.

☞ Page 704 FUNCTIONS

The system clock is turned on/off by the system or turns on/off automatically at the interval specified by the user.

Special relay used for system clock

Special relay used for system clock are as follows (☞ Page 939 System clock)

SM number	Name
SM400	Always On
SM401	Always Off
SM402	After RUN, ON for 1 scan only
SM403	After RUN, OFF for 1 scan only
SM409	0.01 second clock
SM410	0.1 second clock
SM411	0.2 second clock
SM412	1 second clock
SM413	2 second clock
SM414	2n second clock
SM415	2n millisecond clock
SM420	User timing clock No.0
SM421	User timing clock No.1
SM422	User timing clock No.2
SM423	User timing clock No.3
SM424	User timing clock No.4
SM440	On only initial I44 execution after RUN
SM441	On only initial I45 execution after RUN

Precautions

- SM409 to SM415 (System clock) start to measure the time before the CPU module is switched to RUN. Therefore, the time from the first scan after the CPU module has been switched to RUN until switching of the on/off state of the system clock may not match the measurement time of the system clock.
- The on/off status of SM409 to SM415 (System clock) changes even during execution of a program. For this reason, when one program has multiple processes that are performed based on the on/off status of the system clock, these processes are not performed in the execution order of the program. If these processes are required to be performed in the execution order of the program, write a program such that the on/off status of the system clock is transferred to an internal relay at the start of every scan and each processing is performed based on the status in the internal relay.

Special register used for system clock

Special register used for system clock are as follows (☞ Page 985 System clock)

SD number	Name
SD412	One second counter
SD414	2n second clock setting
SD415	2n ms clock setting
SD420	Scan counter

5 WRITING DATA TO THE CPU MODULE

This chapter describes the functions relating to writing data to the CPU module.

5.1 Writing Data to the Programmable Controller

This function writes data specified by the project of the engineering tool to the memory of the CPU module. For details, refer to the following.

GX Works3 Operating Manual

5.2 Online Change

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using the Process CPU (redundant mode), refer to the following as well.

Page 492 FUNCTIONS

- When using the SIL2 Process CPU, refer to the following as well.

Page 704 FUNCTIONS

- When using the Safety CPU, refer to the following as well.

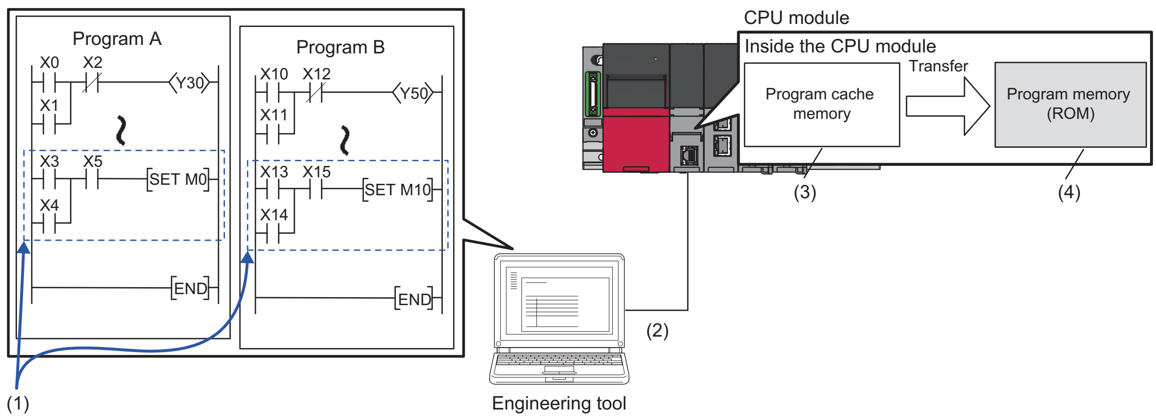
Page 621 FUNCTIONS

The following table lists the types of the online change.

Type		Description	Reference
Online change	Online change (ladder block)	Changes and writes a part of the program and data online.	• Page 125 Online change (ladder block) • GX Works3 Operating Manual
	Online change (SFC block)	Changes, adds, or deletes SFC blocks online.	• GX Works3 Operating Manual • MELSEC iQ-R Programming Manual (Program Design)
File batch online change		Writes data in file units online.	• Page 129 File batch online change • GX Works3 Operating Manual

Online change (ladder block)

This function writes the part of a program edited on the ladder editor using the engineering tool to the running CPU module in units of ladder blocks. Edited contents spanning multiple files or multiple portions can be written to the CPU module at once.



(1) Portion edited in the engineering tool
(2) The changed ladder block is written to the running CPU module.
(3) The program contents in the program cache memory are changed.
(4) After changing the program contents in the program cache memory, they are automatically transferred to the program memory.

Point

For details on the operating procedure of the online change (ladder block) on engineering tools, refer to the following.

GX Works3 Operating Manual

Editable contents

Within a program block, instructions and pointers (P, I) can be added, changed, or deleted. Also, for each program component, program blocks can be added, changed, or deleted. However, if the user try to edit a label, FB, or FUN, the following limitations are applied.

■Editable contents within a program block

To change or delete global label definition and global label, all the programs and FB files using them must match both on the engineering tool and in the CPU module.

■Editable contents within FB definition

- To add or change instructions that refer to a local label or the signal flow memory, the size of the instructions cannot exceed the reserved area capacity*1. The same applies to the standard function blocks, process control function blocks, and module function blocks because those function blocks also have local labels. For details, refer to the following.

Item	Reference
Precautions for adding a local label	Precautions when local labels are added into the MELSEC iQ-R series function blocks (FA-A-0232)
Instructions that refer to the signal flow memory	Page 107 Signal flow memory

- To add a local label, all the programs and FB files referencing the FB file which stores the FB definition to be modified must match both on the engineering tool and in the CPU module.
- For input and output labels (VAR_INPUT, VAR_OUTPUT, and VAR_IN_OUT) and public labels, to add, change, or delete subroutine type FB and FUN interface information*2, all the programs and FB files referencing the FB file storing the FB definition to modify must match both on the engineering tool and in the CPU module.

*1 Reserved area refers to the area used for adding or changing a local label or a local instance when a program is changed online. The area is 48 words for non-latch type local labels, 16 words for latch type labels, and 4 words for signal flow memory by default. The reserved area can be changed for each function block definition. (GX Works3 Operating Manual)

*2 The subroutine type FB and FUN interface information means:
· All FB definition, FUN definitions, and their definition numbers contained in FB files
· Definition names
· The number of all input labels, and their data types
· The number of all output labels, and their data types
· The number of all I/O labels, and their data types
· The number of public local labels in FB definition and their data types
· The implementation method of FB definition

■Editable contents within FUN definition

For input and output labels (VAR_INPUT and VAR_OUTPUT), to add, change, or delete subroutine type FB and FUN interface information, all the programs and FB files referencing the FB file storing the FB definition to modify must match both on the engineering tool and in the CPU module.

■Editable contents for each program component

To add or delete FB or FUN definition to or from an FB file, all the programs and FB files referencing the target FB file must match both on the engineering tool and in the CPU module.

Range changeable in a single session

The following shows the number of steps and number of ladder blocks which can be changed in a single session.

- Number of ladder blocks in a file: 64 blocks
- Maximum number of steps in a ladder block: 65535 steps
- Total number of steps for all blocks (steps before changes + steps after changes): 364K steps

Reserved area for online change

Reserved area for online change can be set in a program file to address the online change (ladder block) which causes a change in the program file size. (GX Works3 Operating Manual)

In addition, if the changed program exceeds the program file capacity (including reserved area for online change) during the online change (ladder block), the reserved area for online change can be set again if there is space available in the program memory.

Required free space in the data memory

If the data memory does not have enough free space when the online change (ladder block) is executed, an error occurs. The free space of the data memory required for the execution of the online change (ladder block) depends on the size of program files to be changed or the usage of labels.

<table>
<tr><th colspan="2">CPU module</th><th>Firmware version</th><th>Required data memory space</th></tr>
<tr><td colspan="2">R00CPU, R01CPU, R02CPU</td><td>All firmware versions</td><td rowspan="2">Total increased size of the relevant program restoration information*1</td></tr>
<tr><td colspan="2" rowspan="2">R04CPU, R08CPU, R16CPU, R32CPU, R120CPU
R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, R120ENCPU</td><td>"31" or later</td></tr>
<tr><td>"30" or earlier</td><td>Size of the program restoration information</td></tr>
<tr><td colspan="2" rowspan="2">Process CPU</td><td>"13" or later</td><td>Total increased size of the relevant program restoration information*1</td></tr>
<tr><td>"12" or earlier</td><td rowspan="2">Size of the program restoration information</td></tr>
<tr><td colspan="2">SIL2 Process CPU</td><td>All firmware versions</td></tr>
<tr><td rowspan="4">Safety CPU</td><td rowspan="2">Standard program</td><td>"13" or later</td><td>Total increased size of the relevant program restoration information*1</td></tr>
<tr><td>"12" or earlier</td><td>Size of the program restoration information</td></tr>
<tr><td rowspan="2">Safety program</td><td>"17" or later</td><td>Total increased size of the relevant program restoration information*1</td></tr>
<tr><td>"16" or earlier</td><td>Size of the program restoration information</td></tr>
</table>

*1 When the setting of [Convert] ⇨ [Online Program Change] ⇨ [Write Program Restore Information] is "Write with Execution Program", the required free space is equal to the size of the program restoration information.

The size of the program restoration information can be checked from the memory capacity calculation of the engineering tool.

[Tool] ⇨ [Confirm Memory Size (Offline)]

Ex.

The following table lists required free space of the data memory (size of program restoration information) for the online change (ladder block) in the R120CPU with a firmware version "30" or earlier.

Target program of online change (ladder block)	Required free space in the data memory	
	When labels are not used	When labels are used (200 global labels and 200 local labels are used)
Ladder program of 1K steps	Approx. 20K bytes	Approx. 40K bytes
Ladder program of 10K steps	Approx. 80K bytes	Approx. 140K bytes
Ladder program of 100K steps	Approx. 550K bytes	Approx. 1000K bytes

Online change (ladder block) during the boot operation

When the online change (ladder block) is performed while booting from the SD memory card, the corresponding files on the booting SD memory card can also be changed.

Setting the initial value for registering/changing label definition

The initial value used when registering/changing label definition can be set. (GX Works3 Operating Manual)

■Initial value setting availability

Indicates whether or not the initial value can be set when adding or changing a label.

○: Available, △: Conditionally available, ×: Not available

Label type		Label addition	Label change
Program block	Global label (standard)	○	△*1
	Global label (safety)	×	×
	Local label (standard)	○	△*1
	Local label (safety)	×	×
	Standard/safety shared labels	×	×
FB definition	Local label (standard)	○*2	△*1
	Local label (safety)	×	×
FUN definition	Local label (standard)	×*3	×*3
	Local label (safety)	×*3	×*3

*1 The initial value can be set only when a new area is assigned again.

*2 A different initial value can be set for each instance.

*3 Since the local label in FUN definition is an undefined value, initialize it by a program in the FUN definition before using it in a program.

Point

When using GX Works3 version 1.000A

- When the label definition (the initial value or any other value) is changed, write the initial label value file to the CPU module after the online change (ladder block). Otherwise, an error occurs when the CPU module is powered off and on or reset, or the module operating status is changed from STOP to RUN.
- To clear all the initial values, delete the initial label value files in the CPU module. If the boot operation is being performed, the initial label value files are stored on the SD memory card. Delete the initial label value files on the SD memory card, and also delete the files set in the module card parameters ("Boot File Setting"). Otherwise, an error occurs when the CPU module is powered off and on or reset, or the module operating status is changed from STOP to RUN.

File batch online change

This function writes programs and other data to the running CPU module in units of files.
For the operating procedure and the execution condition of the file batch online change, refer to the following.
GX Works3 Operating Manual

Writing FB files and the global label setting file

The file batch online change of FB files and the global label setting file is available depending on the model and firmware version of the CPU module. (Page 1139 Added and Enhanced Functions)
When executing the file batch online change of FB files and the global label setting file, configure the setting of the write target file described below before system operation.
However, the setting is not required when the R00CPU, R01CPU, or R02CPU is used.

■Setting of the file to be written

1. Check that SM388 (File batch online change operation setting status) is off.
2. Set "AFBFH" to SD384 (System operation setting).
3. Turn on SM384 (System operation setting request). SM384 automatically turns off. If writing has failed, SM385 (System operation setting error) turns on and an error is stored in SD385 (System operation setting error cause).
4. Check that SM385 is off, and turn off or reset the CPU module.
5. SM388 turns on.

Point

- This setting requires powering off the system or resetting the CPU module. For this reason, configure this setting before system operation when executing the file batch online change of FB files and the global label setting file.
- If the file batch online change is executed after this setting is configured (while SM388 is on), the scan time may increase compared with the case when the file batch online change is executed before this setting is configured (while SM388 is off).
- When using the Process CPU (redundant mode), configure this setting for both systems.

■Procedure for clearing the setting (how to reset the write target file setting)

1. Check that SM388 (File batch online change operation setting status) is on.
2. Set "AFB0H" to SD384 (System operation setting).
3. Turn on SM384 (System operation setting request). SM384 automatically turns off. If writing has failed, SM385 (System operation setting error) turns on and an error is stored in SD385 (System operation setting error cause).
4. Check that SM385 is off, and turn off or reset the CPU module.
5. SM388 turns off.

5.3 Precautions

This section describes the precautions on writing data to the CPU module.

Prohibited operation (Turning off or resetting the CPU modules)

- When writing data to the programmable controller or executing the online change (ladder block), do not turn off or reset the CPU module. Otherwise, the operation does not complete successfully. If doing so, write the data to the programmable controller again.
- Do not power off and on or reset the CPU module when the program memory transfer is not completed. Otherwise, a stop error occurs.

Operation from engineering tools

■Operations cannot be executed simultaneously

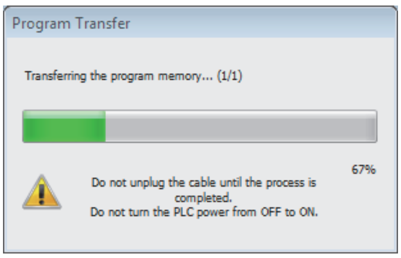
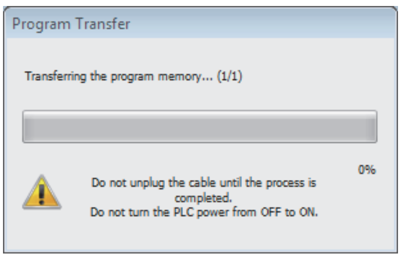
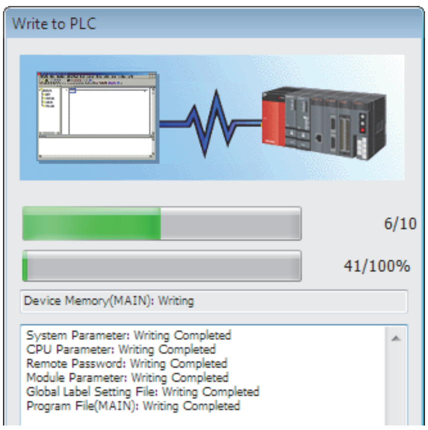
The file batch online change or online change (ladder block) cannot be executed simultaneously with the following operations from the engineering tool.

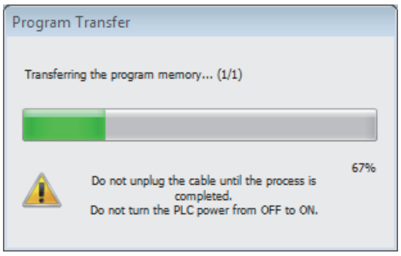
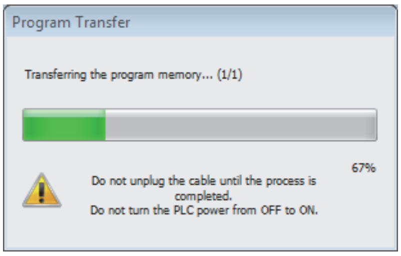
- Write to the programmable controller (excluding device, local device, global label, or local label data).
- File batch online change or online change (ladder block)
- Memory initialization
- Switching safety operation mode*1

*1 Do not execute the file batch online change/online change (ladder block) including the safety program, safety FB, and safety global label, standard/safety shared label for the write target station while the safety operation mode is switching to the safety mode.

■Writing during the program memory transfer

When the program memory transfer starts during writing to the programmable controller or the online change from an engineering tool, writing to the programmable controller or the online change from another engineering tool can be executed. However, if this operation is performed, the progress of the ongoing transfer is reset to 0%. The transfer progress which is reset to 0% resumes when the program memory transfer from another engineering tool starts.

Engineering tool 1	Engineering tool 2	Description
	—	The program memory transfer from the engineering tool 1 starts.
		Writing from the engineering tool 2 starts during the transfer from the engineering tool 1. The progress of the transfer from the engineering tool 1 is reset to 0%.

Engineering tool 1	Engineering tool 2	Description
		When the program memory transfer from the engineering tool 2 starts, the transfer from the engineering tool 1 progresses as well.

If the later program memory transfer (from the engineering tool 2) has completed with an error, the former program memory transfer (from the engineering tool 1) does not complete. In such a case, write the data again instead of powering off and on or resetting the CPU module.

When the online change (ladder block) is used

The following describes the precautions on using the online change (ladder block).

■When deleting the OUT instruction which is on

When deleting the OUT instruction (coil) which is not necessary for control, check that the OUT instruction is off before deleting it. If the OUT instruction is deleted without turning it off in advance, the output will be retained.

■Program file not registered in program setting

A program file which is not registered in parameter setting cannot be written.

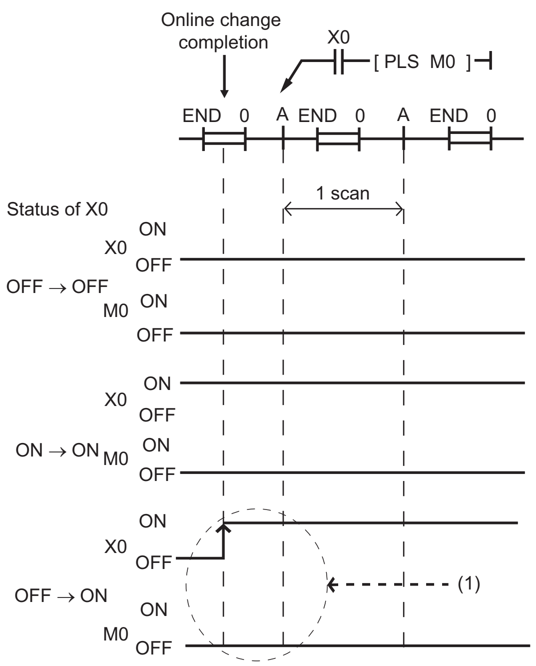
■Instructions which do not operate correctly

The following table lists the instructions which do not operate correctly during the online change (ladder block) or writing data to the programmable controller. In the SFC program, these instructions do not operate correctly only when they are in active steps.

Instructions which do not operate correctly	Description
Rising instruction (PLS and □P instructions)	When a rising instruction exists within the range to be changed, the rising instruction will not be executed even if the execution condition (OFF to ON) is taken at completion of online program change.
Falling instruction (PLF and □F instructions)	When a falling instruction exists within the range to be changed, the falling instruction will not be executed even if the execution condition (ON to OFF) is taken at completion of online program change.
SCJ instruction	If an SCJ instruction exists within the range to be changed and the execution condition is taken, the program will jump without waiting for a single scan.
STMR instruction	If an STMR instruction exists within the range to be changed, the STMR instruction will be executed.

- Rising instruction

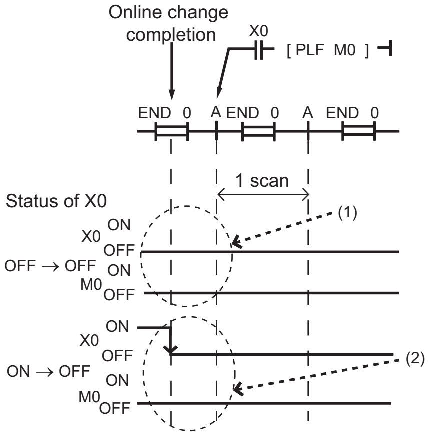
When a rising instruction exists within the range to be changed, the rising instruction will not be executed even if the execution condition (OFF to ON) is satisfied after completion of the online change (ladder block) or writing data to the programmable controller.



(1) The rising instruction will not be executed even if the execution condition is OFF to ON.

• Falling instruction

When a falling instruction exists within the range to be changed, the falling instruction will not be executed even if the execution condition (ON to OFF) is satisfied after completion of the online change (ladder block) or writing data to the programmable controller.

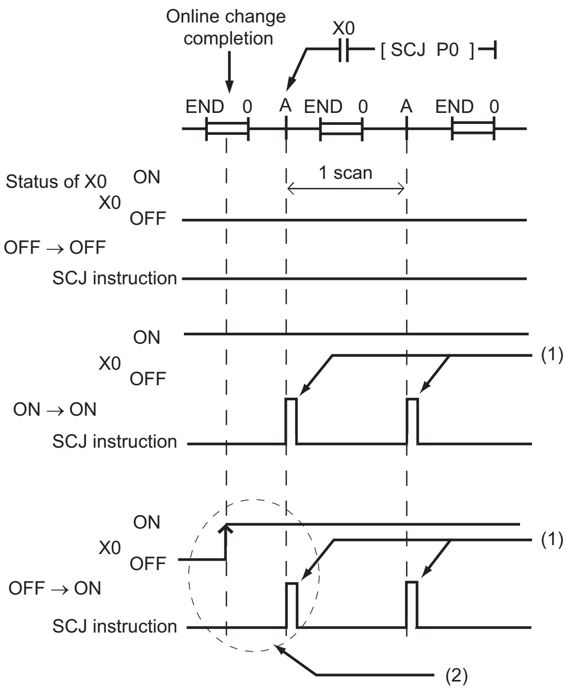


(1) The falling instruction will not be executed even if the execution condition is OFF to OFF.

(2) If a completion of the online program change and a timing of satisfying the execution condition (ON to OFF) occur simultaneously, the falling instruction will not be executed.

• SCJ instruction

If an SCJ instruction exists within the range to be changed and the execution condition is satisfied at the completion of the online change (ladder block) or writing data to the programmable controller, the program will jump without waiting for a single scan.

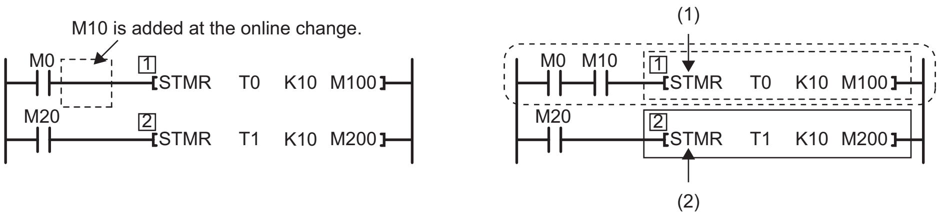


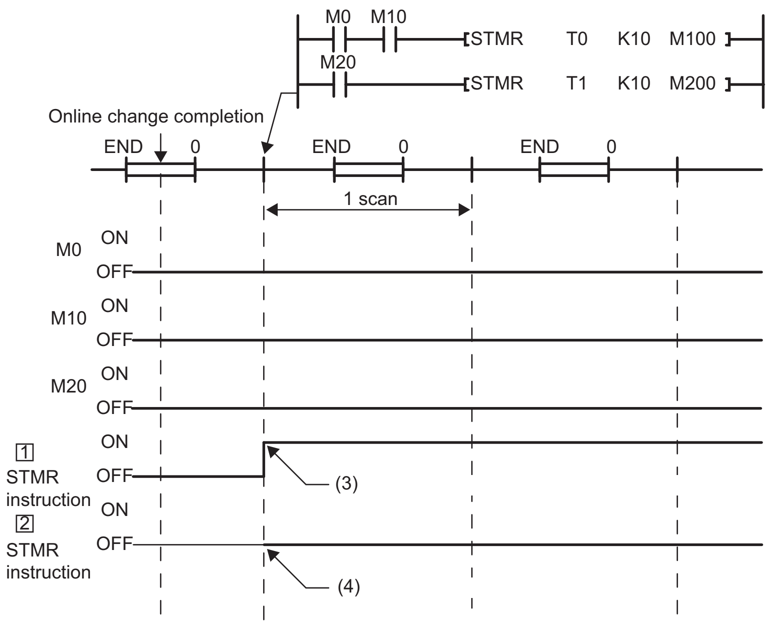
(1) The program jumps to the specified pointer.

(2) The program jumps to the specified pointer without waiting for a single scan.

• STMR instruction

If an STMR instruction exists within the range to be changed, the STMR instruction will be executed.





(1) The STMR instruction functions because the STMR instruction is included in the ladder block where the online change has been executed.

(2) The STMR instruction does not function because the STMR instruction is not included in the ladder block where the online change is executed.

(3) The instruction functions even when M0 and M10 are off.

(4) The instruction does not function.

■Initializing the last execution if the ladder at the online change (ladder block) has an FB call

If a subroutine type FB is called within the changed ladder block, information of the last execution, such as the rising instruction and the falling instruction within the FB definition of the called subroutine type FB, is not initialized.

■When the start-up of an interrupt program is delayed

When the online change (ladder block) is performed, the start-up of the interrupt program may be delayed. Therefore, when the execution time is monitored for the interrupt program that uses the inter-module synchronous interrupt (I44) and multiple CPU synchronous interrupt (I45), an error may be detected in the CPU module. (☞ Page 139 Error detection setting)

■Scan monitoring function during online changes

Some sections are not targeted for the scan monitoring while online changes are being performed, and therefore a WDT error may not be detected if the scan time set in the scan time monitoring time (WDT) setting is exceeded.

■Increase in scan time of the online change (ladder block)

When global labels and local labels are added, or when there are many programs and FB files to be changed or many changes, the scan time may be increased by several dozen milliseconds.

For the Safety CPU, the scan time may be increased by several dozen milliseconds when safety global labels, safety local labels, standard/safety shared labels, safety programs, and safety FBs are included.

■Online change (ladder block) of when another function is performed

The online change (ladder block) cannot be performed during execution of the following functions. Check that the following functions are not being executed and then perform the online change (ladder block).

- CPU module data backup/restoration function
- iQ Sensor Solution data backup/restoration function

■When multiple users execute the online change function to one CPU module

Note the following:

- Use engineering tools with the same version.
- Make the option settings the same in all the engineering tools.
- To prevent program block names from duplicating due to debugs by multiple users when adding a program block or changing a program block name, select "Yes" for "Duplication Check for POU" under [Convert] ⇨ [Online Program Change] in the "Options" window of the engineering tool.
- For editable contents, refer to the following.

☞ Page 126 Editable contents

- Do not make any changes that affect other programs (for example, editing the same program by multiple users, editing FBs/FUNs/global labels, or having a duplicate program block name or a duplicate global pointer). If one user makes such a change to a program and execute the online program change function of the engineering tool, and after that, another user execute the online program change function for another program, the engineering tool will detect a mismatch of data in the programmable controller. In this case, verify the data in the changed program and data in the CPU module, and check the mismatched data. Match the data and write the program by using the write to PLC function of the engineering tool, if needed.

Note that after debugging by multiple users, an appointed person must perform the following operations: Read the project from the CPU module by using the read from PLC function of the engineering tool, rebuild (reassign) all the data, and write the project back to the CPU module.

During the file batch online change

The following describes the precautions on the file batch online change.

■Writing the label data

Write labels using the file batch online change only when new label data is added. When the label data is changed or deleted, write data to the programmable controller or execute the online change (ladder block) after the data is rebuilt (reassigned).

■Falling instruction in the subroutine type FB program

When writing an FB file online, do not use falling instructions in the subroutine type FB program while SM388 (File batch online change operation setting status) is on.

■File batch online change during execution of another function

The file batch online change cannot be executed during execution of the following functions. An error occurs at execution.

- Data logging function (when the storage location is the function memory)
- Data logging file transfer (when the storage location is the function memory)

6 RAS FUNCTIONS

6.1 Scan Monitoring Function

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 492 FUNCTIONS

- When using the SIL2 Process CPU, refer to the following as well.

☞ Page 704 FUNCTIONS

This function detects hardware and program errors of the CPU module by monitoring the scan time. The watchdog timer, an internal timer of the CPU module, is used to monitor the following scan.

- Initial scan (first scan)
- Second and later scans

Scan time monitoring time setting

Set the scan time monitoring time.

[CPU Parameter] ⇨ [RAS Setting] ⇨ [Scan Time Monitoring Time (WDT) Setting]

Window

Displayed items

Item	Description	Setting range	Default
Initial Scan	Sets the scan-time monitoring time (WDT) for the initial scan (first scan).	10 to 2000ms (in units of 10ms)	2000ms
After 2nd Scan	Sets the scan-time monitoring time (WDT) for the second and later scans.	10 to 2000ms (in units of 10ms)	200ms

Watchdog timer reset

The watchdog timer is reset when the END/FEND instruction is executed. While the CPU module is running correctly, if the END/FEND instruction is executed within the set time of the watchdog timer, the watchdog timer does not count up. If the END/FEND instruction cannot be executed within the set time of the watchdog timer due to a hardware error of the CPU module or increase in program execution time due to an interrupt or other causes, the watchdog timer counts up.

Precautions

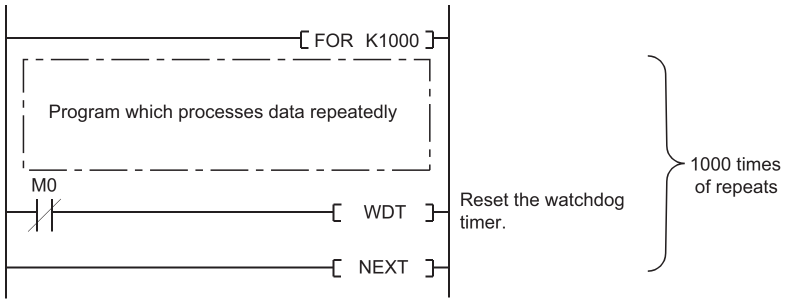
The following lists the precautions on the scan monitoring function.

Measurement error of watchdog timer

Since the watchdog timer produces an error within the range of 0 to 10ms, take this into consideration when setting the scan time monitoring time. For example, if the scan time monitoring time is set to 100ms, an error will occur when the scan time falls within the range 100ms < t < 110ms.

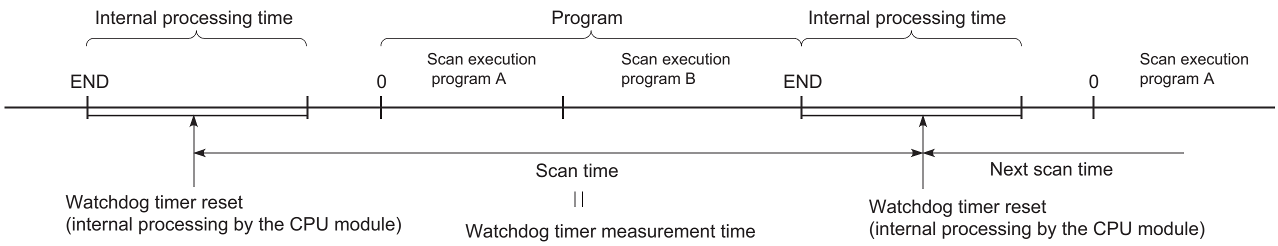
Resetting the watchdog timer when repeatedly executing a program

The watchdog timer can be reset by executing the WDT reset instruction from the program. If the watchdog timer counts up while a program is being repeatedly executed by the FOR instruction and NEXT instruction, use the WDT reset instruction to reset the watchdog timer.



Scan time when the WDT reset instruction is used

Even when the watchdog timer is reset by the WDT reset instruction, the scan time is not reset. The scan time is accumulated until the END instruction is executed.



6.2 Self-Diagnostics Function

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 492 FUNCTIONS

- When using the SIL2 Process CPU, refer to the following as well.

☞ Page 704 FUNCTIONS

- When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

This function (the CPU module itself) checks if a problem exists in the CPU module.

Self-diagnostics timing

If an error occurs when the CPU module is powered on or while it is in the RUN/STOP state, the CPU module detects, and displays it, and stops operation. However, depending on the error occurrence status or the instruction to execute, the CPU module may not be able to detect the error. Configure safety circuits external to the programmable controller to ensure that the entire system operates safely even in such a case.

Check method of error

This section describes the check methods when error occurs.

Check method using the special relay and special register

When the CPU module detects an error, it turns SM0 (Latest self-diagnostic error (including annunciator ON)) and SM1 (Latest self-diagnostic error (not including annunciator ON)) on and stores the error code corresponding to the error definition in SD0 (Latest self-diagnostics error code). If multiple errors are detected, the latest error code is stored in SD0. Use SM0, SM1, and SD0 on the program for the CPU module or mechanical interlock. Besides, the error code up to 16 pieces for the error contents being currently generated will be stored into SD10 to SD25 (Self-diagnostic error number). (The error codes of the 17th error and later are not stored.)

Check method using LED

The error occurrence conditions can be checked through the lighting conditions of ERROR LED. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

Check method using the engineering tool

The error conditions for the overall system, error or event history being currently generated can be checked on the Module diagnostics window. (📖 GX Works3 Operating Manual)

■Existing errors

Up to 16 errors (descriptions of errors) currently existing on the CPU module can be displayed.*1 However, even when an additional error occurs after a stop error, the error information is not updated.

*1 The maximum number of displayable errors is 15 for continuation errors and 1 for stop errors. When 15 continuation errors are displayed and another one occurs, description of the new error is not displayed. Also, when an error with the same code has already been displayed, the date and time of occurrence and detailed information of the relevant error are not updated.

■Error history

Occurred errors are logged in the event history. (☞ Page 148 Event History Function)

The event history is updated only when a battery error occurs, independent of the operating status of the CPU module. Also, when a battery error is detected after the occurrence of a stop error, the information on existing errors is not refreshed, and only the event history is updated.

CPU module operation upon error detection setting

Configure each CPU module operation setting when an error is detected.

Mode when an error is detected

If the self-diagnostic function of the CPU module detects an error, the CPU module can be in one of the following operation status:

■Mode for stopping the operation of CPU module

Operation stops when an error has been detected. During stopping the operation may vary depending on the output mode setting when module parameter error occurs.

- When "Clear" is set: Output for the corresponding module is turned off.
- When "Hold" is set: Output for the corresponding module is held.

Point

For the setting method of module parameter, refer to the manual for each module.

■Mode for continuing the operation of CPU module

If an error has been detected, the program other than programs (instruction) where an error was generated is executed.

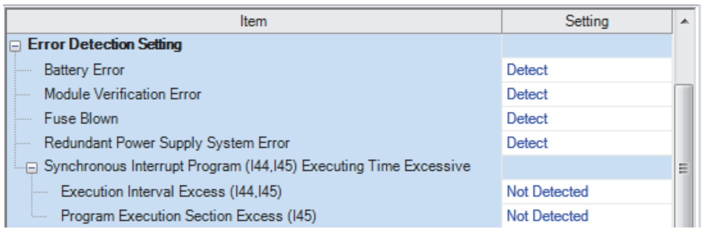
6

Error detection setting

Set whether or not to detect errors.

[CPU Parameter] ⇨ [RAS Setting] ⇨ [Error Detection Setting]

Window



Displayed items

<table>
<tr><th colspan="2">Item</th><th>Description</th><th>Setting range</th><th>Default</th></tr>
<tr><td colspan="2">Battery Error</td><td>Sets whether or not to detect the battery error.</td><td rowspan="6">• Detect
• Not Detected</td><td>• R00CPU, R01CPU, and R02CPU: Not Detected*3
• Other CPU modules: Detect</td></tr>
<tr><td colspan="2">Module Verification Error*1</td><td>Sets whether or not to detect a module verification error.</td><td rowspan="3">Detect</td></tr>
<tr><td colspan="2">Fuse Blown</td><td>Sets whether or not to detect a fuse blown of the controlled module.</td></tr>
<tr><td colspan="2">Redundant Power Supply System Error*2</td><td>Sets whether or not to detect a power supply module error (power off or power supply failure in either system) in the redundant power supply system.*4</td></tr>
<tr><td rowspan="2">Synchronous Interrupt Program (I44, I45) Executing Time Excessive</td><td>Execution Interval Excess (I44, I45)</td><td>Sets whether or not to detect the excessive execution time of synchronous interrupt program (I44, I45).</td><td rowspan="2">Not Detected</td></tr>
<tr><td>Program Execution Section Excess (I45)</td><td>Sets whether or not to detect the program execution section excess errors of synchronous interrupt program (I45).</td></tr>
</table>

*1 When "Not Detected" is set and an operating module is removed, a module verification error is not detected, but a stop error may occur if a program tries to access the removed module. The removed module will not be accessible even when it is re-installed. Therefore, if an access to the re-installed module occurs by switching the operating status of the CPU module from STOP to RUN.

*2 When "Not Detected" is set, values are stored in the special relay and special register (SM150 to SM154/SD150 to SD154) that indicate the status of the redundant power supply module.

*3 When a battery is installed, change the setting to "Detect". Otherwise, the installation status and replacement period of the battery cannot be checked because the battery error cannot be detected.

*4 The power supply module on the extension base unit for the redundant system also becomes a target for the setting.

■Applicable errors to the error detection setting

The following table lists errors for which whether or not to detect the errors can be set.

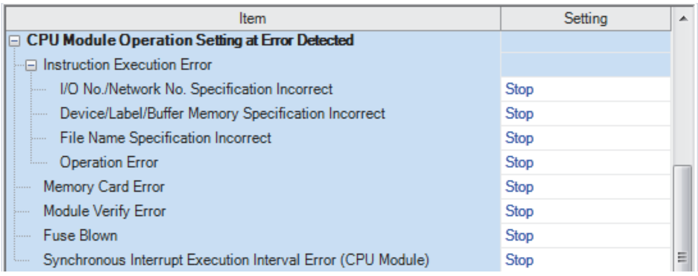
Error name	Error code
Battery error	1090H
Module verification error	2400H, 2401H
Fuse Blown	2420H
Redundant Power Supply System Error	1010H, 1020H
Synchronous Interrupt Execution Interval Error (CPU Module)	1240H, 1241H, 1260H, 1262H

CPU module operation upon error detection setting

Set the CPU module operation upon error detection.

[CPU Parameter] ⇨ [RAS Setting] ⇨ [CPU Module Operation Setting at Error Detected]

Window



Displayed items

Item		Description	Setting range	Default
Instruction Execution Error	I/O No./Network No. Specification Incorrect	Sets the CPU module operation when a specification error of I/O numbers/Network numbers has been detected.	• Stop • Continue	Stop
	Device/Label/Buffer Memory Specification Incorrect	Sets the CPU module operation when a specification error of Device/Label/Buffer memory has been detected.		
	File Name Specification Incorrect	Sets the CPU module operation upon a file name specification error.		
	Operation Error	Sets the CPU module operation upon an operation error.		
Memory Card Error*2		Sets the CPU module operation upon a memory card error.		
Module Verify Error*1		Sets the CPU module operation upon a module verification error.		
Fuse Blown*1		Sets the CPU module operation upon a fuse blown error of controlled module.		
Synchronous Interrupt Execution Interval Error (CPU Module)		Sets the CPU module operation upon a detection of a synchronization signal error on the CPU module.		

*1 For the Process CPU and SIL2 Process CPU, setting "Direct change setting" to "Enable" in "Online module change setting" causes a continuation error regardless of this setting. For the online module change function setting, refer to the following.
MELSEC iQ-R Online Module Change Manual

*2 This item is not displayed on the R00CPU.

6

■Applicable errors to the CPU module operation upon error detection setting

The following table lists the applicable errors to the setting that specifies the CPU module operation of when the specific errors are detected.

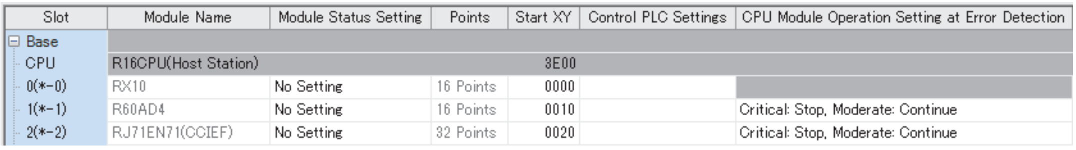
Error name		Error code
Instruction Execution Error	I/O number or network number specification error	2800H, 2801H, 2802H, 2803H, 2804H, 2805H, 2806H, 2807H, 2810H
	Device, label, or buffer memory specification error	2820H, 2821H, 2822H, 2823H, 2824H
	File name specification error	2840H, 2841H, 2842H
	Operation error	3400H, 3401H, 3402H, 3403H, 3404H, 3405H, 3406H, 3420H, 3421H, 3422H, 3423H, 3424H, 3425H, 3426H, 3427H, 3430H, 3440H, 3441H, 34A0H
Memory card error		2120H, 2121H
Module verification error		2400H, 2401H
Fuse Blown		2420H
Synchronous Interrupt Execution Interval Error (CPU Module)		2610H, 2630H

CPU module operation setting

Specify the operation which the CPU module should perform when an error occurs on each intelligent function module.

[System Parameter] ⇨ [I/O Assignment] tab ⇨ [I/O Assignment Setting]

Window



Slot	Module Name	Module Status Setting	Points	Start XY	Control PLC Settings	CPU Module Operation Setting at Error Detection
Base						
CPU	R16CPU(Host Station)			3E00		
0(*-0)	RX10	No Setting	16 Points	0000		
1(*-1)	R60AD4	No Setting	16 Points	0010		Critical: Stop, Moderate: Continue
2(*-2)	RJ71EN71(CCIEF)	No Setting	32 Points	0020		Critical: Stop, Moderate: Continue

Displayed items

Item	Description	Setting range	Default
CPU Module Operation Setting at Error Detection*1	Set the CPU module operation upon the detection of Major or Moderate errors in the configured module.	• Critical: Stop, Moderate: Continue • Critical: Stop, Moderate: Stop • Critical: Continue, Moderate: Continue	Critical: Stop, Moderate: Continue

*1 For the Process CPU and SIL2 Process CPU, setting "Direct change setting" to "Enable" in "Online module change setting" causes a continuation error regardless of this setting. For the online module change function setting, refer to the following.
MELSEC iQ-R Online Module Change Manual

■Applicable errors to the CPU module operation setting

The following table lists the applicable errors to the setting that specifies the CPU module operation of when the specific errors have occurred in each intelligent function module.

Error name	Error code
Module moderate error	1200H
Module major error	2441H, 2442H, 2450H

Stop setting

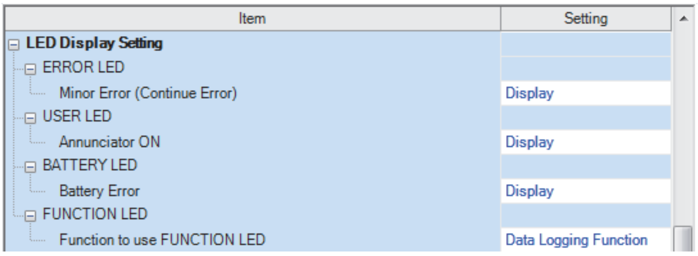
For the multiple CPU system configuration, it can be specified whether or not to stop all CPU modules when a major or moderate error occurs on a CPU module. (Page 329 Stop setting)

LED display setting

Set whether to display or hide the ERROR LED, USER LED, BATTERY LED, and FUNCTION LED.

[CPU Parameter] ⇨ [RAS Setting] ⇨ [LED Display Setting]

Window



Displayed items

Item		Description	Setting range	Default
ERROR LED	Minor Error (Continue Error)	Sets whether or not to display the ERROR LED when a minor error occurs.	• Display • Do Not Display	Display
USER LED	Annunciator ON	Sets whether the USER LED is displayed or not when annunciator (F) ON is detected.		
BATTERY LED*2	Battery Error	Sets whether the BATTERY LED is displayed or not when a battery error occurs.		
FUNCTION LED*1*3	Function to use FUNCTION LED	Sets the function that uses FUNCTION LED. If FUNCTION LED is not used, set "None".	• Data logging function • Memory dump function • None	Data logging function

*1 To specify whether to display or hide the FUNCTION LED, verify the versions of the CPU module and the engineering tool. (☞ Page 1139 Added and Enhanced Functions)

*2 This item is not displayed on the R00CPU, R01CPU, or R02CPU. When the battery error is detected in the R00CPU, R01CPU, and R02CPU, the ERROR LED turns on.

*3 For the R00CPU, the parameter is fixed to "None".
For the Process CPU and SIL2 Process CPU, this does not appear.
For the Safety CPU, whether this appears or not differs depending on the version of the engineering tool. (☞ Page 1139 Added and Enhanced Functions)

For the FUNCTION LED display, the following cases have priority over the function that is set in the above setting.

- When the external input/output forced on/off function is executed (in registration) (☞ Page 182 FUNCTION LED)
- When program restoration information is not written (☞ Page 1101 Checking the program restoration information write status)
- When the device tests with execution conditions are registered (☞ Page 185 Device Test with Execution Conditions)

Error detection invalidation setting

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

Turning on the target bit of SD49 (Error detection invalidation setting) disables detection of the corresponding continuation error.*1 (☞ Page 967 Diagnostic information)

*1 When using the error detection invalidation setting, check the version of the CPU module used. (☞ Page 1139 Added and Enhanced Functions)

The following operations are not disabled even when the detection of the applicable continuation error is disabled in this setting.

- Storage in buffer memory (Un\G770 to Un\G792) (📖 MELSEC iQ-R Ethernet User's Manual (Application))
- Setting of the completion status of the built-in Ethernet function instruction (📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))

Data is not retained in SD49 during power failure. Therefore, register this setting again after turning off and on or resetting the CPU module.

Applicable errors to the error detection invalidation setting

The following table lists errors for which the detection of the continuation error can be disabled.

Error name	Error code
Connection establishment failed	112EH
Socket communications response send error	1133H
TCP connection timeout	1134H
Connection number acquisition error	1155H
Receive buffer securement error	1157H
UDP/IP send failed	1165H
TCP/IP send failed	1166H

6.3 Error Clear

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 492 FUNCTIONS

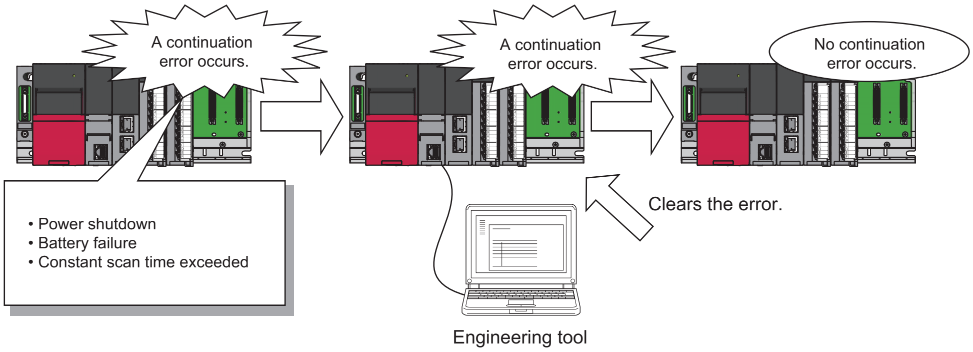
- When using the SIL2 Process CPU, refer to the following as well.

☞ Page 704 FUNCTIONS

- When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

This function clears all the existing continuation errors at once.



Errors that can be cleared

This function can be used to clear only the continuation errors listed in the following table.

Error name	Error code
Power shutoff	1000H
Power shutoff (either of the redundant power supply modules)	1010H
Failure (either of the redundant power supply modules)	1020H
Invalid power supply module	1030H
Power supply module configuration error	1031H
ROM write count error	1080H
Battery error	1090H
Memory card access error	1100H
SNTP clock setting error	1120H
Default gateway/gateway IP address error	1124H
Own node port number error	1128H
Open specification port number error	1129H
Specified IP address error	112DH
Connection establishment failed	112EH
Socket communications response send error	1133H
TCP connection timeout	1134H
IP address error	1152H
Connection number acquisition error	1155H
Receive buffer securement error	1157H
UDP/IP send failed	1165H
TCP/IP send failed	1166H
Unsend data send error	1167H
Redundant IP address error	1180H
PID operation error	11A0H to 11B8H
Module moderate error	1200H, 1210H
Another CPU module moderate error	1220H
Inter-module synchronization processing error	1240H, 1241H
Multiple CPU synchronization processing error	1260H, 1262H
Annunciator ON	1800H

Error name	Error code
Operation continuation error	1810H
Receive queue full	1830H
Receive processing error	1831H
Transient data error	1832H
Constant scan time error	1900H
Network configuration mismatch	1B00H
System consistency check error (operating status)	1B20H
Redundant system error	1B40H, 1B42H, 1B43H
Extension cable failure	1B48H, 1B4AH
Standby system CPU module error	1B60H, 1B61H
Tracking communications disabled	1B70H
Tracking communication error	1B71H, 1B78H
Tracking transfer error	1B80H, 1B81H, 1B82H
Redundant function module error	1BA0H
File name specification error	1BB0H
Program execution time error	1BC0H
System switching error	1BD0H, 1BD1H
Memory card error	2120H, 2121H
Module verification error	2400H, 2401H
Fuse blown error	2420H
Module major error	2441H, 2442H, 2450H
Another CPU module major error	2461H, 2462H, 2470H
Inter-module synchronization signal error	2610H
Multiple CPU synchronization signal error	2630H
I/O number or network number specification error	2800H, 2801H, 2802H, 2803H, 2804H, 2805H, 2806H, 2807H, 2810H
Device, label, or buffer memory specification error	2820H, 2821H, 2822H, 2823H, 2824H
File name specification error	2840H, 2841H, 2842H
Operation error	3400H, 3401H, 3402H, 3403H, 3404H, 3405H, 3406H, 3420H, 3421H, 3422H, 3423H, 3426H, 3430H, 3460H, 3461H, 34A0H

Point

When the write protect switch of the SD memory card is enabled, an error of the memory card access error, which is generated when an event history is held, can be cleared. Note that an error will not be detected again even if the event history is stored due the occurrence of event after resetting an error.

How to clear errors

Errors can be cleared in two ways:

■Using the engineering tool

Clear errors with the module diagnostics function of GX Works3. (📖 GX Works3 Operating Manual)

Point

The event history of error clear using the engineering tool is stored in the CPU module connected.

■Using SM/SD

Clear errors by operating SM/SD.

1. Check SD0 (Latest self-diagnostic error code) to identify what errors are detected.
2. Clear the cause of each of the currently detected continuation errors.
3. Turn on SM50 (Error reset) to clear the errors. When multiple continuation errors have occurred, all the errors are cleared at once.

Precautions

This section describes some precautions to take when using the error clear function:

- Since the function clears all of the currently detected continuation errors at once, errors that should not yet be cleared may be cleared.
- Use the RST instruction to reset each annunciator individually.
- Running the error clear function does not remove the cleared errors from the event history.
- The cause of an error which occurred in a module other than the target CPU module for the error clear cannot be eliminated even though the error is cleared using this function. For example, when "Module verification error" (error code: 2400) or "Module major error" (error code: 2450) occurred, the error cause cannot be eliminated even though the error is cleared in the CPU module using this function. To eliminate the error cause, clear the error of the target module and reset the CPU module.

6.4 Event History Function

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 492 FUNCTIONS

- When using the SIL2 Process CPU, refer to the following as well.

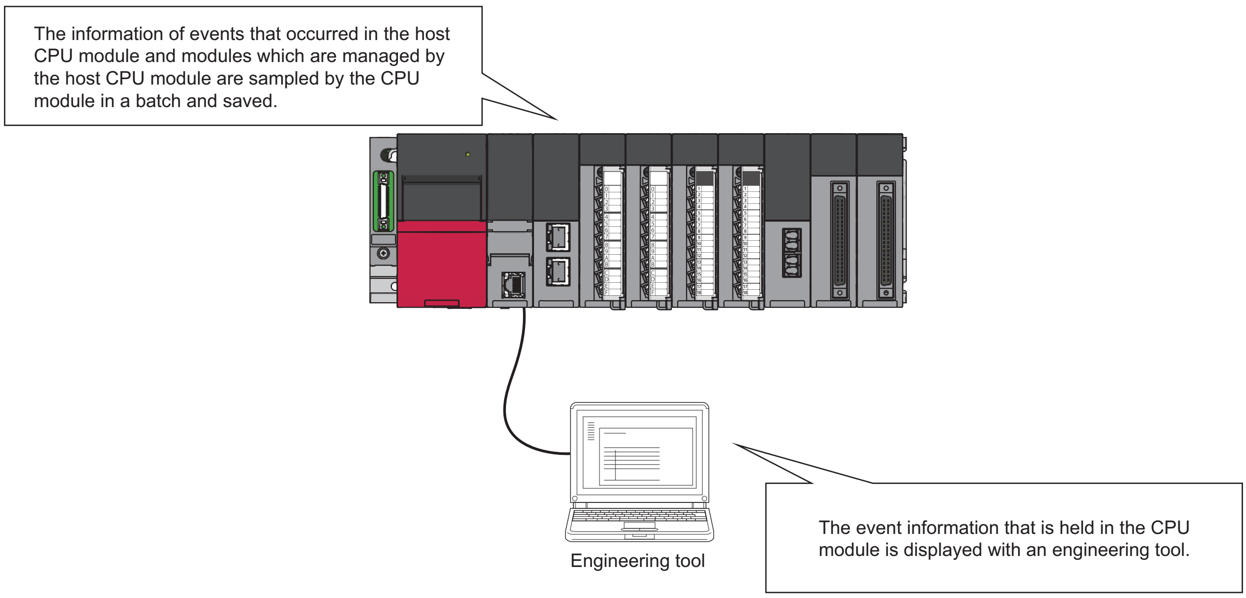
☞ Page 704 FUNCTIONS

- When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

The CPU module collects and stores event information from each module, such as errors detected by the module, operations performed on the module, device/label data write operations[*1], and network errors.[*2]

Once errors and operations are stored, they can be checked chronologically. This function helps to determine the causes of problems that have occurred in the equipment/devices, check the update status of the programmable controller control data, and detect unauthorized access.



*1 When "Save Device/Label Operations" is set in Event History Setting, device/label data write operations are saved. (☞ Page 154 Device/label data write operation saving)
Before using the device/label operation save setting, check the versions of the CPU module and engineering tool. (☞ Page 1139 Added and Enhanced Functions)

*2 If the CPU module is operated online using the engineering tool, there are times when processing performed automatically by the system is saved as an event.

Point

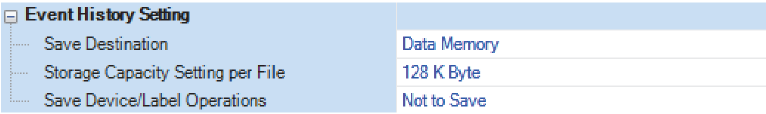
The event history information is constantly collected regardless of the operating state of the CPU module. There are occasions, however, when the event history information cannot be collected due to a major error in a module, a base unit error, a cable failure, or some other cause.

Event history setting

Under normal circumstances, the event history function can be used with its default settings and need not be manually configured. The storage memory and size settings for event history files can be changed as needed. (☞ Page 150 Event history file)

[CPU Parameter] ⇨ [RAS Setting] ⇨ [Event History Setting]

Window



Displayed items

Item	Description	Setting range	Default
Save Destination	Specify the storage location of event history files. (☞ Page 150 Storage memory)	• Data Memory • Memory Card*1	Data Memory
Storage Capacity Setting per File	Specify the storage capacity per event history file. (☞ Page 150 File size)	1 to 2048K bytes (in 1K bytes)	128K Byte
Save Device/Label Operations*2	Specify this item when saving device/label data write operations. (☞ Page 154 Device/label data write operation saving)	• Do not save • Save	Do not save

*1 It cannot be set in the R00CPU.
*2 This item is not displayed on the R00CPU, R01CPU, or R02CPU.

Point

For use in any of the following environments, use of an SD memory card for the save destination memory is recommended because the number of events that occur becomes large.
- When file data write operations and device/label data write operations are performed frequently
- When the communication status changes frequently
- When device data write operations are periodically performed from a programmable controller of another station or from another CPU module*3

The number of writes to the data memory is limited to 100000 times. In any of the above cases, when data memory is used for the save destination memory, be careful about the number of writes to data memory.

*3 When "Save Device/Label Operations" is set in Event History Setting. Note that when device data write operations are performed periodically, the number of events registered in the history becomes large, and event history logging may be restricted. (☞ Page 153 Event history logging restriction)

Logging of the event history

This section describes events saving for the event history.

Event history file

The storage memory and file size for event history files can be changed in event history setting. (☞ Page 149 Event history setting)

■Storage memory

Choose either the data memory or SD memory card.*1

If the storage memory is the SD memory card: when the write protect switch of the SD memory card is enabled, an event history will not be stored. (It is possible to read the event history file in SD memory card using the engineering tool.) Therefore, if the write protect switch of the SD memory card is changed into "Disabled → Enabled" during operation, the write error to the SD memory card occurs when the event for storing into event history is generated (SD memory card available event is registered) after the write protect switch is enabled. An error that has occurred can be checked by using the module diagnostics of the engineering tool immediately after its occurrence. However, these errors are not stored in the event history after powering off and on or reset of the CPU module.

*1 For the R00CPU, only the data memory can be selected as the storage memory.

■File size

The size for event history files can be changed in event history setting (☞ Page 149 Event history setting). If the storage size exceeds the specified size, records are deleted in order from the oldest one and the latest one is stored. An event history file size is obtained from the following calculation formula.

Event history file size = File header size + Event history management information size + (Number of records × Size per event history record)

Element	Size
File header size	20 bytes
Event history management information size	12 bytes
Size per event history record	40 bytes minimum*1

*1 Because the contents of detailed information may differ depending on the event to be saved or the detailed information may include a variable-length file name, the size per event history record is variable.

The number of events to be saved in the event history file differs depending on the event type to be saved. When the event history file size is 128K bytes (default), 1365 events can be registered if programs (whose program name is 8 characters (12 characters including a period and extension)) are written to the CPU module using the engineering tool.

[Calculation formula]

- 128 bytes × 1024 = 131072 bytes
- 131072 bytes - (20 bytes + 12 bytes) = 131040 bytes
- 131040 bytes ÷ 96 bytes = 1365 events

Ex.

The table below shows the size of each element when 100 programs (whose program name is 8 characters (12 characters including a period and extension)) are written to the CPU module by the following operating procedure.

Operating procedure

1. Turn on the power in the STOP state.
2. Write the system parameter, CPU parameter, module parameter, and 100 programs (whose program name is 8 characters (12 characters including a period and extension)) to the CPU module using the engineering tool.
3. Switch the CPU module to the RUN state.

Element		Size (byte)
File header		20
Event history management information		12
Event to be saved	Power-on and reset	40*1
	Operating status change (STOP)	40
	Writing files/folders (SYSTEM.PRM)	96
	Writing files/folders (CPU.PRM)	88
	Writing files/folders (UNIT.PRM)	88
	Writing files/folders (MAIN_001.PRG to MAIN_100.PRG)	9600
	Operating status change (RUN)	40
Total		10024

*1 The size will be 56 bytes for the process CPU with firmware version "06" or later.

6

■When files are created

An event history file is created when:

- The CPU module is turned off and on (if there is no event history file or after the event history settings are changed).
- The CPU module is reset (if there is no event history file or after the event history settings are changed).
- Initialization of the SD memory card (when no event history file exists)*1
- Write of parameters (when no event history file exists, or after an event history setting is changed).

*1 When a parameter is stored in the data memory, the event history file is created on the SD memory card, according to the event history setting.

Point

When a new event history file is generated, the "Event history file generation" (00420) is logged. When a new event history file is generated during the logging restriction of the event history, the "Event history logging restriction" (00421) is also logged.

The following table shows how the event history is treated depending on operation.

Operation	Operation for the event history
Memory initialization	When this event occurs, the event history is stored into the internal memory. If the internal memory reaches the maximum number of event history records it can store, all subsequent events are lost. (☞ Page 152 Loss of event history information)
Event history creation	The event history, which has been stored in the internal memory during absence of the event history file, is stored into the data memory or the SD memory card (If any event was lost, it is logged as "*HST LOSS*").

The following table shows how the event history is treated at removal and installation of an SD memory card when the SD memory card is specified as the storage memory.

Operation	Operation for the event history
Removal of the SD memory card	When this event occurs, the event history is stored into the internal memory. If the internal memory reaches the maximum number of event history records it can store, all subsequent events are lost. (☞ Page 152 Loss of event history information)
Installation of the SD memory card	The event history, which have been stored in the internal memory during absence of the SD memory card, is stored to the SD memory card. If the re-inserted SD memory card contains an event history file of the same file size, the CPU module continues to store the event history information. If the file size is different, the CPU module removes the existing event history file and creates a new event history file.

■When parameters take effect

Any changed parameters take effect when:

- The CPU module is powered off and on
- The CPU module is reset

Point

Any changed parameters written in the storage memory with the CPU module in the STOP state does not take effect when the CPU module operating state is changed from STOP to RUN. In this case, the changed parameters will take effect the next time the CPU module is powered off and on or is reset.

Loss of event history information

If events are detected frequently, or the CPU module is powered off or reset immediately after the detection of events, some events may not be collected and lost. When event loss occurs, "*HST LOSS*" appears in the "Event Code" field of the engineering tool.

Event history logging restriction

When the number of minor events from the CPU module or intelligent function module such as link-up and link-down exceeds the upper limit value, event history logging is restricted (stopped).[*1*2]

When the number of events for which logging is restricted decreases to the lower limit value or below, event history logging is restarted.

The error codes of moderate and major errors that occur in the CPU module are logged in the event history.

Item	Condition
Upper limit value	600 events/minute
Lower limit value	300 events/minute

*1 For models and firmware versions that support the event history logging restriction, refer to the following.
☞ Page 1139 Added and Enhanced Functions

*2 Event history logging is not restricted (stopped) in CPU module events for the following CPU modules.
· The R00CPU, R01CPU, and R02CPU with firmware version "11" or earlier
· The programmable controller CPU with firmware version "43" or earlier (except for R00CPU, R01CPU, and R02CPU)
· The Process CPU with firmware version earlier than "22"
· The Safety CPU with firmware version earlier than "23"

■Events not supported by the logging restriction

The following events are not subjected to the logging restriction.[*1]

- Events of the self-diagnostics for major errors from the intelligent function module

*1 For the following CPU modules, CPU module events are not subjected to the logging restriction, either.
· The R00CPU, R01CPU, and R02CPU with firmware version "11" or earlier
· The programmable controller CPU with firmware version "43" or earlier (except for R00CPU, R01CPU, and R02CPU)
· The Process CPU with firmware version earlier than "22"
· The Safety CPU with firmware version earlier than "23"

■How to check whether event history logging is restricted

When event history logging is restricted, SM1464 (Event history logging restriction status) turns on. SM1466 turns on when the event history logging of the CPU module is restricted and the event category is error (minor error), and SM1467 turns on when the event history logging of the CPU module is restricted and the event category is information or warning.

Modules on which event history logging is restricted can be identified in SD1464 to SD1467 (Module information on event history logging restriction).

Modules from which event history information is collected

Event history information is collected from the CPU module and other modules installed on the same base unit (i.e., the main base unit plus any additional extension base units). Event history information may or may not be collected from devices on the network depending on the specifications of the network modules used to connect to them. Refer to the manuals of the respective modules for more information including the coverage of event history collection regarding devices on the network.

For a multi-CPU system, note that each CPU module logs only events detected on the modules under its control. On Q series modules, only errors of which even type is "System" are stored. (The above applies only to Q series modules that support the module error collection (function).)

Events logged by the CPU module

Information logged in the event history includes operation initiator and other detailed information for troubleshooting purposes.

For events that are logged in the event history on the CPU module, refer to the event history. (☞ Page 905 Event List)

Device/label data write operation saving

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

• This function cannot be used in the R00CPU, R01CPU, and R02CPU.

When "Save Device/Label Operations" is enabled in Event History Setting, device/label data write operations are saved. (☞ Page 149 Event history setting)

Restriction

• Before using the device/label operation save setting, check the versions of the CPU module and engineering tool. (☞ Page 1139 Added and Enhanced Functions)

• When multiple device points or multiple devices/labels are written, some device/label names and written values may not be shown.

■Target operations

The device/label data write operations of the following operations and functions are saved.

Operation and function		Remarks
Operation from the engineering tool	Changing current values of devices and labels from the watch window or with the other methods	The written values are saved up to 128 words.
	Changing the status (active/inactive) of blocks and steps of an SFC program	
	Writing data to the device memory	
Operation using SLMP	Writing device/label data	
Operation using a GOT	Writing device/label data	
Other operations	Writing device data from a programmable controller of another station*1, reading data from programmable controller of another station (with read notification)*2, and writing device data from another CPU module	
	Writing device data using the simple CPU communication function	
	Writing device data using the predefined protocol support function	

*1 When writing the data with write notification, the write notification device is also saved.

*2 The read notification device is saved.

• Applicable commands for writing device/label data through SLMP

The following table shows the applicable commands for writing device/label data through SLMP.

Type	Operation	Command	Subcommand	Remarks
Device	Write	1401	00□1 00□3	The written values are saved up to 128 words.
			00□0 00□2	
	Write Random	1402	00□1 00□3	The number of data sets to be saved varies depending on the device specified.
			00□0 00□2	
	Write Block	1406	00□0 00□2	The written values are saved up to 128 words per block. The number of data sets to be saved varies depending on the devices specified.
Label	Array Label Write	141A	0000	The written element data are saved up to 128 words. The number of data sets to be saved varies depending on the number of array points specified.
	Label Write Random	141B	0000	The number of data sets to be saved varies depending on the label assignment destination. The written strings are saved up to 128 words.
Extend Unit	Write	1601	0000	The written values are saved up to 128 words.

- Applicable instructions for writing device data from a programmable controller of another station or from another CPU module

The following table shows the applicable instructions for writing device data from a programmable controller of another station or from another CPU module.

Major classification	Classification		Instruction name	Remarks
Writing device data from another station	Module dedicated instructions	Writing data to programmable controller of another station	JP.WRITE, GP.WRITE	The written values are saved up to 128 words.
		Writing data to program controller of another station (with write notification)	JP.SWRITE, GP.SWRITE	
		Reading data from programmable controller of another station (with read notification)	JP.SREAD, GP.SREAD	
		Writing data to target station	J(P).RIWT, G(P).RIWT	
Writing device data from another CPU module	Multiple CPU dedicated instructions	Writing device data to another CPU module	D(P).DDWR, M(P).DDWR	

*3 For the display of detailed information, refer to the following.

☞ Page 157 Detailed information

■Target device

The following table shows the target devices. The write operations are also saved when writing data with index modification or indirect specification.

Type	Device
Bit device	X, DX, Y, DY, M, L, F, SM, V, B, SB, T (contact), T (coil), ST (contact), ST (coil), C (contact), C (coil), LT (contact), LT (coil), LST (contact), LST (coil), LC (contact), LC (coil), FX, FY, Jn\X, Jn\Y, Jn\SB, Jn\B, BLn\S, SA\X, SA\Y, SA\M, SA\B, SA\T(contact), SA\T(coil), SA\ST(contact), SA\ST(coil), SA\C(contact), SA\C(coil)
Word device	T (current value), ST (current value), C (current value), D, SD, W, SW, RD, R, ZR, Z, FD, Un\G, Jn\W, Jn\SW, U3En\G, U3En\HG SA\D, SA\W
Double-word device	LT (current value), LST (current value), LC (current value), LZ

Note that for write operation by specifying label names, label memory areas (such as GV, LV, LLV, and UV) may be displayed as device names. For details on the label memory, refer to the following.

📖 GX Works3 Operating Manual

■Target label

The labels to be saved are all labels that can be specified by the engineering tool and SLMP.

■Detailed information

The detailed information of an event history displays a device/label name, the number of write points, and written values.
(☞ Page 906 Detailed information)

Item	Name displayed on window	Description
Device/Label	Device name information	A specified device name is displayed.
	Device name information (user-specified)	
	Device name information (target device)	The name and address of a device actually accessed is displayed. When a device is specified with index modification, if the accessed device is within the specified device range, the device name is displayed. If out of the specified device range, the address is displayed. When a device is specified with indirect specification, the accessed address is displayed.
	Label name information	A specified label name is displayed.
	Label name information (user-specified)	
	Target bit No.	When a bit of a word device is specified with indirect specification, the accessed bit number is displayed. The same applies when a bit is specified with index modification and the accessed bit is out of the specified device range, the accessed bit number is displayed.
Number of write points	Number of points	The number of write points is displayed.
Written value	Written value	The written values for the number of write points is displayed.

Viewing the event history

The event history can be viewed using the menus of the engineering tool. For operating procedures and how to interpret the displayed information, refer to the following:

📖 GX Works3 Operating Manual

Clearing the event history

The event history can be cleared from the event history window. Once the event history is cleared, the CPU module deletes all the event history information stored in the specified storage memory. For operating procedures and other details, refer to the following:

📖 GX Works3 Operating Manual

Point

When clearing of the event history is executed, the "Event history clear" (20200) is logged. When an event history file is executed clear during the logging restriction of the event history, the "Event history logging restriction" (00421) is also logged.

Precautions

Clearing the event history during execution of another function

No event history can be cleared during execution of the following functions. Check that the following functions are not being executed and then clear the event history.

- CPU module data backup/restoration function
- iQ Sensor Solution data backup/restoration function

Reading the event history during execution of another function

No event history can be read during execution of the following function. Check that the following function is not being executed and then read the event history.

- CPU module data restoration function

Logging of event history

When event history logging is restricted, the relevant special relay (☞ Page 153 How to check whether event history logging is restricted) turns on. Eliminate the event causes from the target module. The target module can be identified in SD1464 to SD1467 (Module information on event history logging restriction).

Whether event history logging is restricted or not can also be checked in "Event history logging restricted" (00421) in "Event History" window of the engineering tool.

6

Execution of other functions while event history logging is restricted

When the CPU module is restored by turning on SM1354 while event history logging is restricted, event history logging is restarted after the completion of the restoration.

When event history logging is restarted and event history from a module is frequently logged again, logging the history of events occurring on the target module is stopped (restricted).

Change of the event history setting

The following table shows the operation when the event history setting is changed after the CPU module starts, and written to the CPU module.

Changed item	Operation
Save Destination	With the original file left as is, an event history file is generated in the enabled save destination (drive).
Storage Capacity Setting per File	The event history file is discarded and regenerated.
Save Device/Label Operations	With the original file left as is, EVENT.LOG is generated when the file is not saved, and EVEN2.LOG is generated when the file is saved.

Therefore, to save a past event history, before changing the event history setting, save data by using the [Create File] button in the event history window of the engineering tool.

[Diagnostics] ⇨ [System Monitor] ⇨ [Event History] button ⇨ [Create File] button

Device data write operation from another station or another CPU module

When "Save Device/Label Operations" is set in Event History Setting, many event updates occur. Depending on the timing, events that occurred cannot be checked with the engineering tool.

6.5 Program Cache Memory Auto Recovery Function

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

If the contents of memory of the CPU module have been rewritten by itself due to the factors such as excessive electrical noise, the program cache memory recovers the corresponding areas automatically during the run of the program. This function becomes active with RUN state of the CPU module and works during the run of the program. No special settings are required as the system performs this task automatically.

7 REMOTE OPERATION

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 492 FUNCTIONS

- When using the SIL2 Process CPU, refer to the following as well.

☞ Page 704 FUNCTIONS

- When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

The operating status of the CPU module can be changed using an engineering tool or program, or dedicated instructions from the module. The following types of remote operation are available:

- Remote RUN/STOP
- Remote PAUSE
- Remote RESET

7.1 Remote RUN/STOP

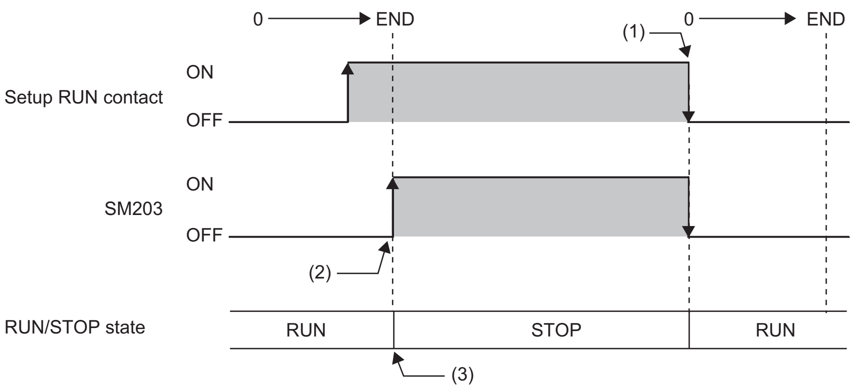
This function externally changes the CPU module status to RUN or STOP with the RUN/STOP/RESET switch of the CPU module set to RUN. This function is useful to RUN or STOP a CPU module located in an unaccessible place, or to RUN or STOP a CPU module located on the control panel by using an external signal.

Executing method

The following methods are available to execute remote RUN/STOP:

Using a contact

Set RUN contact in the RUN-PAUSE contact setting. (☞ Page 163 Setting RUN-PAUSE Contacts) When the END processing for the scan where the RUN contact is turned on is executed, SM203 (STOP contact) turns on to change the CPU module status to STOP and stop the operation. When the RUN contact is turned off, the CPU module exits the STOP state and resumes the program operation from the step 0.



(1) When the RUN contact is turned off, the CPU module status changes to RUN and the program is resumed from the step 0.
(2) SM203 turns on when the END processing for the scan where the set RUN contact is turned on is executed.
(3) When the RUN contact is turned on, the CPU module executes the operation until the END instruction and enters the STOP state.

Using an engineering tool

Perform remote RUN/STOP with the remote operation of the engineering tool. (📖 GX Works3 Operating Manual)

By an external device using SLMP

Perform remote RUN/STOP with the SLMP command. (📖 SLMP Reference Manual)

Using the module dedicated instructions

Perform remote RUN/STOP with the dedicated instructions of a network module. (📖 MELSEC iQ-R Programming Manual (Module Dedicated Instructions))

Precautions

This section describes the precautions on using remote RUN/STOP.

- When remote RUN is performed during execution of the data logging function, it may fail. In that case, wait for a while and retry remote RUN. If remote RUN still cannot be executed, check whether remote RUN is acceptable and retry remote RUN (☞ Page 250 About remote operation)
- When the RUN contact, which is specified in the RUN-PAUSE contact setting, is turned off during execution of the data logging function, it may take time to return to the RUN state.

7.2 Remote PAUSE

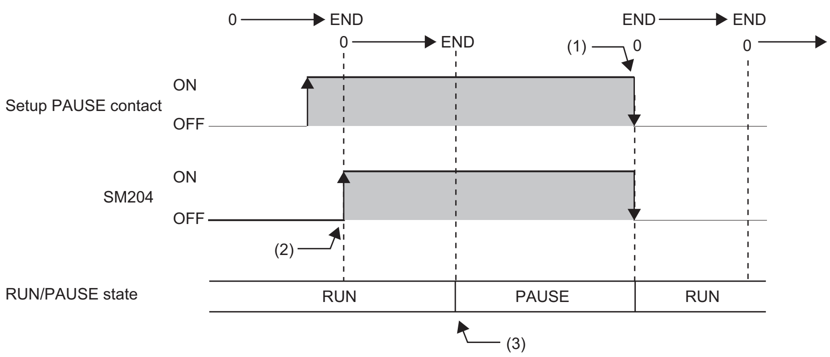
This function externally changes the CPU module status to PAUSE with the RUN/STOP/RESET switch of the CPU module set to RUN. Use this in a process control and other situations to keep the CPU module status in RUN even when changing the status of the output (Y) to STOP.

Executing method

The following methods are available to execute remote PAUSE:

Using a contact

Set PAUSE contact in "RUN-PAUSE Contact Setting". (☞ Page 163 Setting RUN-PAUSE Contacts) When the END processing for the scan where the PAUSE contact is turned on is executed, SM204 (PAUSE contact) turns on. When the CPU module executes until the END instruction for the scan next to that where the PAUSE contact is turned on, its status changes to PAUSE, and the operation stops. When the PAUSE contact is turned off, the CPU module exits the PAUSE status and resumes the program operation from the step 0.



(1) When the PAUSE contact is turned off, the CPU module exits the PAUSE state and resumes the program operation from the step 0.

(2) SM204 turns on when the END processing for the scan where the set PAUSE contact is turned on is executed.

(3) When the CPU module executes until the END instruction for the scan next to that where the PAUSE contact is turned on, its status changes to PAUSE, and the operation stops.

Using an engineering tool

Perform remote PAUSE with the remote operation of the engineering tool. (📖 GX Works3 Operating Manual)

By an external device using SLMP

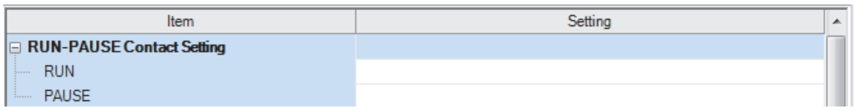
Perform remote PAUSE with the SLMP command. (📖 SLMP Reference Manual)

7.3 Setting RUN-PAUSE Contacts

RUN-PAUSE contacts can be set. RUN-PAUSE contacts are used to perform remote RUN or STOP, or remote PAUSE using a contact.

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [RUN-PAUSE Contact Setting]

Window



Displayed items

Item	Description	Setting range	Default
RUN	Sets the contact that controls RUN for the CPU module.	• R00CPU, R01CPU, and R02CPU: X0 to X1FFF • Other CPU modules: X0 to X2FFF	—
PAUSE	Sets the contact that controls PAUSE for the CPU module.	• R00CPU, R01CPU, and R02CPU: X0 to X1FFF • Other CPU modules: X0 to X2FFF	—

Point

When setting the PAUSE contact, be sure to also set the RUN contact. (The PAUSE contact cannot be set alone.)

7.4 Remote RESET

This function externally resets a CPU module in the STOP state (including that stopped due to an error). Even when the RUN/STOP/RESET switch of a CPU module is set to RUN, the CPU module can be reset in the STOP state.

Point

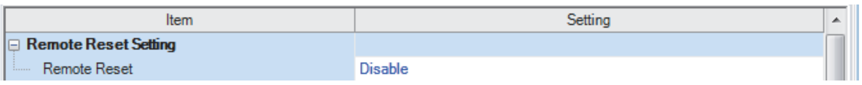
To perform the remote RESET operation when "Execution Target" of the remote operation is set to "All Stations Specified" or "Specify Group No.", ensure that the CPU module set in "Specify Connection Destination Connection" of the engineering tool is in the STOP state. (Even if a CPU module in the STOP state exists on the specified network, the remote RESET does not operate.)

Enabling remote RESET

Performing remote RESET requires setting for enabling remote reset.

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Remote Reset Setting]

Window



Displayed items

Item	Description	Setting range	Default
Remote RESET	Sets whether or not to enable the remote RESET.	• Disable • Enable	Disable

Executing method

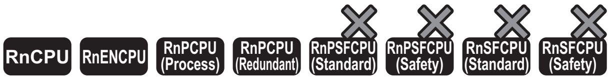
Using an engineering tool

Perform remote RESET with the remote operation of the engineering tool. (GX Works3 Operating Manual)

By an external device using SLMP

Perform remote RESET with the SLMP command. (SLMP Reference Manual)

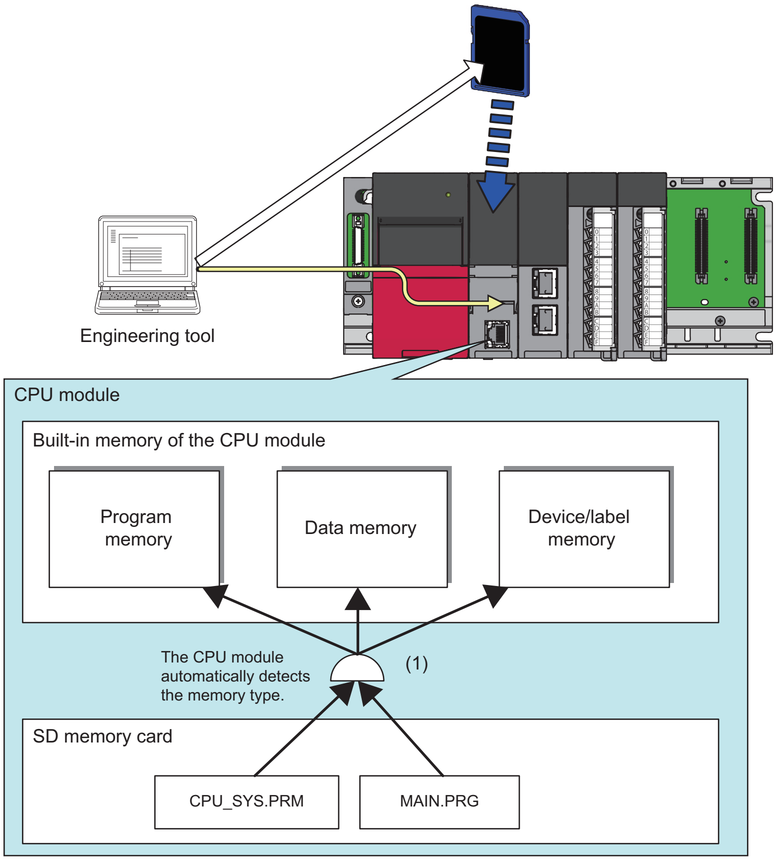
8 BOOT OPERATION



- This function cannot be used in the R00CPU.
- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 492 FUNCTIONS

The files stored on the SD memory card are transferred to the storage memory which is automatically determined by the CPU module when the CPU module is powered off and on or is reset.



(1) The boot operation is executed according to the boot file setting when the CPU module is powered on or is reset.

8.1 Boot Operation Procedure

The boot operation procedure is as follows.

1. Configure the boot setting. (☞ Page 167 Configuring the Boot Setting)
2. Insert an SD memory card to the CPU module.
3. Write the boot setting and boot file to the SD memory card. (☞ Page 168 Writing Boot Settings and Boot Files)
4. Power off and on, or reset the CPU module.

8.2 Specifiable File Types

The files which can be booted are as follows.

- System parameter
- CPU parameter
- Module parameter
- Module extension parameter
- Module extension parameter for protocol setting
- Remote password
- Global label setting file
- Initial global label value file
- Program file
- Initial local label value file
- FB/FUN file
- Device comment file
- Initial device value file

8.3 Allowed Maximum Number of Boot Files

The maximum number of boot file settings which can be specified is 512. However, because more than one file is bootable for a single setting, the maximum number of boot files is the same as the number of files which can be stored in the transfer destination memory.

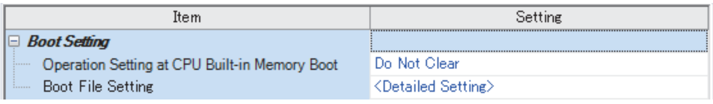
8.4 Configuring the Boot Setting

Configure the necessary settings for the boot operation.

[Memory Card Parameters] ⇨ [Boot Setting]

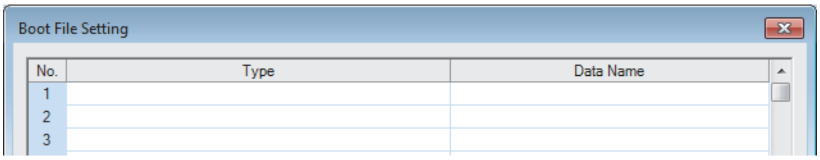
Operating procedure

"Boot Setting" window



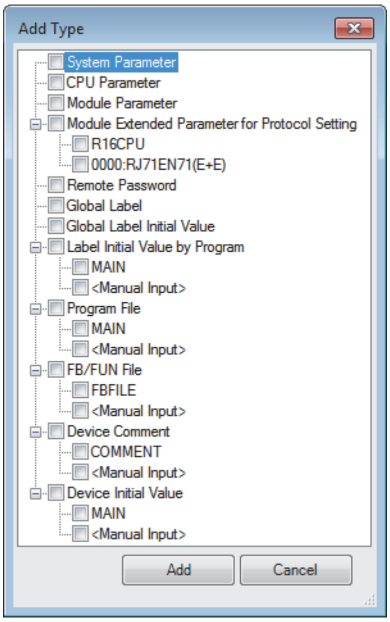
1. Click "Detailed Setting" on the "Boot File Setting" window.

"Boot File Setting" window



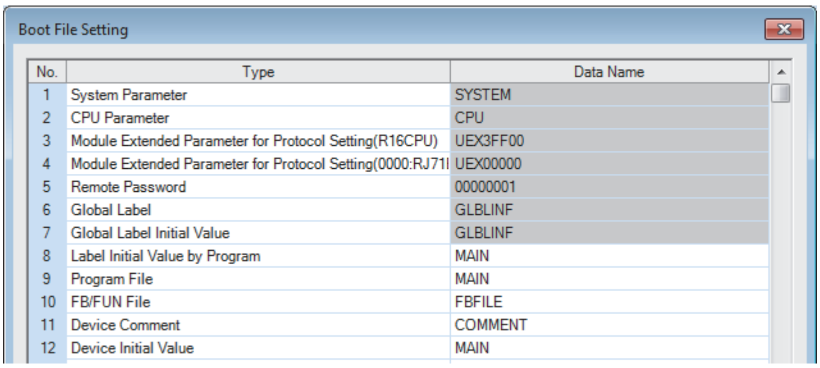
2. Click the "Type" column. The maximum number of boot files that can be specified is the same as the number of files that can be stored in the storage memory.

"Add Type" window



3. Select type for the boot file. (Multiple selection is possible.)

"Boot File Setting" window



4. Set the data name (file name).

8

Displayed items

Item	Description	Setting range	Default
Operation Setting at CPU Built-in Memory Boot	Sets whether or not to clear the CPU built-in memory (program memory and data memory) upon file transfer from the SD memory card.	• Do Not Clear • Clear	Do Not Clear
Boot File Setting	Sets the file used for boot operation from the SD memory card.	—	—

8.5 Writing Boot Settings and Boot Files

Use the following functions of the engineering tool.

Function	Description
Write to PLC	Use this function to write the data to the SD memory card inserted into the CPU module. Click [Online] ⇨ [Write to PLC] on the menu bar.
Write to Memory Card*1	Use this function to write the data to the SD memory card inserted into the computer. Click [Tool] ⇨ [Memory Card] ⇨ [Write to Memory Card] on the menu bar.

*1 The engineering tool with version 1.070Y supports this function.

Point

The boot settings and boot files cannot be written with a user-data-write operation of the engineering tool.

For details, refer to the following.
GX Works3 Operating Manual

8.6 Operation When Security Functions Are Enabled

This section describes the operation when security functions are enabled.

When a security key is set

When a security key is set to the boot target program file and the security key of the program file does not match with that of the CPU module, a boot error occurs. Also, when no security key is written to the CPU module, a boot error occurs as well.

Security key of boot target program file	Security key of CPU module	Security key match/mismatch	Boot program execution
Set	Written	Match	Execute
	Written	Not match	Not execute (boot error)
	Not written	—	Not execute (boot error)

When a file password is set

If a file password is set on both the source boot file and destination file, the file can be transferred only when the passwords match. Furthermore, the file transfer does not work if a file password is set only on either one.

Transferring boot file		Transferred boot file		Password match/ mismatch	Transfer
File	File password setting	File	File password setting		
Existing	Set	Existing	Set	Match	Yes
				Mismatch	No
			Not set	—	No
		Not set	—		Yes
	Not set	Existing	Set		No
			Not set		Yes
		Not set	—		Yes

8.7 Precautions

This section describes the precautions when using the boot operation.

- When a parameter file is set as boot file, it overwrites the parameter file inside the transfer destination CPU module. In addition, if a parameter file is not set as boot file even when stored on the SD memory card, the CPU module operates in accordance with the settings in the parameter file inside it.
- If a program in the program memory is changed online during boot operation, the change is also reflected in the transfer source program on the SD memory card.
- Note the model of the program written on the SD memory card (program specified in the boot file settings) and the model of the CPU module must be the same.
- For the boot file setting, check that the setting is consistent with the program/label setting. When the file is read from the programmable controller after the boot operation without consistency, conversion may not function properly due to lack of consistency in project data.

8

9 MONITOR FUNCTION

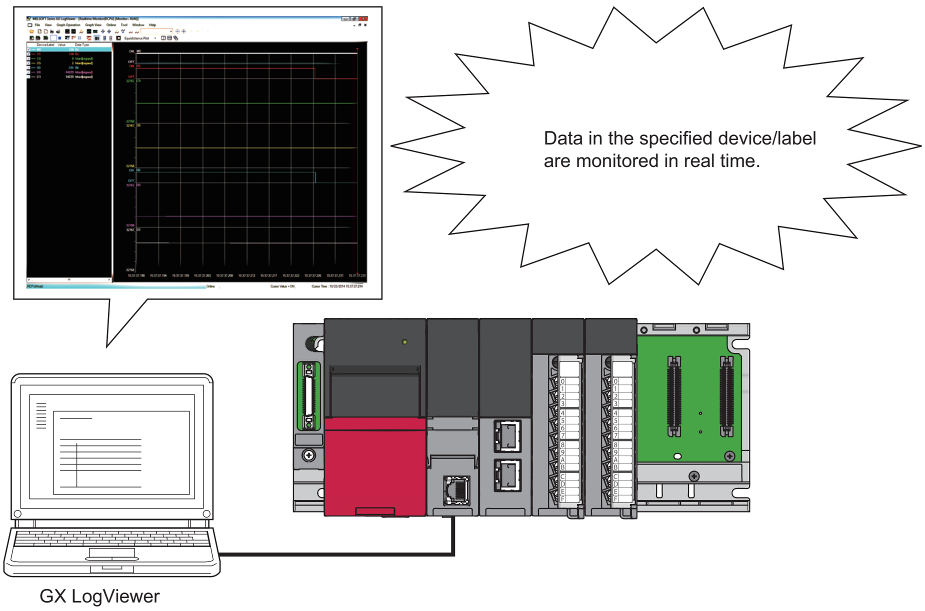
This chapter describes the functions for checking the CPU module operation.

Item	Description	Reference
Circuit monitor	Checks the status of the running program on the program editor.	GX Works3 Operating Manual
Device/buffer memory batch monitor	Checks the current values of the device and buffer memory in a batch.	
Watch	Registers a device and label and checks the current values.	
Program monitor list	Checks the processing time of the running program.	
Interrupt program monitor list	Checks the number of executions of the interrupt program used in the program.	
Real-time monitor	Monitors the contents of a specified device/label of the CPU module in real time at a specified interval or a desired timing.	GX LogViewer Version 1 Operating Manual
Scan time measurement	Measures the processing time of the set program section and displays it on the engineering tool.	• Page 171 Scan Time Measurement • GX Works3 Operating Manual
Scan time clear	Clears values of the scan time at a desired timing without stopping the CPU module.	• Page 942 Fixed scan function information • GX Works3 Operating Manual
Specified program monitor	Checks (monitors) device/label values in the program specified by the engineering tool on the program editor and changes current values.	• Page 176 Specified Program Monitor • GX Works3 Operating Manual

9.1 Real-Time Monitor Function

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

This function monitors the contents of a specified device/label of the CPU module in real time at a specified interval or a desired timing. The function can be set with GX LogViewer, where the value changes of a specified device/label can be shown graphically. Saving the set data and displayed graphs makes it possible to simplify the settings and check the graphs at a later time.



Restriction

• Before executing the function, check the versions of the CPU module and GX LogViewer used. (☞ Page 1139 Added and Enhanced Functions)

For details on the function, refer to the following.

📖 GX LogViewer Version 1 Operating Manual

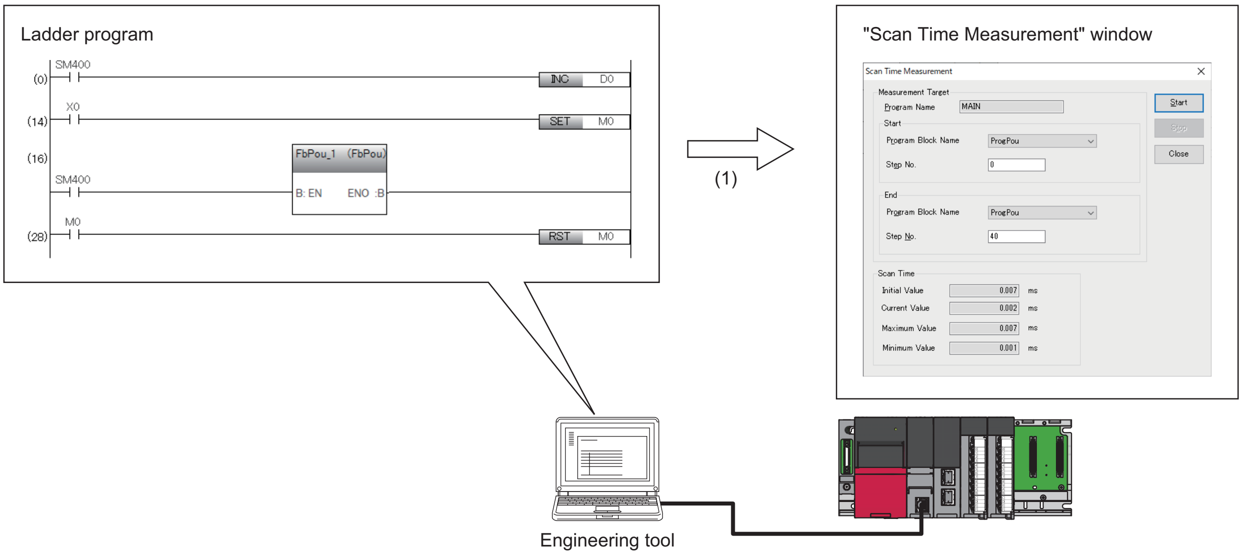
9.2 Scan Time Measurement

RnCPU RnENCPU ~~RnPCPU (Process)~~ ~~RnPCPU (Redundant)~~ ~~RnPSFCPU (Standard)~~ ~~RnPSFCPU (Safety)~~ RnSFCPU (Standard) ~~RnSFCPU (Safety)~~

• When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

This function measures the processing time of the set program section and displays it on the engineering tool. The time required for the subroutine and interrupt programs can be measured.



(1) Displays the scan time of the specified range.

Restriction

Before using scan time measurement, check the versions of the CPU module and engineering tool used.
(☞ Page 1139 Added and Enhanced Functions)

Target programs

The target programs include ladder programs and SFC programs.
For details on operation when each program is measured, refer to the following.
📖 GX Works3 Operating Manual

Execution procedure

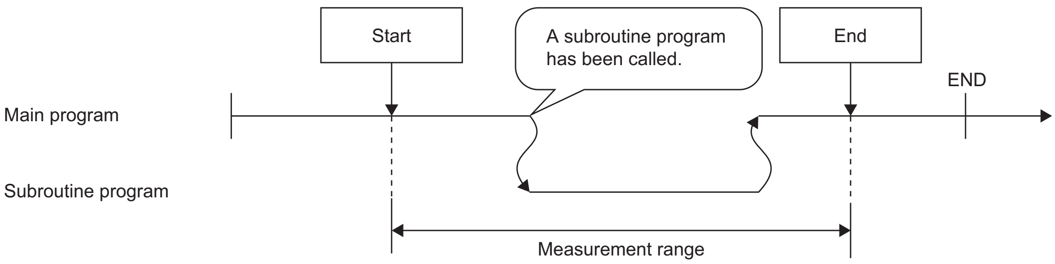
While the target program is displayed, execute this function from the "Scan Time Measurement" window.
For details on the menu operation, refer to the following.
📖 GX Works3 Operating Manual

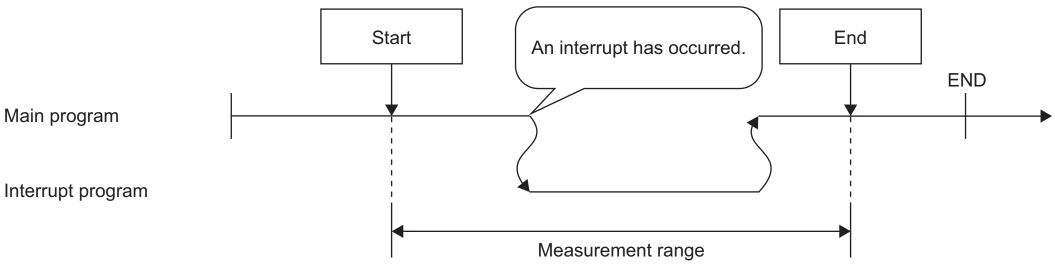
Minimum unit of measurement time

The minimum unit of measurement time is 1μs (0.001ms).

Operating specifications

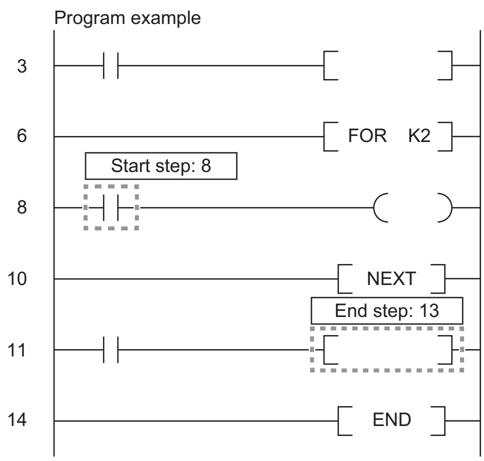
- The scan time is measured while the operating status of the CPU module is the RUN state. When the CPU module is in the STOP state, the current value is 0ms and other values are the ones immediately before the CPU module enters the STOP state. (The values are not updated while the CPU module is in the STOP state.)
- The measurement starts from the start step of the scan subsequent to the scan where the start request receives.
- When a subroutine program is executed in the measurement range specified by the main program, the execution time of the subroutine program is also included in the measurement range. Also, when an interrupt program, fixed scan execution type program, or event execution type program triggered by occurrence of an interrupt is executed in the measurement range specified by the main program, the execution time of an interrupt program, fixed scan execution type program, or event execution type program triggered by occurrence of an interrupt is added accordingly to the scan time of the measurement range.





- When the start step is specified between the FOR and NEXT instructions, scan time measurement is performed with the measurement range from the execution of the first start step instruction to the execution of the end step instruction.

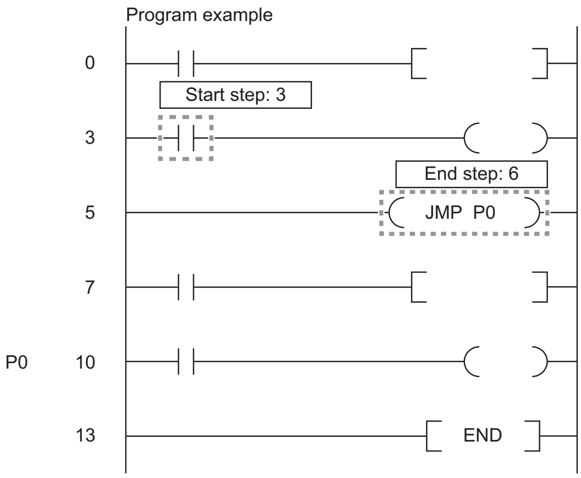
• When the start step is 8 and the end step is 13



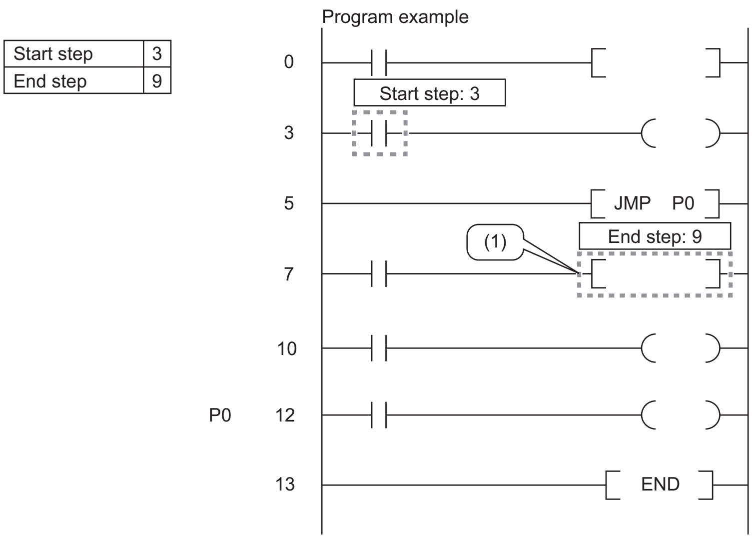
Precautions

- Use the program written to the CPU module. If the program on the engineering tool and the program written to the CPU module do not match, an error occurs. However, if the program is changed using the engineering tool and restored again, the program is judged to be different from the program in the CPU module. In that case, the program must be read from the CPU module before being executed.
- The scan time measurement cannot be executed from multiple external devices simultaneously. Any scan time measurement executed after the first one will result in an error. The scan time measurement executed first will continue to measure normally.
- Measurement may not be executed depending on the program configuration. In that case, the CPU module returns a response without updating the value, and the displayed value is not updated. Therefore, specify the start and end steps in a program configuration in which measurement start and stop can be executed within one scan. The following shows an example of when measurement cannot be executed.

■When the branch instruction such as the JMP instruction is specified for the end step
- When the start step is 3 and the end step is 6



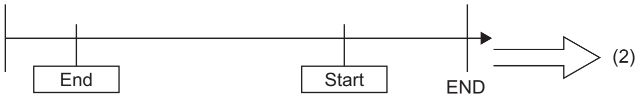
■When the END instruction is executed without executing the end stop after the execution of the start step
- When the start step is 3 and the end step is 9



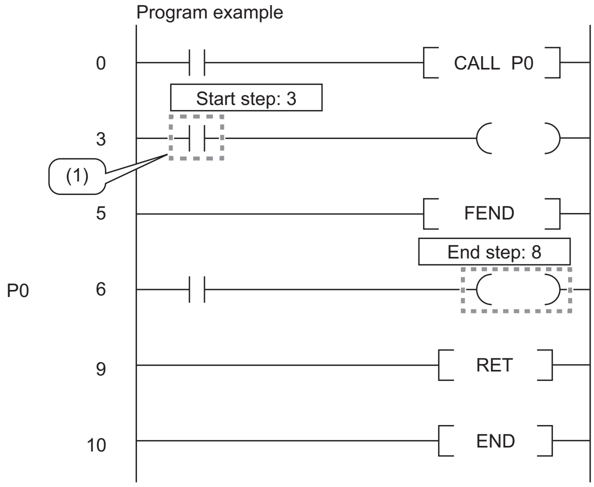
(1) The end stop is not executed due to the JMP instruction.
(2) Measurement cannot be performed because only the start step is executed.

■When the end step is executed before the start step

• When the start step is 3 and the end step is 8



Start step	3
End step	8

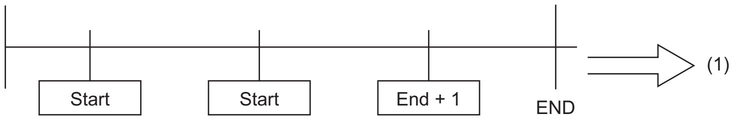


(1) The start step is executed after the execution of the end step by the CALL instruction.

(2) Because the end step is not executed after the execution of the start step, measurement cannot be performed.

■When the END instruction is executed by executing the end step without executing the start step

■When the start step is repeatedly executed (except when the start step exists between the FOR and NEXT instructions (Page 172 Operating specifications))

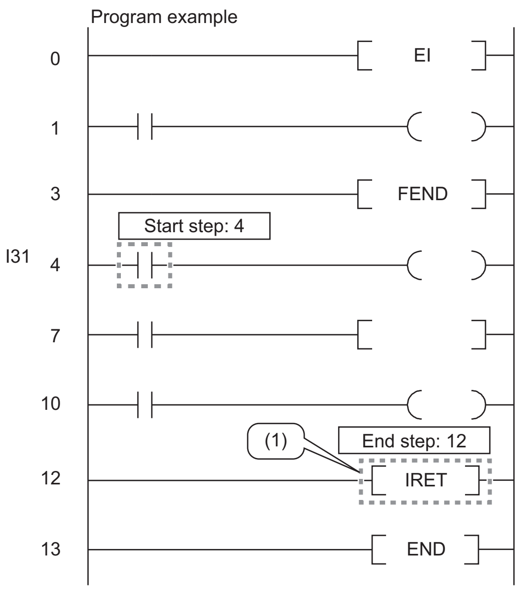


(1) Measurement cannot be performed when the start step is executed repeatedly.

■When the IRET instruction, FEND instruction, BREAK instruction, or RET instruction is specified for the end step

• When the start step is 4 and the end step is 12

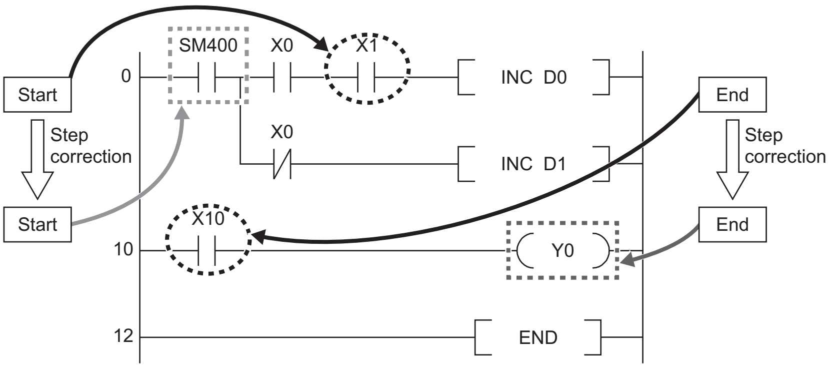
Start step	4
End step	12



(1) When the scan time in an interrupt program is measured, do not specify the IRET instruction for the end step.

• When the start step is not the head step of the ladder block containing the specified step, or when the end stop is not the last step of the ladder block containing the specified step, correct the start/end step with the engineering tool.

Program example



• If the program change is executed by the online program change or the file batch online change while scan time measurement is being executed, scan time measurement is interrupted. The program change by the online program change or the file batch online change is completed normally.
• If scan time measurement is executed while the program change is being executed by the online program change or the file batch online change, scan time measurement cannot be performed. The program change by the online program change or the file batch online change is completed normally.

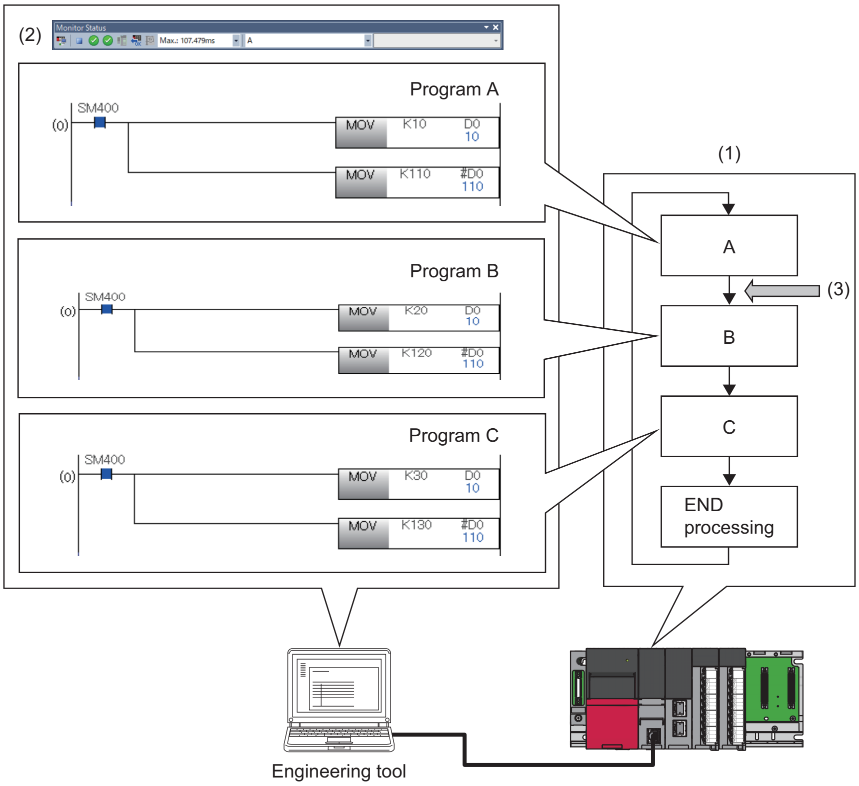
9.3 Specified Program Monitor

RnCPU RnENCPU ~~RnPCPU (Process)~~ ~~RnPCPU (Redundant)~~ ~~RnPSFCPU (Standard)~~ ~~RnPSFCPU (Safety)~~ ~~RnSFCPU (Standard)~~ ~~RnSFCPU (Safety)~~

- When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

This function checks (monitors) device/label values in the program specified by the engineering tool on the program editor and changes current values. Device/label values after the specified program is executed can be checked with this function, enabling debugging on a program basis.



(1) The CPU module is executing three programs whose program names are "A", "B", and "C". (The order of program execution is A → B → C → END processing → A → B → ...)
(2) Specify program A using the monitor status bar.
(3) Global device and global label values, and file register values of each program for program A can be checked. For current value change, values after program A is executed will be changed.

Restriction

To use the specified program monitor, check the versions of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

Setting the engineering tool

When using the specified program monitor, in the option settings of the engineering tool, set "After the execution of specified program" under "Timing to Execute Monitor Function and Modify Value Function" to "Yes".

[Tool] ⇨ [Options] ⇨ [Monitor] ⇨ [Common Item] ⇨ [Timing to Execute Monitor Function and Modify Value Function]

For details, refer to the following.

GX Works3 Operating Manual

When a program for calling a subroutine exists

When a program for calling a subroutine exists, to monitor or perform current value change on a local device on the subroutine side, a program to be specified varies depending on whether SM776 (Local device setting at CALL) is turned on or off.

SM776 setting	Program to be specified
OFF	Specify the call source program.
ON	Specify the call destination program.

Precautions

The precautions for the specified program monitor are mentioned below.

- Monitoring cannot be performed if a program not stored in the CPU module is specified.
- Only one instance of the specified program monitor can be executed with one engineering tool. The specified program monitor cannot be executed on multiple programs.
- When simultaneously executing the specified program monitor on separate personal computers with multiple engineering tools, the specified program monitor can be executed from up to eight engineering tools. When not executing the specified program monitor, there is no limit to the number of simultaneous executions of monitoring and current value change.
- When simultaneously executing the specified program monitor on the same personal computer with multiple engineering tools, whether to execute the monitor function of the engineering tool that was operated later can be selected. If selected, an error occurs for the engineering tool being executed.
- When the initial execution type program, standby type program, fixed scan execution type program, or event execution type program (interrupt occurrence) is specified with the specified program monitor, it is executed in the same timing as the timing when no program is specified.
- When the communication whose target is the device/label access service processing is performed while the specified program monitor is running, the scan time is extended because the processing is performed twice in one scan: after the specified program is performed and at a normal timing such as during the END processing.

Point

The scan time extension can be reduced by preventing other communication while the specified program monitor is running.

- To specify a program whose trigger type for the event execution type program is set to "ON of bit data (TRUE)" or "Passing time" using the specified program monitor, specify in a state in which the trigger is established. If the program is specified in a state in which the trigger is not established, the specified program monitor cannot be executed.
- While the specified program monitor is running, if the CPU module is set to STOP and the execution order of the target program is changed or the target program is deleted by using the program setting of the CPU parameter, do not write to the CPU module and execute RUN. Otherwise, the specified program monitor may not operate for the specified program.

10 TEST FUNCTION

10.1 External Input/Output Forced On/Off Function

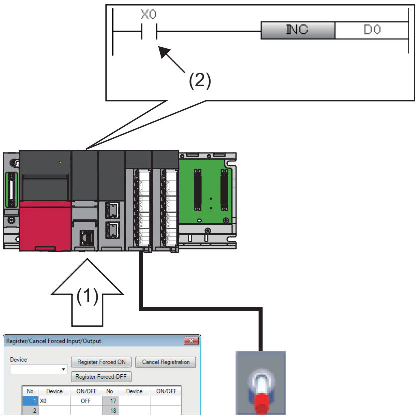
RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using the Process CPU (redundant mode), refer to the following as well.

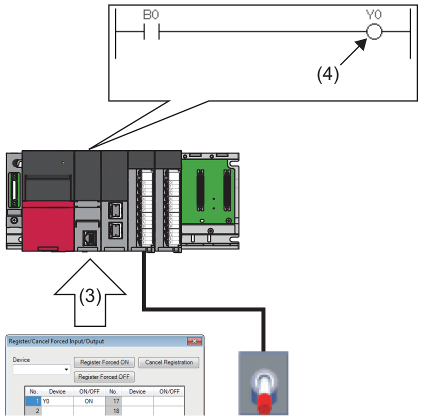
☞ Page 492 FUNCTIONS

External inputs/outputs can be forcibly turned on or off from the engineering tool. This function enables input devices to be turned on or off regardless of the on/off state of the external inputs and enables the external outputs to be turned on and off regardless of the operation result of a program.

(External input)



(External output)



(1) Turn off X0 forcibly.
(2) The input device is turned off regardless of the on/off state of the external input.
(3) Turn on Y0 forcibly.
(4) The external output is turned on regardless of the operation result of the program.

Point

The external input/output forced on/off function is available for MELSEC iQ-R series modules and Q series modules mounted on an extension base unit.

Restriction

Before executing the external input/output forced on/off function, check the versions of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

Devices that allow forced on/off registration

The following lists the devices that allow forced on/off registration.

Device	Range
Input	• R00CPU, R01CPU, and R02CPU: X0 to X1FFF (8192 points) • Other CPU modules: X0 to X2FFF (12288 points)
Output	• R00CPU, R01CPU, and R02CPU: Y0 to Y1FFF (8192 points) • Other CPU modules: Y0 to Y2FFF (12288 points)

Number of device points that allows forced on/off registration

A maximum of 32 points can be registered for input devices and output devices in total.

Inputs/outputs for which forced on/off can be set

The following describes the inputs/outputs for which forced on/off can be set.

■Input

After the refreshed data from the module is reflected, the input devices for which forced on/off registration is set are forcibly turned on or off.

■Output

The following external outputs are turned on or off by refreshing output devices that have been forcibly turned on or off.*1

- Refresh target output of modules mounted on the base unit
- Link refresh target output of network modules
- Link refresh target output of CC-Link IE Field Network Basic

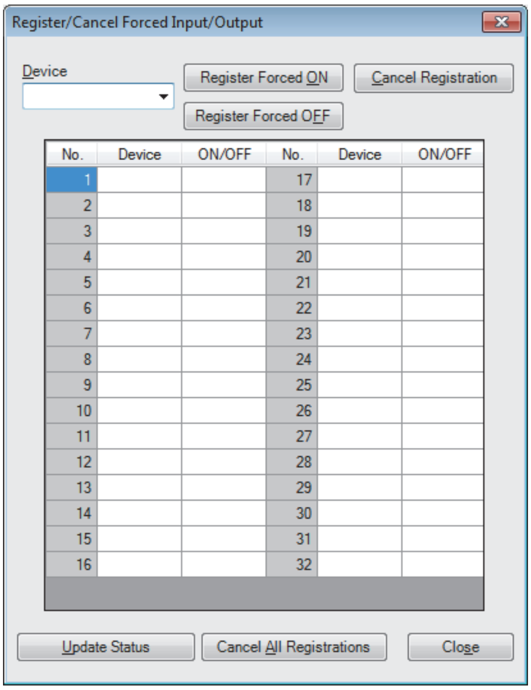
*1 When no modules are mounted on the base unit (no refresh targets), no data is output to the external outputs.

Operation method of forced on/off

Use the engineering tool for the forced on/off operation.

[Debug] ⇨ [Register/Cancel Forced Input/Output]

Window



Displayed items

Item	Description
Device	Enters target devices (X, Y).
[Register Forced ON] button	Registers forced on for the entered devices (X, Y).
[Register Forced OFF] button	Registers forced off for the entered devices (X, Y).
[Cancel Registration] button	Cancels forced on/off for the registered devices (X, Y).
[Update Status] button	Displays the latest on/off states.
[Cancel All Registrations] button	Batch-cancels forced on/off for the registered devices (X, Y).

Behavior in forced on/off registration

The following describes the behavior in forced on/off registration.

■Behavior of an input device

Registering forced on/off turns on or off the input device regardless of the status of the external input. When an input device for which the forced on/off has been registered is changed in the program, the input device is turned on or off in accordance with the operation result of the program.

Operation	Change in the input device in the program	Behavior of an input device
Forced on registration	Changed	• The input device is on when the program operation at every scan starts. • The input device is turned on or off in accordance with the operation result of the program after the program operation at every scan started.*1
	Not changed	The on state is held.
Forced off registration	Changed	• The input device is off when the program operation at every scan starts. • The input device is turned on or off in accordance with the operation result of the program after the program operation at every scan started.*1
	Not changed	The off state is held.

*1 If the forced on/off is registered for a device set to perform link refresh, the device value may be updated at the timing of refresh. Thus, a different value from the operation result may be displayed when the device value is monitored (performed in END processing).

■Behavior of an output device

Registering forced on/off turns on or off the external output regardless of the operation result of the program. When an output device has been changed in the program, the output device is turned on or off in accordance with the operation result of the program. Therefore, the on/off states between the output device and external output may differ.

Operation	Change in the output device in the program	Behavior of outputs	
		Behavior of an output device	Behavior of an external output
Forced on registration	Changed	• The output device is on when the program operation at every scan starts. • The output device is turned on or off in accordance with the operation result of the program after the program operation at every scan started.*1	The external output is turned on regardless of the operation result of the program.
	Not changed	The on state is held.	
Forced off registration	Changed	• The output device is off when the program operation at every scan starts. • The output device is turned on or off in accordance with the operation result of the program after the program operation at every scan started.*1	The external output is turned off regardless of the operation result of the program.
	Not changed	The off state is held.	

*1 If the forced on/off is registered for a device set to perform link refresh, the device value may be updated at the timing of refresh. Thus, a different value from the operation result may be displayed when the device value is monitored (performed in END processing).

CPU module operating status

Forced on/off registration is allowed regardless of the CPU module operating status. However, when a stop error has occurred, the external outputs are turned off regardless of the forced on/off registration setting. While the operating status of the CPU module is STOP due to a stop error, forced on/off is enabled only for the input devices. If the CPU module is powered off and on or is reset, all the forced on/off registration information will be canceled.

Forced on/off timing

The following table lists the timing to reflect the registered data in the forced on/off registration settings to the input/output devices or external outputs.

Inputs/outputs for which forced on/off can be set	Reflection timing for the input devices	Reflection timing for the output devices or external outputs
Input/output of the modules mounted on the base unit	• END processing (at input refresh) • At COM instruction execution (at input refresh) • At instruction execution using the direct access input (DX) (LD, LDI, AND, ANI, OR, ORI, LDP, LDF, ANDP, ANDF, ORP, ORF, LDPI, LDFI, ANDPI, ANDFI, ORPI, ORFI)[*1] • At execution of the RFS instruction and MTR instruction • At execution of instructions used in the interrupt by the system (UDCNT1, UDCNT2, SPD) • At program execution[*2] • At execution of the inter-module synchronization cycle program (I44) • At execution of the multiple CPU synchronization program (I45) and non-execution of the multiple CPU synchronization program (I45) (at END processing)[*3]	• END processing (at output refresh) • At COM instruction execution (at output refresh) • At instruction execution using the direct access output (DY) (OUT, SET, DELTA(P), RST, PLS, PLF, FF, MC, SFT(P))[*1] • At execution of the RFS instruction and MTR instruction • At execution of instructions used in the interrupt by the system (PLSY, PWM) • At program execution[*2] • At execution of the inter-module synchronization cycle program (I44)
Input/output of the CPU module assigned to RX and RY of the CC-Link IE TSN master/local module	• END processing (at link refresh) • At ZCOM instruction execution (at link refresh)	
Input/output of the CPU module assigned to LX and LY of the CC-Link IE Controller Network module or MELSECNET/H network module	• END processing (at link refresh of the CC-Link IE Controller Network module or MELSECNET/H network module) • At COM instruction execution (at link refresh of the CC-Link IE Controller Network module or MELSECNET/H network module) • At ZCOM instruction execution (at link refresh of the CC-Link IE Controller Network module or MELSECNET/H network module)	
Input/output of the CPU module assigned to RX and RY of the CC-Link module	• END processing (at link refresh) • At COM instruction execution (at link refresh) • At ZCOM instruction execution (at link refresh)	
Input/output of the CPU module assigned to RX and RY of the CC-Link IE Field Network module	• END processing (at link refresh) • At COM instruction execution (at link refresh) • At ZCOM instruction execution (at link refresh) • At execution of the inter-module synchronization cycle program (I44)	
Input/output of the CPU module assigned to RX and RY of the CC-Link IE Field Network Basic	• END processing (at link refresh) • At COM instruction execution (at link refresh)	

*1 For the precautions for using the direct access device, refer to the following.
☞ Page 184 Precautions
*2 At input/output refresh execution where input/output refresh is registered for each program, and for interrupt programs.
*3 When an output device or external output is specified as a refresh device in multiple CPU refresh and the forced on/off function is executed, the device is not forcibly turned on or off.

Checking the forced on/off execution status

The execution status of the forced on/off can be checked in the following ways.

■Engineering tool

The execution status can be checked with the [Update Status] button of the engineering tool. (☞ Page 180 Operation method of forced on/off)

■FUNCTION LED

The FUNCTION LED of the CPU module flashes by registering forced on/off (every 200ms).

Point

The LED indicator follows the status of the external input/output forced on/off function even when the LED Display Setting has been set and other functions that use the FUNCTION LED are being executed. (☞ Page 143 LED display setting)

The display status of the FUNCTION LED can be checked with "Module Information List" in the module diagnostics. (GX Works3 Operating Manual)

■Special register

SD1488 (Debug function usage status) can be used to check whether the external input/output forced on/off function is used.
(☞ Page 966 List of Special Register Areas)

Behavior in cancellation of forced on/off

Forced on/off registration can be canceled for each input/output device individually.

■Behavior of the device

Inputs/outputs for which forced on/off can be set		Change in input/output devices in the program	
		Changed	Not changed
Input	Input from the modules mounted on the base unit	The input device is turned on or off in accordance with the on/off state of the modules.	
	Input from RX of CC-Link IE TSN	The input device is turned on or off in accordance with the on/off state refreshed from CC-Link IE TSN.	
	Input from LX of CC-Link IE Controller Network or MELSECNET/H	The input device is turned on or off in accordance with the on/off state refreshed from CC-Link IE Controller Network and MELSECNET/H.	
	Input from RX of CC-Link	The input device is turned on or off in accordance with the on/off state refreshed from CC-Link.	
	Input from RX of CC-Link IE Field Network	The input device is turned on or off in accordance with the on/off state refreshed from CC-Link IE Field Network.	
	Input from RX of CC-Link IE Field Network Basic	The input device is turned on or off in accordance with the on/off state refreshed from CC-Link IE Field Network Basic.	
	Input other than above (input without modules actually mounted)	The input device is turned on or off in accordance with operation result of the program.	The input device is turned on or off in accordance with the registered on/off state.
Output	Output to the modules mounted on the base unit	The operation result of the program is output.	Data is output in accordance with the registered on/off state.
	Output to RY of CC-Link IE TSN		
	Output to LY of CC-Link IE Controller Network or MELSECNET/H		
	Output to RY of CC-Link		
	Output to RY of CC-Link IE Field Network		
	Output to RY of CC-Link IE Field Network Basic		
	Output other than above (output without modules actually mounted)	The output device is turned on or off in accordance with operation result of the program. (Refresh to external output is not executed.)	The output device is turned on or off in accordance with the registered on/off state. (Refresh to external output is not executed.)

■CPU module operating status

Forced on/off registration can be canceled regardless of the CPU module operating status.

■LED status

The following describes the LED status after forced on/off registration is canceled.

Forced on/off registration status of when the registration is canceled	FUNCTION LED status
Registered on/off information is remaining.	Flashing (every 200ms)
No registered on/off information is remaining.	Off[*1]

*1 When other functions that use the FUNCTION LED are being executed, the LED indicator changes its status in accordance with the status of the functions.

Behavior in batch-cancellation of forced on/off registrations

All the forced on/off registrations can be canceled in a batch.

■Behavior of the device

The behavior of the device is the same as that of cancellation of forced on/off (for each device). (☞ Page 183 Behavior of the device)

■CPU module operating status

The behavior of the device is the same as that of cancellation of forced on/off (for each device). (☞ Page 183 CPU module operating status)

■LED status

The following describes the LED status after forced on/off registration is canceled in a batch.

Forced on/off registration status of when the registration is canceled	FUNCTION LED status
No registered on/off information is remaining.	Off*1

*1 When other functions that use the FUNCTION LED are being executed, the LED indicator changes its status in accordance with the status of the functions.

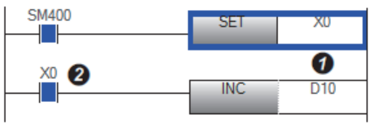
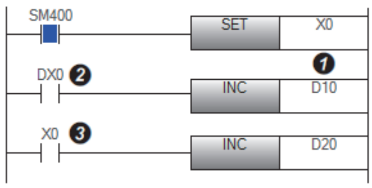
Precautions

The following describes precautions for using the external input/output forced on/off function.

- Multiple engineering tools connected to the network can be used to register forced on/off for the same CPU module. In this case, note the following.
  - The forced on/off state registered last is handled as the on/off state of input/output devices.
  - Since the on/off state displayed in engineering tools may differ from that of the CPU module, update the on/off state displayed in engineering tools.
- If a direct device is used in a program, forced on/off registration is reflected when an instruction is executed.

Ex.

When X0 is forcibly turned off

When the input device (X0) is used	When the direct input device (DX0) is used
 ❶ Set X0 by using the SET instruction. ❷ Turn on X0 because the SET instruction is being executed.	 ❶ Set X0 by using the SET instruction. ❷ Reflect the registered off information. ❸ Turn off X0 although the SET instruction is being executed.

- When the external input/output forced on/off function is executed while the program of high-speed internal timer interrupt (I48, I49) is executed, the interval of programs of high-speed internal timer interrupt is short, and a large number of forced on/off is registered, the scan time will increase and a WDT error may occur.
- Execution of interrupt programs which are executed at a fixed scan may delay depending on the number of forced on/off registrations and the number of refresh points of each refresh processing.
- When forced on/off is registered in the multiple CPU system configuration, registration is allowed for input/output devices*2 and external outputs regardless of the out-of-group I/O fetch setting*1. Note, however, that the devices of other CPU modules and the external outputs controlled by other CPU modules are not turned on or off even if forced on/off is registered from the host CPU for the external outputs controlled by other CPU modules. Only the devices of the host CPU module are turned on or off.

*1 For the out-of-group I/O fetch setting, refer to the following.
☞ Page 328 Out-of-group I/O fetch setting

*2 Except when the on/off state of the input/output devices is changed by using the output instruction

10.2 Device Test with Execution Conditions

RnCPU | RnENCPU | RnPCPU (Process) | RnPCPU (Redundant) | RnPSFCPU (Standard) | RnPSFCPU (Safety) | RnSFCPU (Standard) | RnSFCPU (Safety)

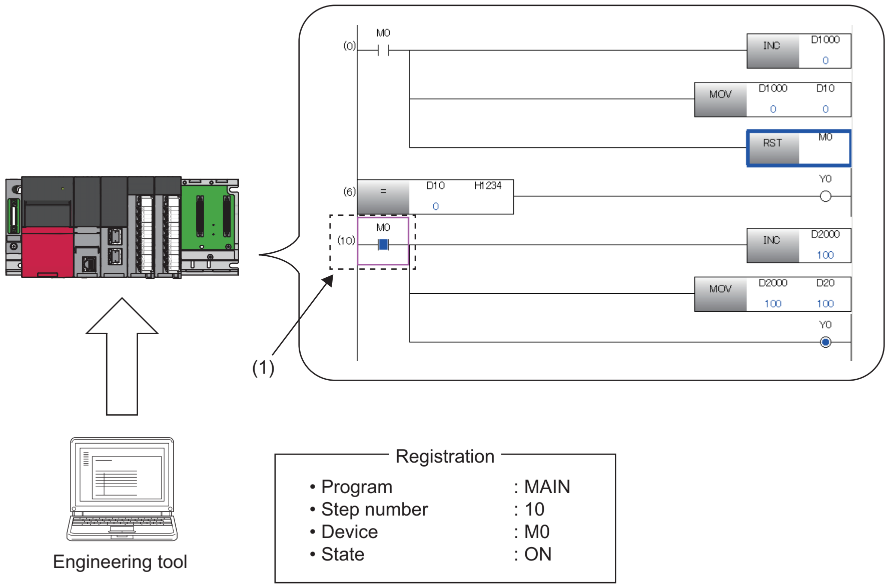
- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 492 FUNCTIONS

- When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

Using the engineering tool, device/label values can be set for each execution of specified steps of programs.
This function enables to debug a specific ladder block without modifying the program even when the program is configured as shown in the example below.



(1) The device is operated in accordance with the registered settings. (M0 of the step number 10 is on.)

Restriction

Before using the device test with execution conditions, check the version of the CPU module and the engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

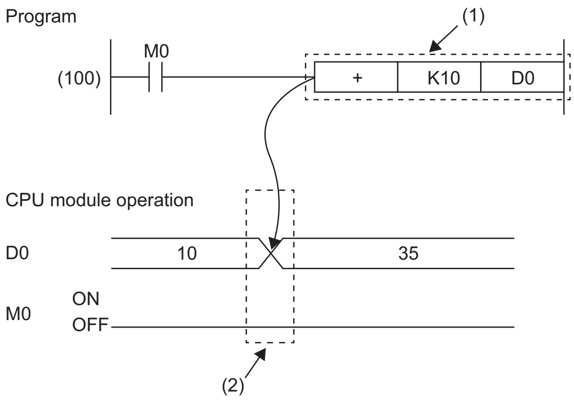
Operation when device test with execution conditions is registered

The device test with execution conditions can forcibly change device/label values (status) of specified locations.

Specify a location of a device/label value (status) to be changed with a program name and a step number. In addition, specify a device/label and its value (status) to be changed with a device/label name and a setting value.

Changes made to the device/label value (status) take effect in the specified step and later. However, when the device/label value (status) is changed due to operations of the program or other factors in the specified step and later, the device/label value (status) is changed accordingly.

In the device test with execution conditions, a device/label value is changed in a specified step regardless of whether or not the instruction execution conditions are satisfied in the specified step. In the following example, the D0 value is changed to 35 regardless of the status (on/off) of M0.



(1) The device test with execution conditions that sets D0 to 35 and is executed before execution of the instruction is registered to the step (100).

(2) The device value in the specified step is changed regardless of the value (on/off) of M0.

Operation when device test with execution conditions is disabled

The device/label value (status) when the registration is disabled is held. (It is not returned to the value (status) before execution of the device test with execution conditions.)

Data that can be set

The following tables list the data that can be set for the device test with execution conditions.

■Devices that can be set

Type	Device*3
Bit device*1	X, DX, Y, DY, M, L, F, SM, V, B, SB, T (contact), ST (contact), C (contact), LT (contact), LST (contact), LC (contact), FX, FY, Jn\X, Jn\Y, Jn\SB, Jn\B
Word device*2	T (current value), ST (current value), C (current value), D, SD, W, SW, RD, R, ZR, Z, FD, Un\G, Jn\W, Jn\SW, U3En\G*4, U3En\HG*4
Double-word device	LT (current value), LST (current value), LC (current value), LZ

*1 For bit devices, digit specification is allowed for K1 to K8 only.
*2 For word devices, bit specification is allowed.
*3 A local device, indirectly-specified device, or index-modified device can also be specified.
*4 The word devices of the other CPU modules, U3En\G and U3En\HG, can be specified in the engineering tool, however, the setting value is not reflected in the CPU module. If U3En\G□ is specified, an error will occur.

■Labels that can be set

Type*1*2	Class	Data type
Global label	• VAR_GLOBAL • VAR_GLOBAL_RETAIN	■Primitive data type • Bit*3 • Word (signed)*4 • Double word (signed) • Word (unsigned)*4 • Double word (unsigned) • Single-precision real number • Double-precision real number • Timer type • Retentive timer type • Counter type • Long timer type • Long retentive timer type • Long counter type ■Array*5 ■Structure*6
Local label	• VAR • VAR_RETAIN	

*1 Only labels that exist in the read project can be specified.
*2 Labels of a program block can be specified.
*3 Digit-specified labels cannot be specified.
*4 Bit specification is allowed.
*5 Specify the array element.
*6 Specify the structure member.

Programs that can be set

Only ladder programs can be set for the device test with execution conditions.

Maximum number of devices/labels that can be set

A total of 32 devices/labels can be set for the device test with execution conditions.

Checking execution status of device test with execution conditions

The execution status can be checked in the following ways.

■Engineering tool

The execution status can be checked by displaying the list of registered settings with the engineering tool.

■FUNCTION LED

The FUNCTION LED of the CPU module flashes (every 200ms) by registering the device test with execution conditions.

Point

When the device test with execution conditions is registered, the FUNCTION LED flashes regardless of the LED Display Setting in the RAS Setting. Even when the LED Display Setting has been set and other functions that use the FUNCTION LED are being executed, the LED indication at the registration of the device test with execution conditions takes priority. The LED indication is reset to the original state when the registration is disabled. (The LED indication follows the status of other functions that use the FUNCTION LED.) (☞ Page 143 LED display setting)

The display status of the FUNCTION LED can be checked with "Module Information List" in the module diagnostics. (📖 GX Works3 Operating Manual)

■Special register

SD1488 (Debug function usage status) can be used to check the usage status of the device test with execution conditions. (☞ Page 1005 Debug function)

Registration of device test with execution conditions

This section describes how to register the device test with execution conditions.

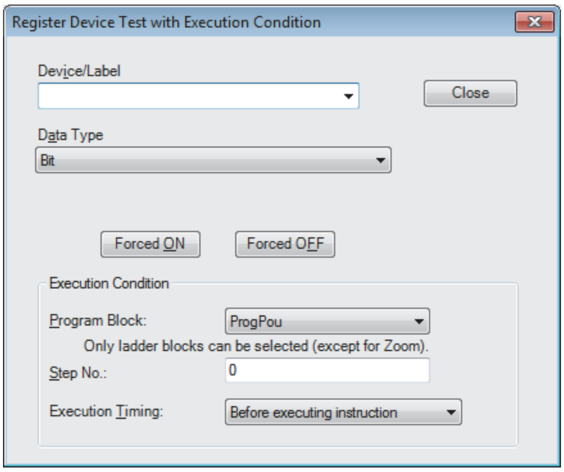
■Registration method

Specify each field in the "Register Device Test with Execution Condition" window.

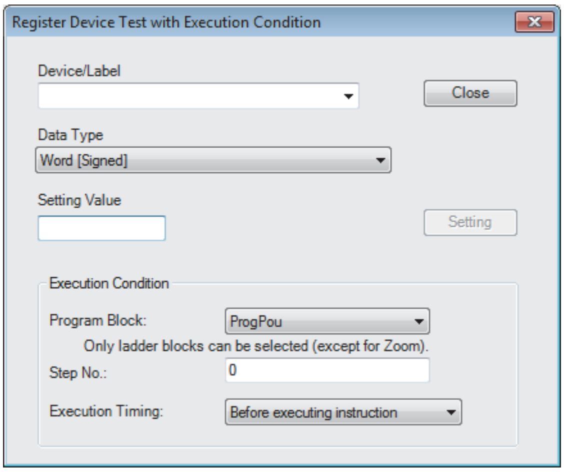
[Debug] ⇨ [Device Test with Execution Condition] ⇨ [Register]

Window

• When bit device/bit type label is specified



• When word device/word type label is specified



Setting data

Item		Description
Device/Label		Set a device or a label to be registered. (☞ Page 187 Data that can be set)
Data Type		Specify a data type. (☞ Page 187 Data that can be set)
[Forced ON] button		These buttons appears when the data type is set to bit data. Click either button to register the forced on/off.
[Forced OFF] button		
Setting Value		This field appears when the data type is set to word data. Enter a setting value in decimal or hexadecimal format. When setting a hexadecimal value, prefix the numerical value with "H". (Example: "H16", "H1F") Click the [Setting] button to register the value.
[Setting] button		
Execution Condition	Program Block	Specify a program block.
	Step No.	Specify a step number for each program. To specify the step number, specify the start step of the instruction.
	Execution Timing	Specify whether to change the device/label value before or after the execution of the instruction of the specified step. (☞ Page 193 Execution timing)

■Registration of multiple device tests with execution conditions to the same step

Multiple device tests with execution conditions can be registered to one step. However, device tests with execution conditions that have the same device/label name and the same execution timing cannot be registered to one step. When such device tests with execution conditions are registered, the existing registration is overwritten with new registration.

Point

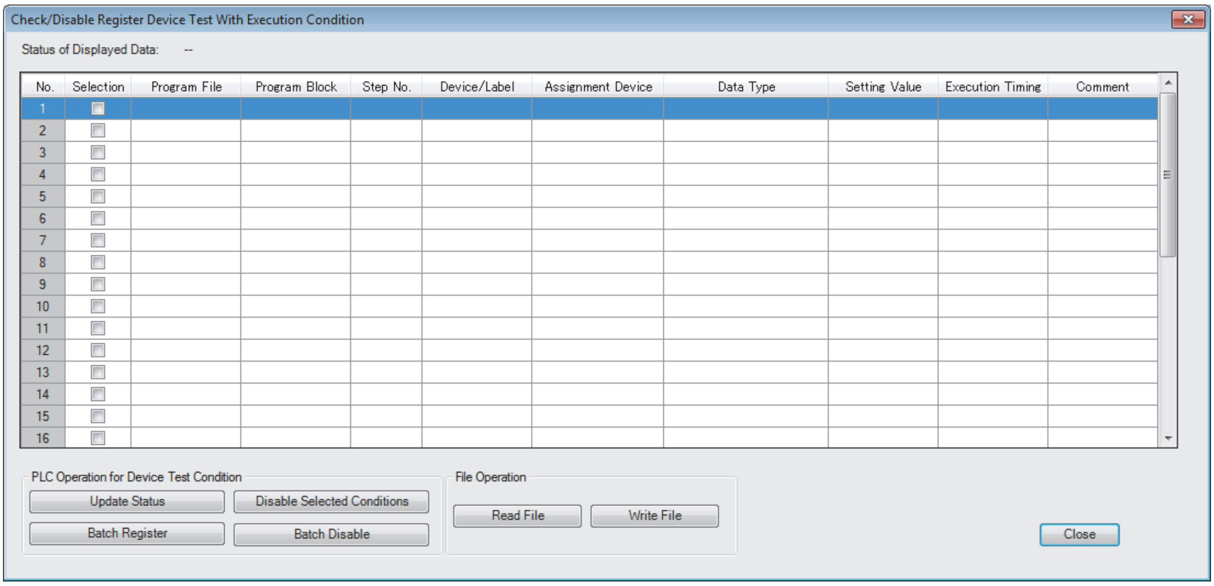
• Two device tests with execution conditions having different execution timing can be registered to one step even when they have the same device/label name.
• Different device modification (bit specification of word device, digit specification of bit device, or index modification) is handled as different device name. Therefore, two device tests with execution conditions having different device modification can be registered to one step.

Checking and disabling settings from list window

From the "Check/Disable Register Device Test with Execution Condition" window, the following operations can be performed: checking the registered settings, disabling selected settings, registering/disabling settings in batch, and reading/writing registered settings from/to a file.

[Debug] ⇨ [Device Test with Execution Condition] ⇨ [Check/Disable Register]

Window



Displayed items

Item		Description
PLC Operation for Device Test Condition	[Update Status] button	Click this button to read the registered settings of the device test with execution conditions in the CPU module. No data is read when this operation is performed with no settings registered.
	[Disable Selected Conditions] buttons	Click this button to disable the selected registration of the device test with execution conditions that has been read from the CPU module by using the [Update Status] button.
	[Batch Register] button	All the existing registered settings of the device test with execution conditions in the CPU module are disabled by clicking this button, and the settings that have been read by using the [Update Status] button or [Read File] button are registered to the CPU module.
	[Batch Disable] button	Click this button to collectively disable the registered settings of the device test with execution conditions in the CPU module.
File Operation	[Read File] button	Click this button to read the registered settings of the device test with execution conditions, which have been saved by using the [Write File] button, and list them on the window.
	[Write File] button	Click this button to save the registered settings of the device test with execution conditions listed on the window into a file in the personal computer.

Point

The registered settings of the device test with execution conditions can be collectively disabled by the following operation in the engineering tool: [Debug] ⇨ [Device Test with Execution Condition] ⇨ [Batch Disable]

■Disabling device test with execution conditions

In addition to the operations from the engineering tool, the following operations can be used to disable the device test with execution conditions.

- Powering off and on
- Resetting the CPU module
- Writing a program to the CPU built-in memory by writing data to the programmable controller while the CPU module is in the STOP state*1*4
- Deleting a program in the CPU built-in memory by deleting data in the programmable controller while the CPU module is in the STOP state*1
- Initializing the CPU built-in memory while the CPU module is in the STOP state*1
- Changing a program in the CPU built-in memory using the following SLMP commands while the CPU module is in the STOP state*1
  - Deleting a file (Delete File) (1822)
  - Copying a file (Copy File) (1824)
- Changing a program in the CPU built-in memory using the following FTP commands while the CPU module is in the STOP state*1
  - Deleting a file (delete, mdelete)
  - Changing a file name (rename)
- Restoring a program using the CPU module data backup/restoration function*1*4
- Changing a program online (the online change (ladder block)*6 or the file batch online change) (☞ Page 194 Operation during online change)
- Writing a global label to the CPU built-in memory by writing data to the programmable controller while the CPU module is in the STOP state*2*4
- Deleting a global label in the CPU built-in memory by deleting data in the programmable controller while the CPU module is in the STOP state*2
- Changing a global label in the CPU built-in memory using the following SLMP commands while the CPU module is in the STOP state*2*4
  - Deleting a file (Delete File) (1822)
  - Copying a file (Copy File) (1824)
- Changing a global label in the CPU built-in memory using the following FTP commands while the CPU module is in the STOP state*2
  - Deleting a file (delete, mdelete)
  - Changing a file name (rename)
- Restoring a global label using the CPU module data backup/restoration function*2*4
- Writing CPU parameters to the CPU built-in memory by writing data to the programmable controller while the CPU module is in the STOP state*3*4
- Deleting CPU parameters in the CPU built-in memory by deleting data in the programmable controller while the CPU module is in the STOP state*5
- Changing the CPU parameter in the CPU built-in memory using the following SLMP commands while the CPU module is in the STOP state
  - Deleting a file (Delete File) (1822)*5
  - Copying a file (Copy File) (1824)*3
- Changing the CPU parameter in the CPU built-in memory using the following FTP commands while the CPU module is in the STOP state*5
  - Deleting a file (delete, mdelete)
- Restoring a CPU parameter using the CPU module data backup/restoration function*3*4

*1 The registered settings of device test with execution conditions for the program to be changed are disabled when the operating status of the CPU module is changed from STOP to RUN after the operation. (They are not disabled at the completion of the operation.)

*2 All the registered settings that specify global labels are disabled when the operating status of the CPU module is changed from STOP to RUN after the operation. (They are not disabled at the completion of the operation.)

*3 When the CPU parameters are changed, all the registered settings that specify local devices or local labels are disabled when the operating status of the CPU module is changed from STOP to RUN after the operation. (They are not disabled at the completion of the operation.)

*4 When the same programs, global labels, or CPU parameters are written, the registered settings are not disabled.

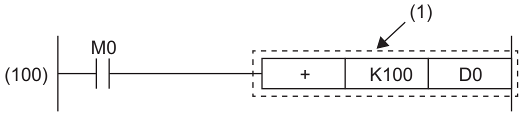
*5 All the registered settings of device test with execution conditions are disabled when the operating status of the CPU module is changed from STOP to RUN after the operation. (They are not disabled at the completion of the operation.)

*6 When the online change (ladder block) is performed while the CPU module is in the STOP state, the device test with execution conditions registered for the program to be changed is disabled when the operating status of the CPU module is changed from STOP to RUN after the operation. (They are not disabled at the completion of the operation.)

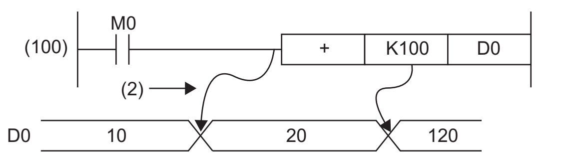
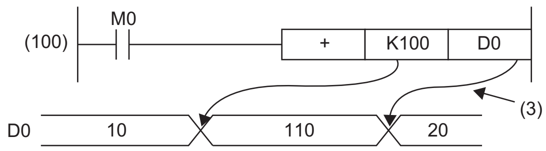
Execution timing

Select whether to change the device/label value before or after the execution of the instruction of the specified step when registering the device test with execution conditions.

Program



(1) The device test with execution conditions that sets 20 in D0 in step (100) is registered.

Setting of execution timing	Operation of CPU module	
Before execution of the instruction in step (101)		(2) A value in D0 is changed to 20.
After execution of the instruction in step (101)		(3) A value in D0 is changed to 20.

When the device test with execution conditions is registered with the step of the specific instruction specified, depending on the setting of the execution timing, the device/label may not be changed even if the specified step is executed.
When the step of the following instructions is specified and the execution timing is set to "Before executing instruction", the device/label value is not changed by the device test with execution conditions even when execution conditions of the instruction are satisfied and the specified step is passed.

Classification	Instruction name
Stop	STOP*1
Jump	CJ*1, SCJ*1, GOEND*1, JMP
Loop	BREAK(P)*1
Ending a program	FEND
Calling a subroutine program	CALL(P)*1, FCALL(P)*1, ECALL(P)*1, EFCALL(P)*1, XCALL*1
Return	RET, IRET

*1 When the execution timing is set to "After executing instruction", the device/label is changed when the execution condition of the instruction is not satisfied.

■FOR instruction, NEXT instruction, FOR to NEXT instruction loop

When the device test with execution conditions is registered with the step for the FOR instruction, the NEXT instruction, or the FOR to NEXT instruction loop specified, the timing to change the device/label differs from that when other instructions are specified.

Instruction for specified step	Specification detail of execution timing	
	Set to "Before executing instruction"	Set to "After executing instruction"
FOR	Executed only once before starting the loop processing	Executed only once after starting the loop processing (Specified devices are changed before the execution of the program that is between the FOR and NEXT instructions.)
NEXT	Executed only once after starting the loop processing (Specified devices are changed after the execution of the program that is between the FOR and NEXT instructions.)	Executed only once after ending the loop processing
FOR to NEXT instruction loop	Executed only once before execution of the instruction at specified step in the FOR to NEXT instruction loop	Executed only once after execution of the instruction at specified step in the FOR to NEXT instruction loop

■END instruction

When the step for the END instruction is specified, the execution timing cannot be set to "After executing instruction".

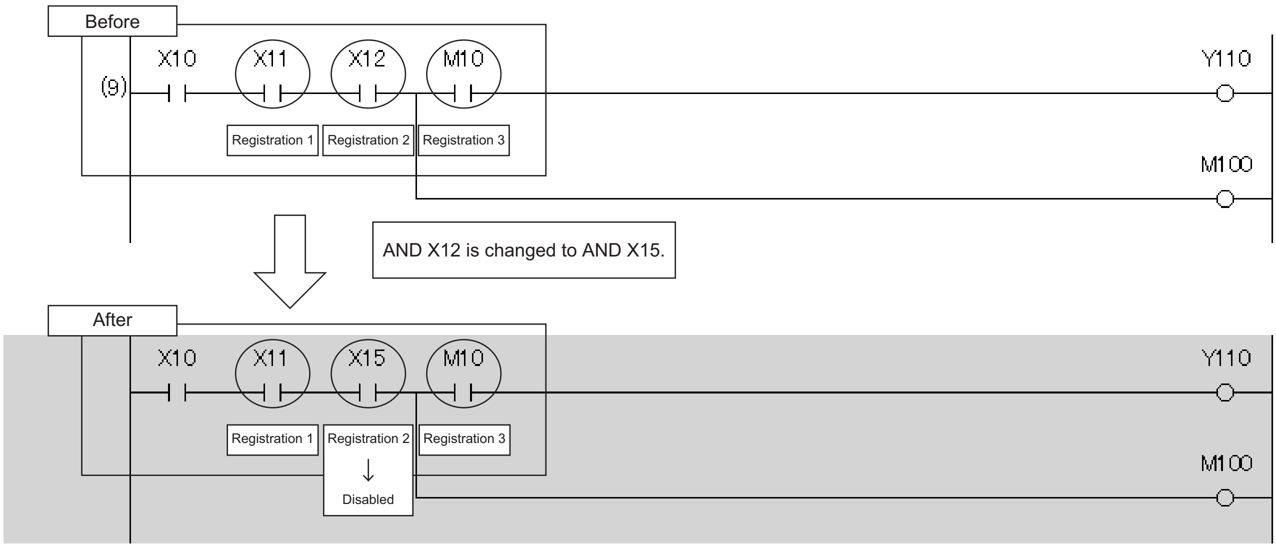
Operation during online change

This section describes the operation performed during the online change of the CPU module to which the device test with execution conditions is registered.

■Online change (ladder block) (without adding or deleting instruction)

If a part to be changed by the online change (ladder block) includes registrations of the device test with execution conditions, such registrations are disabled.

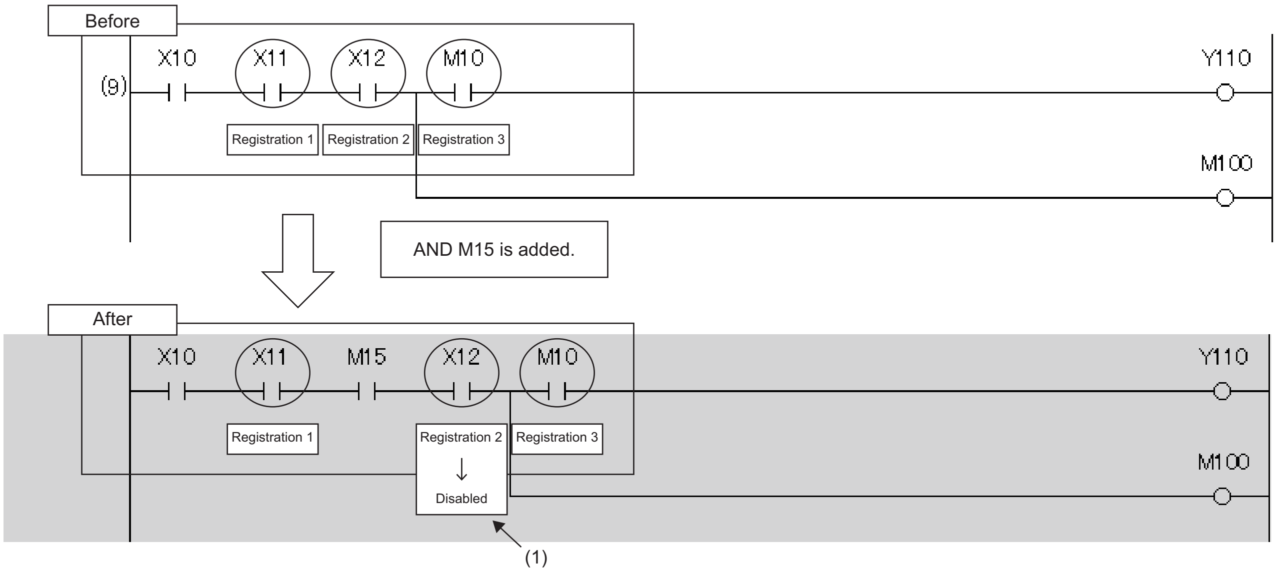
When the Registrations 1 to 3 of the device test with execution conditions are registered for the steps as shown below, the Registration 2 is disabled upon the execution of the online change (ladder block). The Registrations 1 and 3 are not disabled because the steps to which they are registered are not included in the part to be changed.



■Online change (ladder block) (with adding instruction)

When an instruction is added by the online change (ladder block), the registration of the device test with execution conditions of the instruction immediately after the instruction to be added is disabled.

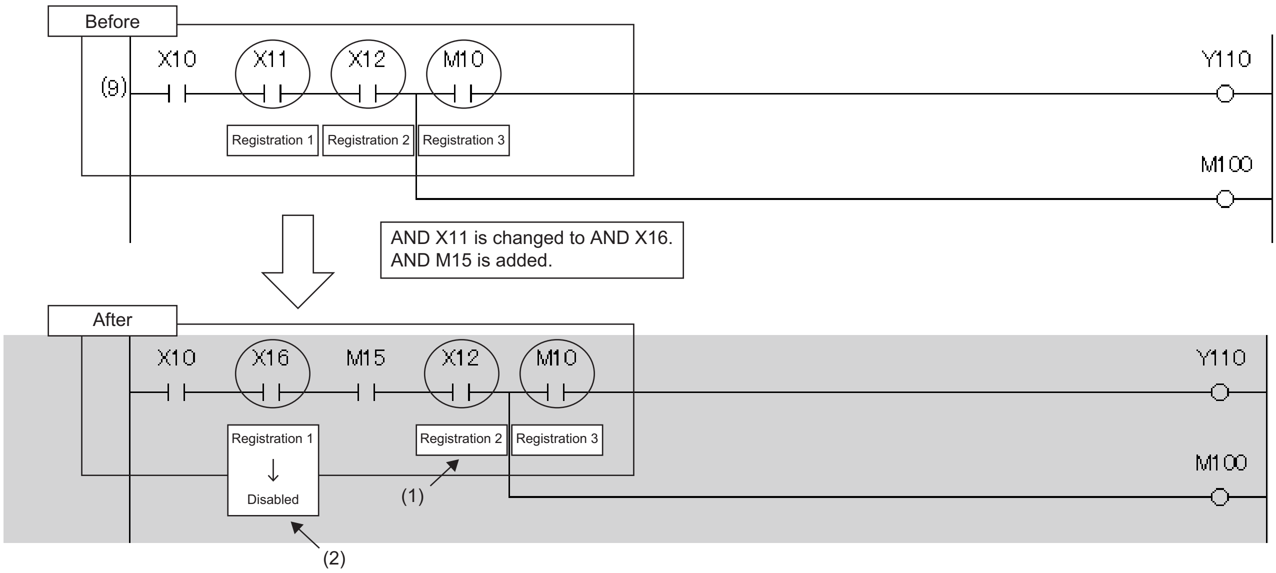
In the following example, an instruction is added by the online change (ladder block). In this case, when the device test with execution conditions is registered to the instruction immediately after the added instruction, the relevant registrations are disabled upon the execution of the online change (ladder block).



(1) The registration of the device test with execution conditions of the instruction immediately after the instruction to be added is disabled.

However, when an instruction to be added is adjacent to an instruction to be changed, the registration of the device test with execution conditions of the instruction immediately after the instruction to be added is not disabled.

In the following example, an instruction to be added is adjacent to an instruction to be changed. Therefore, even when the device test with execution conditions is registered to the instruction immediately after the instruction to be added, the relevant registration is not disabled by the execution of the online change (ladder block).

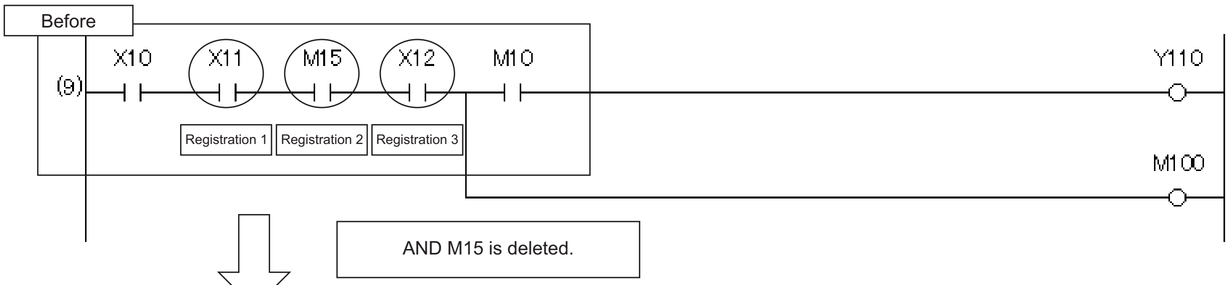


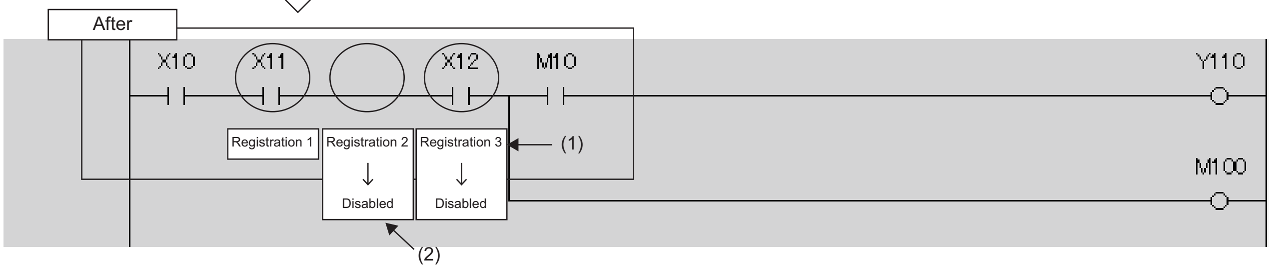
(1) Although this instruction is located immediately after the added instruction, the registration of the device test with execution conditions is not disabled because the added instruction is adjacent to the changed instruction.
(2) The registration of the device test with execution conditions is disabled because the instruction is changed.

■Online change (ladder block) (with deleting instruction)

When an instruction is deleted by the online change (ladder block), registrations of the device test with execution conditions for the deleted instruction and for the instruction immediately after the deleted instruction are disabled.

In the following example, an instruction is deleted by the online change (ladder block). In this case, when the device test with execution conditions is registered to the instruction immediately after the deleted instruction, the relevant registrations are disabled upon the execution of the online change (ladder block).

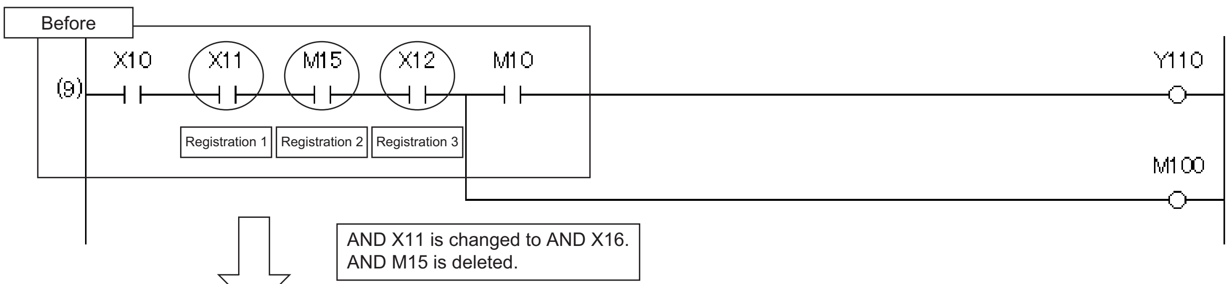


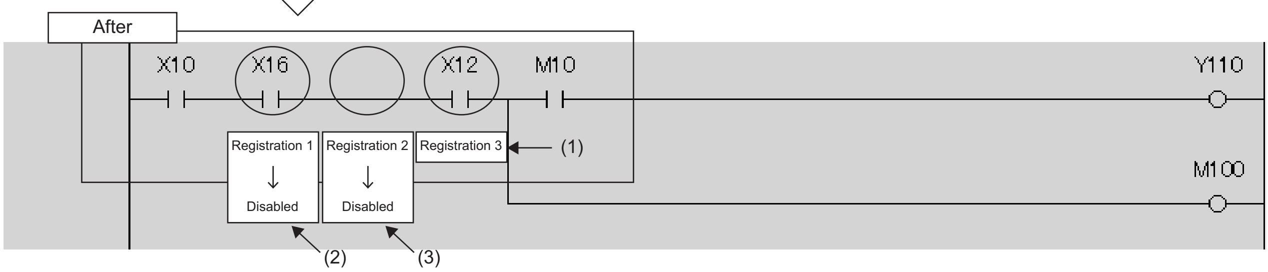


(1) The registration of the device test with execution conditions is disabled because the instruction is located immediately after the deleted instruction.
(2) The registration of the device test with execution conditions is disabled because the instruction where it is registered is deleted.

However, when an instruction to be deleted is adjacent to an instruction to be changed, registration of the device test with execution conditions of the instruction immediately after the instruction to be deleted is not disabled.

In the following example, an instruction to be deleted is adjacent to an instruction to be changed. Therefore, even when the device test with execution conditions is registered to the instruction immediately after the instruction to be deleted, the relevant registration is not disabled by the execution of the online change (ladder block).





(1) Although this instruction is located immediately after the deleted instruction, the registration of the device test with execution conditions is not disabled because the deleted instruction is adjacent to the changed instruction.
(2) The registration of the device test with execution conditions is disabled because the instruction where it is registered is changed.
(3) The registration of the device test with execution conditions is disabled because the instruction where it is registered is deleted.

■Addition/deletion/change of labels by the online change (ladder block)

- When SM940 (Operation setting of the device test with execution conditions) is off, if local labels or program files are added, deleted, or changed and the online change (ladder block) is performed, all the registrations that specify local labels of the relevant program file are disabled. When SM940 is on, the registrations are not disabled.*1
- When SM940 is off, if global labels are added, deleted, or changed and the online change (ladder block) is performed, all the registrations that specify global labels are disabled. When SM940 is on, the registrations are not disabled.*1

*1 If reassignment of local labels/global labels which are used in the program is performed when SM940 is on, registrations including reassigned labels are disabled. When SM940 is on, do not delete or change local labels or global labels registered to device tests with execution conditions. If deleted or changed, registered settings may not be read correctly.

■File batch online change

All the device tests with execution conditions registered to the target program for the file batch online change are disabled. When global labels are added, deleted, or changed and the file batch online change is performed, all the registrations that specify global labels are disabled.

Precautions

This section describes the precautions on the use of the device test with execution conditions.

■Operation when devices/labels cannot be registered

When multiple devices/labels are registered to the device test with execution conditions, none of the devices/labels are registered if there is even one device/label or execution condition (program block, step number, or execution timing) that cannot be registered.

■Operation from multiple engineering tools

Multiple engineering tools connected to the network can be used to register the device test with execution conditions for the same CPU module. However, registration may fail if done simultaneously. If registration fails, register again. When the device tests with execution conditions are registered to the same device/label in the same location, the status (value) that is registered later takes effect. Before registering the device test with execution conditions from multiple engineering tools, update the data using "Update Status".

■Mutual exclusion

This section describes the mutual exclusion between the device test with execution conditions and other functions.

Other functions	Operation
Online change	When the device test with execution conditions is registered or disabled during execution of the online change, an error occurs and registering or disabling the device test with execution conditions fails.
Real-time monitor	When execution of these functions is set for the same step as that specified by the device test with execution conditions, the device test with execution conditions is executed first.
Data logging	

■Device range check

The following table lists the details of the device range check for each specification.

Item	Description
Specification of device name by index modification	When an index-modified device is registered to the device test with execution conditions, the device range check is not performed. Therefore, the device value is not changed under the following conditions. • The index-modified device is across the boundary of the device area. • The index-modified device is out of the device range.
Specification of device name by indirect specification	When an indirectly-specified device is registered to the device test with execution conditions, the device range check is not performed. Therefore, the device value is not changed under the following conditions. • The indirectly-specified device is across the boundary of accessible areas. • The indirectly-specified device is out of the range accessible.
Specification of file register	When file registers are registered to the device test with execution conditions, whether or not the register files are assigned and the range of the file registers are not checked. Therefore, file register values are not changed under the following conditions. • File register files are not assigned. • Specified device numbers of file register is out of the file register range.

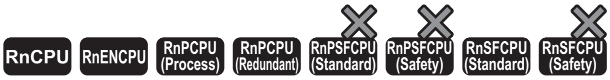
■Specification of local device

When local devices are registered to the device test with execution conditions, the write destination of the device values vary depending on the value of SM776 (Local device setting at CALL) and SM777 (Local device setting in interrupt programs). (☞ Page 945 Instruction related)

■Using together with interrupt programs

When a step number in an interrupt program is specified as an execution condition of a device test with execution conditions, the processing time of the interrupt program increases by the processing time of the device test with execution conditions. For example, when the interrupt interval of I49 set in the parameter of "Interrupt Setting from Internal Timer" under "Fixed Scan Interval Setting" is as short as 0.05ms and a step number in I49 is specified as an execution condition of a device test with execution conditions, the processing time of the interrupt program may exceed the time set in "Fixed Scan Interval Setting". Therefore, a WDT error may occur due to continuous execution of the interrupt program. For how to reduce processing time of interrupt programs, refer to descriptions of interrupt programs. (☞ Page 74 Interrupt Program)

11 DATA LOGGING FUNCTION



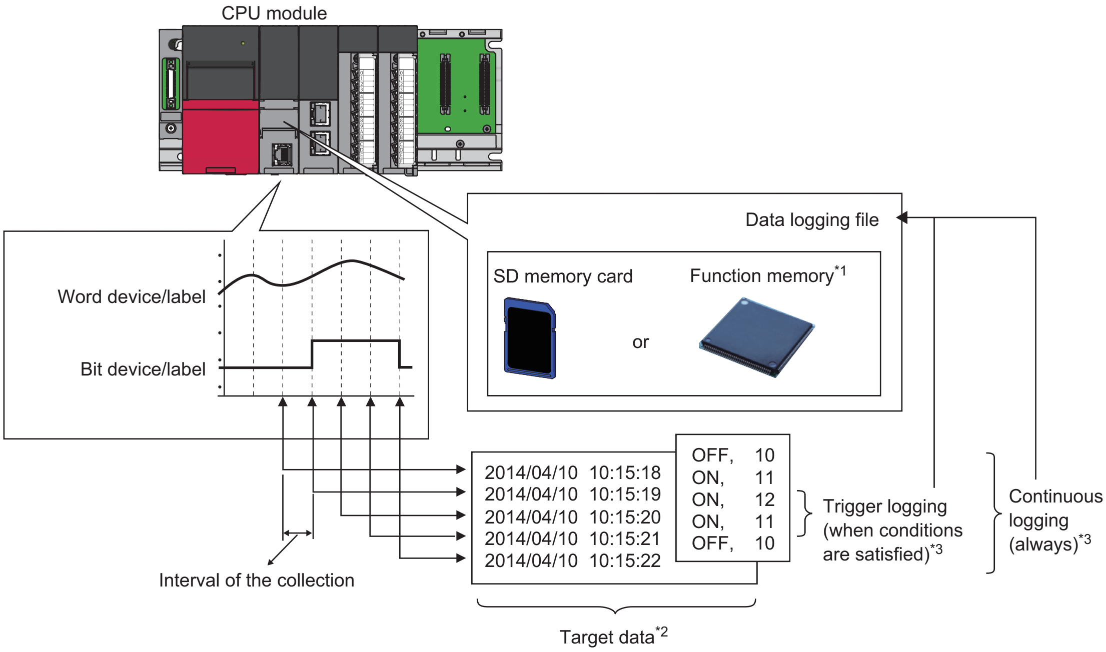
- This function cannot be used in the R00CPU.
- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 492 FUNCTIONS

- When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

This function collects specified data at a specified interval or timing, and stores them in a file.
Either the SD memory card or CPU built-in memory (function memory*1) can be specified as the save destination of the collected data.
The entire data logging function allows the entry of up to 10 data logging settings.



*1 For the CPU modules that can use the function memory, refer to the availability of storage location. (☞ Page 212 Availability)
*2 For details on the data to be collected, refer to the following. (☞ Page 200 Data to Be Collected)
*3 For details on continuous logging and trigger logging, refer to the following. (☞ Page 207 Logging Type)

Point

- Specifying the CPU built-in memory as the save destination and using the data logging file transfer function enables data logging without an SD memory card. In addition, transferring the data logging result file from the CPU built-in memory (function memory) to the data memory after the data logging completion (stop) enables data logging without relying on both the data logging file transfer function and an SD memory card. (☞ Page 216 When the data storage destination memory is the function memory)
- For the description except for the function explanations, refer to the following.

☞ Page 1104 Specifications of the Data Logging Function

- For the procedures for operating and configuring this function in CPU Module Logging Configuration Tool, refer to the following.

📖 CPU Module Logging Configuration Tool Version 1 Operating Manual

11.1 Data to Be Collected

This section describes the data to be collected by data logging.

Number of data points

The data logging function can collect up to 1280 data records. (10 settings × 128 records)[*1*2]

*1 Duplicate data records are counted as distinct.

*2 The number of data points depends on the data type.

Data type

The following table shows the number of data records for each data type.

Data type	Number of data points
Bit	1
Word (signed)	1
Double word (signed)	2
Word (unsigned)	1
Double word (unsigned)	2
Single-precision real number	2
Double-precision real number	4
Time	2
String[*1]	Specified size/2[*2]
Numeric string	Specified size/2[*2]

*1 The entered character code are output.

*2 The specified size can be 1 to 256. If the specified size is an odd number, the number of data records is rounded to the next higher integer. Example: The number of data records is 3 if the specified size is 5.

Data to be collected

The following table lists the data to be collected.

Data	Description
Global device	Set the start number for the target global device.
Local device[*1]	Set the start number for the target local device. To set a local device, set it for "Program name/#Device".
Global label[*1]	Set the label name for the target global label. To set a global label, a project of the engineering tool needs to be read.
Local label[*1]	Set the label name for the target local label. To set a local label, a project of the engineering tool needs to be read. Set the label name for "Program block name/Label name".

*1 When specifying the local device, global label, or local label, check the versions of the CPU module, engineering tool, and CPU Module Logging Configuration Tool. (☞ Page 1139 Added and Enhanced Functions)

Point

- When specifying a label, save the project of the engineering tool that contains the label in advance because the project is imported into CPU Module Logging Configuration Tool.
- Devices and labels can be specified together within a setting. When doing this, check the versions of the CPU module, engineering tool, and CPU Module Logging Configuration Tool. (☞ Page 1139 Added and Enhanced Functions)
- For local devices and local labels, only the local devices or local labels of a single program can be specified in a single data logging setting. To specify local devices or local labels of multiple programs, separate the data logging settings into multiple settings, then specify them.

Devices to be collected

The following table lists the devices that can be specified as the collected data.

Type	Device[*1]
Bit device[*2]	X, DX, Y, DY, M[*4], L, F, SM, V[*4], B, SB, T (contact)[*4*5], T (coil)[*4*5], ST (contact)[*4*5], ST (coil)[*4*5], C (contact)[*4*5], C (coil)[*4*5], LT (contact)[*4*5], LT (coil)[*4*5], LST (contact)[*4*5], LST (coil)[*4*5], LC (contact)[*4*5], LC (coil)[*4*5], FX, FY, Jn\X, Jn\Y, Jn\SB, Jn\B, BLn\S[*6]
Word device[*3]	T (current value)[*4], ST (current value)[*4], C (current value)[*4], D[*4], SD, W, SW, RD, R, ZR, Z, FD, Un\G, Jn\W, Jn\SW, U3En\G, U3En\HG
Double-word device	LT (current value)[*4], LST (current value)[*4], LC (current value)[*4], LZ

*1 An index modified device and indirectly specified device cannot be specified.
*2 For bit devices, digit specification is allowed for K1 to K8 only.
*3 For word devices, bit specification is allowed.
*4 To specify the local device, use "Program name/#Device name". (Example: "MAIN/#M1")
*5 To specify these devices with CPU Module Logging Configuration Tool, use T (contact): TS, T (coil): TC, ST (contact): STS, ST (coil): STC, C (contact): CS, C (coil): CC, LT (contact): LTS, LT (coil): LTC, LST (contact): LSTS, LST (coil): LSTC, LC (contact): LCS, and LC (coil): LCC.
*6 The device can be collected by using the CPU module where the SFC function can be used. (☞ Page 1139 Added and Enhanced Functions)

Labels to be collected

The following table lists the labels that can be specified to be collected.[*1*2]

Type	Class	Data type
Global label	• VAR_GLOBAL • VAR_GLOBAL_RETAIN	■Primitive data type/array[*7] • Bit[*4] • Word (signed)[*5] • Double word (signed) • Word (unsigned)[*5] • Double word (unsigned) • Single-precision real number • Double-precision real number • Time • String • String [Unicode][*9] • Timer type[*6] • Retentive timer type[*6] • Counter type[*6] • Long timer type[*6] • Long retentive timer type[*6] • Long counter type[*6] ■Structure[*8]
Local label[*3]	• VAR • VAR_RETAIN	

*1 A label with a name of 256 characters or less can be specified. When the local label is specified (program block name/label name), the "program block name/" part is counted as the number of characters.
*2 Only labels in the read project of the engineering tool can be specified.
*3 Specify the label with "Program block name/Label name". (Example: "ProgPou/label_w1")
*4 Digit-specified labels cannot be specified.
*5 Bit specification is allowed.
*6 For the label of timer type and counter type, specify the element name. (S: contact, C: coil, N: current value)
Global label example: "label_w1.S"
Local label example: "ProgPou/label_w1.S"
*7 Specify the array element. If the array element is not specified, it is handled as the start of the array ([0]).
*8 Specify the structure member.
*9 Up to 128 characters can be specified.

Point

When specifying a label, save the project of the engineering tool that contains the label in advance because the project is imported into CPU Module Logging Configuration Tool.

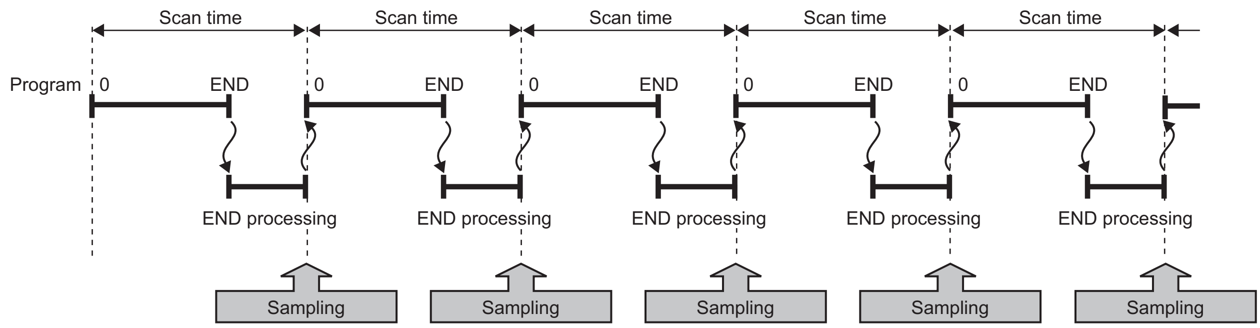
11.2 Data Collection Conditions

This section describes the timing when data is collected and the conditions under which data is collected.

Data collection conditions		Description
Each scan		Collects data during the END processing of each scan.
Time specification	Data collection at specified time interval	Collects data at specified time interval.
	Data collection during the END processing after specified time interval	Collects data during the END processing after specified time interval.
Interrupt occurrence		Collects data at specified time interval after the execution of an interrupt program.
Condition specification	Device/label specification	Collects data when the monitored data meets the specified condition during the END processing.
	Step No. specification	Collects data when the specified condition is met immediately before the execution of the specified step.

Each scan

Data are collected during the END processing of each scan.



Time specification

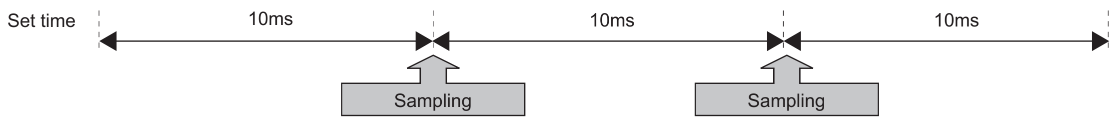
Specify the collection time interval. Specify the timing of data collection using advanced settings as well.

Data collection at specified time interval

Data are collected at specified time interval.

Ex.

10 milliseconds (10ms)

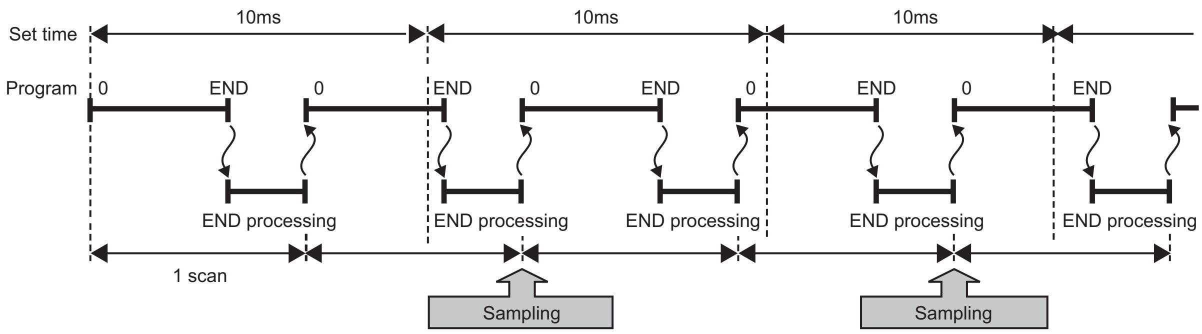


Data collection during the END processing after specified time interval

This option causes data collection to be performed at the timing of the END processing rather than during the course of program execution. Ensure that the "Scan time" is less than "Time specification". If the scan time is longer than the specified time and the collection interval or the collection timing occurs more than once during the same scan, data is collected only once during the END processing. Data collection is performed on a scan by scan basis, which is the same behavior as when "Each scan" is used.

Ex.

10 milliseconds (10ms)

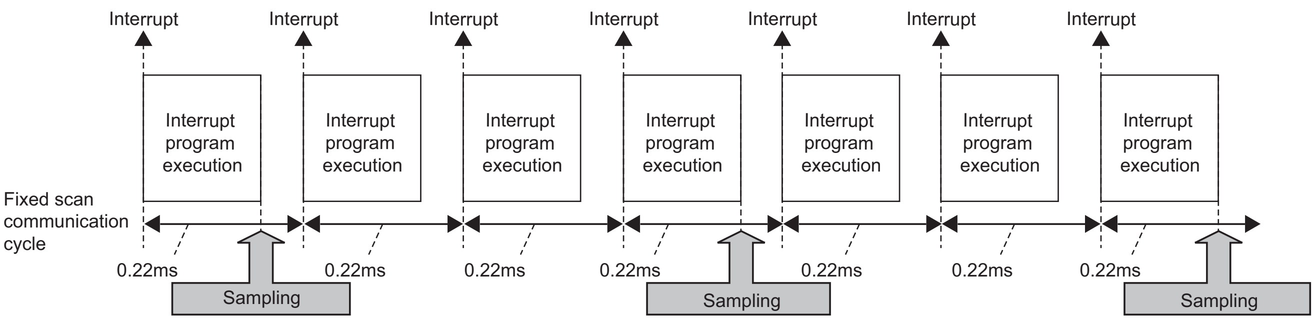


Interrupt occurrence

Data are collected at specified time interval after the execution of an interrupt program. The time interval is specified by "Interrupt cycle specified sampling intervals"× "Interval". Interrupt pointers which can be specified are I28 to I31, I44, I45, I48, and I49.

Ex.

When multiple CPU synchronous interrupt (I45) is used (multiple CPU synchronous interrupt is set to 0.22 milliseconds (0.22ms) and interval to 3):



Point

Since data collection is performed at the timing of the execution of the interrupt program, data is collected only when all of the following conditions are met:

- The specified interrupt meets the operating condition.
- The specified interrupt pointer exists in the program.
- The current state is an EI state and the interrupt mask of the interrupt pointer has been reset.

Condition specification

Specify the data collection timing according to the device/label data conditions and step number. The AND condition using a combination of "Device specification", "Label specification", and "Step No. specification" results in the collection of data at the time when both conditions are established.

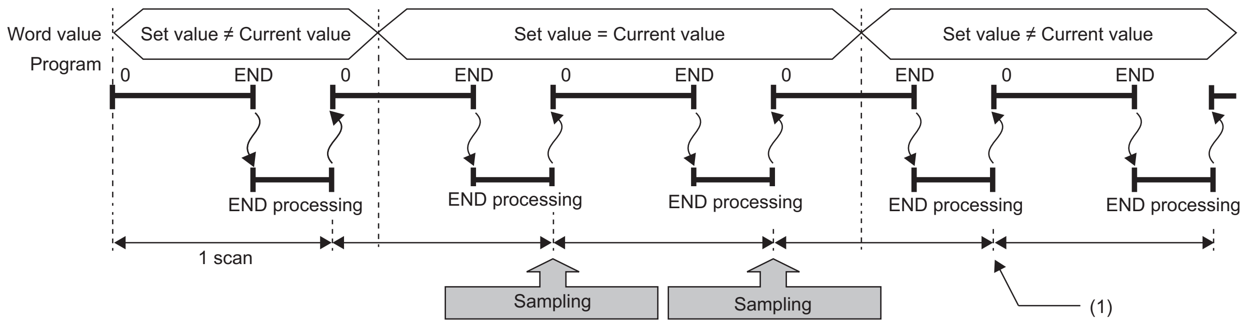
Device/label specification

Data are collected when the monitored data meets the specified condition during the END processing.

■To collect data continuously while the conditions are met

The following conditional formula causes the data logging function to collect data continuously while the conditions are met:

- =: When the current value of the monitored data is equal to the comparison value
- ≠: When the current value of the monitored data is not equal to the comparison value
- ≥: When the current value of the monitored data is equal to or larger than the comparison value
- >: When the current value of the monitored data is larger than the comparison value
- ≤: When the current value of the monitored data is equal to or smaller than the comparison value
- <: When the current value of the monitored data is smaller than the comparison value

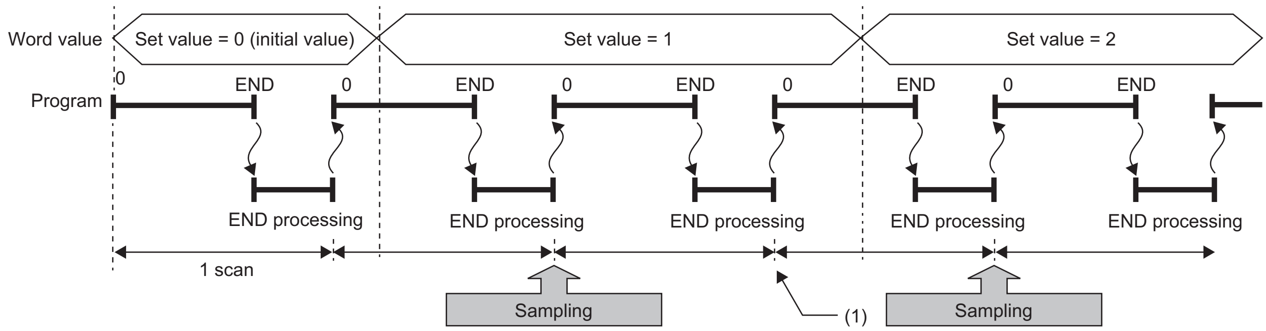


(1) During the END processing, the data logging function does not collect data because the conditions are not met.

■To collect data only when the state changes

The following conditional formula causes the data logging function to collect data only during the END processing for the scans where the conditional formula is met. It does not collect data for any single scan where the conditional formula is not met during the END processing (even if the conditional formula is met before the END processing is initiated).

- ↑: When the specified data turns on
- ↓: When the specified data turns off
- At change: When the current value of the specified data changes



(1) The data logging function does not collect data because there has been no change in state since the last scan.

■Specifying the monitored data

For monitored data, the following devices and labels can be specified.[* 1]

*1 When specifying the local device, global label, or local label, check the versions of the CPU module, engineering tool, and CPU Module Logging Configuration Tool. (☞ Page 1139 Added and Enhanced Functions)

The data types that can be selected include bit/word (unsigned), word (signed), double word (unsigned), and double word (signed).

Type		Device[*2]
Device	Bit device[*3]	X, Y, M[*5], L, F, SM, V[*5], B, SB, T (contact)[*5*6], ST (contact)[*5*6], C (contact)[*5*6], LT (contact)[*5*6], LST (contact)[*5*6], LC (contact)[*5*6], FX, FY, Jn\X, Jn\Y, Jn\SB, Jn\B, BLn\S[*7]
	Word device[*4]	T (current value)[*5], ST (current value)[*5], C (current value)[*5], D[*5], SD, W, SW, RD, R, ZR, Z, FD, Un\G, Jn\W, Jn\SW, U3En\G, U3En\HG
	Double-word device	LT (current value)[*5], LST (current value)[*5], LC (current value)[*5], LZ

*2 An index modified device and indirectly specified device cannot be specified.
*3 For bit devices, digit specification is not supported.
*4 For word devices, bit specification is allowed.
*5 To specify the local device, use "Program name/#Device name". (Example: "MAIN/#M1")
*6 To specify these devices with CPU Module Logging Configuration Tool, use T (contact): TS, ST (contact): STS, C (contact): CS, LT (contact): LTS, LST (contact): LSTS, and LC (contact): LCS.
*7 The device can be collected by using the CPU module where the SFC function can be used. (☞ Page 1139 Added and Enhanced Functions)

Type		Class	Data type
Label[*8*9]	Global label	• VAR_GLOBAL • VAR_GLOBAL_RETAIN	■Primitive data type/array[*15] • Bit[*11] • Word (signed)[*12] • Double word (signed) • Word (unsigned)[*12] • Double word (unsigned) • Timer type[*13*14] • Retentive timer type[*13*14] • Counter type[*13*14] • Long timer type[*13*14] • Long retentive timer type[*13*14] • Long counter type[*13*14] ■Structure[*16]
	Local label[*10]	• VAR • VAR_RETAIN	

*8 A label with a name of 256 characters or less can be specified. When the local label is specified (program block name/label name), the "program block name/" part is counted as the number of characters.
*9 Only labels in the read project of the engineering tool can be specified.
*10 Specify the label with "Program block name/Label name". (Example: "ProgPou/label_w1")
*11 Digit-specified labels cannot be specified.
*12 Bit specification is allowed.
*13 Only the contacts and current values can be specified.
*14 For the label of timer type and counter type, specify the element name. (S: contact, N: current value)
Global label example: "label_w1.S"
Local label example: "ProgPou/label_w1.S"
*15 Specify the array element. If the array element is not specified, it is handled as the start of the array ([0]).
*16 Specify the structure member.

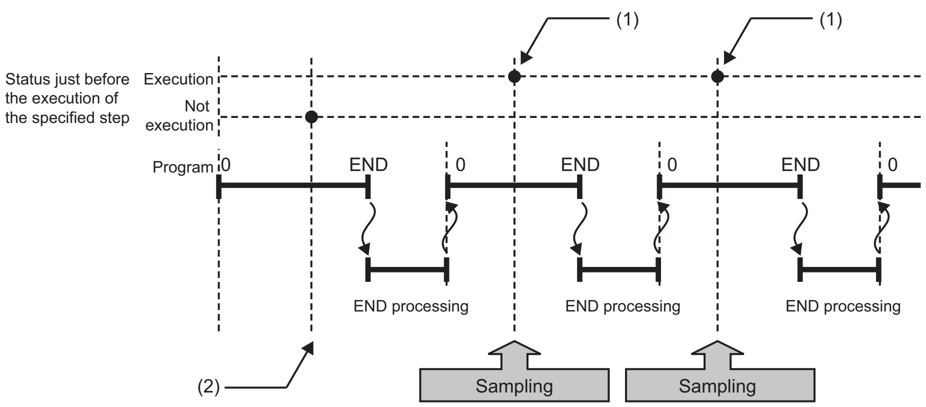
Step No. specification

Collects data when the specified condition is met immediately before the execution of the specified step.

■To collect data continuously while the execution conditions are met

The following execution conditions cause the data logging function to collect data continuously while the execution condition are met:

- Always: The specified step is executed regardless of the state immediately before the execution of it.
- In the specified condition satisfied: The specified step is executed if the state immediately before the execution is a running state.
- In the specified condition not satisfied: The specified step is executed if the state immediately before the execution is not a running state.



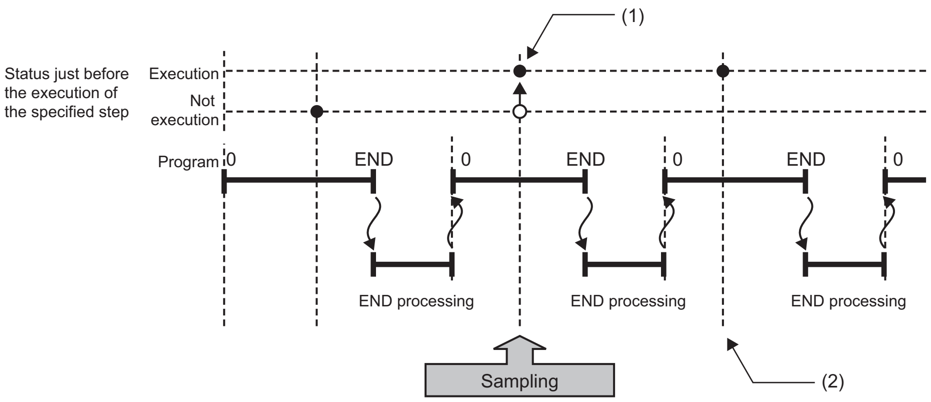
(1) The data logging function collects data because the state immediately before the execution of the specified step is a running state.

(2) It does not collect data because the state immediately before the execution of the specified step is not a running state.

■To collect data only when the execution conditions are met

The following execution conditions cause the data logging function to collect data only when the execution conditions are met:

- On the rising edge of the specified condition: The specified step is executed if the state changes from non-running to running immediately before the execution of it.
- On the falling edge of the specified condition: The specified step is executed if the state changes from running to non-running immediately before the execution of it.



(1) The data logging function collects data because the state changes from running to non-running immediately before the execution of the specified step.

(2) The data logging function does not collect data because there has been no change in state since the last scan.

Point

If the specified step is contained between FOR and NEXT loop of instructions, the data logging function collects the data for only the first iteration of the loop where the specified conditions are met.

Restriction

Since the step number cannot be checked in the following programs, the step number cannot be specified.

- Program having multiple parts
- ST program
- FB program
- SFC program

11.3 Logging Type

The following table describes available methods of data collection:

Logging type	Data collection method	Application
Continuous logging	Continuously collects specified data at specified interval or timing.	Allows the user to continuously monitor the content of specified data.
Trigger logging	Collects specified data at specified interval or timing and extracts a specified number of data records before and after the satisfaction of a trigger condition.	Allows the user to monitor the content of specified data before and after the satisfaction of a trigger condition.

11

Operating procedure for continuous logging

In continuous logging, the CPU module stores specified data in its internal buffer at a specified collection interval or timing and, at the time of a file save operation, it saves the data in a data logging file residing in the storage memory.

1. Write the settings into data memory or SD memory card using CPU Module Logging Configuration Tool.

2. Operate CPU Module Logging Configuration Tool to start data logging. The data logging settings are registered and continuous logging begins. (The special relay (data logging start) turns on.)

3. Data collection finishes upon reaching "Number of files to be saved" specified as part of the "Stop" setting configured in "Operation when exceeds the number of files".

4. Specify the desired file in the storage memory to read the results of data logging.

Point

In continuous logging, data logging files are continuously created, thus allowing the user to read the results of data logging any time without having to wait for the completion of collection.

To stop continuous logging

The user can completely stop data logging by instructing CPU Module Logging Configuration Tool to stop data logging and unregister the data logging settings stored in the CPU module. (The special relay (data logging start) turns off.)

To suspend/resume continuous logging

The user can suspend data logging with the data logging settings remaining intact by doing either of the following:
- Instruct CPU Module Logging Configuration Tool to suspend data logging (the special relay (data logging start) turns off).
- Turn on the special relay (Data logging suspend/resume flag).

To resume continuous logging from suspension, do either of the following:
- Instruct CPU Module Logging Configuration Tool to resume data logging (the special relay (data logging start) turns on).
- Turn off the special relay (Data logging suspend/resume flag).

Operating procedure for trigger logging

In trigger logging, the CPU module stores specified data in its internal buffer at a specified collection interval or timing; it extracts a specified number of data records before and after the satisfaction of a trigger condition and saves the extracted data in a data logging file residing in the storage memory. Note that data collection is performed not only at the specified collection interval or timing but also when a trigger condition is met. In addition, once a trigger condition is met, any subsequent trigger conditions are ignored.

1. Write the settings into data memory or SD memory card using CPU Module Logging Configuration Tool.
2. Instruct CPU Module Logging Configuration Tool to start data logging. The data logging settings are registered and trigger logging begins. (The special relay (data logging start) turns on.)
3. Wait until the trigger condition is met. (Trigger standby)
4. The data specified in CPU Module Logging Configuration Tool is collected. (Trigger condition met)
5. Data collection is completed by collecting as much data as the number of records specified in CPU Module Logging Configuration Tool and writing the collected data into the storage memory.
6. Specify the desired file in the storage memory to read the results of data logging.

Trigger condition

The following table lists the conditions to be used as a trigger.

Trigger condition		Description
Condition specification	Device/label change specification	A trigger occurs when the monitored data meets the specified condition.
	Step No. specification	A trigger occurs when the specified condition is met immediately before the execution of the specified step.
When trigger instruction executed		A trigger occurs when the LOGTRG instruction is executed.

Point

The AND condition using a combination of "Device/Label change specification" and "Step No. specification" results in the collection of data at the time when both conditions are established.

■Device/label change specification

A trigger occurs when the monitored data meets the specified condition.

Conditional formula	Description
↑	When the specified data turns on
↓	When the specified data turns off
=	When the monitored data is equal to the comparison value, regardless of whether or not its current value is equal.
≠	When the monitored data is not equal to the comparison value, regardless of whether or not its current value is equal.
≥	When the monitored data is greater than or equal to the comparison value, regardless of whether or not its current value is equal.
>	When the monitored data is greater than the comparison value, regardless of whether or not its current value is equal.
≤	When the monitored data is less than or equal to the comparison value, regardless of whether or not its current value is equal.
<	When the monitored data is less than the comparison value, regardless of whether or not its current value is equal.
At change	When the current value of the specified data changes

- Specifying the monitored data

For the device and label change specification, monitored data can be configured to be collected from the devices and labels listed in the following table.*1

*1 When specifying the local device, global label, or local label, check the versions of the CPU module, engineering tool, and CPU Module Logging Configuration Tool. (☞ Page 1139 Added and Enhanced Functions)

The data types that can be selected include bit/word (unsigned), word (signed), double word (unsigned), and double word (signed). If double word (unsigned) or double word (signed) is specified, a trigger occurs only when as much data as one double word is written. No trigger occurs when only the upper or lower word of a double word is written.

Type		Device*2
Device	Bit device*3	X, Y, M*5, L, F, SM, V*5, B, SB, T (contact)*5*6, ST (contact)*5*6, C (contact)*5*6, LT (contact)*5*6, LST (contact)*5*6, LC (contact)*5*6, FX, FY
	Word device*4	T (current value)*5, ST (current value)*5, C (current value)*5, D*5, SD, W, SW, RD, R, ZR, Z, FD
	Double-word device	LT (current value)*5, LST (current value)*5, LC (current value)*5

*2 An index modified device and indirectly specified device cannot be specified.
*3 For bit devices, digit specification is not supported.
*4 For word devices, bit specification is allowed.
*5 To specify the local device, use "Program name/#Device name". (Example: "MAIN/#M1")
*6 To specify these devices with CPU Module Logging Configuration Tool, use T (contact): TS, ST (contact): STS, C (contact): CS, LT (contact): LTS, LST (contact): LSTS, and LC (contact): LCS.

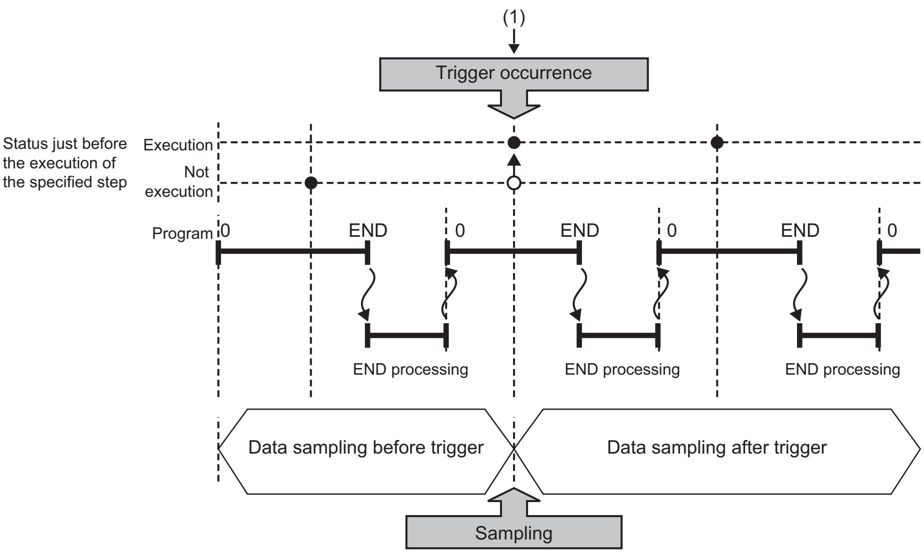
Type		Class	Data type
Label*7*8	Global label	• VAR_GLOBAL • VAR_GLOBAL_RETAIN	■Primitive data type/array*14 • Bit*10 • Word (signed)*11 • Double word (signed) • Word (unsigned)*11 • Double word (unsigned) • Timer type*12*13 • Retentive timer type*12*13 • Counter type*12*13 • Long timer type*12*13 • Long retentive timer type*12*13 • Long counter type*12*13 ■Structure*15
	Local label*9	• VAR • VAR_RETAIN	

*7 A label with a name of 256 characters or less can be specified. When the local label is specified (program block name/label name), the "program block name/" part is counted as the number of characters.
*8 Only labels in the read project of the engineering tool can be specified.
*9 Specify the label with "Program block name/Label name". (Example: "ProgPou/label_w1")
*10 Digit-specified labels cannot be specified.
*11 Bit specification is allowed.
*12 Only the contacts and current values can be specified.
*13 For the label of timer type and counter type, specify the element name. (S: contact, N: current value)
Global label example: "label_w1.S"
Local label example: "ProgPou/label_w1.S"
*14 Specify the array element. If the array element is not specified, it is handled as the start of the array ([0]).
*15 Specify the structure member.

■Step No. specification

A trigger occurs when the specified condition is met immediately before the execution of the specified step.

Execution condition	Description
Always	Executes the specified step regardless of the state immediately before the execution of it.
In the specified condition satisfied	Executes the specified step if the state immediately before the execution is a running state.
In the specified condition not satisfied	Executes the specified step if the state immediately before the execution is not a running state.
On the rising edge of the specified condition	Executes the specified step if the state changes from non-running to running immediately before the execution of it.
On the falling edge of the specified condition	Executes the specified step if the state changes from running to non-running immediately before the execution of it.



(1) The state changes from non-running to running immediately before the execution of the specified step.

Point

If the specified step is contained between FOR and NEXT loop of instructions, only the first iteration of the loop where the specified conditions are met is handled as a trigger.

Restriction

Since the step number cannot be checked in the following programs, the step number cannot be specified.

- Program having multiple parts
- ST program
- FB program
- SFC program

Restriction

When a step number in an interrupt program is specified, the trigger condition may not be met.

■When trigger instruction executed

A trigger occurs when the LOGTRG instruction is executed. (📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))

Restriction

When the LOGTRG instruction is used in an interrupt program, the trigger condition may not be met.

To stop trigger logging

The user can completely stop data logging by instructing CPU Module Logging Configuration Tool to stop data logging and unregister the data logging settings stored in the CPU module. (The special relay (data logging start) turns off.)

To suspend/resume trigger logging

The user can suspend data logging with the data logging settings remaining intact by doing either of the following:

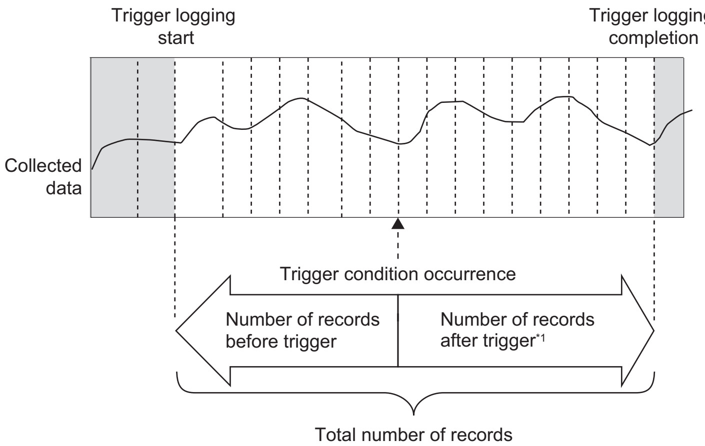
- Instruct CPU Module Logging Configuration Tool to suspend data logging (the special relay (data logging start) turns off).
- Turn on the special relay (Data logging suspend/resume flag).

To resume trigger logging from suspension, do either of the following:

- Instruct CPU Module Logging Configuration Tool to resume data logging (the special relay (data logging start) turns on).
- Turn off the special relay (Data logging suspend/resume flag).

Number of records

Specify the number of records to be collected before and after the satisfaction of a trigger condition.



*1 This number includes the record exactly at the time when the trigger condition is met.

Point

After starting data logging, if the trigger condition is met before data collection of the specified number of records (before trigger) is completed, the number of sampled records will be less than the specified.

11.4 Data Logging File

This section describes data logging files.

Storage format of data logging files

The following storage formats are available for data logging files. For details on the output format, format specifications, and output contents of each file, refer to the data output format. (☞ Page 1109 Data output type)

File format	Application
Unicode text file format	This is a file format which can be opened in generic-purpose application programs such as Excel and Notepad. GX LogViewer is also available for displaying data.
CSV file format*1*2	This is a file format which can be opened in generic-purpose application programs such as Excel and Notepad. GX LogViewer is also available for displaying data. Select this format when saving data together with the output data of MELSEC-Q series.
Binary file format	Comparing the Unicode text file format, the size of files is small and therefore quicker access to files is provided. GX LogViewer is also available for displaying data.

*1 When setting the data logging file storage format to the CSV file format, check the firmware version of the CPU module and the version of the CPU Module Logging Configuration Tool. (☞ Page 1139 Added and Enhanced Functions)

*2 If the data name (device/label name), device comment, or program name contains characters that cannot be converted to ASCII (Shift-JIS), the relevant data is output as a one-byte period (.).

Storage location of data logging files

Select an SD memory card or the function memory for the storage location.

The number of data logging settings and the number of records vary, depending on the storage location. For details, refer to the functional specifications.

Availability

The following table lists available CPU module models by each storage location.

○: Applicable, ×: Not applicable

CPU module	Storage location	
	SD memory card	Function memory
R01CPU, R02CPU	○	×
R04CPU, R08CPU, R16CPU, R32CPU, R120CPU R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, R120ENCPU	○	○*1
R08PCPU, R16PCPU, R32PCPU, R120PCPU	○	×
R08SFCPU, R16SFCPU, R32SFCPU, R120SFCPU	○	×

*1 The function memory can be used in the CPU module with a firmware version "31" or later. (☞ Page 1139 Added and Enhanced Functions)

Restriction

- Data logging into different storage locations (SD memory card and function memory) cannot be executed simultaneously.
- When the storage location is the function memory, up to two data loggings can be executed simultaneously. (Up to 10 data loggings can be set.) The maximum number of data loggings that can be executed simultaneously is ten, depending on the firmware version and production information of the CPU module. (☞ Page 1139 Added and Enhanced Functions)

11.5 States of the Data Logging Function

The data logging function has the data logging state. The data logging state can be checked by CPU Module Logging Configuration Tool.

Data logging states

The following table lists all the possible data logging states.

Data logging states	Description
Stop[*1]	Data logging settings are unregistered and data collection is inactive.
Stop (after collection)	Transition from "Collection completed" to "Stop" has occurred due to the execution of another function[*2].
Stop (after error)	Transition from "Error" to "Stop" has occurred due to the execution of another function[*2].
RUN waiting (no collection)	Data collection has not yet begun because the operating status of the CPU module is not in the RUN state.
Start waiting (no collection)	Data collection is inactive, waiting for the start command.
Pause[*1]	Data logging is suspended and data collection has not yet been started. (The data logging settings remain intact.)
Condition waiting (no collection)	Data logging settings are registered and waiting for the first collection timing.
Collecting	Continuous logging is active and collecting data.
Trigger waiting (Collecting before trigger)	Data logging settings are registered, data collection is being performed, and waiting until the trigger condition is met.
Collecting after trigger	Trigger logging is active and collecting data after the trigger condition is met.
Collection completed[*1]	• Continuous logging: Data collection has finished upon reaching "Number of files to be saved" specified as part of the "Stop" setting configured in "Operation when exceeds the number of files". (The data logging settings remain intact.) • Trigger logging: Trigger logging has finished collecting data as much as the specified number of records. (The data logging settings remain intact.)
Error	Data logging has failed due to the occurrence of an error.

*1 CPU Module Logging Configuration Tool displays "Saving the logging data" as the data logging state until saving of collected data completes. After the completion, the state changes to each of three states.

*2 The execution of another function includes:
· Execution of data logging with the same trigger conditions (trigger conditions = the specified conditions)
· Auto logging
· Online change

LED status

Whether the data logging function is active or not can be checked by the LED of the CPU module.

States of the Data Logging Function	LED status		
	FUNCTION LED	CARD READY LED[*1]	CARD ACCESS LED[*1]
• Data logging settings have been registered by the start operation from CPU Module Logging Configuration Tool. • After registering the auto logging common setting, an SD memory card that stores the setting for which the auto logging function is enabled has been inserted.	On	On	Off
All of the registered data logging sessions are in the state "RUN waiting (no collection)", "Start waiting (no collection)", "Pause", "Condition waiting (no collection)", or "Trigger waiting (Collecting before trigger)".			
One or more of the registered data logging sessions are in the state "Collecting" (including the data being saved) or "Collecting after trigger" (including the data being saved).	Flashes slowly (every one second)	On	On (when the SD memory card is accessed)
All of the registered data logging sessions have finished (or failed due to an error).	Flashes at normal rate (every 200ms)	On	Off

*1 The LED status when the data storage destination memory is the SD memory card.

Point

The FUNCTION LED follows the status of the following LED indications.

• When the external input/output forced on/off function is executed (in registration) (☞ Page 182 FUNCTION LED)

• When program restoration information is not written (☞ Page 1101 Checking the program restoration information write status)

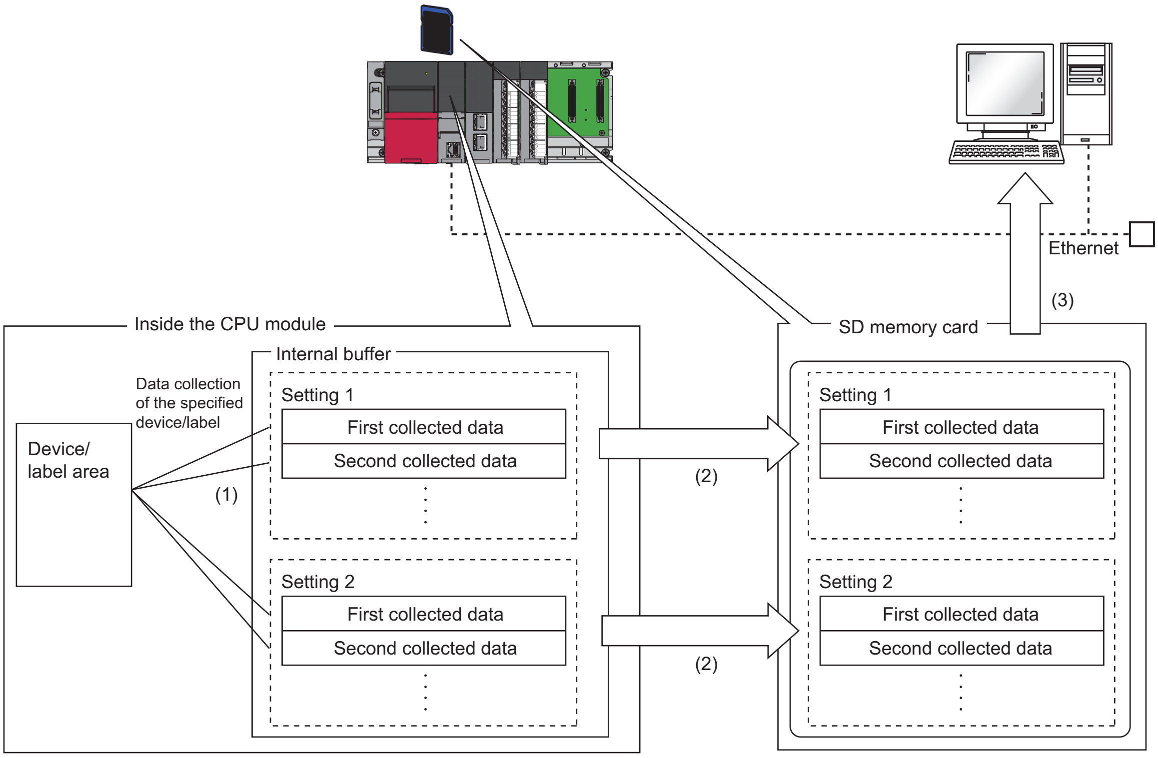
After the external input/output forced on/off function has been executed (after the registration is canceled) and program restoration information has been written, the LED indicator follows the status of the data logging function. (☞ Page 143 LED display setting)

11.6 Steps Until the Collected Data Is Saved

This section describes the steps until the collected data is saved.

When the data storage destination memory is the SD memory card

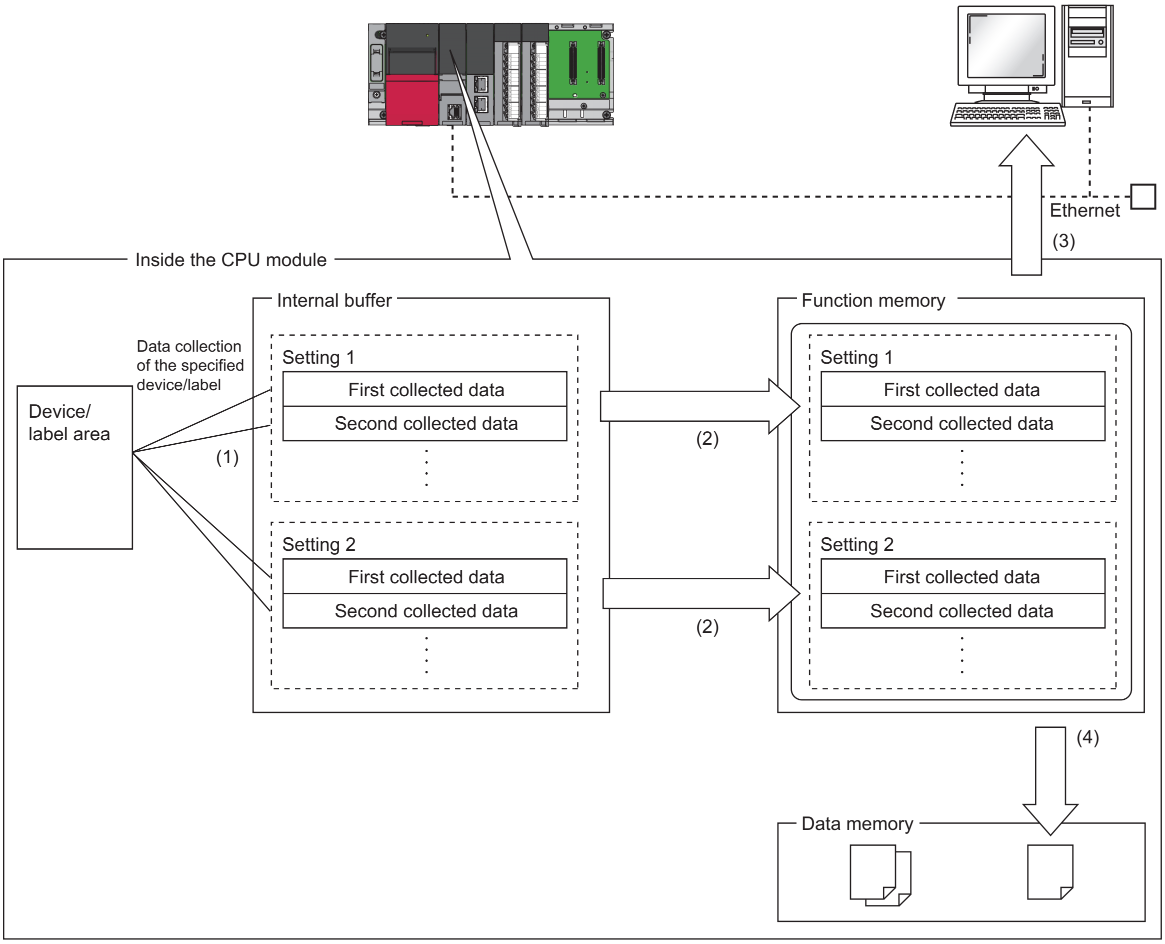
The following figure shows the flow of data when the data storage destination memory is the SD memory card.



(1) The collected data is temporarily stored in the specified internal buffer. (☞ Page 217 Internal buffer)
(2) The data stored in the internal buffer is stored into the SD memory card at the timing of a file save operation.
(3) With the file transfer setting, data logging files can be transferred from the SD memory card to the FTP server. (☞ Page 226 Data Logging File Transfer (Auto Transfer to FTP Server))

When the data storage destination memory is the function memory

The following figure shows the flow of data when the data storage destination memory is the function memory.



(1) The collected data is temporarily stored in the specified internal buffer. (☞ Page 217 Internal buffer)

(2) The data stored in the internal buffer is stored into the function memory at the timing of a file save operation.

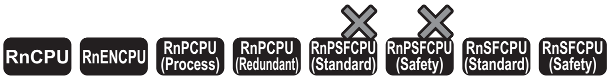
(3) With the file transfer setting, data logging files can be transferred from the function memory to the FTP server. (☞ Page 226 Data Logging File Transfer (Auto Transfer to FTP Server))

(4) Without the file transfer setting, data logging files are transferred to the data memory upon completion or stopping of the data logging (including when an error occurs). (☞ Page 234 Data Logging File Transfer to Data Memory)

Internal buffer

The internal buffer is a system area used to temporarily store collected data.
The collected data is temporarily stored in the internal buffer and stored in the specified data storage destination memory at the time of a file save operation.

Internal buffer capacity setting



• This function cannot be used in the R00CPU.

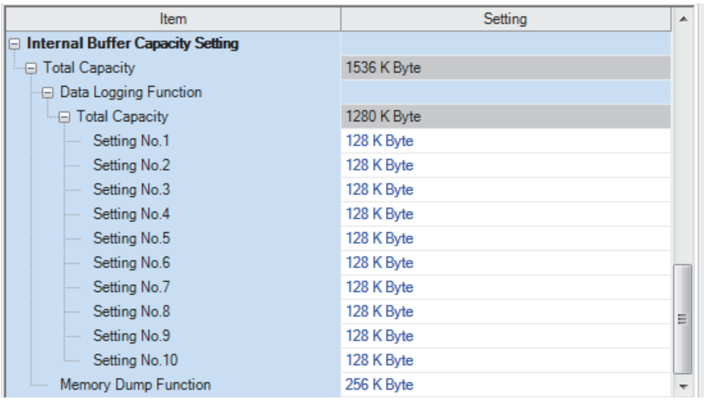
Set the capacity of the area (internal buffer) used by the system to temporarily store the results of data logging and memory dump processing. The capacity can be set individually for each data logging setting number (1 to 10).

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Internal Buffer Capacity Setting]

For trigger logging, increasing the internal buffer capacity allows an increase in the number of collected data before trigger and also helps to prevent processing overflow. If the free space in the internal buffer is still insufficient after increasing the internal buffer capacity, use the following workarounds:

- Increase the data collection interval or timing.
- Reduce the number of data records to be collected.
- Lower the frequency of file switching.

Window



Displayed items

Item		Description	Setting range	Default
Total Capacity		Shows the total of the internal buffer capacity set in the data logging function and the memory dump function.	64 to 3072K bytes	1536K bytes
Data Logging Function	Total Capacity	Shows the total of the internal buffer capacity used for the data logging function.	—	1280K bytes
	Setting No.1 to 10	Sets the internal buffer capacity used for each Setting No. of the data logging settings.	• Each setting range: 32 to 3040K bytes (in increments of 1K bytes)*1 • Total setting range: 32 to 3040K bytes	128K bytes
Memory Dump Function		Sets the internal buffer capacity used for the memory dump function.	32 to 3040K bytes (in increments of 1K bytes)*2	256K bytes

*1 Leaving this field blank allows the setting to be unused (0K bytes).
*2 A reduced capacity of the internal buffer decreases the extension of scan time, but it takes time to complete.

Point

The internal buffer is also consumed in the real-time monitor function. Set the total of the internal buffer to no more than 3072K bytes including the internal buffer to be consumed in the real-time monitor function as well. The internal buffer for the real-time monitor function can be set with GX LogViewer. (GX LogViewer Version 1 Operating Manual)

Restriction

The internal buffer capacity setting cannot be set in the CPU module used or the function that is not supported by the engineering tool.
Check whether the engineering tool supports the function with availability for each model in the function list. (MELSEC iQ-R CPU Module User's Manual (Startup))

Amount of internal buffer consumed

This value can be calculated by multiplying "Number of data points" by 2 bytes. Note, however, that additional space is consumed by columns configured for output, as indicated below:

- Date/time column: 10 bytes
- Data collection interval column: 8 bytes
- Execution step No. column: 10 bytes
- Execution program No. column: 2 bytes
- Index column: 4 bytes

Ex.

When data logging is configured to collect as much data as one setting × 128 records and output all of the columns (i.e., maximum allowable configuration):

128 × 2 + (10 + 8 + 10 + 2 + 4) = 290 bytes

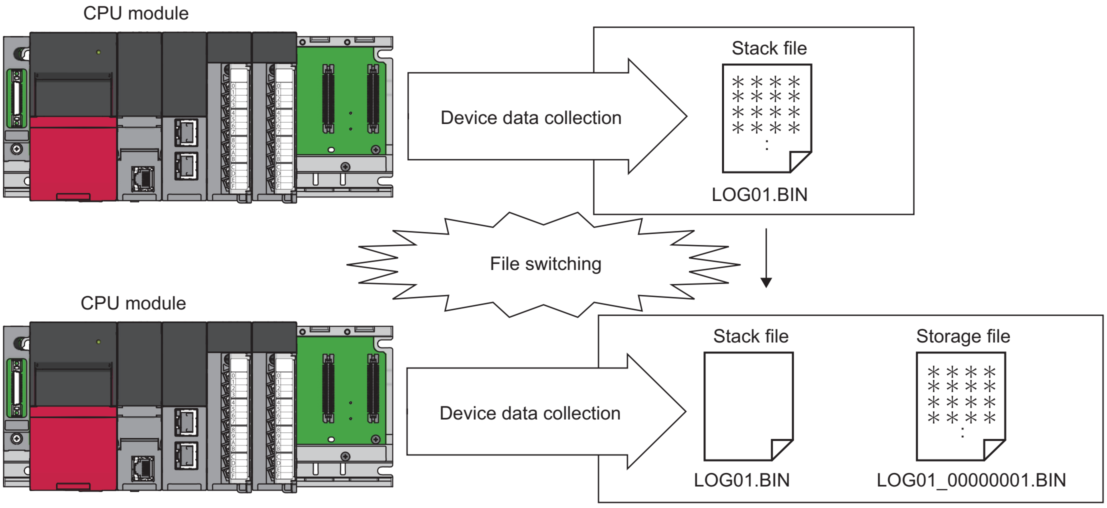
Switching to a storage file

The data collected by data logging is temporarily stored in a stack file. The stack file can be switched to a storage file to free the space in the SD memory card.

How file switching works

File switching works as follows:

1. The CPU module writes collected data into a stack file (such as LOG01.BIN).
2. It changes the file name when the storage file switching condition is met.[*1*2]
3. It creates a new stack file.
4. It continues to write collected data into the newly created stack file.



*1 The file name format can be customized.
*2 The file number of the most recent storage file is stored in the special register (Latest storage file number).

11

File switching condition

The following table shows the conditions to switch files.

File switching condition	Description
Number of records	When the number of records exceeds the specified value, the files are switched. Set the number of records within the following range. • When the data storage destination memory is the SD memory card: 1 to 65500 • When the data storage destination memory is the function memory: 1 to 12000
File size*1	When the file size exceeds the specified value, the files are switched. Set the file size in units of kilobytes within the following range. • When the data storage destination memory is the SD memory card: 10 to 16384K bytes • When the data storage destination memory is the function memory: 10 to 1024K bytes
Condition*2	When the monitored data meets the specified conditions, the files are switched. Specifies the following data. (☞ Page 220 Condition specification) • Monitored data • Conditional formula: ↑, ↓, =, ≠, ≥, >, ≤, <, At value change • Condition values (comparison values)

*1 File switching occurs before the file grows beyond the specified size.

*2 When setting specified conditions, check the firmware version of the CPU module and the version of the CPU Module Logging Configuration Tool. (☞ Page 1139 Added and Enhanced Functions)

However, file switching occurs regardless of the setting when:

- the number of records has reached the maximum number;
- the file size has reached the maximum size;
- the CPU module is stopped or suspended/resumed; or
- data logging is started and there is an existing stack file.

Point

Trigger logging does not require the configuration of these settings because the stack file is automatically switched to a storage file after as much data as the specified number of records is written into the stack file. SM1218 (logging data storage file switching in progress) can be used to check if storage file switching is in progress.

■Condition specification

The stack file is switched to the storage file at the END processing of when the specified conditions are met. Therefore, the storage file is not switched if the conditions are met within one scan.

Even when the file switching conditions are met, the stack file is not switched if the data does not exist in the internal buffer or the stack file does not exist because the data collection conditions are not met or for the other reasons. The stack file is switched to the storage file when the data is recorded in the internal buffer and data is written to the stack file.

- Specifying a conditional formula

The following symbols can be specified for the conditional formula.

Conditional formula	Description
↑	When the specified data turns on
↓	When the specified data turns off
=	When the current value of the monitored data is equal to the comparison value
≠	When the current value of the monitored data is not equal to the comparison value
≥	When the current value of the monitored data is equal to or larger than the comparison value
>	When the current value of the monitored data is larger than the comparison value
≤	When the current value of the monitored data is equal to or smaller than the comparison value
<	When the current value of the monitored data is smaller than the comparison value
At change	When the current value of the specified data changes

- Specifying the monitored data

For monitoring data, the following can be set.

The data types that can be selected include bit/word (unsigned), word (signed), double word (unsigned), and double word (signed).

Type		Device*1
Device	Bit device*2	X, Y, M*4, L, F, SM, V*4, B, SB, T (contact)*4*5, ST (contact)*4*5, C (contact)*4*5, LT (contact)*4*5, LST (contact)*4*5, LC (contact)*4*5, FX, FY
	Word device*3	T (current value)*4*5, ST (current value)*4*5, C (current value)*4*5, D*4, SD, W, SW, RD, R, ZR, FD
	Double-word device	LT (current value)*4*5, LST (current value)*4*5, LC (current value)*4*5

*1 An index modified device and indirectly specified device cannot be specified.

*2 For bit devices, digit specification is not supported.

*3 For word devices, bit specification is allowed.

*4 To specify the local device, use "Program name/#Device name". (Example: "MAIN/#M1")

*5 To specify these devices with CPU Module Logging Configuration Tool, use T (contact): TS, ST (contact): STS, C (contact): CS, LT (contact): LTS, LST (contact): LSTS, and LC (contact): LCS.

Storage file

The CPU module creates a subfolder ("storage file container folder") under the file storage folder and writes storage files to that storage file container folder. One storage file container folder can contain up to 256 storage files. When the files contained in the current storage file container folder reach the maximum number, the CPU module creates a new storage file container folder at the time of next storage file switching and begins writing storage files to that new folder. The number of files that can be contained in one file storage folder is configurable within the range of 1 to 65535.

Point

The base folder name of a storage file container folder is an eight-digit (hexadecimal) number. This number matches the lowest of the serial numbers of the files contained in the directory. Date and time stamps can be appended to the folder name.

■Storage file name

The following describes the storage file name. The base file name is an eight-digit (hexadecimal) serial number.*1

*1 The same number is not used in the same file storage folder. If storage files have already existed when a new storage file is created by switching a stack file to a storage file, the number added one to the largest serial number among existing storage files becomes the name of the new storage file.

The following extra information can be added to the base file name. Up to 64 characters (including an extension and the period) can be a file name, combining any of the following.

Extra information		Details on extra information	Remarks
Simple setting	Storage folder name	Name of a folder where storage files are stored	An underscore (_) is added between each information.
	Date	Date information in YYYYMMDD format • YYYY: Year (four digits) • MM: Month (two digits) • DD: Day (two digits)	
	Time	Time information in hhmmss format • hh: Hour (two digits) • mm: Minute (two digits) • ss: Second (two digits)	
Optional setting	String	Any string*2	—
	Date	Date information added by specifying the following strings • YYYY: Year (four digits) • YY: Year (two digits) • MM: Month (two digits) • DD: Day (two digits)	
	Day of week	Day of the week information added by specifying the following strings • ddd: Day of week (three digits) (Sunday: Sun, Monday: Mon, Tuesday: Tue, Wednesday: Wed, Thursday: Thu, Friday: Fri, Saturday: Sat)	
	Time	Time information added by specifying the following strings • hh: Hour (two digits) • mm: Minute (two digits) • ss: Second (two digits)	
	Device value*3	The data value of the specified device can be added to the beginning of the file name by the number of digits specified (in the range of 1 to 5).	

*2 When using an above formatted string as it is, enclose a character string with double-quotation marks (" ") to add it.
Example: When adding the character string "address" to the file name, "address" → address_00000001.bin can be used.
However, when a character string that contains double quotation marks (" ") is specified, the maximum number reduces by the number of the double quotation marks.

*3 When adding a device value, check the firmware version of the CPU module and the version of the CPU Module Logging Configuration Tool. (☞ Page 1139 Added and Enhanced Functions)

• Date type

A date type can be selected from the following.

Add date type	Description
Date to establish file switching condition	Date and time information when the storage file switching condition is met is added.
File creation date	Date and time information when the file is created as a stack file (when the previous file switching is executed) is added.

Point

- Processing of file switching may take time depending on the setting. In this case, a date and time, which is closer to present than the timestamp of the first record in the data logging file, is added even though "File creation date" is selected for "Add date type".
- When "File creation date" is selected for "Add date type", a second information (two digits) added to a file name is always even number.
- After a data logging stops (completes) and restarts, the serial number (eight digits) of the base file name starts from 1 again in the following two cases: where the setting to delete data logging files after transfer is completed by using the data logging file transfer function or where the CPU built-in memory is specified as the file storage destination for transferring the files to the data memory (without using the data logging file transfer function). If the storage file name consists of the serial number only, the existing files in the transfer destination may be overwritten. It is recommended to add a date and time to the file name.

- Device value

The data value of the specified device can be added to the beginning of the file name by the number of digits specified (in the range of 1 to 5).

The timing to obtain the device value varies depending on the add date type. (☞ Page 222 Storage file name)

The date and time when the file switching condition is met is added when "Date to establish file switching condition" is selected, and the date and time when the stack file is created is added when "File creation date" is selected.

If the device value cannot be obtained upon file switching or stack file creation, no data is added to the storage file name.

The following table describes the types of the data that can be selected for adding to the file name.

Data type	Effective data	Remarks
Word (unsigned)	Fixed decimal format (0 to 65535)*1	—
String	Single-byte alphabetical characters*2	Characters in the file format that was specified in the storage format for data logging files.

*1 If the number of digits in the device data value exceeds the specified number of digits, only the specified number of digits are displayed from the small end. If the number of digits is insufficient, the remainder is supplemented with 0.
Example 1: When the device value is K123 and five digits are specified, "00123" is displayed.
Example 2: When the device value is K12345 and three digits are specified, "345" is displayed.

*2 If the number of characters in the device data value exceeds the specified number of characters, only the specified number of characters are displayed from the beginning. If the number of characters is insufficient, the remainder is not supplemented. If invalid data is included in the specified number of characters, valid characters are displayed.
Example: When the device value is "ABC" and five characters are specified, "ABC" is displayed.
Example: When the device value is "ABCDE" and three characters are specified, "ABC" is displayed.

The following describes the devices that can be specified for a device value.

Type	Device*3
Word device	T (current value), ST (current value), C (current value), D, SD, W, SW, RD, R, ZR, Z

*3 An index modified device, indirectly specified device, and local device cannot be specified.

When the maximum number of storage files to be saved is exceeded

Either "Overwrite" or "Stop"*1 can be selected as the action to take when the maximum number of storage files is exceeded.

*1 This settings is not configurable for trigger logging.

■When "Overwrite" is selected

When the storage file switching condition is met after the specified maximum number of storage files is exceeded, the CPU module deletes the file with the lowest serial number and creates a new file that has a serial number incremented by one from the highest serial number, allowing data logging to continue. In addition, if deleting the file with the lowest serial number results in an empty folder, the CPU module deletes that folder as well.

■When "Stop" is selected

As described in the following table, the action differs depending on when the specified maximum number of storage files is exceeded.

Occurrence timing	Occurrence condition	Behavior
When data logging is started	There exist more storage files than the specified maximum number when data logging is started.	• If an attempt is made to register the data logging settings by the start operation from CPU Module Logging Configuration Tool, an error occurs, resulting in failure to run data logging. • If an attempt is made to register*1 the data logging settings from outside CPU Module Logging Configuration Tool, a special relay area (data logging error) turns on and a special register area (data logging error cause) stores its error cause, resulting in failure to run data logging.
While data logging is running	The specified maximum number of storage files is reached due to file switching upon the satisfaction of the storage file switching condition.	Data logging stops and enters into the completion state with the data logging settings remaining intact. A special relay area (Data logging end) turns on to indicate that data logging is completed.

*1 When an attempt is made to register the data logging settings again, the CPU module enters into the data logging completed state. A special relay area (Data logging end) turns on to indicate that data logging is completed.

Point

The files deleted by "Delete files completed transfer" in the data logging file transfer setting are counted as the number of storage files. (Temporary stored files are counted as the number of storage files although they are not stored in the SD memory card or function memory.)
When "Overwrite" is selected, the file being transferred may be deleted with the data logging file transfer setting. It is recommended to set "Stop"

11.7 Missing Data

The term "missing data" means that some of the collected data is missing, resulting in data discontinuity.

Conditions under which missing data occurs

Missing data occurs under the following conditions:

Item	Description
Processing overflow	Processing overflow has occurred due to failure to keep up with the specified collection interval/timing.*1
Operations for the CPU module	The CPU module has been stopped and run with "Operation at transition to RUN" set to "Auto Start".
	The CPU module has been turned off and on with "Operation at transition to RUN" set to "Auto Start".
	The CPU module has been reset and run with "Operation at transition to RUN" set to "Auto Start".
Operation from engineering tools, CPU Module Logging Configuration Tool, and external devices via protocols such as FTP, SLMP, and MC	• When the CPU module is suspended and restarted, and operation for displaying the logging state is performed from CPU Module Logging Configuration Tool • File read*2, write, delete, or verification

*1 Failure to collect data at the specified collection interval/timing due to the execution of a long-running instruction (such as FMOV) does not cause processing overflow or missing data.

*2 The following operation also is included:
· Online operation which displays data by operation such as read from the programmable controller performed from an engineering tool (retrieval and display of a list of files on the CPU module)
· View of the event history (retrieval of the event history from the CPU module)

Processing overflow

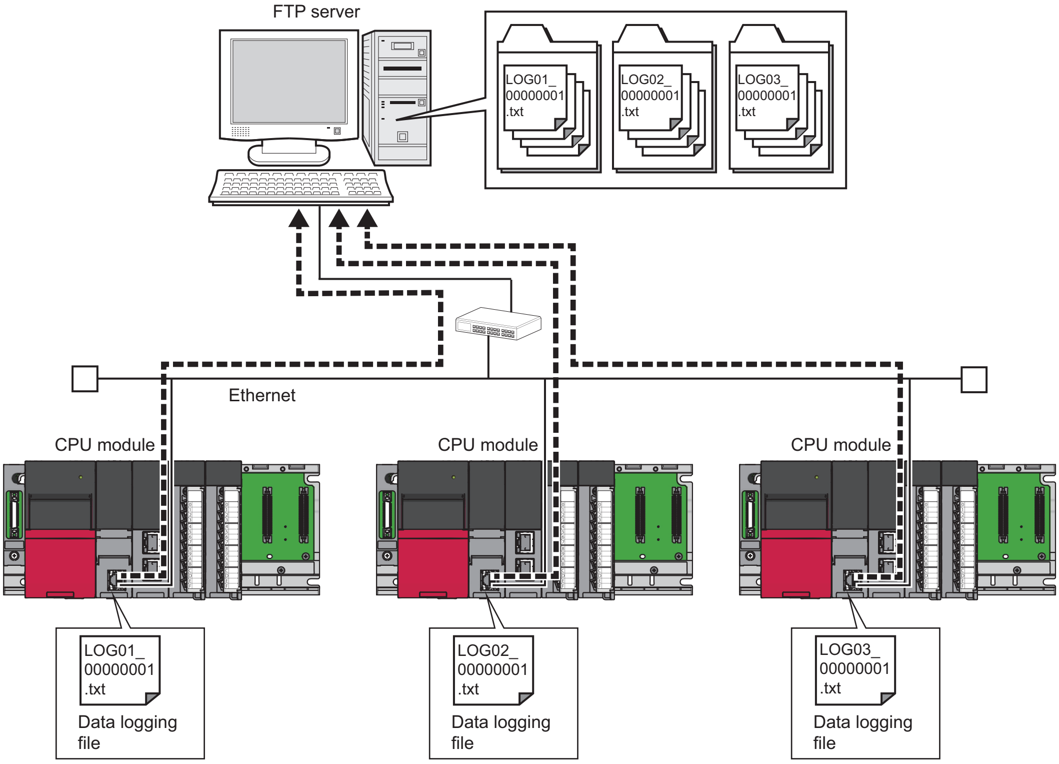
In normal cases when the usage of the internal buffer reaches the specified maximum capacity, the CPU module overwrites the data stored in the storage memory on a first-in first-out basis. If the internal buffer becomes full before all of the data stored in it is saved to the storage memory, however, the CPU module does not overwrite the existing data and stops storing data in the internal buffer, thus resulting in missing data. This situation is referred to as processing overflow. Upon the occurrence overflow, the special register (Number of processing overflow occurrences) stores the number of times when processing overflow occurred.

11.8 Data Logging File Transfer (Auto Transfer to FTP Server)

RnCPU RnENCPU ~~RnPCPU (Process)~~ ~~RnPCPU (Redundant)~~ ~~RnPSFCPU (Standard)~~ ~~RnPSFCPU (Safety)~~ ~~RnSFCPU (Standard)~~ ~~RnSFCPU (Safety)~~

- This function cannot be used in the R00CPU.

This function automatically transfers data logging files to the FTP server.
An SD memory card as the temporary storage destination is not required by setting the CPU built-in memory (function memory*1) as the data storage destination while using this function.



*1 For the CPU modules that can use the function memory, refer to the availability of storage location. (☞ Page 212 Availability)

Restriction

Before executing the function, check the versions of the CPU module and CPU Module Logging Configuration Tool used. (☞ Page 1139 Added and Enhanced Functions)

Point

An FTP server is required for the data logging file transfer function. For details on the server, refer to the manual for the server used.

The operation of this function is checked with the following FTP server.

Item	Supported operating system
FTP server whose operation is checked by Mitsubishi	Microsoft® Internet Infomation Services(IIS) The supported operating systems are as follows: • Microsoft® Windows® 10 • Microsoft® Windows® 8.1 • Microsoft® Windows® 8 • Microsoft® Windows® 7

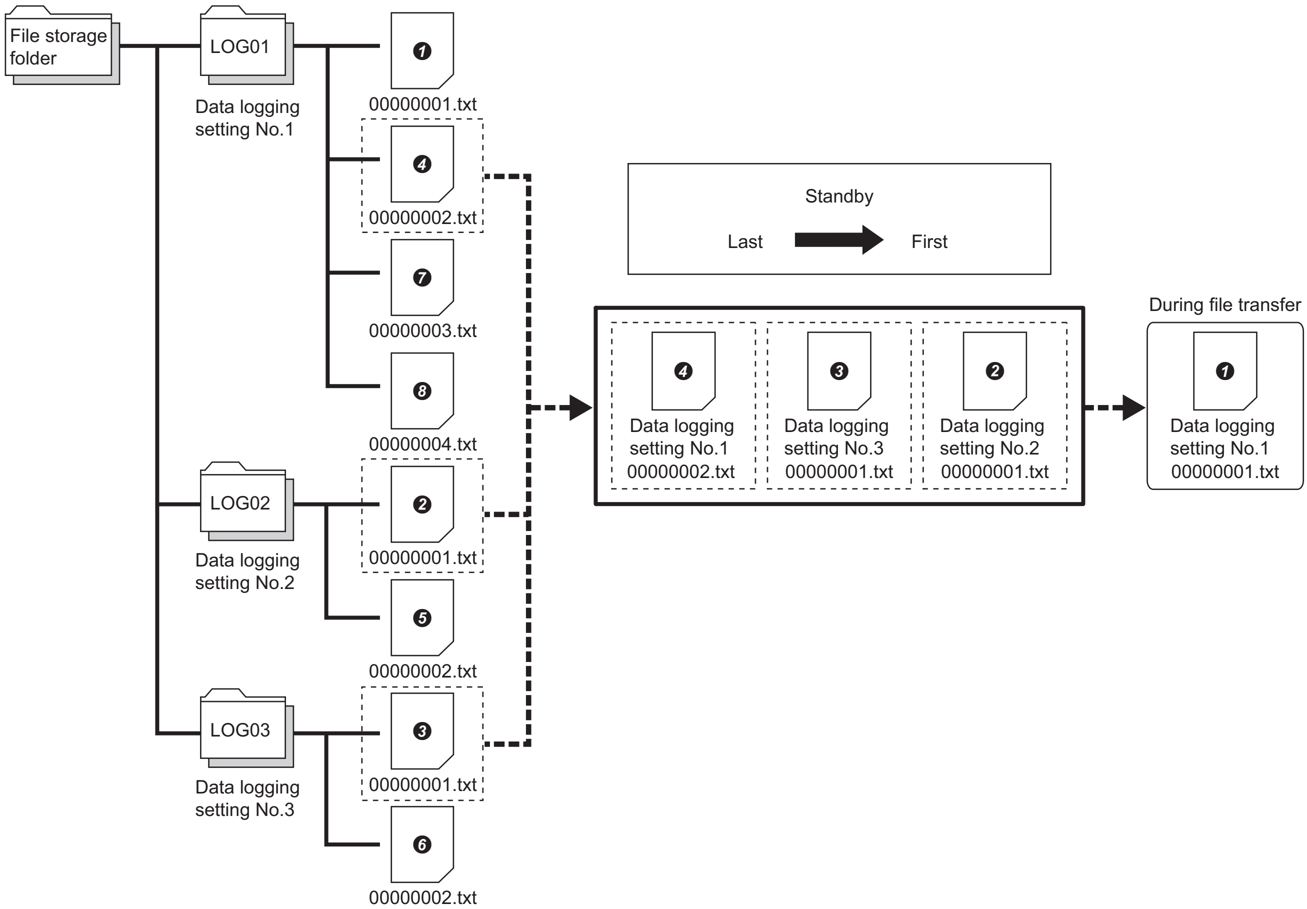
Transfer specifications and start timing of the data logging file

Data logging files are transferred at the file switching timing in the data logging function.

■Transfer specifications of the data logging file

- Data logging files are transferred one by one from each setting number (folder).
- If multiple files exist in multiple setting numbers, a file which has the smallest serial number in the lowest setting number is transferred.
- Only one file per setting number can be the standby file.

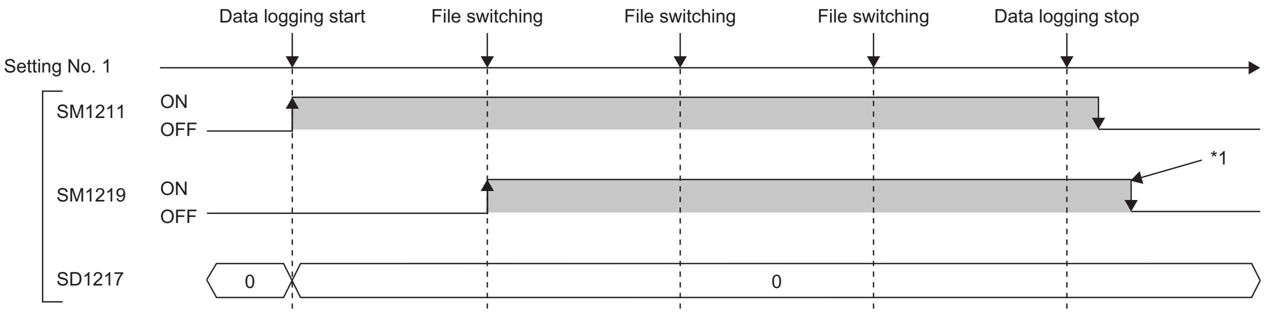
For example, in the following data structure, files are transferred from the file ❶. (Files are transferred in order from ❶ → ❷ →...→ ❼ → ❽.)



- If an error occurs during a file transfer, the data logging file caused the error is omitted from the standby files. A following file in the same setting number becomes a new standby file.
- If the file switching timing occurs during a file transfer, the file that has been being transferred becomes a standby file.
- If the data logging operation is completed or the user has operated CPU Module Logging Configuration Tool to stop data logging, the data logging file transfer is stopped at the completion of the transfer of the stored data logging files.

■Start timing of file transfer

After data logging is started, the transfer of the files created at the file switching timing in the data logging function starts. When the transfer is started, special relay areas SM1219 to SM1309 (Data logging file transfer execution status flag) for each setting number turn on. They turn off after all the files are transferred.



- SM1211: Data logging setting No.1 Data logging start
- SM1219: Data logging setting No.1 Data logging file transfer execution status flag
- SD1217: Data logging setting No.1 Data logging file transfer error cause

*1 SM1219 does not turn off until all the files are transferred.

Procedure for file transfer

■FTP server setting

Set the login name, password, and home directory to the FTP server. Authorize the user of the data logging file transfer function to read/write files. (Manual for the server used)

■Engineering tool setting

1. Set the IP address of the CPU module. Set the subnet mask and default gateway as necessary.

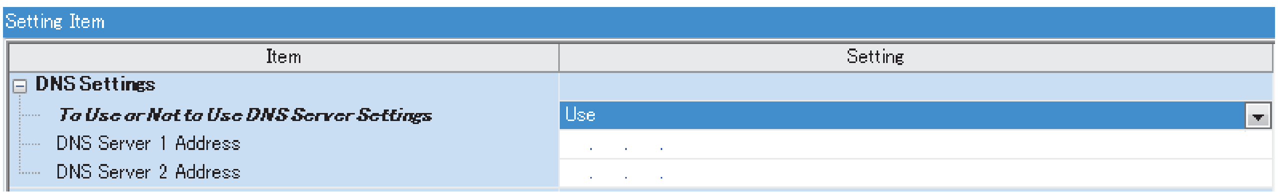
[Navigation window] ⇨ [Parameter] ⇨ CPU module ⇨ [Module Parameter] ⇨ [Basic Settings] ⇨ [Own Node Settings] ⇨ [IP Address]

2. To specify an FTP server with the server name, set "To Use or Not to Use DNS Server Settings" to "Use" by using the engineering tool.

[Navigation window] ⇨ [Parameter] ⇨ CPU module ⇨ [Module Parameter] ⇨ [Application Settings] ⇨ [DNS Settings]

3. Set an address of the DNS server.

Window

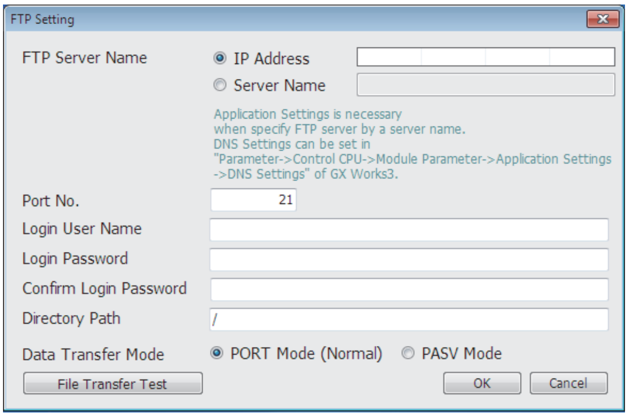


Displayed items

Setting item	Description	Setting range	Default
DNS server 1 IP address	Sets the IP address of the DNS server 1 in the decimal format.	0.0.0.1 to 223.255.255.254	Blank
DNS server 2 IP address	Sets the IP address of the DNS server 2 in the decimal format.	0.0.0.1 to 223.255.255.254	Blank

■Setting of CPU Module Logging Configuration Tool

1. Select the "Transferring files to the FTP server" check box in "File transfer" of CPU Module Logging Configuration Tool.
2. Click the [Server Setting] button.



3. Configure the server setting. To specify the FTP server with the server name, the DNS setting is required. (☞ Page 228 Engineering tool setting)
4. Click the [File Transfer Test] button to execute the file transfer test to the FTP server. Before operating the system, execute the file transfer test and ensure the connection with the FTP server. (☞ Page 230 File transfer test)
5. Set the timeout time and other items on the "File transfer" window. (☞ Page 231 Setting on the "File transfer" window)
6. Write the setting from CPU Module Logging Configuration Tool.
7. The transfer is started at the logging file switching timing. (☞ Page 228 Start timing of file transfer)

File transfer test

Check the communication status and settings by transferring a test file from the CPU module to the FTP server. The file transfer to the FTP server can be checked before system operation.

■Procedure for the file transfer test

The following describes the procedure for the file transfer test.

1. Configure the transfer destination server setting in CPU Module Logging Configuration Tool.
2. Click the [File Transfer Test] button in the "FTP Setting" window to execute the file transfer test.
3. Check the execution result.
4. Check that the test file is transferred to the FTP server. (☞ Page 230 Structure of a test file)

■Structure of a test file

The following table lists structures of a test file to be transferred to the FTP server.

Item	Description	Example
Test file name	MELSEC_CPU_FTP_TEST_**.txt ** indicates the data logging setting number (two digits, with zero-padding).	For setting No.1 MELSEC_CPU_FTP_TEST_01.txt
Contents of the test file	The IP address of the CPU module, test execution date, and test execution time are described. For the date and time, the clock data in the CPU module is used.	For IP address: 192.168.3.39, date: October 01, 2017, time: 11:22:33 192.168.3.39_20171001_112233

■File transfer test specification

- The FTP server connection request timeout time is fixed to 10 seconds.
- Even if the file transfer test fails due to a communication error, the retry is not executed.
- Even if the file transfer test fails, the error is not stored in the file transfer error history.

■Precautions

- File transfer tests cannot be executed simultaneously with another configuration tool. Execute the file transfer test after the file transfer test from another configuration tool is completed.
- If the engineering tool is operated or monitored from the same connection destination of the same computer during the file transfer test, the operation or monitoring will be executed after the file transfer test completion.

Setting on the "File transfer" window

■FTP server connection request timeout time

Set the waiting time from when the connection request from the CPU module to the FTP server is sent to when the response is received. If no response is received from the FTP server within the connection request timeout time, an error occurs.

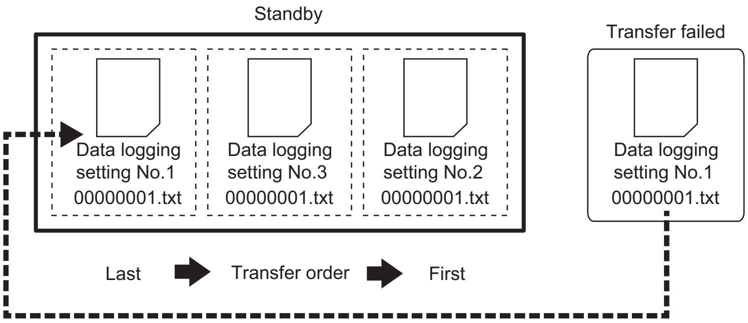
■File transfer retry time

Set the time to retry the file transfer when the file transfer fails due to an error caused by communication failure such as the network error between the CPU module and the FTP server.

The data logging file failed to be transferred is in the standby state and the retry starts. The file is resent until the file transfer retry time elapses. Even when the file transfer fails again due to a communication error, the file will be on standby for the reverse transfer again without an error of the data logging file transfer function.

Ex.

When a data logging file of the data logging setting No.1 has not been transferred due to a network failure



The retry ends when the network is recovered and retry of the file transfer succeeds.

■Matching the folder structure of the transfer destination FTP server with the CPU module

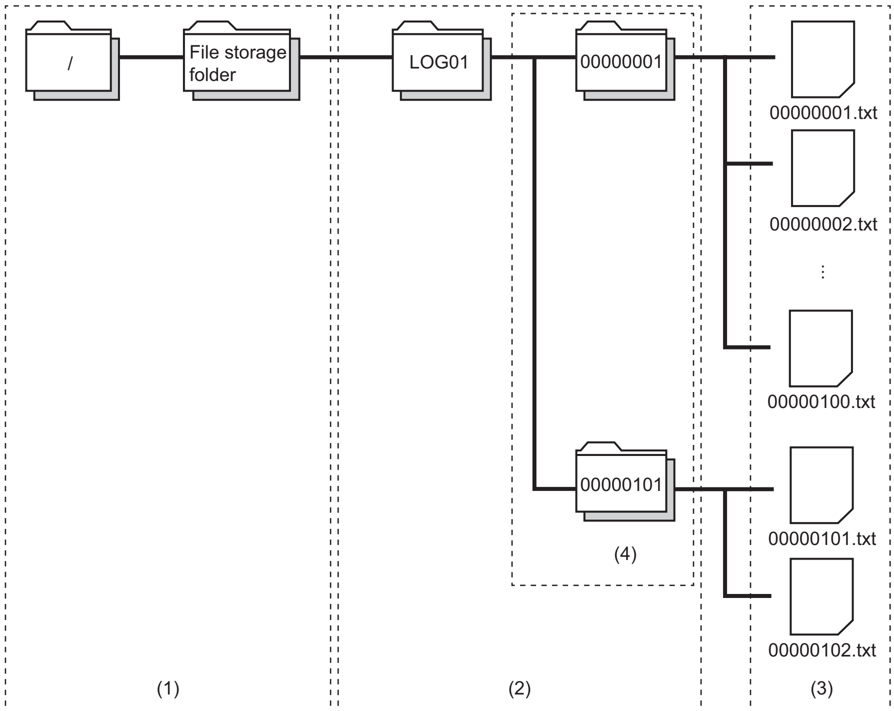
When the data logging file is transferred, the directory is automatically created in the specified folder path of the FTP server so that the structure is the same as the storage destination (☞ Page 1108 Folder configuration).

If the same file exists, the file will be overwritten.

- When the folder structure is matched with the CPU module

The folder path (1) specified in the server setting, folder structure (2), logging files transferred (3).

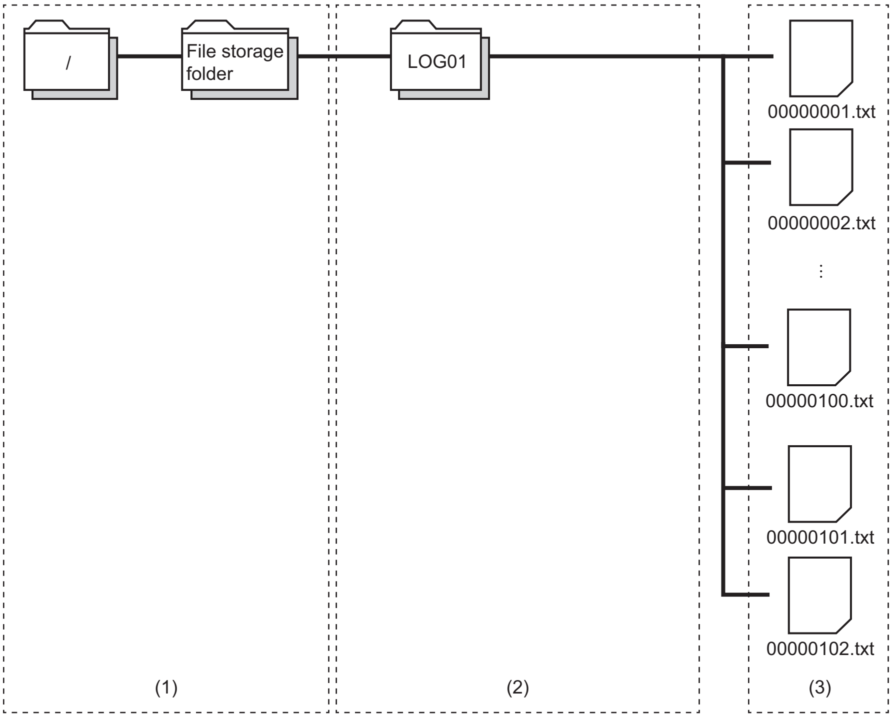
In the folder structure (2), subdirectories (4) are created.



• When the folder structure is not matched with the CPU module

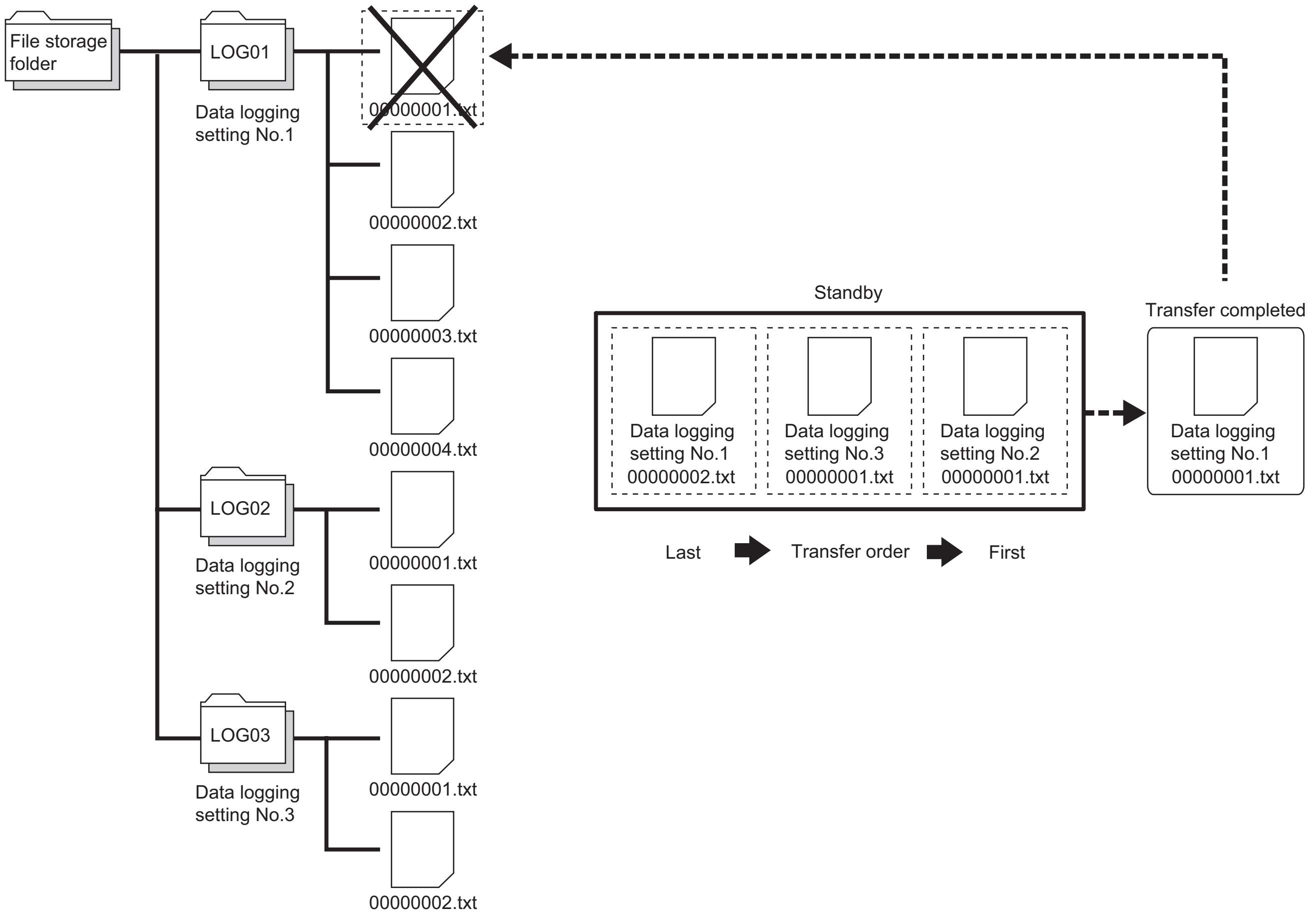
The folder structure of the FTP server consists of the storage destination structures of the data logging file excluding the subdirectory.

The folder path (1) specified in the server setting, folder structure (2), logging files transferred (3).



■Deleting files completed transfer

At the completion of data logging file transfer, transferred files are automatically deleted. If there is no file in the saved file storage folder (subdirectory) as a result of the file deletion, the saved file storage folder (subdirectory) is deleted while data logging is stopped.



Point

- If the function memory is specified as the data storage destination, files are deleted after the transfer regardless of the setting.
- Even if "Delete files completed transfer" is specified, data logging stops when the number of files stored by the data logging function exceeds the maximum value of the number of files to be saved. To consecutively execute data logging, set a large number of files to be saved.

Data logging file transfer status

The file transfer status, the data logging name, and the IP address of the file transfer destination FTP server can be checked. They can be checked on the data logging file transfer status window of CPU Module Logging Configuration Tool.

File transfer error log

Error history including the date and time of error occurrence, data logging No., and the error codes can be checked. They can be checked on the file transfer error log window of CPU Module Logging Configuration Tool.

Stopping the data logging file transfer

After data logging is stopped, the data logging file transfer stops when no standby file for the data logging file transfer is left. However, after data logging is stopped, if the data logging file transfer is being retried and does not stop, the file transfer in progress can be stopped by following the procedure below.

1. Set the data logging setting number for SD1203 (Data logging file transfer stop information). Multiple data logging setting numbers can be set for SD1203.
2. Turn on SM1203 (Data logging file transfer stop request). When multiple data logging setting numbers are specified for SD1203 and SM1203 is turned off and on, SM1203 turns off after the data logging file transfer of all the data logging setting numbers stops.
3. At the completion of the file transfer stop processing, an error is stored in each data logging file transfer error cause of the files being transferred and in the standby (retry) state.

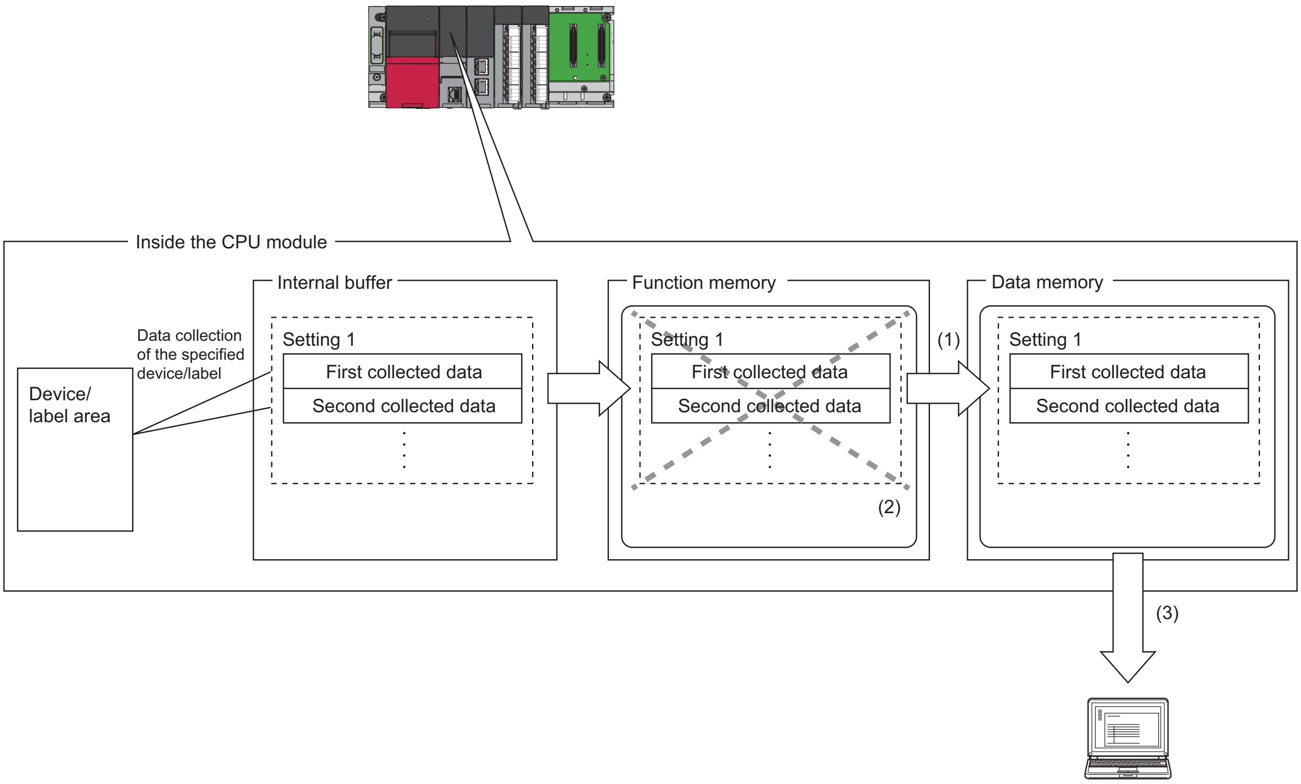
For the file transfer stopped by the stop request, files which have not been transferred cannot be transferred even if the data logging of the same setting number is restarted and the file transfer is executed. To transfer files which have not been transferred, the procedure differs depending on the data storage destination. When the data storage destination is the SD memory card, transfer the files to the server by using the FTP server function. When the data storage destination is the function memory, transfer the files to the data memory by using SM653 (File transfer to data memory request) and transfer the files to the server by using the FTP server function.

Point

- When the data logging status of the data logging setting No. which has been set by SD1203 (Data logging file transfer stop information) is other than stop, or the data logging file transfer is not enabled in the data logging setting, turning off and on SM1203 (Data logging file transfer stop request) does not execute the stop processing and SM1203 turns off.
- When the data logging is in the collection completed state or when an error occurs, file transfer cannot be stopped by SM1203 (Data logging file transfer stop request). To stop file transfer, stop the data logging function.

11.9 Data Logging File Transfer to Data Memory

When the function memory*1 is specified for the data storage destination memory without setting of the file transfer, the data is transferred from the function memory to the data memory at the logging completion or stop (including when an error occurs) as shown below.



(1) At the logging completion or stop (including when an error occurs), the data is transferred from the function memory to the data memory. When data logging is suspended or waiting for RUN without collection, the data is not transferred to the data memory.
(2) Data in the function memory is automatically deleted after completion of the transfer.
(3) The data logging files transferred to the data memory can be read by engineering tool or displayed by GX LogViewer.

*1 For the CPU modules that can use the function memory, refer to the availability of storage location. (☞ Page 212 Availability)

Behavior when the free space in the data memory is full

When multiple data logging files exist in the function memory and the free space in the data memory is full by only transferring some of the files, an error occurs and the files remaining are not transferred. In this case, files that have not been transferred are not deleted from the function memory.

Starting data logging during the data transfer to the data memory

The data logging cannot be started during the data transfer to the data memory. An error occurs when starting the data logging. (☞ Page 253 Starting data logging during the data transfer to the data memory)

An error during the data transfer to the data memory is stored in SD1217 (Data logging file transfer error cause). (☞ Page 998 Data logging function)

11.10 Setting Behavior at the Time of Transition to RUN

After the data logging settings are registered by the start operation of data logging, set the behavior of data logging when the following user operations to switch the operating status of the CPU module to RUN are performed (transition to RUN).

- Powering off and on and switching the operating status to RUN
- Resetting and switching the operating status to RUN
- Switching the operating status from STOP to RUN

Point

The logging operation can be set individually for each setting number (1 to 10).

Behavior at the time of a transition to RUN

The behavior can be selected from the following.

■Auto start

The data logging automatically starts after the user operation of starting data logging in CPU Module Logging Configuration Tool and switching the operating status of the CPU module to RUN.

[Online] ⇨ [Logging Status and Operation]

■Start by user operation

The data logging status becomes "Start waiting (no collection)" after the user operation of starting data logging in CPU Module Logging Configuration Tool and switching the operating status of the CPU module to RUN. To start data logging, operate CPU Module Logging Configuration Tool to start data logging again.

[Online] ⇨ [Logging Status and Operation]

Point

When the auto logging is used, the behavior of the data logging is always "Auto Start", even if the behavior at transition to RUN is set to "Start by User Operation". (☞ Page 236 Auto Logging)

Data logging behavior that occurs after operating status of CPU module has changed

Data logging does not continue when the operating state of the CPU module changes from RUN to STOP or PAUSE after it has been started. The data logging status changes to "RUN waiting (no collection)" and data collection is stopped.

11.11 Auto Logging

When inserting an SD memory card, which holds data logging setting, into the CPU module, the data logging automatically starts based on the data logging setting information on the SD memory card.

How to use auto logging

This section describes how to use auto logging.

1. Prepare an SD memory card that contains data logging settings as well as common settings (auto logging common settings) in which auto logging is enabled.
2. Insert the SD memory card prepared in step 1 into the CPU module while it is running.
3. When the SD memory card is inserted, data logging starts automatically. (CARD READY LED and CARD ACCESS LED turn on.)
4. Check that auto logging is completed on the engineering tool or using the LED*1 on the CPU module.
5. Remove the SD memory card.

*1 When "Data Logging Function" is set in the LED indicator setting, the FUNCTION LED flashes every 200ms. If the executed function has higher priority of the FUNCTION LED than the function set in the LED indicator setting, the data logging function status is not displayed. (☞ Page 143 LED display setting)

Point

- Even when the data logging stop operation is performed after auto logging starts, auto logging does not end until when the SD memory card is removed.
- To use auto logging, the auto logging common setting file and data logging setting file for operation are required in the SD memory card.
- Write the auto logging common setting file to the SD memory card only when using auto logging. When auto logging is not used, delete the auto logging common setting file.

Auto logging start conditions

Auto logging starts in the following cases.

■Inserting the SD memory card with the auto logging setting while the CPU module is running

Auto logging starts when the SD memory card with the auto logging setting is inserted while the CPU module is running.
(When the CPU module is in the STOP state, auto logging starts by changing the state from STOP to RUN.)
If data logging is in progress before the SD memory card with the auto logging setting is inserted, auto logging does not start.

■Inserting the SD memory card with the auto logging setting while the CPU module is off or being reset, and powering on or resetting the CPU module

Auto logging starts when the CPU module is powered on or reset after the SD memory card with the auto logging setting is inserted while it is off or being reset. If data logging is in progress, powering on or resetting the CPU module executes auto logging instead of the data logging.

Conditions for auto logging completion

Auto logging completes when a completion condition is met as described in the following table. These completion conditions can also be configured in combination with each other. When they are configured in combination, auto logging completes as soon as one of the conditions is met.

Completion condition	Description
Data logging stop	Choose one of the following: • When all data loggings stop • When any of the data loggings stops
Complete with timer	Auto logging is stopped when the specified time has elapsed after the start of data logging. Elapsed time setting range: 1 second to 86400 seconds (unit: second)

■When "Data logging stop" is selected

Auto logging is assumed to be completed if all the settings or any of auto logging stops*1.

*1 Here the term "stop" means one of the following:
· Continuous logging: When the "Number of files to be saved" setting configured in the save setting has been exceeded and data logging has been completed.
· Trigger logging: When as much data as the number of records specified in "Number of records" has been collected, the collected data has been written to the SD memory card, and data logging has been completed.
· When the user has operated CPU Module Logging Configuration Tool to stop data logging.

Point

- If "When any of the data loggings stops" is selected, the behavior of any other data logging sessions than stopped is the same as the behavior that occurs upon the elapse of the time configured using "Complete with timer".
- When completing auto logging for continuous logging, do not select "Overwrite" for the operation at the time when the number of files exceeds the save setting because doing so results in failure to stop data logging; instead, select "Stop".

■When "Complete with timer" is selected

When the operating time since the start of data logging reaches the specified time, the CPU module completes auto logging by moving all the data collected so far from the internal buffer to the SD memory card. If trigger logging has not yet collected as much data as the number of records specified "Number of logging lines", however, the CPU module does not store any data including the collected data.

Point

The timer is cleared to zero when auto logging is suspended and resumed by stopping and running the CPU module or turning off and on the CPU module or resetting it before the completion of auto logging. If auto logging is suspended by the turning on of the special relay (Data logging suspend/resume flag), the timer continues to run even while data logging is suspended.

Conditions under which auto logging does not start

Doing any of the following operations once auto logging is completed does not start auto logging:

- Turn off and on the power
- Reset
- STOP to RUN state

If data logging is in progress before the SD memory card with the auto logging setting is inserted, auto logging does not start as well.

Behavior of auto logging at the time of a transition to RUN

If some, not all, of data logging sessions configured using auto logging have been completed when entering into the RUN mode, only the uncompleted data logging sessions are started.

Behavior of auto logging that occurs when an error is generated

When an error is generated, auto logging behaves as follows:

■When an error is generated at the start of auto logging (Registration of the data logging settings has failed.)

If auto logging cannot be started (registration has failed), it behaves in the same way as when it is completed.

■When an error has occurred during the execution of auto logging

An error that occurs during the execution of auto logging does not constitute the auto logging completion condition since it does not prevent data logging from being resumed. If such an error is resulting from an online change during the execution of auto logging, however, it constitutes the auto logging completion condition because it prevents data logging from being resumed.* 1

*1 Applies only when "When all data loggings stop" is selected as part of the auto logging completion condition.

11.12 SD Memory Card Replacement

SD memory cards can be replaced using the SD memory card forced disable function even while data logging is in progress.
MELSEC iQ-R CPU Module User's Manual (Startup)
Only the data saving to an SD memory card is stopped while this function is being executed. The data collection keeps working. (Data collection continues in accordance with the settings registered when data logging is started.)
With the setting for the file transfer, an error occurs when the SD memory card is replaced during the file transfer. (Page 873 Codes of errors detected by other than the self-diagnostic function (4000H to 4FFFH))

Point

If SD memory card replacement causes processing overflow, make adjustments by changing the collection interval, internal buffer capacity, or other settings.

Behavior during SD memory card replacement

If the internal buffer becomes full during the time between SD memory card replacement and the resumption of data writes to the SD memory card, processing overflow occurs resulting in missing data.

Storage file numbers after SD memory card replacement

The numbering of the first storage file created after SD memory card replacement differs depending on the storage file switching condition, as described in the following table.

Storage file switching condition	Storage file numbers after SD memory card replacement
Overwrite	Numbering continues from the number of the last storage file contained in the replaced SD memory card.
Stop	Numbering begins at 00000001.*1

*1 While the data logging file transfer function is used, numbering begins at 00000001 only when the "Delete files completed transfer" is disabled in the "File transfer" setting of CPU Module Logging Configuration Tool.

Point

If the new SD memory card contains a "LOGGING" folder and its subfolders, folder deletion takes time, possibly resulting in missing data. Ensure that the new SD memory card does not contain a "LOGGING" folder.

Logging state during SD memory card replacement

SD memory cards can be replaced without depending on the current data logging state. SD memory card replacement results in the deletion of the LOGGING folder if the data logging state is other than below:

- Stop
- Stop (after collection)
- Stop (after error)

Operations during SD memory card replacement

If one of the following operations is performed during the time between the removal and installation of SD memory cards, any data collected during that time will not be stored in the new SD memory card.

- STOP to RUN state*1
- Turn off and on the power*1
- Reset*1
- Suspend data logging
- Stop data logging

*1 An error is generated if data logging was previously running based on the setting file contained in the replaced SD memory card.

Operations after SD memory card replacement

If the SD memory card was replaced while data logging was running based on the data logging setting file contained in the SD memory card, the data logging setting file contained in the new SD memory card is used when data logging is started by one of the following operations. If the new SD memory card does not contain the data logging setting file, data logging is not started.

- STOP to RUN state
- Power-on to RUN state
- RESET to RUN state
- Data logging start

11

Stack file remaining in the replaced SD memory card

Replacing an SD memory card that contains a stack file may result in the storage file remaining in the replaced SD memory card along with storage files. If the stack file is remaining in the replaced SD memory card, recover the latest data contained in the stack file by doing the following:

- Retrieve the data from the stack file and combine the data with a storage file.
- Save the stack file as a storage file.

11.13 SD Memory Card Life When the Data Logging Function Is Used

An SD memory card has a life (restriction on writing data). The following shows the calculation method of an SD memory card life when the data logging function is used. Note that the actual life of the card varies depending on the use conditions and environment. Therefore, use the calculated life as a rough standard for the replacement of the card.

Calculation formula of SD memory card life

SD memory card life (year) = Total size of data that can be written (G bytes) ÷ Size of data to be written per year (G bytes/year)

Total size of data that can be written

Capacity × Number of writes*1

*1 For the capacity of applicable SD memory cards and the number of writes, refer to the following.
MELSEC iQ-R Module Configuration Manual

Size of data to be written per year

The size of data to be written per year is obtained by the following formula.

Size of data to be written per year (G bytes/year) = ((DS1*1 + 6144) × DN1 + ··· + (DSn*1 + 6144) × DNn + (DCS1*1 + 6144) × DCN1 + ··· + (DCSn*1 + 6144) × DCNn) ÷ 1073741824

*1 Round up DSn and DCSn to a multiple of 512.

DSn, DNn, DCSn, and DCNn are obtained as follows.

■Data logging data size per record (DSn)

Binary file output format: Refer to the data. (Page 1121 Binary file output format)

Unicode text file format: Refer to the data row. (Page 1109 Unicode text file output type)

■Number of records for data logging per year (DNn)

Continuous logging: DNn = 60 × 60 × 24 × 365 ÷ Collection interval and timing (seconds)*1 × Operating rate*2

Trigger logging: DNn = Total number of records*3

*1 The value that is determined depending on the condition set in "Sampling" when "Continuous logging" is selected for the logging type. (When the value is determined in milliseconds, convert the value into seconds.)

*2 Calculate the ratio using the operating time per year of the CPU module. For example, if the operating time per year is 5000 hours, the operating rate is calculated as follows: 5000 ÷ (24 × 365) = 0.57.

*3 The value set in "Number of logging lines" when "Trigger logging" is selected for the logging type.

■Header size of data logging (DCSn)

Binary file output format: Refer to the header. (☞ Page 1121 Binary file output format)

Unicode text file format: Refer to the file information row to device comment row. (☞ Page 1109 Unicode text file output type)

■Number of file switching times for the data logging per year (DCNn)

Calculate this number with an estimated number according to the save setting of the data logging and system operations. For example, when 1000 records are set in "Number of records" of "File switching timing" in the save setting and "Each scanning cycle" is specified for "Sampling interval" in the sampling setting, the time interval of the file switching is obtained by multiplying the scan time by 1000. Therefore, the number of file switching times for the data logging per year is obtained by the following formula: 60 × 60 × 24 × 365 ÷ (Scan time (second) × 1000)

11.14 Errors Generated During Data Logging

No diagnostic error occurs if an error occurs during data logging, the SM applicable to the special relay (data logging error) setting No. turns on, and the error cause is stored in the SD applicable to the special register (data logging error cause) setting No.

11.15 Special Relay and Special Register Used by the Data Logging Function

For details on the special relay and special register areas used by the data logging function, refer to the following:

- Special relay: Special relay areas relating to the data logging function (☞ Page 949 Data logging function)
- Special register: Special register areas relating to the data logging function (☞ Page 998 Data logging function)

11.16 Precautions to Take When Using the Data Logging Function

This section describes precautions to take when using the data logging function.

Mutual exclusion of the data logging function

This section describes the mutual exclusion of the data logging function.

■When another function is executed during the execution of the data logging function

The following table lists the cases where another function is executed during the execution of the data logging function.[*1]

Function that has been already executed	Function to be executed later	Behavior
Data logging function	Data logging function	When the data logging is started using CPU Module Logging Configuration Tool to the same data logging setting number via another route, the data logging to be executed later cannot be executed. However, the data logging to be executed later can be executed to a data logging setting number different from the data logging setting number currently being executed.
		When specifying multiple trigger conditions, both data condition and label change specification cannot be specified as the trigger conditions.
		For the execution of multiple data loggings, the data logging settings stored in different target memory areas cannot be executed at the same time.
	Auto logging	The auto logging cannot be executed during the execution of the data logging. (Even though an SD memory card where the auto logging setting is written is inserted, the auto logging does not start.)
	Online change (ladder block)	Although the online change (ladder block) is completed, the data logging may stop depending on the settings for the data logging function and the files to be changed. (☞ Page 243 Operation when online change is executed while data logging is in progress)
	File batch online change	Although the file batch online change is completed, the data logging may stop depending on the settings for the data logging function and the files to be changed. (☞ Page 243 Operation when online change is executed while data logging is in progress)
	Writing data to the programmable controller	Although the data is properly written to the programmable controller, the operation when the operating status of the CPU module changes from STOP to RUN differs depending on the target file for the change. (☞ Page 244 Operation when data logging function and another function are executed together and also the status is changed from STOP to RUN)
	File transfer (FTP server) function	Although the file transfer (FTP server) function is completed, the operation when the operating status of the CPU module changes from STOP to RUN differs depending on the target file for the change. (☞ Page 244 Operation when data logging function and another function are executed together and also the status is changed from STOP to RUN)
	SLMP	Although SLMP is completed, the operation when the operating status of the CPU module changes from STOP to RUN differs depending on the target file for the change. (☞ Page 244 Operation when data logging function and another function are executed together and also the status is changed from STOP to RUN)
	CPU module data backup function	The CPU module data backup function cannot be executed while a logging setting file is being written/deleted or a logging setting is being registered/cleared.
	CPU module data restoration function	The CPU module data restoration function cannot be executed while a logging setting file is being written/read/deleted or a logging setting is being registered/cleared.
	iQ Sensor Solution data backup/restoration function	The iQ Sensor Solution data backup/restoration function cannot be executed while a logging setting file is being written/deleted or a logging setting is being registered/cleared.

Function that has been already executed	Function to be executed later	Behavior
Data logging function	Function specified in the internal buffer capacity setting*2	If the internal buffer capacity setting is changed to execute the subsequent function, attempting to start the subsequent function results in an error. The data logging continues to function normally.
	Function not specified in the internal buffer capacity setting	If the condition "Total capacity that is set in the internal buffer capacity setting + Internal buffer capacity that is set in other than the internal buffer capacity setting > 3072K bytes" is satisfied, attempting to start the subsequent function results in an error. The data logging continues to function normally.
		If the internal buffer capacity setting is changed to execute the subsequent function, attempting to start the subsequent function results in an error. The data logging continues to function normally.
Auto logging	Data logging function	• Another data logging cannot be executed during the execution of the auto logging. • Even if data logging is started by using CPU Module Logging Configuration Tool, another data logging cannot be executed until the SD memory card with the auto logging setting is removed.
Data logging function (when the storage location is the function memory)	File batch online change	The file batch online change cannot be executed during data logging (when the storage location is the function memory).
Data logging file transfer (when the storage location is the function memory)	File batch online change	The file batch online change cannot be executed during data logging file transfer (when the storage location is the function memory).
Data logging function (Transfer to the data memory)	CPU module data backup function	The CPU module data backup function cannot be executed during data logging file transfer to the data memory.
	CPU module data restoration function	The CPU module data restoration function cannot be executed during the data logging file transfer to the data memory.

*1 The data logging function is being executed in the following states where the data logging status remains intact or when the save status is "Saving".
·RUN waiting (no collection)
·Condition waiting (no collection)
·Start waiting (no collection)
·Pause
·Collecting
·Trigger waiting (Collecting before trigger)
·Collecting after trigger
In the states other than the above, although the functions can be executed, the registration of the data logging setting executed first is canceled when the data logging function or another function is executed.

*2 The data logging function is not included here.

■Operation when online change is executed while data logging is in progress

If online change (ladder block or file batch online change) is executed while the data logging function is in progress, the operation differs depending on the setting details for the data logging settings.

- When SD940 (Stop direction at file change on label specification) is off

Function to be executed later		Operation based on the setting details of data logging in progress*1				
Function	**Change target file**	**Global device**	**Local device**	**Global device**	**Local device**	**Step No.**
• Online change (ladder block) • File batch online change	Program files, FB files	Continue	Continue	Continue	Continue	Stop*2
	Global label setting file	Continue	Continue	Continue	Continue	Continue

*1 This is applicable for when the device and label are specified as "target data", "collection conditions", or "trigger conditions" or when the step No. is specified as "collection conditions" or "trigger conditions".

*2 During the execution of a "function to be executed later", if the change involves only a local label name change or the addition of a local label definition, an error may not occur, allowing the operation to continue.

- When SD940 (Stop direction at file change on label specification) is on

Function to be executed later		Operation based on the setting details of data logging in progress*3				
Function	**Change target file**	**Global device**	**Local device**	**Global device**	**Local device**	**Step No.**
• Online change (ladder block) • File batch online change	Program file	Continue	Continue	Continue	Stop*4*5	Stop*4
	FB file	Continue	Continue	Continue	Continue	Stop*4
	Global label setting file	Continue	Continue	Stop*4	Continue	Continue

*3 This is applicable for when the device and label are specified as "target data", "collection conditions", or "trigger conditions" or when the step No. is specified as "collection conditions" or "trigger conditions".

*4 During the execution of a "function to be executed later", if the change involves only a local label name change or the addition of a local label definition, an error may not occur, allowing the operation to continue.

*5 An error occurs if a program file which includes the specified local label is written.

■Operation when data logging function and another function are executed together and also the status is changed from STOP to RUN

The following table shows the operations where the CPU module operating status is changed from STOP to RUN after another function is executed during data logging.

- When SD940 (Stop direction at file change on label specification) is off

Function to be executed later		Operation based on the setting details of data logging in progress*1				
Function	**Change target file**	**Global device**	**Local device**	**Global device**	**Local device**	**Step No.**
• Writing data to the programmable controller • File transfer (FTP server) function • SLMP	CPU parameter file	Continue	Stop*3	Continue	Stop*3	Continue
	Program file	Continue	Continue	Continue	Continue	Continue*2
	Global label setting file	Continue	Continue	Continue	Continue	Continue

*1 This is applicable for when the device and label are specified as "target data", "collection conditions", or "trigger conditions" or when the step No. is specified as "collection conditions" or "trigger conditions".

*2 The operation stops if the step No. does not exist when the CPU module operating status is changed from STOP to RUN.

*3 The operation stops when the CPU module operating status changes from STOP to RUN.

- When SD940 (Stop direction at file change on label specification) is on

Function to be executed later		Operation based on the setting details of data logging in progress*4				
Function	**Change target file**	**Global device**	**Local device**	**Global device**	**Local device**	**Step No.**
• Writing data to the programmable controller • File transfer (FTP server) function • SLMP	CPU parameter file	Continue	Stop*6	Continue	Stop*6	Continue
	Program file	Continue	Continue	Continue	Stop*6	Continue*5
	Global label setting file	Continue	Continue	Stop*6	Continue	Continue

*4 This is applicable for when the device and label are specified as "target data", "collection conditions", or "trigger conditions" or when the step No. is specified as "collection conditions" or "trigger conditions".

*5 The operation stops if the step No. does not exist when the CPU module operating status is changed from STOP to RUN.

*6 The operation stops when the CPU module operating status changes from STOP to RUN.

■When the data logging function is executed during the execution of another function

The following table lists the cases when the data logging function is executed during the execution of another function.

Function that has been already executed	Function to be executed later	Behavior
Online change (ladder block)	Data logging function	Although the online change (ladder block) is completed, the data logging may stop depending on the settings for the data logging function and the files to be changed. (☞ Page 245 Operation when data logging is executed while online change is in progress)
File batch online change		Although the file batch online change is completed, the data logging may stop depending on the settings for the data logging function and the files to be changed. (☞ Page 245 Operation when data logging is executed while online change is in progress)
CPU module data backup function		While the CPU module data backup function is being executed, a logging setting file cannot be written/deleted or a logging setting cannot be registered/cleared.
CPU module data restoration function		While the CPU module data restoration function is being executed, a logging setting file cannot be written/read/deleted or a logging setting cannot be registered/cleared.
iQ Sensor Solution data backup/restoration function		While the iQ Sensor Solution data backup/restoration function is being executed, a logging setting file cannot be written/deleted or a logging setting cannot be registered/cleared.
Function specified in the internal buffer capacity setting*1		If the internal buffer capacity setting is changed to execute data logging, attempting to start data logging results in an error. The function already in execution continues to function normally.
Function not specified in the internal buffer capacity setting		If the internal buffer capacity setting is changed to execute data logging, attempting to start data logging results in an error. The function already in execution continues to function normally.
CPU module data backup function*2	Data logging function (Transfer to the data memory)*3	While the CPU module data backup function is being executed, a data logging file cannot be transferred to the data memory.
CPU module data restoration function*2		While the CPU module data restoration function is being executed, a data logging file cannot be transferred to the data memory.

*1 The data logging function is not included here.

*2 The CPU module data backup/restoration function executed during data logging

*3 It is executed when the trigger logging data collection is completed or data collection for the specified number of storage files is completed.

■Operation when data logging is executed while online change is in progress

The following table shows the operation where data logging is started while online change (ladder block or file batch online change) is in progress. The operation differs depending on the data logging settings.

Function that has been already executed		Operation based on the setting details of data logging in progress*1				
Function	**Change target file**	**Global device**	**Local device**	**Global device**	**Local device**	**Step No.**
• Online change (ladder block) • File batch online change	Program files, FB files	Can be executed	Can be executed	Cannot be executed	Cannot be executed	Cannot be executed
	Global label setting file	Can be executed	Can be executed	Cannot be executed	Cannot be executed	Cannot be executed

*1 This is applicable for when the device and label are specified as "target data", "collection conditions", or "trigger conditions" or when the step No. is specified as "collection conditions" or "trigger conditions".

■When a file operation related to the data logging is performed during the execution of the data logging function

The following table lists the cases when a file operation related to the data logging is performed during the execution of the data logging function.

Target file	File operation	Behavior
Data logging setting file/common setting file	Write/delete	During execution of the data logging function, data cannot be written/deleted to/in the data logging setting file/common setting file being used.
	Folder delete	Folders cannot be deleted from the $MELPRJ$ folder in which the data logging setting files and the common setting file are stored.
	Initialize	During execution of the data logging function, the memory storing the data logging setting files and the common setting file being used cannot be initialized.
Data logging file	Write/delete/folder delete	During execution of the data logging function, data cannot be written/deleted to/in or folders cannot be deleted from the data logging file being used.
	Initialize	During execution of the data logging function, the memory storing the data logging file being used cannot be initialized.

■When a file operation related to the data logging is executed during the execution of online change

If a writing operation of the data logging setting file is executed while online change (ladder block or file batch online change) is in progress, the operation differs depending on the setting details for the data logging settings. The following table shows the operation where data logging is started while online change (ladder block) or file batch online change is in progress.

Function that has been already executed		Operation based on the setting details of data logging in progress*1				
Function	Change target file	Global device	Local device	Global device	Local device	Step No.
• Online change (ladder block) • File batch online change	Program files, FB files	Can be executed	Can be executed	Cannot be executed	Cannot be executed	Can be executed
	Global label setting file	Can be executed	Can be executed	Cannot be executed	Cannot be executed	Can be executed

*1 This is applicable for when the device and label are specified as "target data", "collection conditions", or "trigger conditions" or when the step No. is specified as "collection conditions" or "trigger conditions".

Locations from which data logging can be performed

Data logging cannot be performed from multiple locations to the same setting number. The CPU module supports data logging performed concurrently at a maximum of 10 locations assigned to setting numbers 1 to 10.

Retention and clearance of data logging settings

After the data logging is started, the registered data logging settings are latched. Thus, if the data logging is started (registered) before powering off and on or resetting the CPU module, register the settings again when performing the following operations to the CPU module; Powering off and on and switching the operating status to RUN; Resetting and switching the operating status to RUN; Switching the operating status from STOP to RUN.

As a result, data logging can be executed again with the registered data logging setting.

However, the registration of the data logging settings is canceled in the following cases. (The corresponding SM number to the data logging setting number among relevant special relay areas (data logging preparation) turns off.)

- The CPU module is turned off and on or is reset without an SD memory card that contains the data logging setting file.
- The replaced SD memory card does not contain the data logging setting file and the CPU module is turned off and on or is reset.*1

*1 If the data logging setting file contained in the replacement (new) SD memory card is different from that contained in the replaced (old) SD memory card, register the data logging setting file contained in the replacement SD memory card.

It is necessary to register the data logging settings again by the user operation of starting data logging in CPU Module Logging Configuration Tool.

Behavior that occurs when trigger logging is resumed

If data logging is stopped or collection is suspended before the completion of trigger logging and subsequently data logging is run again, data collection begins from the initial state before trigger logging, rather than continuing from the last time.

Stopping/suspending data logging using CPU Module Logging Configuration Tool

After data logging is stopped or suspended from CPU Module Logging Configuration Tool, all the data in the internal buffer are saved into the target memory. If a small number of records or a small file size is specified as part of the storage file switching condition, saving data to the target memory may take longer.

Error that occurs when the data logging is started

Behavior when an error occurs at the start of the multiple data logging settings at the same time is as follows:

- If the start operation is performed by using CPU Module Logging Configuration Tool, the CPU module runs data logging for the setting files that have been successfully registered.
- If the auto logging is started, the CPU module does not run any data logging session.

Behavior upon change of the internal buffer capacity

When the internal buffer capacity is changed during execution of the data logging function, note that:

- If the internal buffer capacity for the setting number of the running data logging is left empty to disable the capacity, an error occurs when the data logging is stopped and restarted (write to the programmable controller does not cause an error).
- If the internal buffer capacity of the setting number of the running data logging is changed to a smaller value, data may be lost when the data logging is stopped and restarted.

Trigger condition at the start operation of data logging

Ensure that the trigger condition is not met during the registration of the data logging settings by the start operation of data logging. If the trigger condition is met, the data logging settings cannot be registered.

When file register is specified as device/label in condition specification

After registration of data logging, do not change the file name and block number of the file register file in the following cases; the file register is specified as the device or the global label where the file register is assigned as the label for the condition specification in the "Sampling" and "Trigger" settings. Doing so may possibly result in failure to successfully collect data logging results.

When the data collection condition is set to "Time specification"

If "Time specification" is selected for the data collection condition, data collection is performed as an interrupt processing operation and therefore special care should be taken when configuring the data collection interval, the data collection timing, and the data logging processing time per scan. In the following cases, scans may take a longer time, possibly resulting in a WDT error:

- The collection interval and the collection timing are so short that data logging is frequently performed during a single scan.
- The data to be collected is so much that the data logging processing time per scan is long.

Numbering of the storage files used during data logging

■If one or more numbered storage files already exist in the file storage folder

The new file is given a file name that uses a number incremented by one from the highest number among the existing files. When the setting for files that has been completed to transfer is specified to delete in the file transfer setting, numbering to new storage files starts from 1 because no files exist in the folder at the time of restarting data logging.

■If one or more storage file folders exist but no storage files in them

A new file is stored into a folder with the lowest number and given the same number as the folder.
However, if there are 258 or more folders, a new folder is created and the file in it is given the same number as the new folder.
When the file that has been completed to transfer is specified to delete in the file transfer setting, the existing folders are deleted if no files exist in the folders when the data logging stops. (The existing folders are not deleted while the data logging is in progress.)

Behavior that occurs while collected data is stored in the target memory

If one of the following operations is performed while collected data is stored in the target memory, any unsaved data is cleared and not reflected to the results:

- Powering off and on the CPU module
- Reset

If one of the following operations is performed, unsaved data continues to be stored in the target memory:

- Changing the operating status of the CPU module from RUN to STOP
- Suspending the data logging by turning on of the special relay areas SM1312 to SM1321 (Data logging setting No.1 to 10 Data logging suspend/resume flag)
- Stopping/suspending data logging from within CPU Module Logging Configuration Tool
- Issuing the LOGTRGR instruction

Creating files and folders

Under the "LOGGING" folder that contains data logging setting files and data logging files, do not attempt to create files or folders using a personal computer or other device. Doing so may result in deletion of files and folders.

When collection is performed at the specified time

When "Time specification" is selected for the data collection condition and the collection at the specified time is selected rather than data collection during the END processing, check the collection interval by referring to the information reported in the collection interval column. Do not rely on the information reported in the date/time column because it may be incorrect due to clock accuracy error.

Changing the clock data

Whatever changes, such as advancing or reverting the clock, are made to the clock data of the CPU module during data logging, the CPU module performs data collection at the specified collection interval/timing, but the date/time column in the output file reports the changed clock data.

Events that are not recognized as a trigger condition

For trigger logging, the following events are not recognized as a trigger condition:

- A second trigger condition is met after the first trigger condition is met.
- The data condition specified as part of the trigger condition is met within the I49 interrupt program.

Point

If a data condition is specified as part of the trigger condition, any trigger condition met during the execution of the I49 interrupt program will not be recognized as a trigger condition. In this case, using I48 instead of I49 allows for avoiding the timing when the condition is not met.

Access to an SD memory card or the function memory

If data logging is performed with a setting that the data collection interval is short or the number of records to be collected is large, access (read/write) to an SD memory card or the function memory occurs so frequently that a delay occurs in completing the access. To avoid such a delay, use the following workarounds:

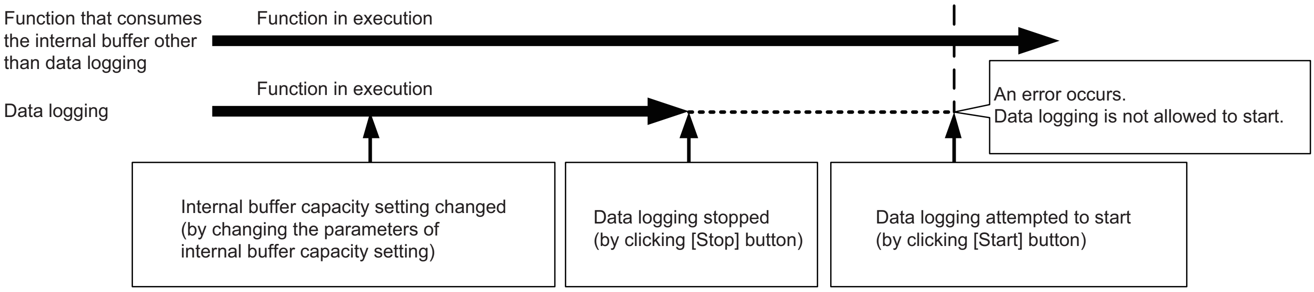
- Increase the data collection interval/timing.
- Reduce the number of data records to be collected.
- Lower the frequency of file switching.

Access to the CPU module during data logging

When the multiple data logging settings are started at the same time (when the multiple data logging settings are registered again at the same time), a time-out error may occur in communications or a dedicated instruction issued from a peripheral to the CPU module. Take measures such as increasing the time-out time period of the peripheral, reducing some data logging settings, and stopping the other functions.

Behavior at parameter change when functions consuming the internal buffer are active

If the internal buffer capacity setting is changed during the execution of the functions that consume the internal buffer, attempting to start data logging results in an error, where the data logging fails to start.



CPU module operation when registering the data logging

Note that the operating status of the CPU module is not changed until the following data logging registration or data save processing is completed. (The operating status may not be switched to STOP immediately.)

- While saving the data in the internal buffer by changing the operating status of the CPU module from RUN to STOP or operation to stop/pause data logging from CPU Module Logging Configuration Tool
- When the multiple data logging settings are started at the same time (when the multiple data loggings are registered again at the same time)
- When the data logging is started with any unused folders remaining in the storage memory

The waiting time for the operation status change of the CPU module is shortened by reducing the number of data logging settings and deleting unused folders.

Unicode text file

To view the data, use a software application with Unicode support. The data may be displayed incorrectly in a software application with no Unicode support. To open a Unicode text file with multiple lines of comments in Microsoft® Excel, be sure to drag and drop the file on the Microsoft® Excel window; otherwise, the content of the file may be displayed incorrectly in Microsoft® Excel.

Registration of the available devices

■User device, system device, file register, and index register

The data logging settings can be registered when devices of the CPU module exist. When a device number outside the range is specified, an error occurs at the registration.

■Module access device (buffer memory) and link direct device

The data logging settings can be registered when the buffer memory and devices of the target module exist. When the target module is not mounted or a device number outside the range is specified even with the target module mounted, an error occurs at the registration.

■Local device

The data logging settings can be registered when the target program name and devices in the target program exist. When a device number outside the range of the target program is specified, an error occurs at the registration.

■Local index register, local long index register, and file register in which "Use File Register of Each Program" is set

The program name cannot be specified for the following devices.

- Local index register
- Local long index register
- File register in which "Use File Register of Each Program" is set

To perform data logging of the above devices, transfer the data of the above devices to the global device on the program using the engineering tool beforehand.

For the data logging device, specify the global device where the data is transferred. The data to be logged is the data specified in the data collection condition.

File operation during execution of data logging

This section describes file operation during execution of data logging.

Target file	File operation	Behavior
Data logging setting file	Write	During execution of data logging, it is not possible to write/delete data to/in the data logging setting file being used.
	Delete	
	Initialize	During execution of data logging, it is not possible to initialize the memory storing the data logging setting file being executed.
	Folder delete	Folders cannot be deleted from the $MELPRJ$ folder in which the data logging setting file is stored.
Data Logging File	Write	During execution of data logging, it is not possible to write/delete data to/in and delete folders from the data logging setting file being used.
	Delete	
	Folder delete	
	Initialize	During execution of data logging, it is not possible to initialize the memory storing the data logging setting file being executed.

About remote operation

When remote RUN is performed while the data logging function is in the following execution status, the remote RUN may fail. In that case, wait for a while and retry remote RUN. If remote RUN still cannot be executed, check whether remote RUN is acceptable and retry remote RUN.

Execution state of data logging function	The situation to accept remote RUN
Saving data in the internal buffer into a storage location in progress	No special relay (Data logging data saving in progress) is on.
After the start operation of data logging by CPU Module Logging Configuration Tool (Registration of the data logging setting in progress)	The special relay (data logging preparation) and the special relay (data logging start) corresponding to the setting number of the data logging setting, which is being registered in the way shown in the left column, are on.
After data logging is started by auto logging (Registration of the auto logging setting in progress)	M1200 (Auto logging setting file and registration status) is on.

RUN operation through switching operation or the RUN contact

During execution of data logging, when the status of the CPU module is switched from STOP to RUN with the RUN/STOP/RESET switch, or when the RUN contact that is specified in the RUN-PAUSE contact setting turns off, it may take time to return to the RUN state.

Using together with interrupt programs

When occurrence of an interrupt is specified as the data collection condition of data logging, the processing time of the interrupt program increases because the processing time of the data logging function is added.
When the sampling method of the recording function is set to use the trigger instruction in an interrupt program, the processing time of the interrupt program also increases because the sampling processing time for the recording function is added.
For this reason, a WDT error may occur if an interrupt interval of I49 set in the parameter of "Interrupt Setting from Internal Timer" under "Fixed Scan Interval Setting" is too short (such as 0.05ms) and occurrence of I49 is specified as the data collection condition, since the processing time of the interrupt program becomes longer than the set interrupt period and the END instruction cannot be executed due to the continuous execution of the interrupt program. For how to reduce processing time of interrupt programs, refer to Saving/restoring of the file register (R) block number. (☞ Page 86 Saving/restoring of the file register (R) block number)

Data logging using the function memory as the data storage destination

■Power-off or reset during data logging

During data logging using the function memory as the data storage destination, do not power off or reset the CPU module. During data logging, if the CPU module is powered off or reset before completion of the data transfer to the data memory (before data logging is completed or stopped), all the data logging data (data logging files) in the function memory are deleted. When the data logging file transfer function is used, all data logging data in the standby state are deleted as well.
When data logging is restarted after power-off or reset, a new data logging file is given a file name that uses a number incremented by one from the end number of data logging file in the function memory before power-off or reset.

■Starting data logging during the file batch online change

Do not start the data logging using the function memory as the data storage destination during the file batch online change. Otherwise, an error occurs at the start operation of data logging.

Data logging file transfer (FTP server auto transfer)

■Operations and functions that cannot be performed

While the following operation or function is being executed, the data logging file transfer function cannot be executed.

- Auto logging

■Data collection performance

The performance of the data collection is decreased compared to when the data logging file transfer function is not used. As a result, missing may occur in the data logging setting in which no missing has occurred. When a missing has occurred, the frequency of missing may be increased. (☞ Page 225 Missing Data)

■SD memory card during data logging file transfer

If the SD memory card is specified as the data storage destination of data logging files, an SD memory card cannot be replaced during data logging file transfer. (☞ Page 238 SD Memory Card Replacement)
If the SD memory card forced disable function is executed or the SD memory card is removed during data logging file transfer, a file transfer error occurs.

■Powering off and on or a reset operation during file transfer

If the CPU module is powered off and on or is reset during data logging file transfer, data logging files being transferred may remain in the FTP server. The files being transferred and in the standby (retry) status are not transferred again.

■File transfer at power-off or reset

The files being transferred and in the standby (retry) status when the CPU module is powered off or reset are not transferred again after the power-on or reset.

■Error at "Overwrite" operation of the data logging file

When "Overwrite" is selected for the operation at the time when the number of files exceeds the maximum number of files to be saved, the file being transferred may be overwritten and a file transfer error may occur if the Ethernet line is busy.
To ensure the file transfer, configure the following settings.

1. Set a large value for "Number of files to be saved".
2. Set "Stop" for "Operation when exceeds the number of files".
3. Configure the setting to delete files that complete data logging file transfer. (☞ Page 232 Deleting files completed transfer)

■Port number when using the socket communications function

For the data logging file transfer function, the own station port numbers F230H to FFFEH are used. Thus, do not specify the own station port numbers F230H to FFFEH for the connection establishment instruction (SP.SOCOPEN) of the socket communications function while the data logging file transfer function is executed. Otherwise, the instruction may be completed with an error.

■SLMP communications

For the SLMP communications, port numbers set with the parameters take a priority. Thus, the SLMP communications are not affected by this function even if F230H to FFFEH are specified for the own station port number with the parameters.

■File transfer processing time

The file transfer processing time differs depending on the Ethernet line load ratio (network congestion), the operating status and system configuration of other communication functions.

■Communications during the data logging file transfer

Since the Ethernet communication load is high during the data logging file transfer, the behavior is as follows.

- When other communication functions such as MELSOFT connection and SLMP communications are executed using UDP, data may be lost at UDP reception and timeout and other errors may occur. Use TCP for communications while this function is executed.
- When other communication functions such as MELSOFT connection and SLMP communications are executed, their completion are delayed.

■A data logging file with a file transfer error

If a communication error, a file access error, or a file transfer stop error occurs during the file transfer, the data logging files being transferred may remain in the FTP server. Do not refer to those data logging files since the data is not guaranteed. (Data logging files in which a file transfer error has occurred can be checked in the file transfer error history.)

■Data logging stop operation at logging out from the FTP server

Even when the data logging file is transferred successfully, a data logging file transfer error occurs if the operation to stop data logging file transfer function is performed before logging out from the FTP server or logout fails.

■Timing for DNS setting

To enable the DNS setting, the CPU module needs to be powered off and on or reset. Therefore, when DNS is set during logging, data in the function memory is deleted if the function memory is specified as the data logging file storage destination. Configure the DNS setting before the logging starts.

■When the function memory is specified for the data storage destination

When the function memory is specified for the data storage destination, do not power off or reset the CPU module until the file transfer is completed. Doing so deletes all the data logging files which have not been transferred (standby) in the function memory.

■File access during the data logging file transfer

- Do not read files in the SD memory card during the file transfer. The transferred files can be read from the server. If a data logging file in the CPU module is read by using GX LogViewer during the data logging file transfer, the file being read may be deleted by the setting to delete the transferred files and a file read error may occur.
- When a file is accessed (read/write) during the data logging file transfer, the completion for the file access may be delayed depending on the number of files.

■Starting data logging during the file transfer

Since the data logging of the same setting number cannot be started during the file transfer to the FTP server, start data logging after completion of the file transfer. Otherwise, an error occurs at the start operation of data logging.

Data transfer to the data memory

■Free space in the data memory

When the transfer to the data memory is set, delete data by user data operation in the engineering tool to free up space in the data memory for storing the transferred data logging files.

When a file transfer error occurs due to out of data memory space during transfer to the data memory, free up the required space and turn on SM653 (File transfer to data memory request) to transfer the data to the data memory again.

Check the required free space for transfer to the data memory as follows.

Required free space for transfer to the data memory = Function memory capacity*1 - Function memory free area capacity*2.

*1 The size can be checked in SD648/SD649 (Function memory capacity).

*2 The size can be checked in SD650/SD651 (Function memory free space capacity).

The data logging file in the function memory are not deleted by transferring data to the data memory by using SM653. Thus, after transferring data to the data memory by using SM653, it is recommended to turn on SM652 (Function memory clear request) and delete unnecessary files in the function memory for the next data logging.

■When the CPU module is in the STOP state and data logging is in the pause state

When the CPU module is in the STOP state, and the data logging is in the pause state, data is not transferred to the data memory. Turn on SM653 (File transfer to data memory request) as necessary and transfer data to the data memory again.

■Power-off or reset during the data transfer to the data memory

- When an error occurs during the data transfer to the data memory, all data logging files in the function memory are deleted when the CPU module is powered off or reset. Turn on SM653 (File transfer to data memory request) as necessary and transfer data to the data memory again.
- During the data transfer to the data memory, a temporary file "LOGGING_T.TMP" is created and file name is changed after completion of the file transfer. Therefore, if the CPU module is powered off or reset during the transfer, the temporary file may remain. In that case, delete the file by user data operation in the engineering tool.

■Starting data logging during the data transfer to the data memory

Since the data logging of the same setting number cannot be started during the data transfer to the data memory, start data logging after completion of the file transfer. Otherwise, an error occurs at the start operation of data logging.

Data in the CPU module when a device/label is specified

Before starting the data logging, write the following data to the CPU module from the engineering tool.

Device/label specification	Data required to be written
When a local device is specified	The CPU parameter including the program name specified with CPU Module Logging Configuration Tool
When a global label is specified	Project data that is read using CPU Module Logging Configuration Tool (global label setting file)
When a local label is specified	• Project data that is read using CPU Module Logging Configuration Tool (the program file with the corresponding program name) • The CPU parameter including the program name specified with CPU Module Logging Configuration Tool

In the following cases, data cannot be written with CPU Module Logging Configuration Tool.

Device/label specification	Description
When a local device is specified	• The CPU parameters do not exist in the CPU module. • The program name specified is not set in the program setting in the CPU parameters.
When a global label is specified	• The global label setting file does not exist in the CPU module. • The global label setting file in the CPU module and the project that is read using CPU Module Logging Configuration Tool (global label setting file) do not match.
When a local label is specified	• The CPU parameters and the program file which has the corresponding program name do not exist in the CPU module. • The program name is not set in the program setting in the CPU parameters. • The program file which has the corresponding program name in the CPU module and the project that is read using CPU Module Logging Configuration Tool (the program file which has the corresponding program name) do not match.

CPU parameter at the start operation of data logging

When performing data logging in which the local device or local label specifying the program name (execution order) in the program setting of the CPU parameter is specified, do not change and write the CPU parameter to the CPU module after writing the data logging setting file. An error occurs during data logging registration after the CPU parameter is written.
Change the internal buffer capacity in the CPU parameter used for data logging before writing the data logging setting file.

Change of the file when a label is specified

Do not perform the following operations after the registration of the data logging setting by starting the data logging or between the completion of writing the data logging setting file and the registration of the data logging setting.

Label specification	Description
When a global label is specified	Writing the global label setting file where a global label is added, changed, or deleted.
When a local label is specified	Writing a program file where a program is changed (including addition, change, and deletion of a local label).

These operations change assignment of labels, and thus the data logging may not be performed to the specified label. If a file is changed, perform either of the following operations.

- Import the project of the engineering tool to CPU Module Logging Configuration Tool again, and then write the data logging setting again.
- Read the file from the CPU module with "Online Data Operation" of the engineering tool. Save the project with the engineering tool and import the project to CPU Module Logging Configuration Tool again, and then write the data logging setting again.

Point

Using SD940 (Stop direction at file change on label specification) can prevent the data logging to a different file.
Turning on the bit of SD940 corresponding to the data logging setting No. being executed generates an error at the following timings.

- When the data logging is being executed: At writing a program file or global label setting file
- When the data logging is stopped or paused: At the registration of the data logging after a program file or global label setting file is written

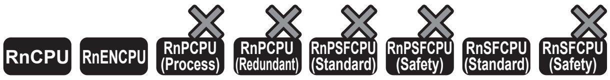
For details on SD940, refer to the following.
☞ Page 996 Latch area

12 DEBUG FUNCTION

This chapter describes the functions used for debugging.

Item	Description	Reference
Online change (ladder block)	Changes and writes a part of the program and data online.	☞ Page 125 Online change (ladder block)
Memory dump function	Stores device values of the CPU module at any given timing.	☞ Page 255 Memory Dump Function

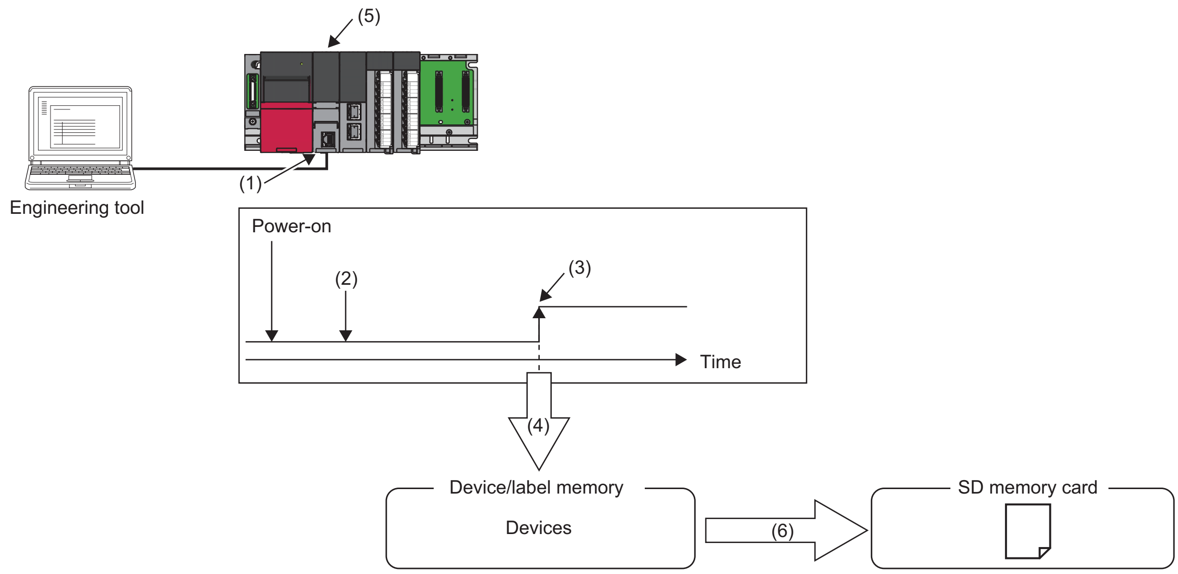
12.1 Memory Dump Function



- This function cannot be used in the R00CPU.
- When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

This function stores device values of the CPU module at any given timing. Checking data at the desired timing through the function facilitates the analysis of troubles, the occurrence of which depends on a particular condition.



(1) Perform memory dump settings.
(2) Enters a wait state for the trigger after the memory dump setting file has been written.
(3) Establishment of the trigger condition
(4) Start of data collection
(5) The LED on the front of the CPU module allows memory dump status to be checked. (☞ Page 261 LED status)
(6) The memory dump file is stored in the SD memory card. (☞ Page 260 Memory dump file)

Restriction

Before executing the function, check the versions of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

Object data

This section describes the data to be collected by memory dump.

Data to be collected

Of the devices listed below, all devices that are within the range specified in the device settings are subject to the collection.

Type	Device*1
Bit device	X, Y, M, L, B, F, SB, V, T (contact), T (coil), LT (contact), LT (coil), ST (contact), ST (coil), LST (contact), LST (coil), C (contact), C (coil), LC (contact), LC (coil), FX, FY, SM, BLn\S*3
Word device	T (current value), ST (current value), C (current value), D, W, SW, FD, SD, R*2, ZR*2, Z, RD
Double-word device	LT (current value), LST (current value), LC (current value), LZ

*1 Includes local devices as well. Note that to the memory dump file, the program names of collected local devices are output.
*2 All file registers that exist in the device/label memory are collected.
*3 The CPU module where the SFC function can be used supports this device. (Page 1139 Added and Enhanced Functions)

Trigger condition

The following table lists the conditions to be used as a trigger. Set the trigger condition in the memory dump settings. (GX Works3 Operating Manual)

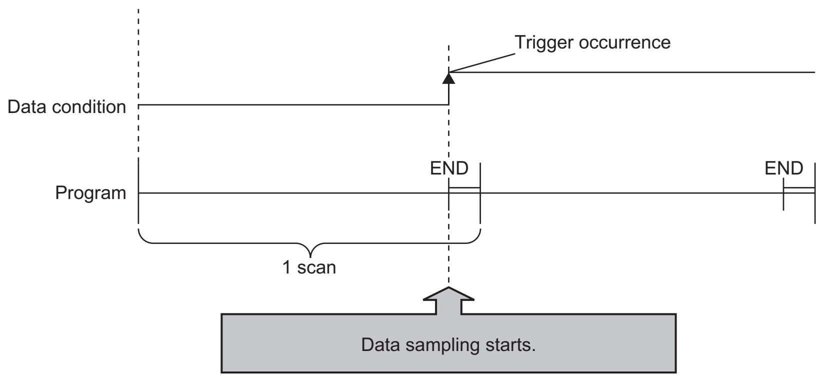
Trigger condition	Description
Device specification	Data are collected when the specified monitoring target data (bit data) turns on during the END processing.
Error code specification	Data are collected when an error occurs in the specified CPU module.

Point

- On the occurrence of consecutive triggers, if data collection due to the previous occurrence of trigger is completed and the status is "Trigger-wait not collected", the next trigger is recognized as a trigger again. Note that events other than the above are not recognized as a trigger condition.
- A trigger can be generated with trigger conditions combined. (Page 258 Combining trigger conditions)

Device specification

Data are collected when the specified monitoring target data (bit data) turns on during the END processing. Even though the value of the monitoring target data changes during a single scan, if the value during the END processing is same as that during the last END processing, it is not recognized as a trigger. Completion bits (including error completion bits) used in dedicated instructions cannot be used as a trigger.



For monitoring data, the following devices can be specified.

Type	Device*1
Bit device	X, Y, M, L, F, SM, V, B, SB, T (contact)*3, ST (contact)*3, C (contact)*3, LT (contact)*3, LST (contact)*3, LC (contact)*3, FX, FY
Word device*2	D, SD, W, SW, R, ZR, FD, RD

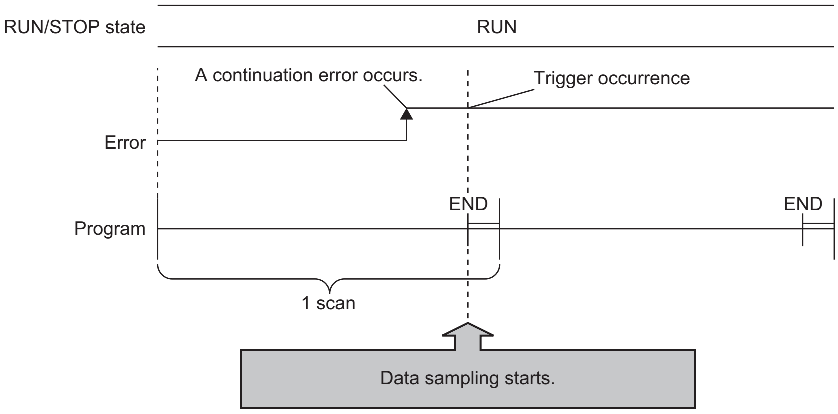
*1 A local device, index modified device, and indirectly specified device cannot be specified.
*2 Word devices allow bit specification only.
*3 In the engineering tool, specify TS as T (contact), STS as ST (contact), CS as C (contact), LTS as LT (contact), LSTS as LST (contact), and LCS as LC (contact).

Error code specification

With a specified error code of the CPU module as a trigger, data is to be collected. The occurrence timing of trigger varies depending on the error type: continuation error or stop error.

■At the occurrence of a continuation error

The occurrence timing of trigger is at the time of END processing of the scan where an error has occurred.



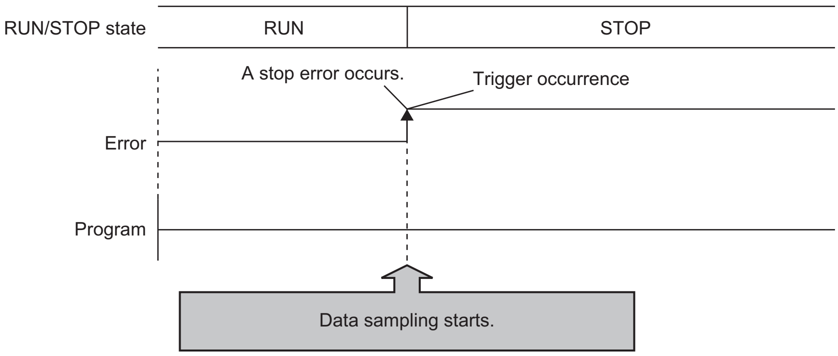
In the following situations, the occurrence of a continuation error that is specified as a trigger condition is not recognized as a trigger, with no data collection:

- After the occurrence of a continuation error that is specified as a trigger condition, the same error, or a continuation error, has occurred again.
- In the situation that a continuation error of 15 or more has occurred, a continuation error that is specified as a trigger condition has occurred.

Being recognized as a trigger requires the error to be cleared.

■At the occurrence of a stop error

The occurrence timing of trigger is at the time of occurrence of an error.

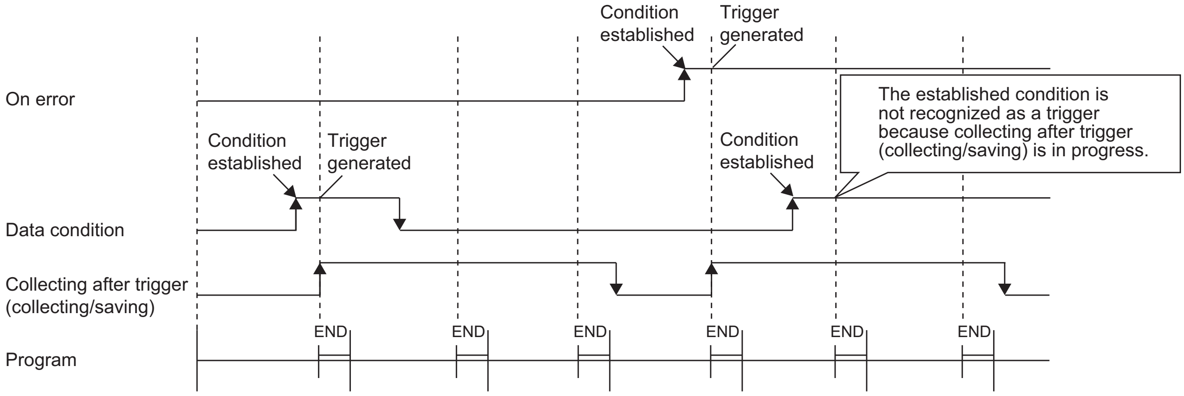


Point

To specify an annunciator number as a trigger condition, employ device specification to specify any desired annunciator number.

Combining trigger conditions

A trigger can be generated with trigger conditions combined. This combination is based on an OR condition. The establishment of a condition, either device specification or error code specification, results in data collection.



Procedure for memory dump

This section describes the procedure for memory dump. Note that each operation of the memory dump function is performed with the engineering tool.

[Debug] ⇨ [Memory Dump]

For how to view and operate the window, refer to the following.

GX Works3 Operating Manual

1. Configure the memory dump settings by the menu operation in the engineering tool.

2. Writing the memory dump setting file results in a wait state for the trigger. Whether the CPU module is in a RUN state, STOP state (a stop error also included*1), or PAUSE state, a wait state for the trigger results.

*1 Limited to where the trigger condition is device specification.

Point

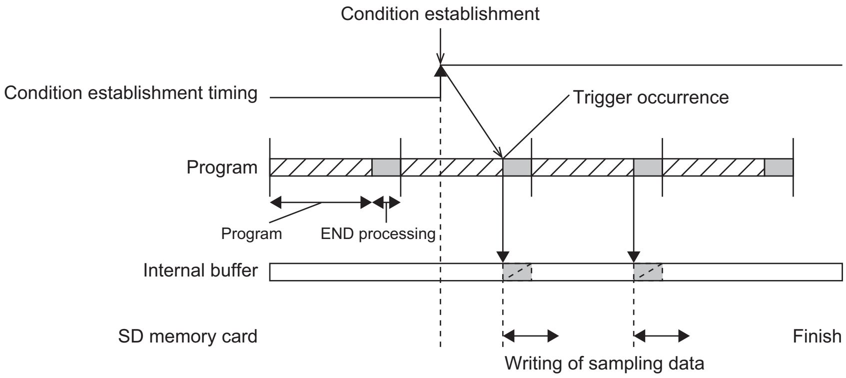
- If the memory dump setting file is stored, putting the CPU module back into operation (powering off and on, or resetting) results in a wait state for the trigger.
- The engineering tool allows the memory dump status to be checked. The event history also allows checking whether or not in a wait state for the trigger.

3. Establishment of the trigger condition initiates data collection, saving the memory dump file to the SD memory card.

4. The contents of the memory dump file (collected device data) can be checked with the engineering tool.

Flow of data collection

Collected data is stored in the internal buffer, where the data is partitioned at END processing and saved in the SD memory card.



The size of data to be collected per END processing is determined according to the capacity setting of the internal buffer (☞ Page 217 Internal buffer capacity setting) and the individual area size of devices to be collected. Data is collected and partitioned into the groups per END processing, as in the table below. If the capacity of an individual area exceeds the internal buffer capacity, collected data in the individual area is further partitioned. The first 1K byte in the internal buffer is used as the system area, and thus the collection size per END processing is determined to a value of the internal buffer capacity minus 1K byte.

Partition area		Remarks
Device area		If the total value of devices exceeds the internal buffer capacity, the amount by which the internal buffer capacity is exceeded is collected at the next END processing in order from the top device, not collected at the current END processing.
Local device (at each program)		If the total value of local devices of programs concerned exceeds the internal buffer capacity, the amount by which the internal buffer capacity is exceeded is collected at the next END processing in order from the top device, not collected at the current END processing.
File register (at each file)		—
Refresh memory	Refresh memory area	—

Effect on the scan time

The greater the number of collected points, the longer the scan time. For the increase in scan time due to the collected points, refer to the processing time. (☞ Page 1058 Memory dump function)

Memory dump file

This file stores data that is collected through memory dump (collection result by memory dump). Data collected by one execution is saved in one file. The memory dump file is saved in a binary format and stored under the "MEMDUMP" folder.

Save file name

The file name can be arbitrarily set within a range of 64 characters (extension and period included) together with an auto-assigned number (00 to 99). Specify the save file name in the memory dump settings. (GX Works3 Operating Manual)

Ex.

MEMDUMP_00*1

*1 Between a specified file name (MEMDUMP) and an auto-assigned number (00), the single-byte underscore (_) is added.

When the memory dump function is registered, the debug folder (DEBUG (fixed)) and the memory dump folder (MEMDUMP (fixed)) are created in the SD memory card. The memory dump file (result file) is stored in the memory dump folder. One folder can contain a maximum of 100 files. If any file does not exist in creating a save file, the file with the number 00 is created. If any file already exists in creating a save file, the behavior is as follows:

Number of files	Behavior
For less than 100	Creates a file*2 assigning the number obtained by adding 1 to the number of the file where the creation date and time is the latest.
For 100 (maximum)	Deletes the file where the creation date and time is the oldest and creates a new file using the deleted number as it is.

*2 If the corresponding file number is 99, a file with file number 00 is created.

States of the memory dump function

The state of the memory dump function is reflected in the memory dump status. The engineering tool allows the memory dump status to be checked. (GX Works3 Operating Manual)

Memory dump status

The following table lists the memory dump status.

Memory dump status	Description
Trigger-wait not collected	A state that data is not yet collected and establishment of the trigger condition is being waited
Collecting after trigger	A state that collection of the data after trigger is in progress (includes a state that collected data is being saved in the target memory)
Collection completed	A state that collection of a specified data is completed
Error	A state that a memory dump error occurs and memory dump fails

12

LED status

Whether the memory dump function is active or not can be checked by the LED of the CPU module.

State of the memory dump function	LED status		
	FUNCTION LED	CARD READY LED	CARD ACCESS LED
Trigger-wait not collected	On	On	Off
Collecting after trigger	Flashes slowly (every one second)	On	Turns on when the SD memory card is accessed
Collection completed	Flashes at normal rate (every 200ms)	On	Off

Point

To let the FUNCTION LED indicate the status while the memory dump function is used, setting "LED Display Setting" in "RAS Setting" of the CPU parameter is required. (Page 143 LED display setting)

For the FUNCTION LED indication, the LED indicator follows the status of the following LED indications.

- When the external input/output forced on/off function is executed (in registration) (Page 182 FUNCTION LED)
- When program restoration information is not written (Page 1101 Checking the program restoration information write status)

After the external input/output forced on/off function has been executed (after the registration is canceled) and the program restoration information has been written, the LED display follows the status of the memory dump function.

Sizes of files used for the memory dump function

This section shows the sizes of files used for the memory dump function.

Capacity of the memory dump setting file

The capacity of the memory dump setting file varies depending on the length of the save file name. The following formula is used for the calculation:

Capacity of memory dump setting file = (((Number of characters of save file name*1 × 2 bytes + 1201 bytes (fixed)) + 3) ÷ 4)*2 × 4

*1 Except for the period and extension.
*2 The remainder is discarded.

Capacity of the memory dump file

The capacity of the memory dump file is given by the total of the following items:

Capacity of memory dump file = Volume of header + Volume of data of program file name + Volume of device data + Volume of local device data + Volume of data of file register file name + Volume of file register data

■Volume of header

The volume of header is given by:

Volume of header = 1088 bytes (fixed)

■Volume of data of program file name

The volume of data of program file name is given by the total of the following items. Note that this data is always created in the memory dump file regardless of the settings of CPU parameters.

Volume of data of program file name = 16 bytes (fixed) + (Number of programs × (2 bytes (length of program file name) + 130 bytes (program file name))

■Volume of device data

The volume of device data is given by the total of the following items. Note that this data is always created in the memory dump file regardless of the settings of CPU parameters.

Volume of device data = 682 bytes (fixed) + Volume of collected device data

The volume of collected device data is given by the following:

Volume of collected device data = (Total number of points of bit devices ÷ 8) + (Total number of points of word devices × 2) + (Total number of points of word devices × 4)

■Volume of local device data

The volume of local device data is given by the total of the following items. Note that this data is not created in the memory dump file unless local devices are set in the CPU parameters.

Volume of local device data = 16 bytes (fixed) + (Number of programs × 4 bytes) + (Number of programs × Volume of local device contents)

Item	Calculation method
Volume of local device contents	418 bytes (fixed) + Volume of collected local device data
Volume of collected local device data	(Total number of points of bit devices ÷ 8) + (Total number of points of word devices × 2) + (Total number of points of word devices × 4)

■Volume of data of file register file name

The volume of data of file register file name is given by the total of the following items. Note that this data is not created in the memory dump file unless a file register file exists in the device/label memory.

Volume of data of file register file name = 16 bytes (fixed) + (Number of file register files × (2 bytes (length of file register file name) + 130 bytes (file register file name)

■Volume of file register data

The volume of file register data is given by the total of the following items. Note that this data is not created in the memory dump file unless a file register file exists in the device/label memory.

Volume of file register data = Number of file register files × (148 bytes (fixed) + (Number of file registers × 2))

Special relay and special register used in the memory dump function

For details, refer to the following.

- Special relay: Special relay relating to the memory dump function (☞ Page 954 Memory dump function)
- Special register: Special register relating to the memory dump function (☞ Page 1005 Memory dump function)

Precautions for the memory dump function

This section describes precautions to take when using the memory dump function.

Mutual exclusion of the memory dump function

12

The mutual exclusion of the memory dump function is as follows.

The following table lists the cases when another function is executed during the execution of the memory dump function[*1].

Function that has been already executed	Function to be executed later	Behavior
Memory dump function	CPU module data backup function	The CPU module data backup function cannot be executed while memory dump is being registered/cleared.
	CPU module data restoration function	The CPU module data restoration function cannot be executed while a memory dump file or memory dump setting file is being read or memory dump is being registered/cleared.
	iQ Sensor Solution data backup/restoration function	The iQ Sensor Solution data backup/restoration function cannot be executed while memory dump is being registered/cleared.
	Function specified in the internal buffer capacity setting[*2]	If the internal buffer capacity setting is changed to execute the subsequent function, attempting to start the subsequent function results in an error. The memory dump continues to function normally.
	Function not specified in the internal buffer capacity setting	If the condition "Total capacity that is set in the internal buffer capacity setting + Internal buffer capacity that is set in other than the internal buffer capacity setting > 3072K bytes" is satisfied, attempting to start the subsequent function results in an error. The memory dump continues to function normally.
		If the internal buffer capacity setting is changed to execute the subsequent function, attempting to start the subsequent function results in an error. The memory dump continues to function normally.

*1 The state where the memory dump function is in execution includes the memory dump status of "Collecting after trigger" or the save status of "Saving in progress".

*2 The memory dump function is not included.

The following table lists the cases when the memory dump function is executed during the execution of another function.

Function that has been already executed	Function to be executed later	Behavior
CPU module data backup function	Memory dump function	While the CPU module data backup function is being executed, memory dump cannot be registered/cleared.
CPU module data restoration function		While the CPU module data restoration function is being executed, a memory dump file or memory dump setting file cannot be read or memory dump cannot be registered/cleared.
iQ Sensor Solution data backup/restoration function		While the iQ Sensor Solution data backup/restoration function is being executed, memory dump cannot be registered/cleared.
Function specified in the internal buffer capacity setting[*3]		If the internal buffer capacity setting is changed to execute memory dump, attempting to start memory dump results in an error. The function already in execution continues to function normally.
Function not specified in the internal buffer capacity setting		If the internal buffer capacity setting is changed to execute memory dump, attempting to start memory dump results in an error. The function already in execution continues to function normally.

*3 The memory dump function is not included.

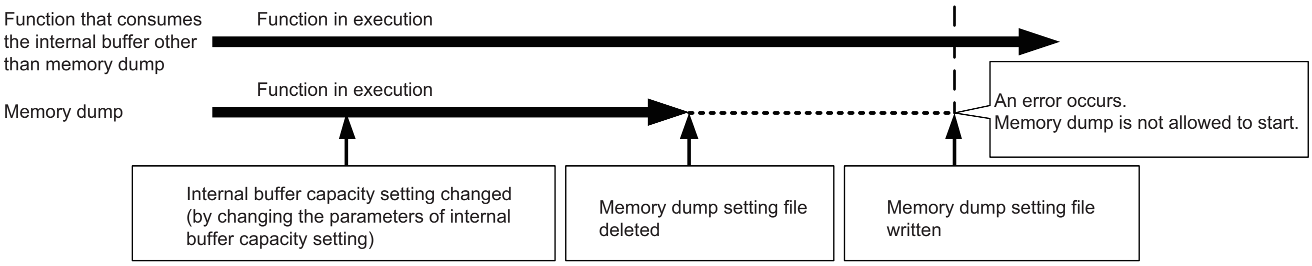
The following table shows the cases where the file operation related to the memory dump function is executed while the memory dump function is in execution.*4

Target file	File operation	Behavior
Memory dump setting file	Write	Settings that are subsequently written during the execution of the memory dump function are reflected after the completion of save, not reflected immediately.
	Delete	If the memory dump setting file is subsequently deleted during the execution of the memory dump function, the memory dump settings are cleared after the completion of save.
	Initialize	Initialization fails on the memory dump setting file during the execution of the memory dump function.
	Folder delete	Folder delete fails on the folder in which the memory dump setting file is stored.
Memory dump file	Write, read, delete, initialize, and folder delete	Write, read, delete, initialize, and folder delete fails on the memory dump file during the execution of the memory dump function.

*4 The state where the memory dump function is in execution includes the memory dump status of "Collecting after trigger" or the save status of "Saving in progress".

Behavior at parameter change when functions consuming the internal buffer are active

If the internal buffer capacity setting is changed during the execution of functions that consume the internal buffer, attempting to start memory dump results in an error, where the memory dump fails to start.



Operation on each individual file

Write, read, and delete are possible on each file. In addition, folder/file all delete and folder delete are possible on the memory or folder in which files are stored. The following table shows whether each operation is possible or not depending on the execution status of memory dump.

○: Operation possible, ×: Operation not possible

File type	Operation to be performed							
	Read		Write/delete		Folder/file all delete		Folder delete	
	Not during execution*1	During execution*1	Not during execution*1	During execution*1	Not during execution*1	During execution*1	Not during execution*1	During execution*1
Memory dump setting file	○	○	○	○	○	×	×	×
Memory dump file	○	×	○	×	○	×	○	×

*1 The state where the memory dump function is in execution includes the memory dump status of "Collecting after trigger".

Where to carry out memory dump

Concurrent execution from multiple sources is not allowed. In the CPU module, execution at a time from only one source is possible.

Trigger condition during the registration of memory dump

If a trigger condition is established during the registration of memory dump, the memory dump settings can be registered. A second establishment of the trigger condition is recognized as a trigger condition there.

When file register is specified as specification device in the condition specification

After registering memory dump, do not change the file name of the file registers and the block number of the file registers. Doing so may result in a failure to successfully collect the memory dump results.

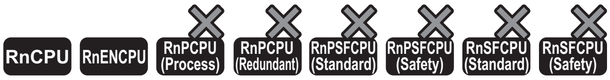
Creating files and folders

Under the "MEMDUMP" folder containing memory dump files, do not create any files or folders using a personal computer or other device. Doing so may result in deletion of files and folders.

Access to the SD memory card

The SD memory card is so frequently accessed that a delay occurs in completing the access to the SD memory card (read/write).

13 DATABASE FUNCTION

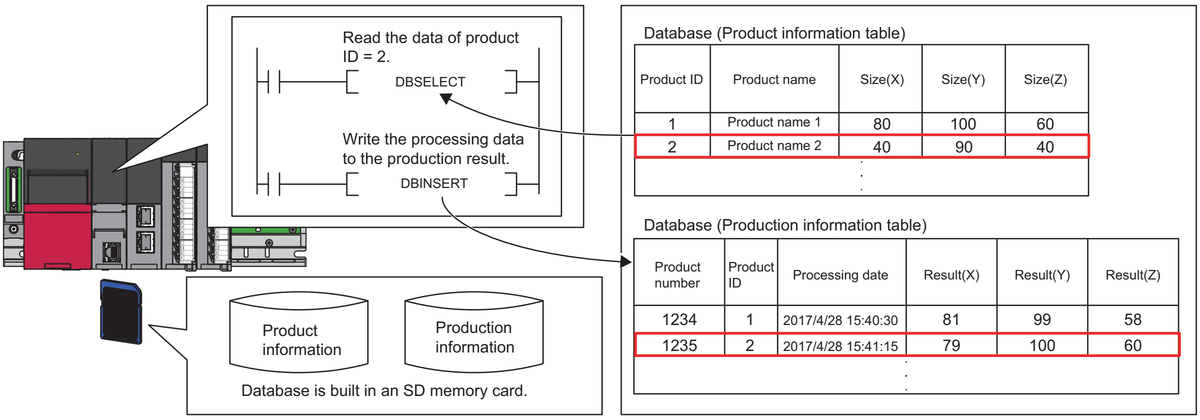


• This function cannot be used in the R00CPU, R01CPU, and R02CPU.

This function manages the data such as product information or production information as a database in an SD memory card of the CPU module.

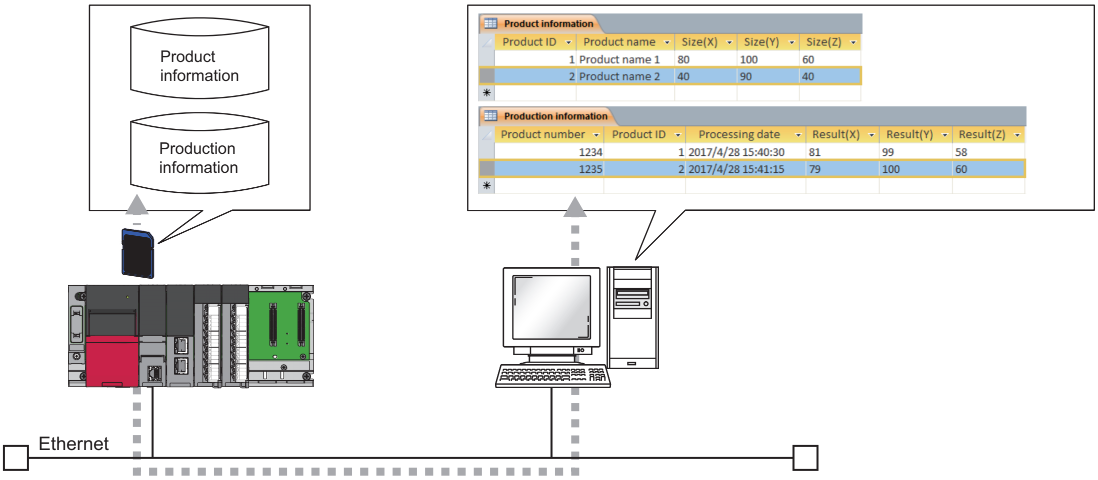
Database access instruction

The database access instructions allow the databases of product information and production information to be created, input, updated, edited, searched, transacted, and rolled back.



CPU module database access (from external device) function

The CPU module database access function enables operations such as table creation and record operation/search of the database built in an SD memory card inserted into the CPU module.



13.1 Specifications

Database specifications

The following table lists specifications of databases which the user can create.

Item	Description
The maximum number of fields	128 fields per table*1
The maximum number of records	No limitation (they can be created up to the capacity of the SD memory card)*2*4
The maximum number of tables	32 tables per database*3
The maximum number of databases	No limitation (they can be created up to the capacity of the SD memory card)
The maximum number of concurrently connectable databases (number of connections)	4
The maximum number of characters of a field name	32 single-byte alphabetical characters
The maximum number of characters of a table name	32 single-byte alphabetical characters
Corresponding data type	☞ Page 267 Data type

*1 For the programmable controller CPU with firmware version earlier than "28", up to 16 fields can be created per table.
*2 For the programmable controller CPU with firmware version earlier than "28", up to 100000 records can be created per table.
*3 For the programmable controller CPU with firmware version earlier than "28", up to 16 tables can be created per database.
*4 For rough standards of the access time depending on the database size, refer to the following.
☞ Page 1073 Database function processing time

■Data type

The following table lists the data types that can be used in the table created by using the database access instructions and the corresponding data types for the CPU module database access function.

Data type that can be used in the table created by using the database access instruction	Range	Corresponding data type for the CPU module database access function
BOOL: Bit	0, 1	BOOLEAN: Bit
WORD: Unsigned 16-bit numerical value	0 to 65535	INT: Unsigned 16-bit numerical value
DWORD: Unsigned 32-bit numerical value	0 to 4294967295	BIGINT: Unsigned 32-bit numerical value
INT: Signed 16-bit numerical value	-32768 to 32767	INT: Signed 16-bit numerical value
DINT: Signed 32-bit numerical value	-2147483648 to 2147483647	BIGINT: Signed 32-bit numerical value
REAL: Single-precision real number	E±1.175495-38 to E±3.402823+38	REAL: Single-precision real number
LREAL: Double-precision real number	E±2.22507385850721-308 to E±1.79769313486231+308	DOUBLE PRECISION: Double-precision real number
STRING: String (Shift-JIS code)	Number of supported characters: 1 to 124	NLSCHAR: String (Unicode)
WSTRING: String (Unicode)	Number of supported characters: 1 to 124	NLSCHAR: String (Unicode)

The following table lists the data types that can be used in the table created by the CPU module database access function and the corresponding data types for the database access instructions.

Data type that can be used in the table created by the CPU module database access function	Range	Corresponding data type for the database access instruction
BOOLEAN: Bit	0, 1	BOOL: Bit
INT: Signed 16-bit numerical value	-32768 to 32767	INT: Signed 16-bit numerical value
BIGINT: Signed 32-bit numerical value	-2147483648 to 2147483647	DINT: Signed 32-bit numerical value
REAL: Single-precision real number	E±1.175495-38 to E±3.402823+38	REAL: Single-precision real number
DOUBLE PRECISION: Double-precision real number	E±2.22507385850721-308 to E±1.79769313486231+308	LREAL: Double-precision real number
NLSCHAR: String (Unicode)	Number of supported characters: 1 to 124	WSTRING: String (Unicode)

Point

- To input the data to the field of STRING using the CPU module database access function, use the data type of NLSCHAR. (Although the data is managed as Unicode in the database of the CPU module, it is converted to Shift-JIS code on the device of the CPU module.)
- To input the data in the data type of WORD or DWORD to the field using the CPU module database access function, the data must be within the range of WORD: 0 to 65535 or DWORD: 0 to 4294967295.

Available operations

The following table lists the operations that can be performed with the database functions.

○: Can be performed, ×: Cannot be performed

Operation		Description	Database access instruction		CPU module database access function	
			Availability and instruction symbol		Availability and application	
Database	Connection	Connects to the specified database.	○	DBOPEN(P)	○	Microsoft Access®, Excel®, user-created application
	Disconnection	Disconnects from the specified database.	○	DBCLOSE(P)	○	
	Import	Imports the specified data from the specified Unicode text file to create a database.	○	DBIMPORT(P)	×	—
	Export	Exports the specified data in the specified database into a Unicode text file.	○	DBEXPORT(P)	×	—
Table	Add	Adds a table to the database.	×	—	○	Microsoft Access®
Record	Access	Accesses the specified table in the database.	×	—	○	Microsoft Access®
	Add	Adds a record to the specified table of the database.	○	DBINSERT(P)	○	Microsoft Access®
	Update	Updates the value of the specified record on the specified table in the database.	○	DBUPDATE(P)	○	Microsoft Access®
	Acquire (Search)	Acquires the value of the specified record on the specified table in the database.	○	DBSELECT(P)	○	Excel®
	Delete	Deletes the specified record on the specified table in the database.	○	DBDELETE(P)	○	Microsoft Access®
Transaction	Start	Starts a transaction for the specified database.	○	DBTRANS(P)	○	Microsoft Access® (The transaction range depends on the application.)
	Commit	Commits a transaction for the specified database.	○	DBCOMMIT(P)	○	Microsoft Access® (The transaction range depends on the application.)
Rollback		Performs rollback on the specified database.	○	DBROLBAK(P)	○	Microsoft Access® (The transaction range depends on the application.)
Database operation by a desired SQL			×	—	○	User-created application

Point

- For details on the database access instructions, refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

- For the available SQL commands for the CPU module database access function, refer to the following.

☞ Page 1127 List of Available SQL Commands for CPU Module Database Access Function

Restriction

- The user cannot execute multiple database access instructions simultaneously. If they are executed simultaneously, instructions other than the first one will be completed with an error.
- Even though the operation can be performed for the database of the CPU module by the SQL command, an error occurs on the application side when it cannot be performed in the application. In this case, check the details of the error and perform the operation within the range of the specifications of the application again.

13.2 Database Access Instruction

Usage procedure

This section describes the procedure to use the database function.

Creating databases

To construct a database on an SD memory card, create a Unicode text file which defines the configuration of the database and its tables, store the file on the SD memory card, and execute the DBIMPORT(P) instruction, specifying the created Unicode text file.

1. Create a Unicode text file which defines the configuration of a database and its tables. (☞ Page 271 Creating Unicode text files)

2. Write the created Unicode text file to the SD memory card. For details on how to write Unicode text files (write of user data), refer to the following.

📖 GX Works3 Operating Manual

3. By executing the DBIMPORT(P) instruction with the created Unicode text file specified, the database folder is created on the same layer as the Unicode text file and the database is constructed. (☞ Page 274 Folder configuration of databases)

Point

The DBEXPORT(P) instruction enables the user to check the contents of a database by exporting it to a Unicode text file. Also, when the user wants to add more than one record or field to a database, the user can change the configuration of the database by exporting it to a Unicode text file, editing the file, and executing the DBIMPORT(P) instruction to get the database back to the system.

Operating databases

First, the user must execute the DBOPEN(P) instruction to connect to the database which the user wants to work on. Also, after finishing database operation, the user must execute the DBCLOSE(P) instruction to disconnect from the database.

1. Execute the DBOPEN(P) instruction to connect to the database which the user wants to work on (connect to the database by specifying the folder which was created in Step 3 of the database creation procedure). When the execution is finished, a value including the "database ID number" is returned.

Point

Database ID numbers are necessary to perform database operation. Therefore, please make a note of the database ID number of each database.

2. Once the connection to the database is established, perform operation (addition, update, search, or deletion) of the database. To perform database operation, the user must specify the "database ID number" of the target database in each instruction.

3. When database operation is finished, execute the DBCLOSE(P) instruction to disconnect from the database.

Point

The user can use a transaction when the user wants to perform multiple operations for a database as a set and update the database at once. (☞ Page 274 Transactions for databases)

Creating Unicode text files

When the DBIMPORT(P) instruction is executed, a database is created on an SD memory card, according to setting details in the tab-delimited format of Unicode text file. The user must create Unicode text files on an SD memory card.

Setting details of Unicode text file

The following table lists items which must be specified in a Unicode text file to create a database.

Item	Description
Database name	Specify a database name, using single-byte alpha-numeric characters. • Within 32 characters • Not case-sensitive • The following characters cannot be used: " # % * + , / : ; < = > ? [\ ] | ' { } & ~ $ @ ^
Table definition start tag	Specify the tag, <table>, using a single-byte lower-case characters. When defining more than one table, enclose each table with <table> and </table>.
Table name	Specify a table name, using single-byte alpha-numeric characters. • Within 32 characters • Not case-sensitive • The following characters cannot be used: " # % * + , / : ; < = > ? [\ ] | ' { } & ~ $ @ ^ • The maximum number of tables is 32.[*3]
Field name	Specify the name of each field of each record, using single-byte alpha-numeric characters, and in the tab-delimited format. • The maximum number of characters which can be used for each field name is 32. • Case-sensitive • The following characters cannot be used: " and ' • The maximum number of fields is 128.[*3]
Data type	Specify the data type of each field in the tab-delimited format. The data types which can be specified are as follows. They must be specified using upper-case characters. • BOOL • WORD • DWORD • INT • DINT • REAL • LREAL • STRING: Specify the number of characters after a colon (:) (For example, when the number of characters is 16, the user can specify like that: "STRING: 16"). The range of the number of characters is 1 to 124.[*1*2] • WSTRING: Specify the number of characters after the colon (:) The range of the number of characters is 1 to 124.[*1]
Key constraint	Specify the key constraint on each field, using single-byte alphabetical characters. Between key constraints, a tab must be inserted as a delimiter. One of the following can be specified: 0: None 1: Primary key (used to uniquely identify each record) When a primary key is specified, its key name also must be specified after a colon (:) (e.g. "1: pk1"). Key names must be up to 16 single-byte alphabetical characters and are case sensitive. 2: NOT NULL constraint (When this setting is selected, setting values cannot be left unspecified by skipping specification of primary keys or foreign keys. (This value is used to set a restriction.) 3: Foreign key (used to refer to the field of another table.) When a foreign key is specified, a table name, which is to be associated with the key name, also must be specified after a colon (:), like that: "3: Key Name: Table Name" (e.g. "3: fk1: tb2"). Key names must be up to 16 single-byte alphabetical characters and are case sensitive.
Index	Specify 0 (single-byte numeric character) when any indexes are not specified. When an index is specified, the name of the index and a field name must be specified using a colon (:) between them, like that: "Index Name: Field Name" (e.g. "id1: field1"). • Index names must be up to 16 single-byte alphabetical characters and are case sensitive. • The maximum number of indexes is 128.[*3] • For fields with a primary or foreign key constraint, indexes are automatically set, and it is not necessary to specify indexes by this setting (if specified, the DBIMPORT(P) instruction is completed with an error).
Setting value	Specify the setting value of each field in the tab-delimited format. • Characters or numbers which will be set to each field must match with the specified data type (☞ Page 271 Creating Unicode text files, ☞ Page 272 Format of setting values in Unicode text) • Individual records are set by row.
Table definition end tag	Specify the tag, </table>, using a single-byte lower-case characters.

*1 The termination character, NULL, is not counted in the number of characters.

*2 The length of characters is represented by single-byte characters. For double-byte character strings, one double-byte character should be counted as two single-byte characters.

*3 For the programmable controller CPU with firmware version earlier than "28", the maximum number is 16.

Point

Specifications of the character code for Unicode text files are as follows:

- Character encoding schema: UTF-16 (Little-Endian)
- BOM: Yes

■Format of setting values in Unicode text

Item	Description
WORD, DWORD, INT, DINT	The value must be specified in the decimal format (e.g. 0, 1, 111, -111).
BOOL	The value must be 0 or 1.
REAL, LREAL	The value must be specified in the following exponential format (e.g. 1.0e-01, 1.0E+01): Mantissa + E*1 + Exponent part*2

*1 "E" can be specified in lower case.

*2 The exponent part must always have a sign.

Configuration of Unicode text files

```
Database name
<table>
Table name 1
Field name 1 [Tab] Field name 2 [Tab] ... [Tab] Field name  n
Data type 1 [Tab] Data type 2 [Tab] ... [Tab] Data type n
Key restriction 1 [Tab] Key restriction 2 [Tab] ... [Tab] Key restriction n
Index 1 [Tab] Index 2 [Tab] ... [Tab] Index n
Set value 1 [Tab] Set value 2 [Tab] ... [Tab] Set value n
 :

Set value k [Tab] Set value 2 [Tab] ... [Tab] Set value n
</table>
<table>
Table name 2
(The subsequent setting is the same as the setting of Table name 1.)
</table>
```

[Tab] in the figure represents a tab character and it is entered using the [Tab] key on the keyboard in practice. Line breaks, comments, tabs, and spaces are processed as follows.

Item	Description
Line break	• The line break code is CR+LF. • Lines consisting of only line breaks are ignored.
Comment	A row beginning with "//" is regarded as a comment. The maximum number of characters in a comment row is 2048 (including two characters for the line break code).
Tab	Tabs cannot be used for database names, table names, field names, key constrains, and setting values. Tabs are recognized as delimiters.
Space	• Spaces between tabs and printable characters are not recognized as a part of settings. • Spaces between printable characters are recognized as spaces.

Ex.

Example of the Unicode text file format (Database name: database1, Table name: product-info1)

Data type: INT
Key restriction: Main key

Data type: STRING, Number of characters: 124
Key restriction: None

proid1	proinfo1	proinfo2	· · · · · · · · ·	proinfon
1	abcd	efgh		pqrs
1001	abcd	efgh		pqrs tuv
· · ·				
1010	abcd	efgh		pqrs

```
//Database name
database1
<table>
//Table name
product-info1
//Field name
proid1 [Tab] proinfo1 [Tab] ... [Tab] proinfon
//Data type
INT [Tab] STRING: 124 [Tab] ... [Tab] STRING: 124
//Key restriction
1: pk1 [Tab] 0 [Tab] ... [Tab] 0
//Index
0
//Record
//First record
1 [Tab] abcd [Tab] efgh [Tab] ... [Tab] pqrs
//Second record
1001 [Tab] abcd [Tab] efgh [Tab] ... [Tab] pqrs tuv
:
:
//nth record
1010 [Tab] abcd [Tab] efgh [Tab] ... [Tab] pqrs
</table>
```


Point

Unicode text files can be created using a text editor or spreadsheet software program. The method for saving Unicode text files in Notepad or Excel is as follows:

- In the text editor (Notepad, an accessory of Windows® 7 or later), select [Save As] from the [File] menu, and select "Unicode" as the character code to save the file.
- In the spreadsheet software program (Excel 2010), select [Save As] from the [File] menu, and select "Unicode" as the file type to save the file. When a Unicode text file is created using a spreadsheet software program, tabs are automatically inserted by saving the file. Therefore, it is not necessary to insert tabs into the file in the spreadsheet software program.

Transactions for databases

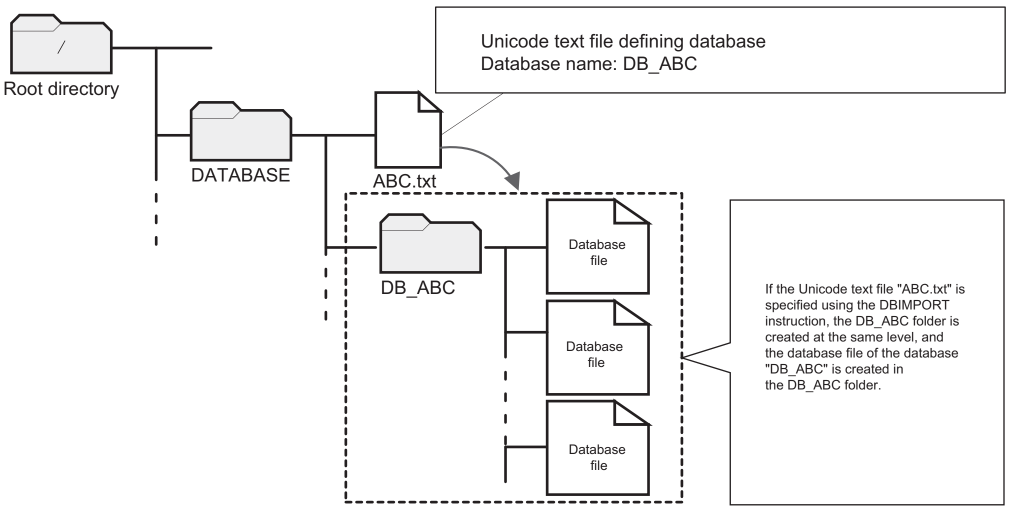
Use a transaction (the DBTRANS(P) instruction and the DBCOMMIT(P) instruction) to perform multiple operations for a database as a set and update the database at once. The DBTRANS(P) instruction starts a transaction, and the DBCOMMIT(P) instruction groups the results of the following instructions together to update the target database at once.

- The target of a transaction must be a single database. If a transaction is executed for more than one database, transaction instructions for the second and later databases are completed with errors.
- When the CPU module is powered off or is reset in the period from the start of a transaction (execution of the DBTRANS(P) instruction) to the commit of the transaction (execution of the DBCOMMIT(P) instruction), the database is automatically rolled back to the status before the start of the transaction. However, if a database access instruction is completed with an error, the database is not roll backed to the previous status.
- To cancel the running transaction processing and reset the database to the status before the start of the transaction, perform a rollback (the DBROLBAK(P) instruction). A rollback is an operation which resets the target database to the previous status by executing the DBROLBAK(P) instruction in the period from the start of the transaction to the commit of the transaction. When the DBCLOSE(P) instruction is executed without the DBCOMMIT(P) instruction or the DBROLBAK(P) instruction, the transaction is committed during execution of the DBCLOSE(P) instruction.

1. Execute the DBOPEN(P) instruction to connect to a database.
2. Execute the DBTRANS(P) instruction to start a transaction.
3. Operate the database.
4. To commit the transaction, execute the DBCOMMIT(P) instruction. When committing the transaction is not necessary, execute the DBROLBAK(P) instruction to restore to the state before the start of the transaction (rollback).
5. Execute the DBCLOSE(P) instruction to disconnect from the database.

Folder configuration of databases

The database folder for storing database files is generated at the same level in the folder hierarchy as the Unicode text file which is specified in the DBIMPORT(P) instruction. Also, the database folder is given the same names as the database. The length of a database folder path name must be up to 128 characters, including the drive name and a colon (:) prefixed to the path name.



Operation of files/folders in the database folder

Files and folders in the database folder are used for database access instructions. They should not be changed, deleted, and newly created. Otherwise, database access instructions may be completed with errors. In that case, delete the relevant database folder.

Timing of database update

Once execution of a database access instruction is completed, the target database is updated. However, during transaction, the database is not updated each time execution of an instruction is completed. Instead, all changes during the transaction are applied to the database at once when the DBCOMMIT(P) instruction is executed.

Precautions

This section describes precautions on using the database function.

Free space of SD memory card required for execution of database access instruction

To execute a database access instruction, the SD memory card must have at least 10MB of free space. When a database access instruction is performed and if the SD memory card does not have enough free space, the instruction is completed with an error.

When the SD memory card is write-protected

Before using database access instructions, check that the write protection of the SD memory card used is disabled. If a database access instruction is performed while the SD memory card is write protected, the instruction is completed with an error.

When the CPU module database access function is used

The database access instructions can be executed while SM1498 (CPU module database start-up flag) is on.

When the database is being operated by another function

Before creating a database by using the DBIMPORT(P) instruction, check that the same database is not being operated by another function.

If the DBIMPORT(P) instruction is executed specifying a name of a database in the SD memory card while the same database is operated by another function, the DBIMPORT(P) instruction may be completed with an error and the database operation in another function may fail. After the DBIMPORT(P) instruction is completed with an error, if the database operated by another function cannot be operated, the database may be broken. Power off and on or reset the CPU module, and build the database again by using the DBIMPORT(P) instruction.

Combination of the firmware version and database

For the combination of the firmware version of the programmable controller CPU that creates a database and that of the programmable controller CPU that can operate the database, refer to the following.

☞ Page 293 Combination of the firmware version and database

Removal of the SD memory card

Do not remove the SD memory card while connecting to a database by using the DBOPEN(P) instruction or executing a database access instruction. Otherwise, the database access instruction is completed with an error.

13.3 CPU Module Database Access (from External Device) Function

The CPU module database access function operates a database, that is built in an SD memory card inserted into the CPU module, from an application on a personal computer through the Ethernet port of the CPU module.
To use the CPU module database access function, install CPU Module Database Access Driver into a personal computer. (CPU Module Database Access Driver Installation Instructions)

Point

In the CPU module database access function, the CPU module operates as the ODBC server and the personal computer operates as the ODBC client.
ODBC (Open Database Connectivity) is the standard API that allows applications to connect with the database management system (DBMS) to acquire and write the data.

Restriction

Before using the CPU module database access function, check the version of the CPU module and the engineering tool used. (Page 1139 Added and Enhanced Functions)

Usage procedure

This section describes the procedure for operating the database by using the CPU module database access function.

Procedure	Description
Construction of usage environment	Construct the environment to use the CPU module database access function.
Database access setting	Configure this setting to create or change a database of the CPU module.
Database operation	Operate a database of the CPU module from a personal computer.

Construction of usage environment

■Engineering tool

1. Module parameter setting of the CPU module

Set whether or not to use the CPU module database access function, login name, and password in the module parameter of the CPU module. (Page 278 Built-in database access setting)

2. Update of the built-in database access setting in the CPU module

Write the module parameter to the CPU module, and then power off and on or reset the CPU module.

■CPU Module Database Access Driver

1. Obtaining CPU Module Database Access Driver

For CPU Module Database Access Driver, please consult your local Mitsubishi representative.

2. Installing CPU Module Database Access Driver

Install CPU Module Database Access Driver into the personal computer. (CPU Module Database Access Driver Installation Instructions)

Point

The installer of CPU Module Database Access Driver has the 32-bit or the 64-bit version. Choose the installer depending on the supported bit of the application (Microsoft Access, Excel, user-created application) used for the CPU module database access function.
For example, when using Microsoft Access with 32-bit version, use the 32-bit installer of CPU Module Database Access Driver even though the 64-bit OS is used.

Database access setting

1. Creating a database

Create a database in an SD memory card of the CPU module. (☞ Page 279 Creating a database)

2. Adding a database to the ODBC data source

Add the created database of the CPU module to the ODBC data source of the personal computer. (☞ Page 280 Adding a database to the ODBC data source)

Database operation

1. Checking the start-up of the ODBC server

Check that SM1498 (CPU module database start-up flag) is on.

When SM1498 is on, the ODBC server of the CPU module is active. The ODBC server starts and SM1498 turns on when:

- An SD memory card storing the database folder is inserted to the CPU module, and the CPU module is powered off and on or is reset.
- An SD memory card storing the database folder is inserted to the CPU module powered-on when SD1498 (Start-up status of CPU module database) is 0H.

Whether or not the ODBC server has been started up can be checked in SD1498 (Start-up status of CPU module database).

Point

The active ODBC server stops when:

- The SD memory card is disabled (SD memory card forced disable) or the SD memory card is removed.
- The CPU module is powered off.
- The data memory is initialized.
- Free space of the SD memory card is less than 10MB.

2. Connecting to a database

Start an application (Microsoft Access, Excel, or the user-created application) on a personal computer and connect it to a database specified with the application. (☞ Page 282 Application example)

3. Operating a database

Operate an application on a personal computer to operate a database. (☞ Page 282 Application example)

Restriction

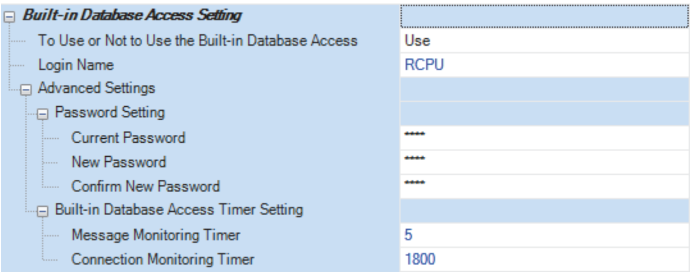
The SD memory card is accessed when the ODBC server is started up. This may lengthen the processing time of other functions that use the SD memory card.

Built-in database access setting

To use the CPU module database access function, set "To Use or Not to Use the Built-in Database Access" of the module parameter to "Use" with the engineering tool.

[Navigation window] ⇨ [Parameter] ⇨ CPU module ⇨ [Module Parameter] ⇨ [Application Settings] ⇨ [Built-in Database Access Setting]

Window



Displayed items

Item			Description	Setting range	Default
To Use or Not to Use the Built-in Database Access			Set whether to use the CPU module database access function.	• Use • Not Use	Not Use
Login Name			A user name to be used for the authentication at the access to a database	1 to 31 characters*1*2	RCPU
Advanced Settings	Password Setting	Current Password	A password to be used for the authentication at the access to a database	0 to 31 characters*1	RCPU
		New Password	A new password to be set	0 to 31 characters*1	—
		Confirm New Password	A password for confirmation. The password can be changed only when this password matches a new password.	0 to 31 characters*1	—
	Built-in Database Access Timer Setting	Message Monitoring Timer	A monitoring timer for until the first message is sent from the personal computer after the connection establishment	1 to 16383s	5
		Connection Monitoring Timer	A monitoring timer for until the CPU module disconnects the personal computer after the CPU module recognizes that the personal computer stops accessing to the CPU module	1 to 16383s	1800

*1 Single-byte alphabetical characters and special characters (excluding " # % * + , / : ; < = > ? [\ ] | ' { } & ~ $ @ ^) can be input.
*2 A space cannot be used as the first character.

Creating a database

This section describes the procedure for creating a database to be used with the CPU module database access function. A database can be created with the following methods.

- Storing the database folder
- DBIMPORT(P) instruction (a database access instruction)

Storing the database folder

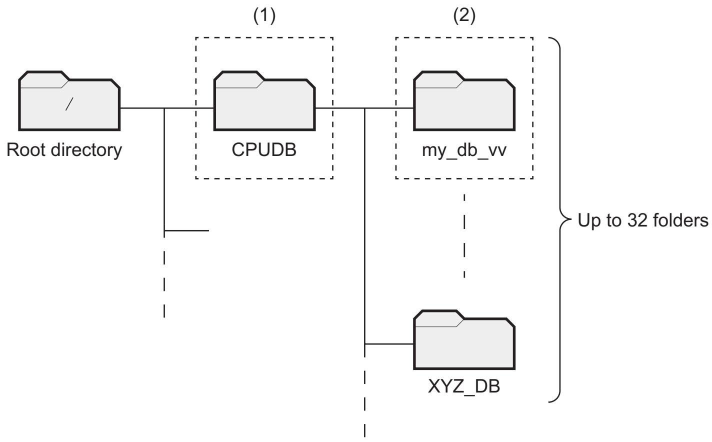
Store the database folder in an SD memory card.

1. For the database folder, please consult your local Mitsubishi representative.

2. Create the CPUDB folder directly under the root directory of the SD memory card from the personal computer. (1)

3. Store the database folder (default database name: my_db_vv*1) in the CPUDB folder. (2)

*1 "vv" indicates the version of the database folder.



- The name of the database folder stored is recognized as the database name. The maximum number of characters in the folder path to the database folder is 128. The characters that can be used for a database name is the same as the Unicode text file specified by using the DBIMPORT(P) instruction. (☞ Page 271 Setting details of Unicode text file)
- Only the CPUDB folder directly under the root directory can be used as the storage location of the database. In the CPUDB folder, store database folders only.
- Up to 32 folders can be stored in the CPUDB folder.
- If data other than the database folder or 33 database folders or more is stored in the CPUDB folder, starting up of the ODBC server is failed and an error code is stored in SD1498 (Start-up status of CPU module database).

■Precautions

Do not change, delete, or create a file and folder in the database folder from the personal computer. Otherwise, the CPU module database access function and database access instructions are completed with an error. If the above mentioned operations are performed from the personal computer, delete the database folder in the CPU module and create the new one.

DBIMPORT(P) instruction (a database access instruction)

Use the DBIMPORT(P) instruction and create a database in an SD memory card according to the setting of the Unicode text file.

- For details on the Unicode text file, refer to the following.

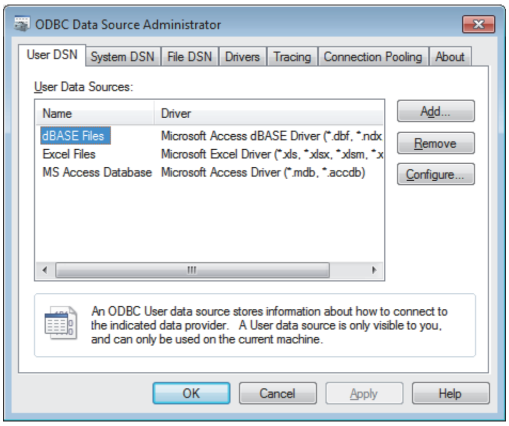
☞ Page 271 Setting details of Unicode text file

- For details on the DBIMPORT(P) instruction, refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

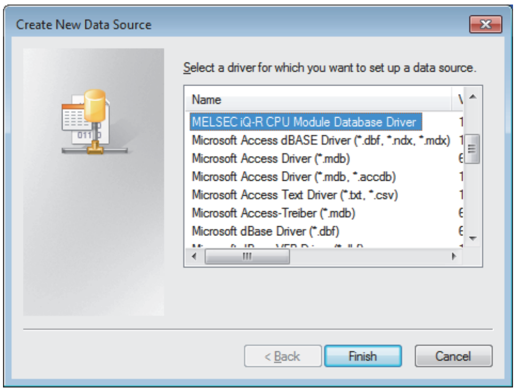
Adding a database to the ODBC data source

This section describes the procedure for adding a created database of the CPU module to the ODBC data source in the personal computer. (Examples of Windows 7 are shown below. Names of windows and menus may differ depending on the version of the OS.)

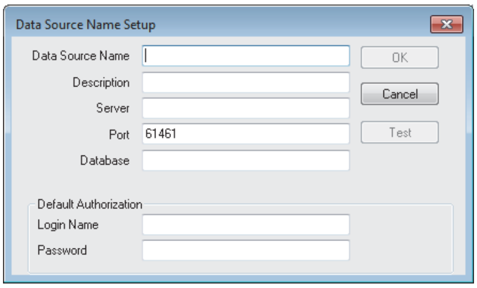


1. Select [Start] ⇨ [Control Panel] ⇨ [System and Security] ⇨ [Administrative Tools] ⇨ [Data Sources (ODBC)] on the personal computer and start the ODBC Data Source Administrator.

2. Click [User DSN] ⇨ the [Add] button to display the "Create New Data Source" window.



3. Select "MELSEC iQ-R CPU Module Database Driver", and click the [Finish] button.



4. Set the database of the CPU module on the "Data Source Name Setup" window. (☞ Page 281 Data Source Name Setup)

Point

When the 64-bit OS is used, the ODBC Data Source Administrator of a 32-bit or 64-bit version can be used. Choose the appropriate one depending on the supported bit of the application used with the CPU module database access function.

For the 64-bit OS, the ODBC Data Source Administrator can be started from the following. (When the start-up drive is C)

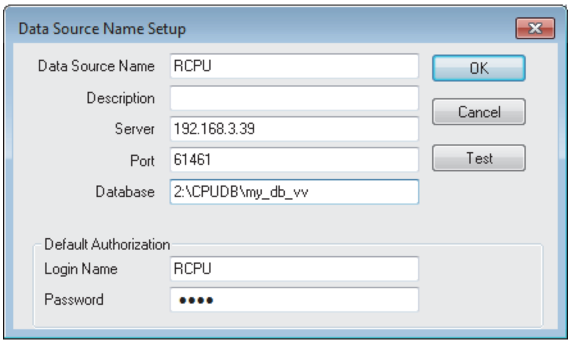
64-bit version: C:\WINDOWS\System32\odbcad32.exe

32-bit version: C:\WINDOWS\SysWOW64\odbcad32.exe

Data Source Name Setup

Set the database of the CPU module connected from the personal computer.

Window



Displayed items

Item		Description
Data Source Name		Set the identification name (any character string) for specifying the connection target database from the application. Input 1 to 32 characters in single-byte or double-byte alphanumeric characters. Characters except for * ! ( ) = \ ? , [ ] can be used.
Description		Input the comment. Input none, up to 1023 single-byte alphanumeric characters, or up to 512 double-byte characters.
Server		Set the IP address of the CPU module to be connected.
Port		Set 61461. (Default setting is 61461. Do not change the value.)
Database		Set the absolute path of the database to be connected. For example, input the following for the database folder, my_db_vv, stored in the CPUDB folder directly under the root directory of the SD memory card. 2:\CPUDB\my_db_vv ("2:\" indicates the root directory of the SD memory card. "vv" indicates the version of the database folder.)
Default Authorization	Login Name*1	Set the login user to access the database which is set. Input the user name which is set in the module parameter. (☞ Page 278 Built-in database access setting)
	Password*1	Input the password which is set in the module parameter. (☞ Page 278 Built-in database access setting)

*1 If they are not set, input them at login to connect to the database.

Point

Click the [Test] button to perform a connection test of the database of the CPU module in accordance with the setting.

Application example

This section describes the examples of database operations of the CPU module from applications in the personal computer using the CPU module database access function. (Names of windows and menus may differ depending on the version of the OS and application.)

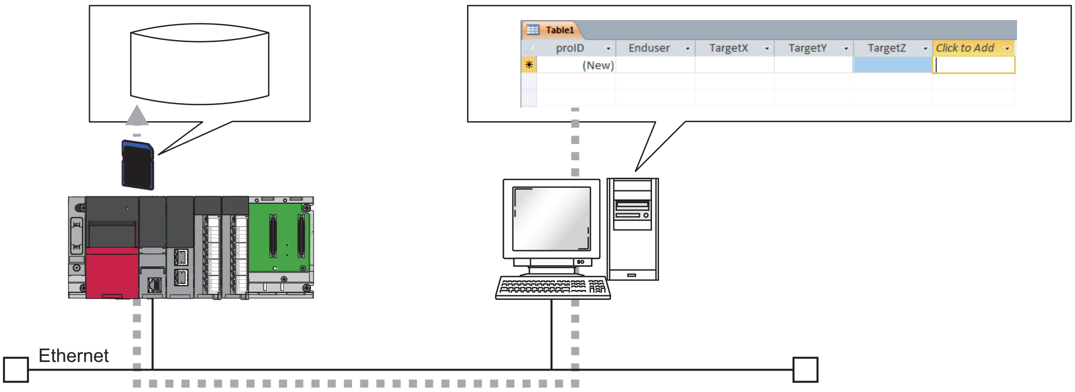
Item	Description	Application	Reference
Table creation	Create a table in a database of the CPU module from the personal computer.	Microsoft Access	Page 282 Creating a table by using an application
Record operation	Edit a database of the CPU module from the personal computer.	Microsoft Access	Page 285 Record operation from an application
Record search	Search a database of the CPU module from the personal computer.	Excel	Page 287 Record search from an application
Database operation with the user-created application	Create a database access application using the ODBC classes of Microsoft .NET Framework.	User-created application	Page 289 User-created application

The following table lists the applications whose connection to the database has been checked.

Application	Version
Microsoft Access	Microsoft Office 2010, 2013, 2016
Excel	
User-created application	Application created by using Microsoft .NET Framework 4.5

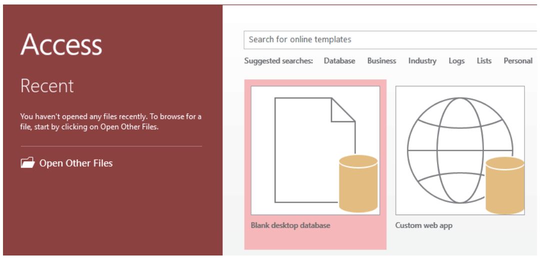
Creating a table by using an application

The following provides an example to create a recipe table in a database of the CPU module by using Microsoft Access.

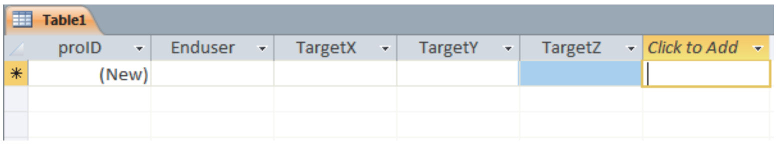


13

1. Start Microsoft Access and select "Blank database".



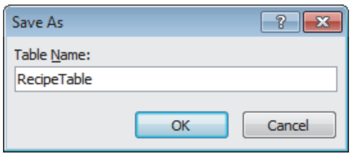
2. Add the following fields.



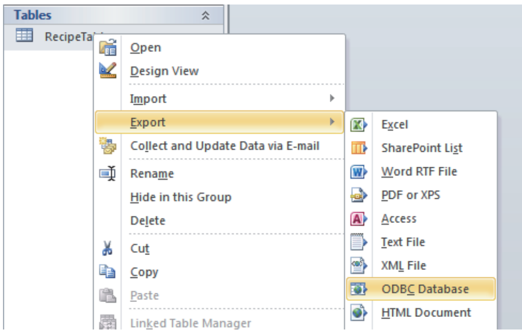
Field name	Data type	Data type for when the data is exported to the database of the CPU module
proID*1	AutoNumber type: Long integer	INT
Enduser	Numeric type: Long integer	INT
TargetX	Numeric type: Single-precision floating point type	REAL
TargetY	Numeric type: Single-precision floating point type	REAL
TargetZ	Numeric type: Single-precision floating point type	REAL

*1 Rename the ID field.

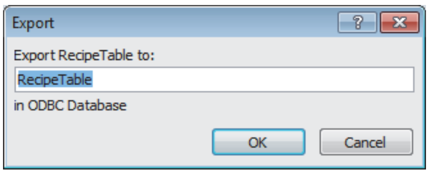
3. Name the table "RecipeTable" and save it. After saving, close the table.

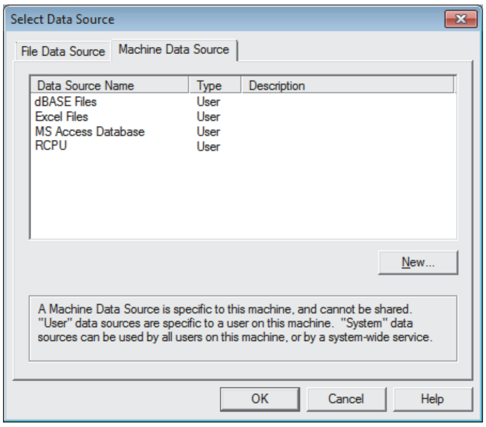


4. Right-click on RecipeTable, and select [Export] ⇨ [ODBC Database].



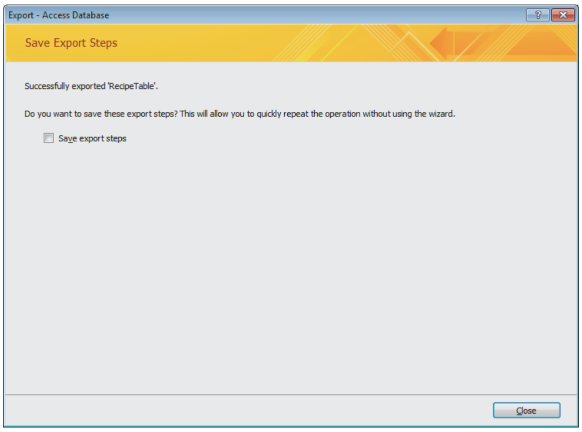
5. When the window to specify the export destination is displayed, click the [OK] button.





6. On the window for selecting the data source, click the [Machine Data Source] tab and select the data source of the CPU module.

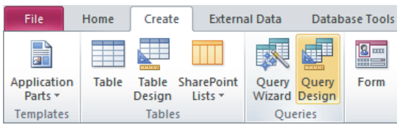
- When the default user is not set to the ODBC data source, input the user name and password.
- When no data source of the CPU module is displayed in the window, add a database of the CPU module to the ODBC data source. (☞ Page 280 Adding a database to the ODBC data source)



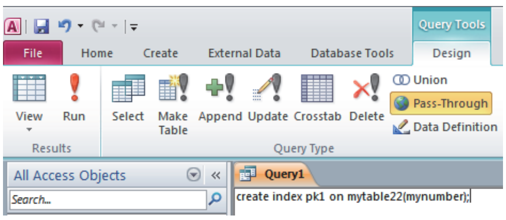
7. The result of export is displayed, and the table is added to the database of the CPU module.

Point

To set the details that cannot be set with Microsoft Access, send desired SQL commands to the database of the CPU module.



Clicking the [Create] tab ⇨ [Query Design] of Microsoft Access displays the "Show Table" window, so click the [Close] button. The [Query Tools] tab becomes enabled. Select [Pass-Through].



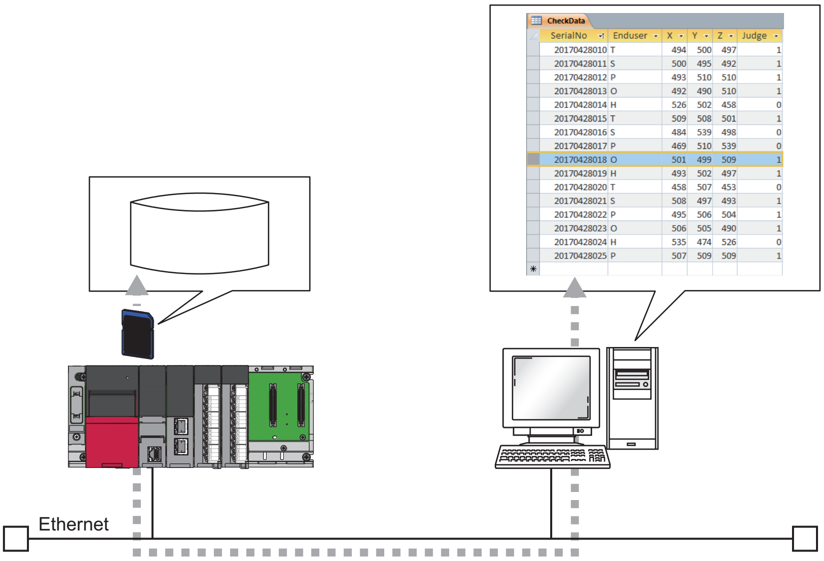
Selecting [Pass-Through] enables the query tab and allows input of SQL commands to be sent to the database of the CPU module directly.

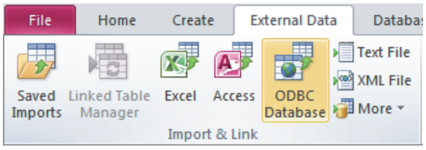
For available SQL commands for the CPU module database access function, refer to the following.

☞ Page 1127 List of Available SQL Commands for CPU Module Database Access Function

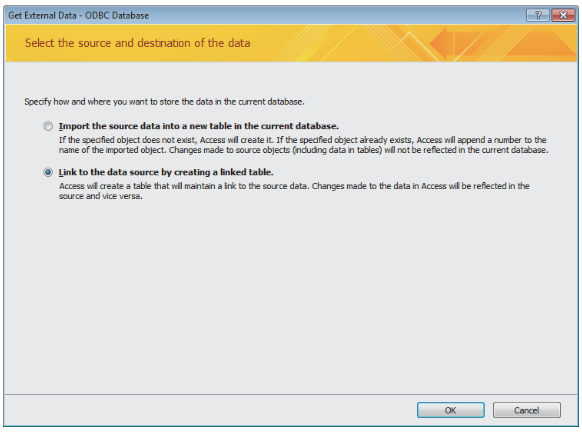
Record operation from an application

The following provides an example of record operation such as synchronizing, writing, and deleting of the data by connecting to the database of the CPU module by using Microsoft Access.

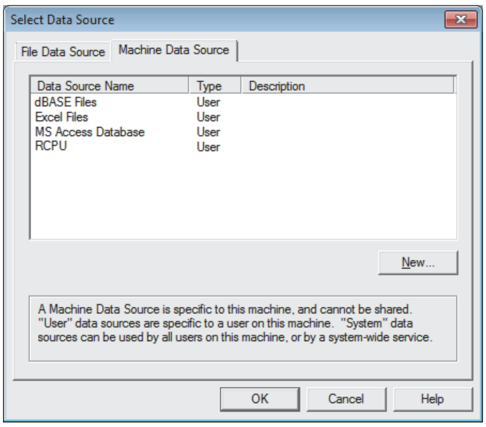




1. Start Microsoft Access and select the [External Data] tab ⇨ [ODBC Database].



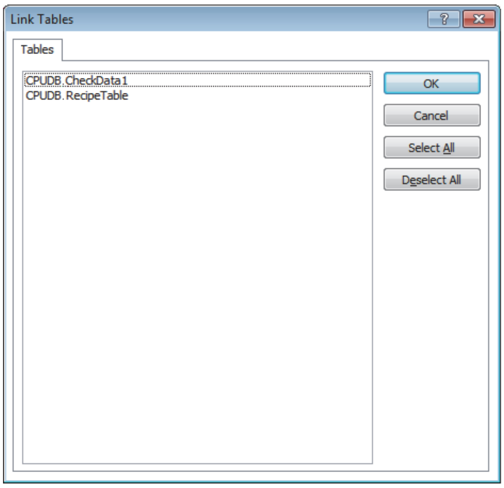
2. When the window to select the method for storing the data in the database, select "Link to the data source by creating a linked table", and click the [OK] button.



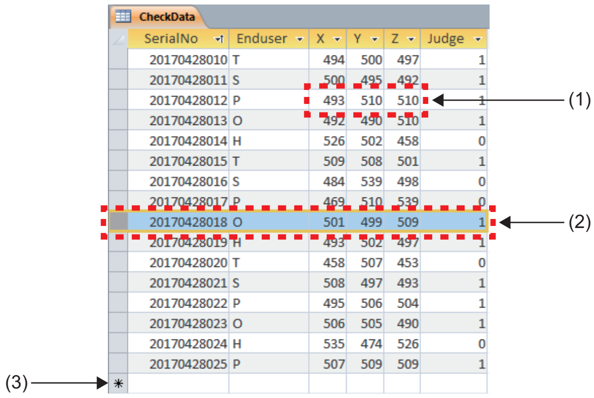
3. On the window for selecting the data source, click the [Machine Data Source] tab and select the data source of the CPU module.

- When the default user is not set to the ODBC data source, input the user name and password.
- When no data source of the CPU module is displayed in the window, add a database of the CPU module to the ODBC data source. (☞ Page 280 Adding a database to the ODBC data source)

4. On the window for selecting table, select a table to be operated and click the [OK] button to display the contents of the table.



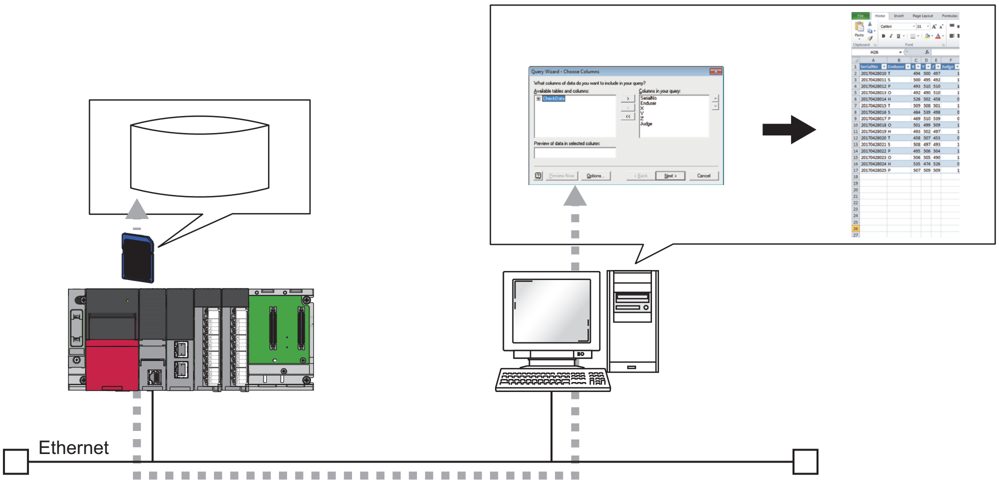
5. When the value is changed or the record is deleted, the database of the CPU module is changed according to the operation.



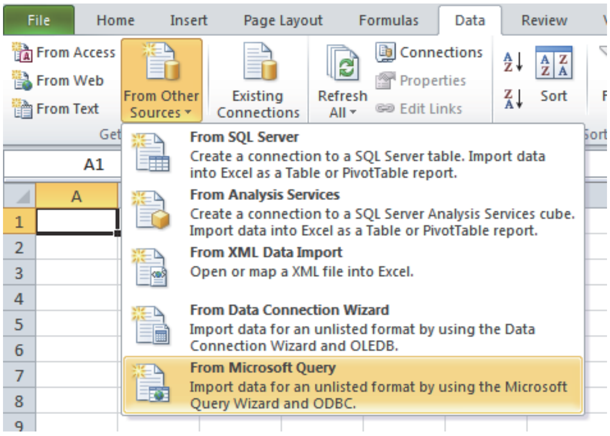
(1) Select a cell to change the value.
(2) Select a row to delete the record.
(3) Add a new record.

Record search from an application

The following provides an example to search the record that matches the specified condition from the production data stored in the database by connecting to the database of the CPU module by using Excel.

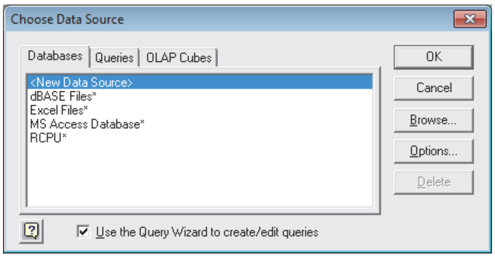


1. Start Excel, and select the [Data] tab ⇨ [From Other Sources] ⇨ [From Microsoft Query].

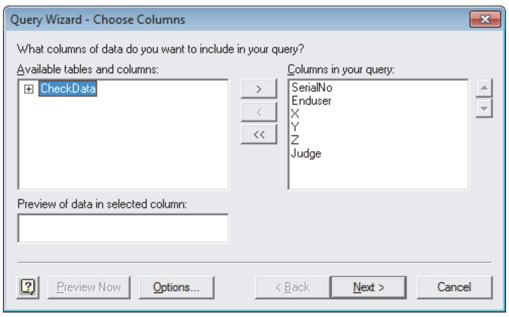


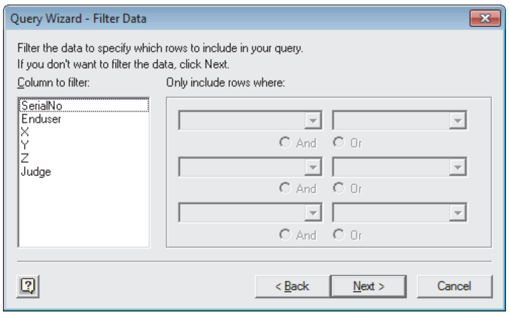
2. On the window for selecting the data source, select the data source of the CPU module.

- When the default user is not set to the ODBC data source, input the user name and password.
- When no data source of the CPU module is displayed in the window, add a database of the CPU module to the ODBC data source. (☞ Page 280 Adding a database to the ODBC data source)

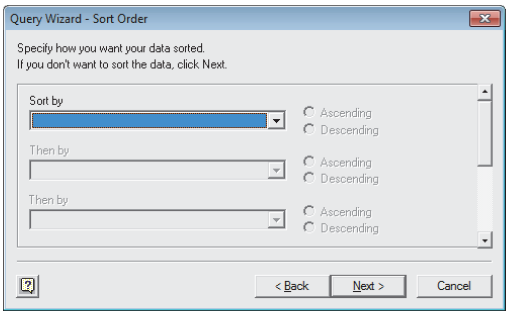


3. On the window for selecting columns of the query wizard, select the field to be output.

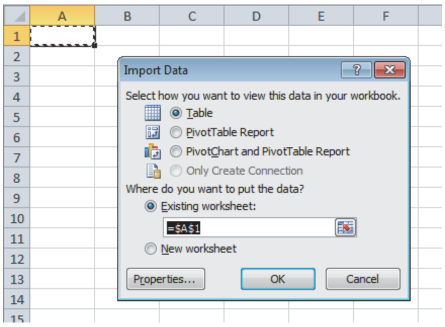




4. On the window for filtering data of the query wizard, set the search condition.



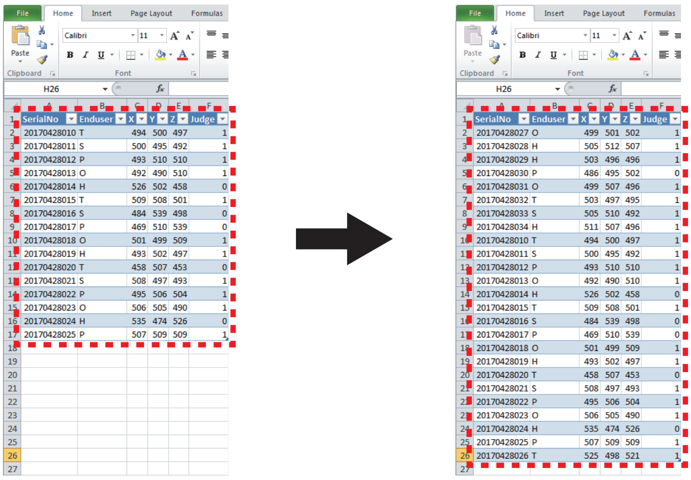
5. On the window for setting the sort order of the query wizard, set the condition to sort the output data.



6. Set the output destination. Then, the search result is output.

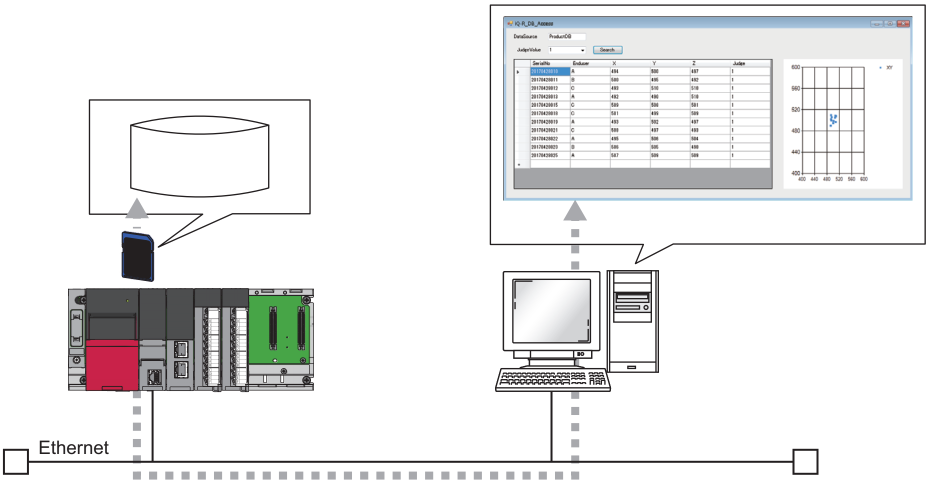
Point

When the file used in the above procedure is saved, the search condition is also saved.
Click the [Data] tab ⇨ [Refresh All] in Excel to acquire the data in the same condition from the database of the CPU module once more.



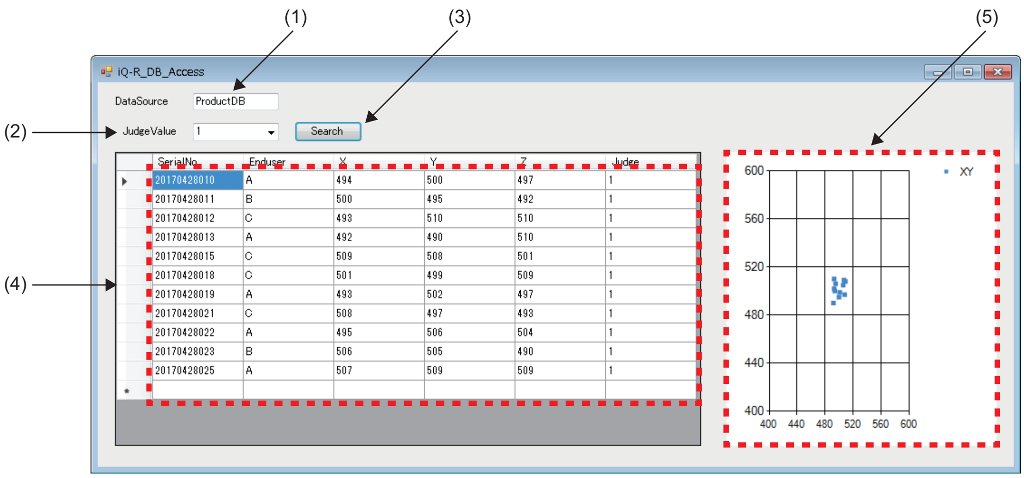
User-created application

The following describes a sample program for accessing the database of the CPU module with the SQL command using the ODBC class of Microsoft .NET Framework.



■Overview of the sample program

This sample program performs the condition search in the database of the CPU module using the value of the Judge field as its key, and displays the acquired search result in a list and graph (scatter plot).



(1) Specify the database source name to be connected.
(2) Specify the Judge field value to be searched.
(3) Click the [Search] button to search the record with the same value as the one specified by "JudgeValue" in the database specified by "DataSource".
(4) The acquired record is displayed in the list.
(5) The X value and Y value of the acquired record are displayed in a graph (scatter plot).

■Database configuration of the sample program

The following describes the database configuration to be searched by the sample program.

Item	Description
Server name (IP address of CPU module)	192.168.3.39
Database folder path	2:\Database\SampleDB
Database name	SampleDB
Table name to be searched	CheckData

The following table lists the records of CheckData table.

SerialNo	Enduser	X	Y	Z	Judge
20170428010	A	494	500	497	1
20170428011	B	500	495	492	1
20170428012	C	493	510	510	1
20170428013	A	492	490	510	1
20170428014	B	526	502	458	0
20170428015	C	509	508	501	1
20170428016	A	484	539	498	0
20170428017	B	469	510	539	0
20170428018	C	501	499	509	1
20170428019	A	493	502	497	1
20170428020	B	458	507	453	0
20170428021	C	508	497	493	1
20170428022	A	495	506	504	1
20170428023	B	506	505	490	1
20170428024	C	535	474	526	0
20170428025	A	507	509	509	1

■Source code of the sample program

The following describes the source code of the sample program.

- Development environment: Visual Studio 2015
- Programming language: C#

```
namespace iQ_R_DB_Access
{
    public partial class FrmMain : Form
    {
        public FrmMain()
        {
            InitializeComponent();

            // Range setting of X-/Y-axis on a graph
            chart1.ChartAreas[0].AxisX.Minimum = 400;
            chart1.ChartAreas[0].AxisX.Maximum = 600;
            chart1.ChartAreas[0].AxisY.Minimum = 400;
            chart1.ChartAreas[0].AxisY.Maximum = 600;

            // Initial value (all) set to the judge value of search target
            cmbJudge.SelectedIndex = 2;
        }

        /// <summary>
        /// Operation when the Search button is clicked
        /// Records that match the judge value specified by the combo box are acquired and displayed in a list and a scatter plot of X and Y field values.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnSearch_Click(object sender, EventArgs e)
        {
            // The number of fields (the number of fields on the CheckData table)
            int len = 6;

            // Acquiring the data source name
            string strConnect = txtDataSourceName.Text.ToString();
            // ODBC connection information management class
            OdbcConnectionStringBuilder o = new OdbcConnectionStringBuilder();
            o.Dsn = strConnect;
            // Creating the connection object
            OdbcConnection cn = new OdbcConnection(o.ConnectionString);

            // SQL statement
            string strQuery = GetSQL();
            // Creating the command object
            OdbcCommand cmd = new OdbcCommand(strQuery, cn);


            // Initializing the list
            dtRecord.Rows.Clear();
            // Initializing the graph
            chart1.Series[0].Points.Clear();

            try
            {
                // Connecting to DB
                cn.Open();

                // Creating the data reader object
                OdbcDataReader reader = cmd.ExecuteReader();
```

13

```
            // Processing the search results one record at a time
            for (int recordnum = 0; reader.Read(); recordnum++)
            {
                // Adding a blank row to the list
                dtRecord.Rows.Add();

                Setting the number of rows to be inserted into the list
                recordnum = dtRecord.Rows.Count-2;
                Storing acquired records into the list one field at a time
                for (int i = 0; i < len; i++)
                {
                    // Adding search results into the list
                    dtRecord.Rows[recordnum].Cells[i].Value = reader.GetValue(i);
                }

                // Plotting values of X and Y fields on the graph
                chart1.Series[0].Points.AddXY(Convert.ToInt32(dtRecord.Rows[recordnum].Cells[2].Value),
                                    Convert.ToInt32(dtRecord.Rows[recordnum].Cells[3].Value));
            }
        }
        catch (OdbcException ex)
        {
            MessageBox.Show(ex.ToString());
        }
        finally
        {
            // Disconnecting from DB
            cn.Close();
        }
    }

    /// <summary>
    /// Generating the SQL statement from the value in the combo box
    /// </summary>
    public string GetSQL()
    {
        string strSQL = "";

        switch (cmbJudge.SelectedIndex)
        {
            case 0:
                // The search target is records of Judge = 0.
                strSQL = "SELECT SerialNo, Enduser, X, Y, Z, Judge FROM CheckData WHERE Judge=0";
                break;
            case 1:
                // The search target is records of Judge = 1.
                strSQL = "SELECT SerialNo, Enduser, X, Y, Z, Judge FROM CheckData WHERE Judge=1";
                break;
            case 2:
                // Acquiring all the records
                strSQL = "SELECT SerialNo, Enduser, X, Y, Z, Judge FROM CheckData";
                break;

            default:
                strSQL = "SELECT * FROM CheckData";
                break;
        }
        return strSQL;
    }
  }
}
```

Precautions

This section describes the precautions for using the CPU module database access function.

Database creation

When creating a database in the CPUDB folder, use characters that can be specified only. If the characters that cannot be specified are used to create the database, the table cannot be accessed by the database access instructions.
For the characters that can be specified, refer to the following.
☞ Page 271 Setting details of Unicode text file

Combination of the firmware version and database

The following table lists the combination of the firmware version of the programmable controller CPU that creates the database and that of the programmable controller CPU where the operation of the database is available.
○: Available, ×: Not available

Database	Availability	
	The programmable controller CPU with firmware version earlier than "28"	The programmable controller CPU with firmware version "28" or later
Database created with the programmable controller CPU with firmware version earlier than "28"	○	○*1
Database created with the programmable controller CPU with firmware version "28" or later	×	○

*1 If the database that has been operated with the programmable controller CPU with firmware version "28" or later is operated with the programmable controller CPU with firmware version earlier than "28", the operation is completed with an error.

Number of connections that can be operated simultaneously

Up to four connections can be operated with the CPU module database access function simultaneously.
However, if an application such as Microsoft Access uses several connections, four applications may not connect with the database of the CPU module simultaneously. If an attempt was made to connect with the database with all the connections used, an error occurs in the application and the database cannot be connected. In this case, end unused applications connected with the database and connect again.

Several simultaneous accesses

When several applications access the same database in the CPU module, the database processing becomes slow.

■Access to the same table

When the database access instructions or the CPU module database access function access the same table, the function executed first locks the table, and an error may occur in the function executed later. For the database access instruction, an error code is stored in the completion status.

■Definition change of the same database

When the database definition is changed with "CREATE TABLE" or "DROP TABLE" in a single database from several applications using the CPU module database access function, the application executed first locks the database and an error may occur in the function executed later.

Completion with an error during database access

Do not power off or reset the CPU module during the access to a database of the CPU module. Otherwise, the change is not reflected on the database that is being executed.

Files created with this function

Do not create, change, and delete the ODBC server setting file (netserver.cfg), database path file (dbmaintainpath.txt), and error database check file (ErrorDB.txt) created with the CPU module database access function. Otherwise, the CPU module database access function or database access instruction is completed with an error.
If the above mentioned operations are performed, delete the database folder in the CPU module and create the new one.

Removal of the SD memory card

To remove the SD memory card while the ODBC server is active, execute the SD memory card forced disable and check that CARD READY LED is off before removing the SD memory card. (Failure to do so may result in a failure of databases.)
Since databases stop after the execution of the SD memory card forced disable, databases of the CPU module cannot be accessed. (The error code is stored in SD1498.)
To access the databases of the CPU module again, insert the SD memory card and power off and on or reset the CPU module.

Free space of the SD memory card

To use the CPU module database access function, the SD memory card must have at least 20MB of free space.
When the free space of the SD memory card becomes 20MB or less while the CPU module database access function is used, SM1497 (Memory card free space flag for CPU module database) turns on. When the free space of the SD memory card becomes 10MB or less, the ODBC server of the CPU module stops and the error code is stored in SD1498 (Start-up status of CPU module database).
When the ODBC server stops because the SD memory card does not have enough free space, secure the free space of the SD memory card using the online operation and power off and on or reset the CPU module. Powering off and on or resetting the CPU module starts the ODBC server.

When the SD memory card is write-protected

To use the CPU module database access function, the user must ensure that write protection of the SD memory card is off.
When the SD memory card is write-protected, the CPU module database access function cannot update the database and the error code is stored in SD1498.

When the load of the CPU module is high

When the CPU module database access function is used with high circuit load to the Ethernet port of the CPU module by other functions or high access load to the SD memory card, a timeout error may occur in the application on the personal computer side. In this case, eliminate the cause of the error and execute this function.

Errors

■Errors detected by the CPU module

The CPU module starts the ODBC server based on the setting of the module parameter. When the CPU module fails to start the ODBC server, the error code is stored in SD1498 (Start-up status of CPU module database). For the error code to be stored, refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

■Errors detected by the personal computer

When an error occurs during the access to the database of the CPU module, the CPU module returns the error message to the personal computer. In this case, check the error message displayed on the personal computer and eliminate the cause of the error. The error causes are as follows.

- The built-in database access setting of the module parameter is disabled.
- The ODBC server of the CPU module is not started. (SM1498 is not on.)
- The communication between the Ethernet port of the CPU module and the personal computer is not established.
- There is no database to be accessed.
- Multiple accesses are simultaneously performed to an access target database.
- A broken database is being accessed.
- The database is accessed when the circuit load of the Ethernet port of the CPU module or the access load to the SD memory card are high.

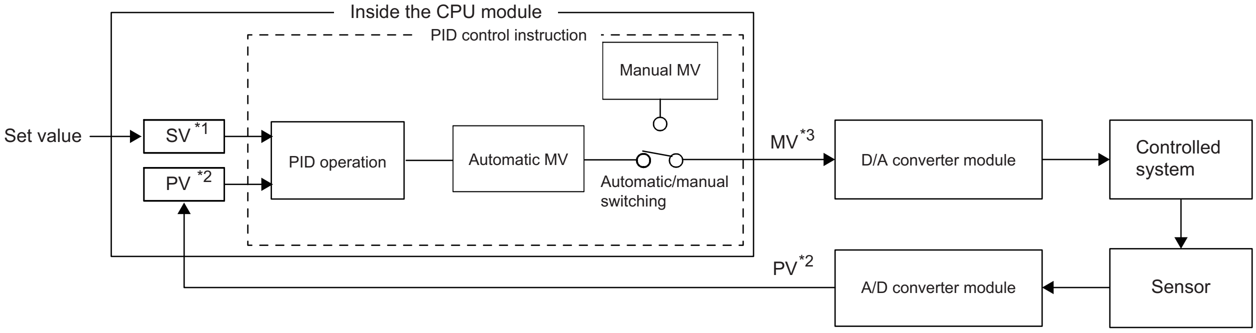
14 PID CONTROL/PROCESS CONTROL FUNCTION

This chapter describes the PID control/process control function.

14.1 PID Control Function

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

PID control is applicable to process control in which factors such as flow rate, velocity, air flow volume, temperature, tension, and mixing ratio must be controlled. The control for maintaining the control object at the preset value is shown in the diagram below: PID control via the PID control instructions is implemented by combining the CPU module with the A/D converter module and D/A converter module.



*1 SV: Set value
*2 PV: Process value
*3 MV: Manipulated value

In the PID control, the PID operation is executed to compare the value set beforehand (SV) with the digital value (process value (PV)) in which the analog value measured by a sensor is read from the A/D converter module.
PID operation utilizes proportional operation (P), integral operation (I), and derivative operation (D) in combination, thereby calculating the manipulated value (MV) quickly and accurately so that a process value (PV) is to be equivalent to a set value (SV). A larger difference between the process value (PV) and set value (SV) results in a faster speed to reach the set value (SV) quickly by increasing the manipulated value (MV) and a smaller difference between the process value (PV) and set value (SV) results in a slower speed to reach the set value (SV) accurately by decreasing the manipulated value (MV). The calculated manipulated value (MV) is written to the D/A converter module and output to the external device.

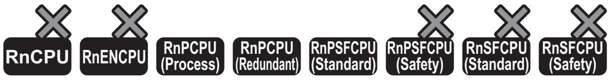
Point

For the PID control, use the PID operation instruction. To execute the same PID control as that of the MELSEC-Q series and MELSEC-L series, use the PID control instructions.
For details on the comparison between the PID operation instruction and PID control instructions, instruction specifications, PID control, and programming, refer to the following.
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

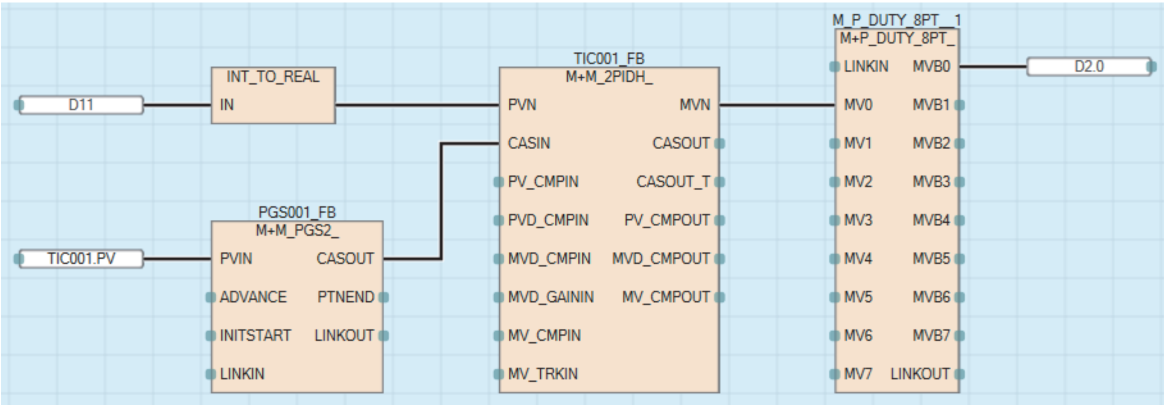
14.2 Process Control Function

This chapter describes the process control function.

Process control by using process control function blocks



A process control function block is the function block whose functions are extended for the process control. A process control program can be easily created by using process control function blocks.



Process control function blocks have features as follows.

- A program can be easily created by placing and connecting FB elements using various types of function block provided for the process control.
- Since the initial value of the function block can be set in the "FB Property" window of the engineering tool, the program for the initial value setting is not required.
- An argument of a function block can be specified using a label without considering the address of a device.
- The execution status of a tag FB can be checked and controlled by accessing the tag data from the faceplate of the engineering tool.

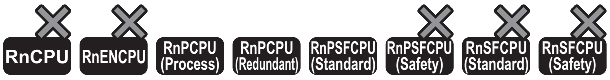
Point

For details on the process control function block, refer to the following.

MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions)

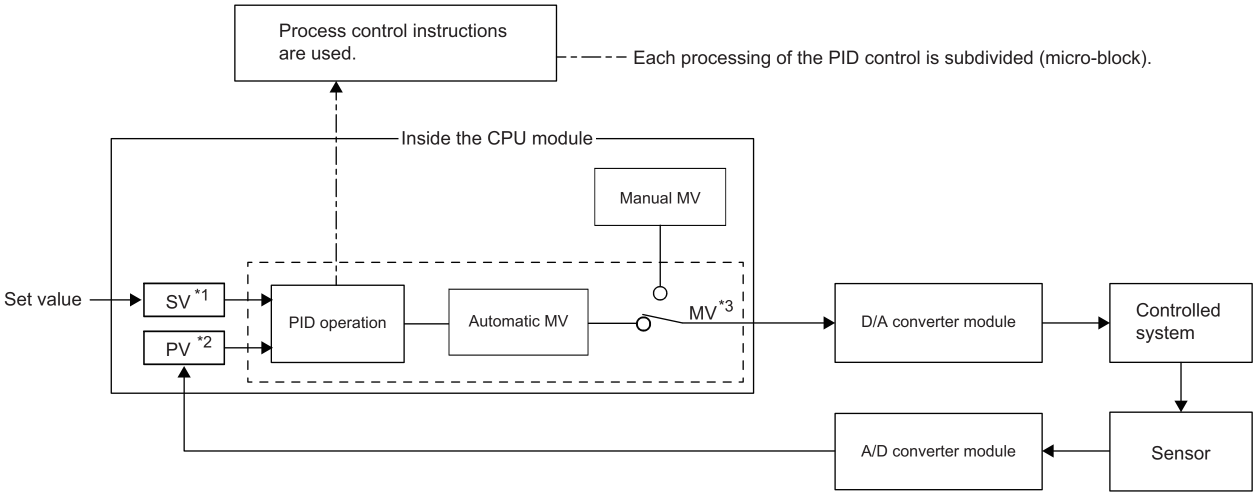
14

Process control by using process control instructions



This function performs various types of process control by using process control instructions that support loop control, such as two-degree-of-freedom PID control, sample PI, and auto tuning, in combination.

Since each processing of the PID control is subdivided and multiple process control instructions are used in combination, this function performs more accurate and sensitive control than the PID control function.



*1 SV: Set value

*2 PV: Process value

*3 MV: Manipulated value

Process control instructions have the following features:

- Increased efficiency of system adjustment: Multiple process control instructions are combined to perform PID control, and operation can be checked for each instruction individually.
- Application to a wide range of control: Another process control instruction can be added to the existing control loop, which consists of multiple process control instructions, as an option.
- Configuration of safety system: Alarms are automatically detected in the system.
- Auto tuning: The auto tuning instruction automatically calculates PID constants that are most suitable for the control system.

Point

- When the process control function is used, the process control by the process control function block is recommended. For the process control function block, refer to the following.

MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions)

- For the specifications of process control instructions, basic loop types, and programming details, refer to the following.

MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions)

15 CPU MODULE DATA BACKUP/RESTORATION FUNCTION

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- This function cannot be used in the R00CPU, R01CPU, and R02CPU.
- When using the Process CPU (redundant mode), refer to the following as well.

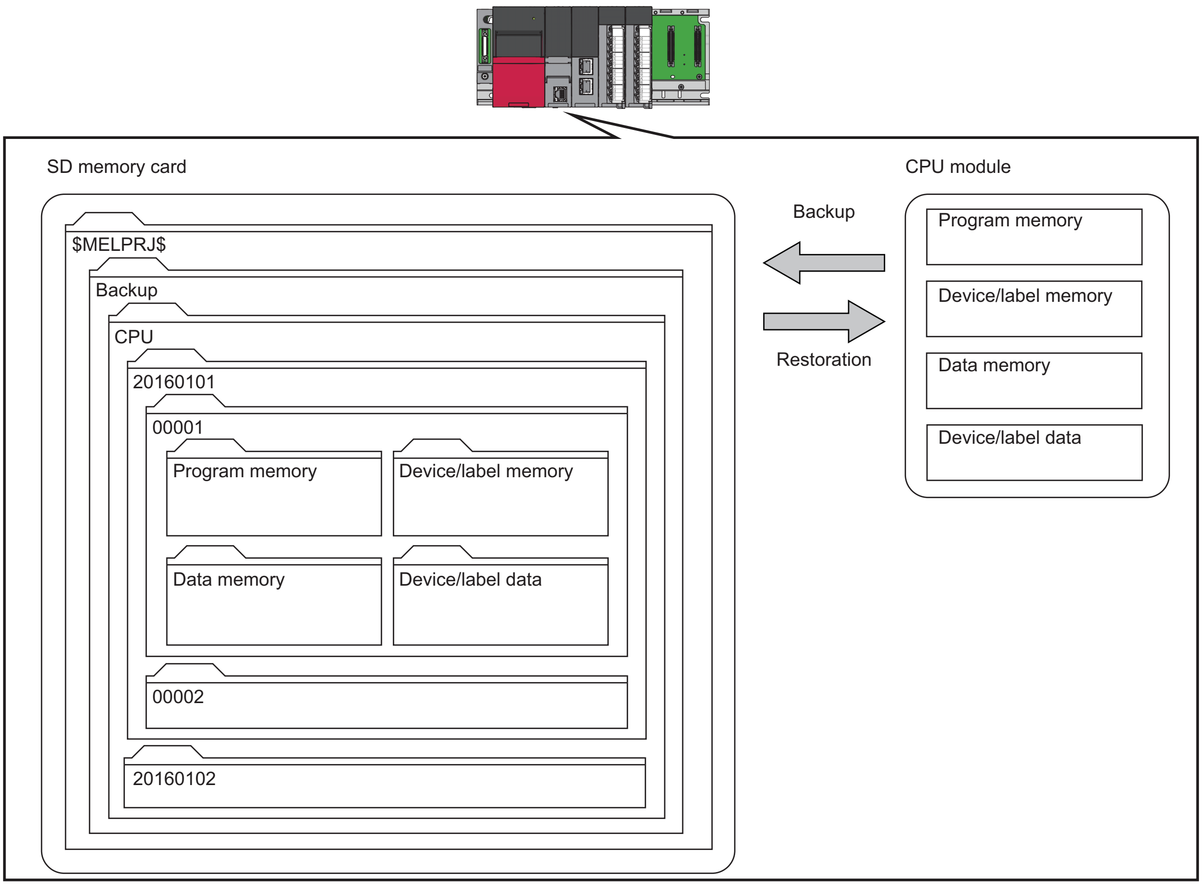
☞ Page 492 FUNCTIONS

- When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

This function backs up data such as program files, a parameter file, and device/label data*1 of a CPU module to an SD memory card. The data backed up in the SD memory card can be restored as required.

*1 Module access devices and buffer memory are excluded.



The following table lists the backup and restoration methods of the CPU module data backup/restoration function.

Function		Availability			Reference
		Programmable controller CPU	Process CPU	Safety CPU	
Backup function	Backup processing triggered by turning on SM1351	○	○	○	Page 308 Backup processing triggered by turning on SM1351
	Automatic backup using SD944	○	○	○	Page 309 Automatic backup using SD944
Restoration function	Restoration processing triggered by turning on SM1354	○	×	×	Page 318 Restoration processing triggered by turning on SM1354
	Automatic restoration using SD955	○	○	×	Page 319 Automatic restoration using SD955
	Restoration by enabling the automatic restoration setting	×	×	○	Page 645 Restoration function
	Automatic restoration with the SD CARD OFF button*2	×	○	×	Page 320 Automatic restoration with the SD CARD OFF button

*2 When using the automatic restoration with the SD CARD OFF button, set it before the backup processing. After setting, execute each backup processing. (☞ Page 311 Settings for automatic restoration with the SD CARD OFF button)

Point

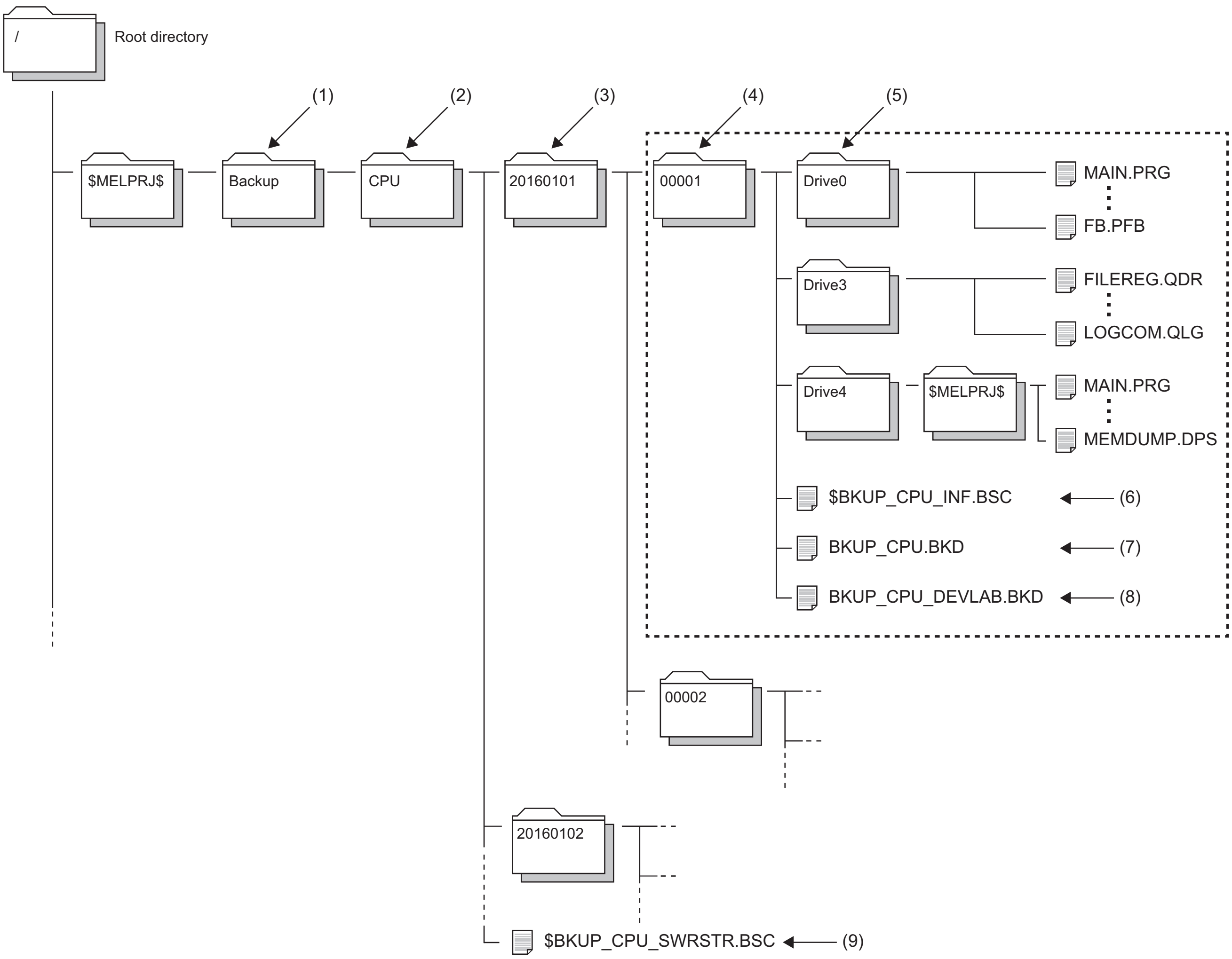
The restoration function modifies programs, parameters, or device/label data of the CPU module. After restoration, check the restored data carefully before an actual operation. (Check the data with an engineering tool.)

Restriction

When using the CPU module data backup/restoration function, check the versions of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

Backup data

Backup data is saved in an SD memory card. The following shows the folder structure of backup data.



No.	Folder type	Folder name	Number of storable folders	Description
(1)	Backup data folder	Backup (Fixed)	1	A folder for storing all backup data
(2)	CPU data folder	CPU (Fixed)	1	A folder for storing backup data of the CPU module
(3)	Date folder	Automatically determined*1 Folder name format: YYYYMMDD • YYYY: Year when the data was backed up (four digits) • MM: Month when the data was backed up (two digits) • DD: Day when the data was backed up (two digits)	Depends on the capacity of the SD memory card used*2 or 1 to 100 folders	Folders for storing backup data by date For the setting of the upper limit value for the number of CPU module backup data, the number of backup data indicates the number of date folders. (☞ Page 306 Setting the upper limit value for the number of CPU module backup data)
(4)	Number folder	Automatically determined*1 Folder name: Sequentially numbered from 00001 to 32767 (five digits)	Depends on the capacity of the SD memory card used*2	Folders for storing information per backup data. Each backup data created on a date is stored in sequentially numbered folders.
(5)	Drive folder	Drive0 (Fixed), Drive3 (Fixed), and Drive4 (Fixed)	One set of the folders in a number folder	Folders for storing folders/files stored in each drive of the backup target CPU module by each drive
(6)	System file for backing up CPU module data	$BKUP_CPU_INF.BSC	One folder in a number folder	Files for storing the information required at restoration of data, such as a list of backup data and identification information of the CPU module
(7)	Backup data file for backing up CPU module data	BKUP_CPU.BKD	One folder in a number folder	The following data is stored. • Data on operations of the data logging setting • Data for restarting the SFC program from the block and step where the processing was stopped

No.	Folder type	Folder name	Number of storable folders	Description
(8)	Device/label data file for backing up CPU module data	BKUP_CPU_DEVLAB.BKD	One folder in a number folder	Device/label data is stored.
(9)	System file for automatic restoration with the SD CARD OFF button	$BKUP_CPU_SWRSTR.BSC	One folder in a CPU folder*3	Setting information of automatic restoration with the SD CARD OFF button is stored.

*1 Date folders and number folders are automatically named by the CPU module.
*2 The maximum number of storable folders is 32767.
*3 When bit 2 of SD955 (Restoration function setting) is set to on at the backup processing, the file is generated if it is not in the CPU folder, and the setting information stored is updated if it is. (When bit 2 of SD955 is set to off at the backup processing, the file is not generated.)

Backup/restoration target data

Backup target data is all target data in the CPU module. (☞ Page 302 Backup/restoration target files)

Restoration target data is set with SD954 (Restoration target data setting). (☞ Page 316 Restoration target data)

■Backup/restoration target drives

Target drives are Drive0 (Program memory), Drive3 (Device/label memory), and Drive4 (Data memory).

Drive2 (SD memory card) is not a target drive.

■Backup/restoration target files

The following table lists backup/restoration target files.

○: Available, ×: Not available

File type		Backup/restoration
Program		○
FB file		○
CPU parameter		○
System parameter		○
Module parameter		○
Module extension parameter		○
Module-specific backup parameter		○
Memory card parameter		×
Device comment		○
Device initial value		○
Global label setting file		○
Initial label value file	Initial global label value file	○
	Initial local label value file	○
File register		○
Event history		○
Device data storage file		○
General-purpose data		○
Data logging setting file	Common setting file	×
	Individual setting file	○
Memory dump setting file		○
Remote password		○
System file for the iQ Sensor Solution data backup/restoration function		×
Backup data file for the iQ Sensor Solution data backup/restoration function		×
Device station parameter file		○
ODBC server setting file		○
Recording setting file		○

All folders/files in the CPU built-in memory are backup/restoration targets.

■Files created during backup

The following files are created during backup.

File type	File name	Extension
System file for backing up CPU module data	$BKUP_CPU_INF	BSC
Backup data file for backing up CPU module data	BKUP_CPU	BKD
Device/label data file for backing up CPU module data	BKUP_CPU_DEVLAB	BKD

The following are the sizes of files that are created at data backup.

File type	File size
System file for backing up CPU module data	40 + ((N1 × 34) + (N2 × 34) + (N3 × 34)) + M bytes · N1: Number of target drives · N2: Number of target files · N3: Number of target folders · M: Total name size of target files/folders (bytes) (including ".", extensions, but no drive symbols)
Backup data file for backing up CPU module data	• When the data logging setting has been registered: 30608 bytes • When the data logging setting has not been registered: 30200 bytes
Device/label data file for backing up CPU module data	1087398 + (number of programs × 142) + (number of file registers × 134) + S1 + S2*1 + S3 bytes*2 • S1: Total size of the set devices (global devices) = (N1 + N2 × 2) × 18 + ((G1 ÷ 16) + G2 + (G3 × 2) + (G4 + (G4 ÷ 16) × 2) + (G5 × 2) + (G5 ÷ 16) × 2) + (G6 × 8)) × 2 • N1: Number of device types used from M, L, B, F, SB, V, S, D, W, SW, U3E0\HG, U3E1\HG, U3E2\HG, U3E3\HG, Z, LZ, and RD • N2: Number of device types used from T, ST, C, LC, LT, and LST • G1: Total number of points of M, L, B, F, SB, V, and S • G2: Total number of points of D, W, SW, U3E0\HG, U3E1\HG, U3E2\HG, U3E3\HG, Z, and RD • G3: Number of points of LZ • G4: Total number of points of T, ST, and C • G5: Number of points of LC • G6: Total number of points of LT and LST • S2: Total size of the set local devices = 16 + number of programs × (8 + ((N3 + N4 × 2) × 18) + ((L1 ÷ 16) + L2 + (L3 × 2) + (L4 + (L4 ÷ 16) × 2) + ((L5 × 2) + (L5 ÷ 16) × 2) + (L6 × 8)) × 2 • N2: Number of device types used from T, ST, C, LC, LT, and LST • N3: Number of local device types used from T, ST, C, LC, LT, and LST • L1: Total number of points of the local devices M and V • L2: Total number of points of the local devices D and Z • L3: Number of points of the local device LZ • L4: Total number of points of the local devices T, ST, and C • L5: Number of points of the local device LC • L6: Total number of points of the local devices LT and LST • S3 = (Label area capacity (word) + latch label area capacity (word) + module label assignment area capacity (word)) × 2

*1 This is added only when local devices are used.
*2 The total file size is adjusted by the system so that the size will be a multiple of 4 bytes. (Up to 3 bytes are added.)

15

■The number of CPU module backup data that can be stored in an SD memory card

The number of CPU module backup data that can be stored in an SD memory card is 32767. This number is equal to the maximum number of storable folders.

The number of files that can be backed up and restored (the number of backup source data files) depends on the maximum number of files of each model or drive.

■Backup/restoration target device data

○: Available, ×: Not available

Classification	Device name	Symbol	Backup*5	Restoration*5
User device	Input	X	○	○
	Output	Y	○	○
	Internal relay	M	○*3	○*3
	Link relay	B	○	○
	Annunciator	F	○	○
	Link special relay	SB	○	○
	Edge relay	V	○*3	○*3
	Step relay	S	○	○
	Timer	T	○*3	○*3
	Retentive timer	ST	○*3	○*3
	Long timer	LT	○*3	○*3
	Long retentive timer	LST	○*3	○*3
	Counter	C	○*3	○*3
	Long counter	LC	○*3	○*3
	Data register	D	○*3	○*3
	Link register	W	○	○
	Link special register	SW	○	○
	Latch relay	L	○	○
System device	Function input	FX	○	×
	Function output	FY	○	×
	Function register	FD	○	×
	Special relay	SM	○	○*1*2
	Special register	SD	○	○*1*2
Link direct device	Link input	Jn\X	×	×
	Link output	Jn\Y	×	×
	Link relay	Jn\B	×	×
	Link special relay	Jn\SB	×	×
	Link register	Jn\W	×	×
	Link special register	Jn\SW	×	×
Module access device	Module access device	Un\G	×	×
CPU buffer memory access device	CPU buffer memory access device	U3En\G	○	○*2
		U3En\HG	○	○*2
Index register	Index register	Z	○*4	○*4
	Long index register	LZ	○*4	○*4
File register	File register	R/ZR	○	○
Refresh data register	Refresh data register	RD	○	○
Nesting	Nesting	N	×	×
Pointer	Pointer	P	×	×
	Interrupt pointer	I	×	×
Other devices	Network No. specification device	J	×	×
	I/O No. specification device	U	×	×
	SFC block device	BL	○	×
	SFC transition device	TR	×	×

*1 Whether to restore these areas can be set to the bit 14 of SD955 (Restoration function setting).
*2 Areas used by the system may be overwritten after restoration.
*3 Includes local devices.
*4 Includes the local index register.
*5 Device data may be overwritten depending on the mounting status (I/O refresh) of each module or the refresh settings.

■Backup/restoration target label data

○: Available, ×: Not available

Classification	Backup*2	Restoration*2
Global label (including module labels)	○	○*1
Global label with latch specified	○	○
Local label	○	○
Local label with latch specified	○	○

*1 For module labels, data may be overwritten to the write areas from a module to the CPU module when the refresh settings have been made.
*2 Device data may be overwritten depending on the mounting status (I/O refresh) of each module or the refresh settings.

Progress of the backup/restoration processing

The progress of the backup/restoration processing can be checked in SD1350 (Number of uncompleted folders/files of CPU module data backup/restoration) or SD1351 (Progression status of CPU module data backup/restoration). However, the progress of the automatic restoration cannot be checked with the special register.

Special register	Description
SD1350	Displays the number of remaining backup/restoration target folders and files. • When the backup/restoration processing is started, the total number of backup/restoration folders and files is stored. • When the backup/restoration processing is completed, 0 is stored.
SD1351	Displays the progress of the backup/restoration processing in percentage (0 to 100%).*1

*1 When program files are restored, the progress in SD1351 stops while data is being written to the program memory in the restoration processing because the data is transferred from the program cache memory to the program memory. The progress of data transfer to the program memory can be checked in SD629 (Program memory write (transfer) status).

15

15.1 Backup Function

This function backs up data of a specified CPU module in an SD memory card.

Point

The backup function operates even when the CPU module is in the RUN state.
When executing the backup function with the CPU module in the RUN state, do not change device/label data during execution of the function. Doing so may cause data inconsistency of the device/label data and the contents of the backup data may unintentionally change.

Setting the upper limit value for the number of CPU module backup data

When the backup processing has not been executed (when no backup data folder (CPU data folder) exists in the SD memory card), the upper limit value for the number of CPU module backup data can be set.
The number of CPU module backup data in the upper limit value setting is the number of date folders.
The upper limit value for the number of CPU module backup data is enabled by turning on the bit 5 of SD944 (Enable the upper limit value for the number of CPU module backup data). The set value can be checked in SD960 (Upper limit status for the number of CPU module backup data).
When the bit 5 of SD944 is turned off, no upper limit value setting is applied.

Special relay/Special register	Description
SM960	This relay specifies an operation of backup when the number of CPU module backup data reaches the upper limit value. (This relay is valid only when the bit 5 of SD944 is on.) • Off: The oldest date folder is deleted, and the backup operation continues. • On: The backup operation does not continue if the upper limit value for the number of CPU module backup data is exceeded. (In this case, the backup is completed with an error.)
Bit 5 of SD944	Set to enable or disable the upper limit value for the number of CPU module backup data. • Off: Disable (No limit (Date folders are created within the maximum capacity of the SD memory card.)) • On: Enable
SD960	This register displays the value (1 to 100) set in SD1353. If the bit 5 of SD944 is off, 0 is stored in this special register (SD).
SD1353	Set the upper limit value (1 to 100) for the number of CPU module backup data.

Restriction

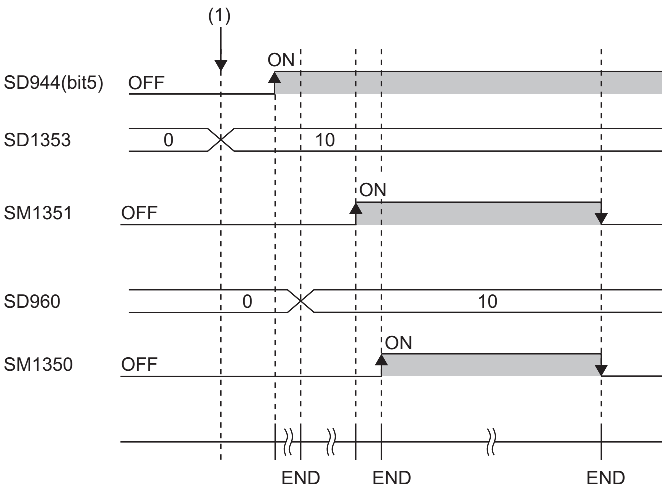
If data folders are created more than the capacity of the SD memory card before the number of them reaches the upper limit value for the number of CPU module backup data, the backup processing is completed with an error and the data cannot be backed up to reach the upper limit value.

■Operation of the special relay and special register

The following figure shows the operations of the special relay and special register of when the upper limit value for the number of CPU module backup data has been set.

Check the following at the timing on when the bit 5 of SD944 (Enable the upper limit value for the number of CPU module backup data) is turned on, and enable the upper limit value for the number of CPU module backup data.

- The backup processing has not been executed (when no backup data folder (CPU data folder) exists in the SD memory card).
- The value set in SD1353 (Upper limit value setting for the number of CPU module backup data) is within the range (1 to 100).



(1) The upper limit value for the number of CPU module backup data is set. (0 → 10)

15

Backup processing triggered by turning on SM1351

Data in the CPU module is backed up at a desired timing.

Operating procedure

Data in the CPU module is backed up by turning on SM1351.

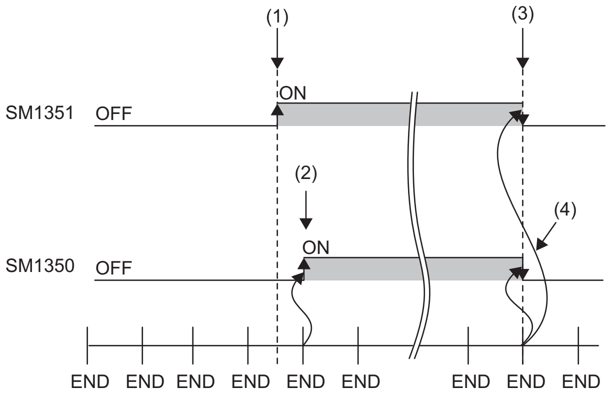
1. To set the upper limit value for the number of CPU module backup data, follow the steps below.

- Set SD1353 (Upper limit value for the number of CPU module backup data).
- Set SM960 (Upper limit operation setting flag for the number of CPU module backup data).
- Turn on the bit 5 of SD944 (Backup function setting).

For the upper limit value for the number of CPU module backup data, refer to the following.

☞ Page 306 Setting the upper limit value for the number of CPU module backup data

2. Turn on SM1351 (CPU module data backup execution request).



(1) A backup execution request is sent.
(2) The system turns on SM1350 (CPU module data backup status flag).
(3) The system turns off SM1351 after the backup processing is completed.
(4) The system turns off SM1350.

If the backup processing is completed with an error and SM953 (CPU module data backup error check flag) turns on, check SD953 (Backup error cause), take actions, and then back up the data again as required.

Point

The execution status of the backup processing can be checked in SD1350 (Number of uncompleted folders/files of CPU module data backup/restoration) and SD1351 (Progression status of CPU module data backup/restoration). (☞ Page 305 Progress of the backup/restoration processing)

Automatic backup using SD944

Data in the CPU module can be automatically backed up at a preset execution timing. Set the execution timing of the automatic backup with SD944 (Backup function setting). Multiple execution timing settings can be set.

Bit pattern of SD944	Execution timing
Bit 0: On	On the time set in SD948 and SD949 on the day set in SD947
Bit 1: On	On the time set in SD950 and SD951 on the day of the week set to SD952
Bit 15: On	When a stop error has occurred in the CPU module

Point

Since the special register set for the automatic backup is a latch area, setting data is held.

Retrying the automatic backup

Set whether to retry the automatic backup when the automatic backup is executed during execution of an operation or function that cannot be executed (☞ Page 314 Operations and functions that cannot be performed).

The retry interval is three minutes and the number of retries is 10.

Special relay/Special register	Description
SM961	This relay turns on when the retry of the automatic backup is completed with an error after the retry of the automatic backup for the number of retries was attempted. This relay turns off at the start of the automatic backup. (This relay does not turn off when SM1351 (CPU module data backup execution request) is on.) • Off: Retry not executed/Retry being executed • On: Retry failed
SM1356	This relay turns on during execution of the retry of the automatic backup. This relay turns on at the start of the retry of the automatic backup and turns off when the retry of the automatic backup is started or the retry of the automatic backup for the number of retries is attempted without execution of an operation or function that cannot be executed. • Off: Retry is not being executed. • On: Retry is being executed.
Bit 10 of SD944	Set whether to retry the automatic backup. • Off: Retry is not executed. • On: Retry is executed.

The setting of the bit 10 of SD944 (Backup function setting) cannot be changed during execution of the automatic backup.

The retry setting of the automatic backup is enabled at the following timing.

- When the bit 0, bit 1, or bit 15 of SD944 turns on

Operating procedure (when date and time are specified)

Data is automatically backed up on the specified date and time.

1. Set the upper limit value for the number of CPU module backup data. (The setting method and operating procedure for the upper limit value are the same as those for the upper limit value for the backup processing triggered by turning on SM1351.) (☞ Page 308 Operating procedure)

2. Set the date and time with SD947 to SD949.

Special register	Description
SD947	Set the date when the data is to be automatically backed up.
SD948	Set the time (hour) when the data is to be automatically backed up.
SD949	Set the time (minute) when the data is to be automatically backed up.

3. To retry the automatic backup, turn on the bit 10 of SD944 (Backup function setting). For the retry of the automatic backup, refer to the following.

☞ Page 309 Retrying the automatic backup

4. Turn on the bit 0 of SD944.

If the backup processing is completed with an error and SM953 (CPU module data backup error check flag) turns on, check SD953 (Backup error cause), take actions, and then back up the data again as required.

Restriction

- In months that does not have the specified date, the automatic backup is not executed. For example, when SD947 has been set to "31", the months when the automatic backup is executed are January, March, May, July, August, October, and December.
- If the time less than 1 hour from the start time of daylight saving time has been set with the daylight saving time function, the automatic backup is not executed.

Operating procedure (when time and day of the week are specified)

Data is automatically backed up on the specified time on the specified day of the week.

1. Set the upper limit value for the number of CPU module backup data. (The setting method and operating procedure for the upper limit value are the same as those for the upper limit value for the backup processing triggered by turning on SM1351. (☞ Page 308 Operating procedure))

2. Set the time and day of the week with SD950 to SD952.

Special register	Description
SD950	Set the time (hour) when the data is to be automatically backed up.
SD951	Set the time (minute) when the data is to be automatically backed up.
SD952	Set the day of the week when the data is to be automatically backed up. b0: Sunday, b1: Monday, b2: Tuesday, b3: Wednesday, b4: Thursday, b5: Friday, b6: Saturday

3. To retry the automatic backup, turn on the bit 10 of SD944 (Backup function setting). For the retry of the automatic backup, refer to the following.

☞ Page 309 Retrying the automatic backup

4. Turn on the bit 1 of SD944.

If the backup processing is completed with an error and SM953 turns on, check SD953, take actions, and then back up the data again as required.

Restriction

If the time less than 1 hour from the start time of daylight saving time has been set with the daylight saving time function, the automatic backup is not executed.

Operating procedure (when a stop error has occurred in the CPU module)

Data is automatically backed up when a stop error occurs in the CPU module.

1. Set the upper limit value for the number of CPU module backup data. (The setting method and operating procedure for the upper limit value are the same as those for the upper limit value for the backup processing triggered by turning on SM1351. (☞ Page 308 Operating procedure))

2. To retry the automatic backup, turn on the bit 10 of SD944 (Backup function setting). For the retry of the automatic backup, refer to the following.

☞ Page 309 Retrying the automatic backup

3. Turn on the bit 15 of SD944.

If the backup processing is completed with an error and SM953 turns on, check SD953, take actions, and then back up the data again as required.

Restriction

If a major error has occurred, the automatic backup may not be performed.

Settings for automatic restoration with the SD CARD OFF button

When using the automatic restoration with the SD CARD OFF button, set it before the backup processing. After setting, execute each backup processing.

15

Restoration setting

Turn on bit 2 of SD955 (Automatic restoration with SD CARD OFF button).

Bit 2 of SD955	Setting of the automatic restoration with the SD CARD OFF button
OFF	Disable
ON	Enable

Optionally, set the other restoration settings. (☞ Page 316 Restoration target data, ☞ Page 316 Restoration of the special relay and special register, ☞ Page 317 Operation setting after restoration)

Point

The bit 13 of SD955 (Latest data) is not used because the automatic restoration with the SD CARD OFF button restores the latest backup data out of multiple backups saved.

Settings

Turn on bit 2 of SD955 (Restoration function setting).

Backup after setting

Refer to each backup function and execute the backup processing. (☞ Page 308 Backup processing triggered by turning on SM1351, ☞ Page 309 Automatic backup using SD944)

Checking backup errors

When an error has occurred, a diagnostic error is not detected and an error code is stored in SD953 (Backup error cause). (☞ Page 805 List of error codes)

Precautions

The following describes the precautions for the backup function.

Prohibited operation during execution of the backup processing

Do not perform the following operations during execution of the backup processing.

- Removing and inserting the SD memory card
- Powering off or resetting the CPU module

The above mentioned operations leave the backup data in the SD memory card in an incomplete state which is middle of the backup processing. Do not use these data for a restoration. If these data are used, the restoration is completed with an error.

Suspending backup processing

The following operation can suspend a backup processing.

- Setting the SD memory card forced disable

Suspending a backup processing leaves the backup data in the SD memory card in an incomplete state which is middle of the backup processing. Do not use these data for a restoration. If these data are used, the restoration is completed with an error.

Device/label data

To execute the backup processing, do not change device/label data during execution of the processing. Since device/label data is divided into multiple scans and backed up, changes in the device/label data may cause data inconsistency.

When labels accessible from external devices have been set in the CPU module

For the programmable controller CPU with firmware version "24" or earlier, the backup cannot be executed if labels accessible from external devices have been set in the CPU module by access label settings from external devices.
Check that no such labels have been set in the CPU module and then execute the backup processing.

When parameter settings were changed before execution of the backup processing

When programs or parameter settings were changed, check that operations are performed with the new programs and parameter settings and then execute the backup function. If the backup processing is executed without the check of the operations with the new programs and parameter settings, the restoration processing may not be executed.

Changing backup target data

Do not change backup target data in the CPU module during execution of the backup processing. If the target data was changed during execution of the backup processing, the changes will not applied.

Special relay and special register that function as flags to execute other functions

Before executing the backup processing, turn off the special relay and special register that function as flags to execute other functions. If the backup processing is executed when they are on, the corresponding function request may turn on and the function may be executed at the restoration of data in the special relay and special register.

Data protected by security functions

■File password function

Unlock the file passwords of the files in the backup target CPU module. If any files to which file passwords have been set exist in the CPU module, the files are not backed up.

■Security key authentication function

Locked programs are backed up in the locked state regardless of whether security keys have been written or not.

Timing for setting the upper limit value for the number of CPU module backup data

Set the upper limit value for the number of CPU module backup data before execution of the backup processing.
When the backup processing has been executed (a CPU data folder exists in the SD memory card) and the bit 5 of SD944 (Enable the upper limit value for the number of CPU module backup data) is turned on, an error will occur.
Even though the backup processing has been executed, turning off the bit 5 of SD944 can disable the upper limit value for the number of CPU module backup data. To set the upper limit value for the number of CPU module backup data again, replace the SD memory card with another one where no CPU data folder exist, or turn off and on the bit 5 of SD944 after deleting the backup data folder.

SFC program status

Do not change the status of the SFC program, such as step active status and transition conditions during execution of the backup processing. If the status of the SFC program was changed, the backup processing is completed with an error.

Time required for completing the backup processing

The backup processing takes more time depending on the size of data or the number of folders/files stored in the CPU module.

Operations and functions that cannot be performed

While the following operations or functions are being executed, the backup processing cannot be executed.

The following operations and functions cannot be executed during execution of the backup processing.

Operation or function		
Operation from the engineering tool	Initializing the CPU built-in memory/SD memory card	
	Clearing values (Devices, labels, file registers, latches)	
	Writing data to the programmable controller (including online change of files)	
	Deleting data in the programmable controller	
	User data operation	Writing user data
		Deleting user data
		Creating a folder
		Deleting a folder
		Changing a folder name
	Online change (online change (ladder block))	
	Event history function (Clearing event history)	
	File password function	
	Security key authentication function (Writing/deleting a security key to/in the CPU module)	
	Predefined protocol support function (Writing protocol setting data)	
	Memory dump function (Registering/clearing memory dump)	
	Firmware update function (Firmware update using the engineering tool)	
Operation using the CPU module logging configuration tool	Data logging function (Writing/deleting a logging setting file, registering/clearing a logging setting, stopping a logging)	
	Deleting a logging file	
Others	• SLMP • MC protocol	Clearing the remote latch (Remote Latch Clear)
		Creating a new file (New File)
		Writing data to a file (Write File)
		Deleting a file (Delete File)
		Copying a file (Copy File)
		Changing a file attribute (Change File State)
		Changing file creation date (Change File Date)
		Opening a file (Open File)[*1]
	File transfer from an Ethernet-equipped module (FTP server)	Writing a file (put, mput, pm-write)
		Deleting a file (delete, mdelete)
		Changing a file name (rename)
		Changing a file attribute (change)
	File transfer function (FTP server) of the built-in Ethernet function	
	File transfer function (FTP client) of the built-in Ethernet function	
	Changing an IP address	
	iQ Sensor Solution data backup/restoration function	
	System operation setting with SD384	
	Transfer to the data memory with special relay	
	Data logging file transfer to data memory[*2]	
	Online module change function	

*1 The operation performed only for opening a file to write data

*2 It is executed when the trigger logging data collection is completed or data collection for the specified number of storage files is completed.

Backup during execution of the backup processing

The backup processing triggered by turning on SM1351 or automatic backup cannot be executed during execution of the backup processing. (The latter backup processing is ignored.)

Data logging function and backup

When the CPU built-in memory (function memory) is specified for the data storage destination memory in the data logging function, the function memory is cleared after the CPU module is powered off or the RESET state is cleared. Thus, data logging files are also cleared. To prevent data logging files from being lost, back up the files after the completion of file transfer by the data logging file transfer function.

When the data is written to the programmable controller after rebuilt all (reassignment)

When the data is written to the programmable controller after rebuilt all (reassignment), power off and on or reset the programmable controller and execute backup processing.

15.2 Restoration Function

This function restores backup data in the SD memory card to the CPU module.

Restoration target folder

Set restoration target data among backup data in the SD memory card with SD956 (Restoration target date folder setting) to SD958 (Restoration target number folder setting). The latest backup data can be restored with the bit 13 of SD955 (Restoration function setting).

Special register	Description
Bit 13 of SD955	Set the restoration function setting with bit patterns. Off: Data specified with the restoration target folders is restored. On: The latest data is restored.*1
SD957, SD956	Specify a date folder of the restoration target data in BCD. SD957 (upper): Year, SD956 (lower): Month and date
SD958	Specify the folder number (00001 to 32767) of restoration target data.

*1 The latest data is the backup data with the largest number in the latest date folder.

Restoration target data

Restoration target data is set with SD954 (Restoration target data setting).

Value of SD954	Restoration target data setting
0	All target data
1	Only device/label data
2	All target data excluding device/label data

Restoration of the special relay and special register

Set whether to restore the special relay and special register with the bit 14 of SD955.

Bit 14 of SD955	Restoration target data setting
Off	The special relay and special register are not restored.
On	The special relay and special register are restored.

However, the special relay/special register areas listed below are not restored even when the bit 14 of SD955 is on.

CPU module	Special relay/Special register
Programmable controller CPU	Special relay/special register areas for the CPU module data backup/restoration function • SM953, SM959, SM961, SM1350, SM1351, SM1353, SM1354, SM1356, SD953, SD959, SD1350, SD1351, SD1353
Process CPU	Special relay/special register areas for the redundant function • SM953, SM959, SM961, SM1350, SM1351, SM1353, SM1354, SM1356, SD953, SD959, SD1350, SD1351, SD1353, SM1630, SM1632, SM1633, SM1634, SM1635, SM1636, SM1637, SM1643, SM1644, SM1645, SM1680, SM1681, SM1682, SM1683, SM1684, SM1754, SD1643, SD1644, SD1645, SD1646, SD1648, SD1649, SD1650, SD1680, SD1681, SD1682 to SD1688, SD1689, SD1690 to SD1720, SD1721, SD1722 to SD1752
Safety CPU	Special relay/special register/safety special register areas for the functions of the Safety CPU. • SM953, SM959, SM961, SM1350, SM1351, SM1353, SM1354, SM1356, SD953, SD959, SD1350, SD1351, SD1353, SA\SD1089 to SA\SD1097, SA\SD1104 to SA\SD1223, SA\SD1240 to SA\SD1247, SA\SD1256 to SA\SD1263, SA\SD1272 to SA\SD1279, SA\SD1288 to SA\SD1295, SA\SD1304 to SA\SD1311, SA\SD1320 to SA\SD1327, SA\SD1336 to SA\SD1343, SA\SD1352 to SA\SD1359, SA\SD1367, SA\SD1384 to SA\SD1387, SA\SD1400 to SA\SD1403, SA\SD1409 to SA\SD1412

Operation setting after restoration

As an operation after the restoration processing, set whether to operate the CPU module in the status at the backup processing or to operate the CPU module in the initial status with the bit 15 of SD955. The following table lists the operations of each item to be performed according to the operation setting after restoration.

Item	Operation setting after restoration	
	Operating the CPU module in the status at data backup (bit 15 of SD955 = On)	Operating the CPU module in the initial status (bit 15 of SD955 = Off)
Device initial value	The device initial value is not set after the restoration processing.	The device initial value is set after the restoration processing. (The device data at data backup will be overwritten with the device initial value.)
Initial values of global/local labels	The initial values of global/local labels are not set after the restoration processing.	The initial values of global/local labels are set after the restoration processing. (The label data at data backup will be overwritten with the initial value of the initial global/local value.)
SFC program*1	When "Resume Start" was selected before data backup, the SFC program is resumed after restoration processing.	The SFC program is not resumed after restoration processing even though "Resume Start" was selected before data restoration.
Event history	The event history at data backup is set.	The event history at data backup is not set, a new file is created and a restoration event is registered.

*1 When the firmware version of the CPU module at the backup processing is different from that of at the restoration processing, the SFC program will be started from the block 0 and step 0 regardless of the operation setting after the restoration.

Note that this setting is invalid since the device initial value file, initial global label value file, initial local label value file, event history file, and SFC program are not restored when the value in SD954 (Restoration target data setting) is 1 (restoration target data are only device/label data).

Point

For the operation setting after restoration, specify the operation at a completion of restoration.
When the CPU module is switched from STOP to RUN, values of devices are changed according to the operation of the device memory at an operating status change of the CPU module. (☞ Page 96 Operation Processing When Operating Status Is Changed)

Restoration processing triggered by turning on SM1354

Backup data is restored at a desired timing.

Point

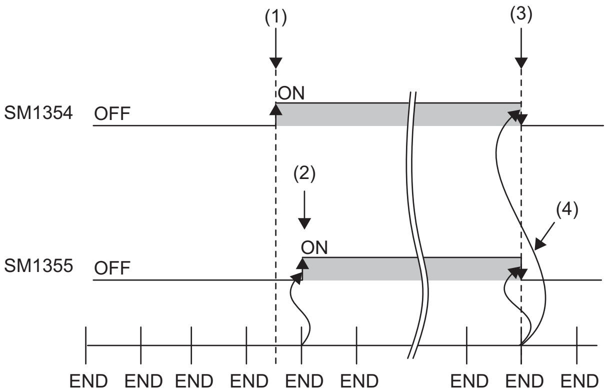
Use the restoration function by turning on SM1354 to check the backup data and to test before actual operation.
To operate the system using the backup data, use the automatic restoration with SD955. (☞ Page 319 Automatic restoration using SD955)

Restriction

The restoration processing triggered by turning on SM1354 (CPU module data restoration execution request) can be executed only when the CPU module is the STOP state.

Operating procedure

1. Set restoration target data with SD954 (Restoration target data setting).
2. Set a restoration target folder in the areas from SD956 (Restoration target date folder setting) to SD958 (Restoration target number folder setting). (These settings are not required when the bit 13 of SD955 is on.)
3. Set each setting with the bit 13 to 15 of SD955.
4. Set the CPU module to the STOP state.
5. Turn on SM1354.



(1) A restoration execution request is sent.
(2) The system turns on SM1353 (CPU module data restoration status flag).
(3) The system turns off SM1354 after the restoration processing is completed.
(4) The system turns off SM1353.

If the restoration processing is completed with an error and SM959 (CPU module data restoration error check flag) turns on, check SD959 (Restoration error cause), take actions, and then restore the data again as required.

Point

The execution status of the restoration processing can be checked in SD1350 (Number of uncompleted folders/files of CPU module data backup/restoration) and SD1351 (Progression status of CPU module data backup/restoration). (☞ Page 966 List of Special Register Areas)

Automatic restoration using SD955

Backup data is automatically restored when the CPU module is powered on or is reset.

Initialization at the automatic restoration

Set whether or not to initialize drives other than the SD memory card at execution of the automatic restoration with the bit 1 of SD955 (Restoration function setting).

This setting is enabled only when SD954 (Restoration target data setting) is set to 0 (All target data).

Bit 1 of SD955	Restoration target data setting
Off	Not initialized
On	Initialized

Operating procedure

1. Set the data to restore in SD954.
2. Set a restoration target folder in the areas from SD956 (Restoration target date folder setting) to SD958 (Restoration target number folder setting). (These settings are not required when the bit 13 of SD955 is on.)
3. Set values to the bit 1 and bits from 13 to 15 of SD955.
4. Turn on the bit 0 of SD955.
5. Power off and on or reset the CPU module.

If the restoration processing is completed with an error and SM959 (CPU module data restoration error check flag) turns on, check SD959 (Restoration error cause), take actions, and then restore the data again as required.

Point

- Since the special register set for the automatic restoration is a latch area, setting data is held.
- SD955 holds its setting even after the CPU module is powered off and on or is reset. Thus, if the CPU module is powered off and on or is reset while the bit 0 of SD955 is on, the automatic restoration is executed again. Not to perform the automatic restoration when the CPU module is powered off and on or is reset next time, turn off the bit 0 of SD955 after a restoration is completed and then power off and on or reset the CPU module.

Automatic restoration with the SD CARD OFF button

Backup data is automatically restored when the CPU module is powered on or is reset while the SD CARD OFF button is pressing.

Point

When the automatic restoration using SD955 and the automatic restoration with the SD CARD OFF button are executed at the same time, the automatic restoration with the SD CARD OFF button is prioritized. Even if bit 0 of SD955 (Automatic restoration) is set to on, the automatic restoration with the SD CARD OFF button is executed.

Restoration setting

The automatic restoration with the SD CARD OFF button only restores data using the SD CARD OFF button (the special register area is not operated from the engineering tool or the GOT). Set it before the backup processing.*1 (☞ Page 311 Settings for automatic restoration with the SD CARD OFF button)

*1 The restoration settings are restored to the SD memory card as a system file for the automatic restoration with the SD CARD OFF button at backup.

Point

The automatic restoration with the SD CARD OFF button restores the latest backup data out of multiple backups saved. When restoring a specific backup data, prepare the data and the SD memory card in which the system file for the relevant backup data only and the automatic restoration with the SD CARD OFF button is stored.

Operating procedure

Power on or reset the CPU module while pressing the SD CARD OFF button.*1*2*3

*1 Do not execute the automatic restoration with the SD CARD OFF button and the firmware update function at the same time. Otherwise, the automatic restoration with the SD CARD OFF button does not operate.

*2 If the system file for the automatic restoration with the SD CARD OFF button is not stored to the SD memory card, 300CH error is detected.

*3 Release the SD CARD OFF button within 10 seconds after the READY LED flashes. If the switch is pressed more than 10 seconds, restoration may fail.

Checking restoration errors

- When an error occurs in the restoration processing triggered by turning on SM1354, a diagnostic error is not detected and an error code is stored in SD959 (Restoration error cause). (☞ Page 805 List of error codes)
- When an error occurs at the automatic restoration using SD955, a diagnostic error is detected. The error code is stored also in SD959. (☞ Page 805 List of error codes)

Precautions

The following describes the precautions for the restoration function.

Prohibited operation during execution of the restoration processing

Do not perform the following operations during execution of the restoration processing.

- Removing and inserting the SD memory card
- Powering off or resetting the CPU module

The above mentioned operations leave the data in the CPU module in an incomplete state which is middle of the restoration processing. Do not run the CPU module with this incomplete state. Doing so may cause an unintended operation. Restore the data again or format each drive in the CPU module, and clear devices/labels before writing programs or parameters to the programmable controller.

Suspending the restoration processing

The following operation can suspend a restoration processing.

- Setting the SD memory card forced disable

Suspension during a restoration leaves the data in the CPU module in an incomplete state which is middle of the restoration processing. Do not run the CPU module with this incomplete state. Doing so may cause an unintended operation. Restore the data again or format each drive in the CPU module, and clear devices/labels before writing programs or parameters to the programmable controller.

15

Model name of the CPU module to which data is restored

Restore the data to the CPU module whose model name is the same as that of the CPU module where the backup data has been stored. If not, the data cannot be restored.

When labels accessible from external devices have been set in the CPU module

For the programmable controller CPU with firmware version "24" or earlier, the restoration cannot be executed if labels accessible from external devices have been set in the CPU module by access label settings from external devices.
Check that no such labels have been set in the CPU module and then execute the restoration processing.
For the programmable controller CPU with firmware version "25" or later, the restoration by turning on SM1354 cannot be executed if labels accessible from external devices have been set in the CPU module or in a backed up data from external devices. When the restoration is required, execute the automatic restoration using SD955.

Automatic restoration using SD955 and functions that cannot be executed

Do not set both of the automatic restoration using SD955 and the boot operation at a time.
If the CPU module is powered off and on or is reset with both of the automatic restoration using SD955 and the boot operation set, the automatic restoration using SD955 does not function.

Status of the restoration destination CPU module

If the status of the restoration destination CPU module differs from that of the CPU module at the backup processing (such as programs or parameters), the restoration may not be executed.
When the backup data to be restored is backed up in a different status from that of the restoration destination CPU module, store 0 (All the target data) to SD954 (Restoration target data setting) and execute the automatic restoration.

When the same name folder or file exists in the restoration target CPU module

If the name of a folder or file in the restoration target CPU module and the name of a folder or file in backup data are identical, the folder or file in the module will be overwritten by that in the backup data.

Changing the operating status during execution of restoration

During execution of the restoration processing, the CPU module remains in the STOP state even though the RUN/STOP/RESET switch is changed from the STOP to RUN position or the remote RUN or the remote PAUSE is executed. If the operating status of the CPU module is changed, the status will changes to the set status after the restoration processing is completed.

Time required for completing the restoration processing

The restoration processing takes more time depending on the number of backup data sets (folders), file size, and the number of files in the SD memory card.

In a multiple CPU system, if the automatic restoration has taken time and an error has occurred in another CPU module, another error may occur in the CPU module to which the automatic restoration was executed after the completion of the restoration.

In that case, shorten the time for restoration. Change the setting of SD954 (Restoration target data setting) to only the device data that is to be cleared at a system start-up, and execute the automatic restoration again.

Monitoring stop at restoration

Stop monitoring before executing the restoration processing.

When the restoration processing is executed, programs, parameters, and device/label values may not be properly monitored because they are changing.

Reflecting restored data

Some of the parameters are reflected only when the CPU module is powered off and on or is reset. If the restoration processing is executed while the CPU module is in the STOP state and then is switched to the RUN state, the CPU module may not operate with the backup data. In that case, power off and on or reset the CPU module. For device/label data, since device/label data except for latch-specified devices/labels is initialized when the CPU module is powered off and on or is reset, restore the device/label data again as required.

Abnormal completion of restoration

Since the restoration processing will be completed with an error, do not execute the restoration processing in the following cases.

- The name of a file in the restoration destination CPU module and the name of a file in backup data are identical, and a file password has been set.
- Data in a backup folder has been deleted. (Do not delete the data in backup folders that are likely to be used for restoration.)
- Backup data has problems. (Backup data has been changed or the CPU module was powered on and off during execution of the backup processing.)

Instruction executed at the rising/falling edge when data is restored

When the program file is restored by turning on SM1354, the execution status of instructions is undefined and the instructions executed at the rising/falling edge may not work properly.

To operate the system by restoring the backup data, use the automatic restoration with SD955.

When the special relay and special register are restored

When a restoration is operated with the setting to restore the special relay and special register, the system will be operated with the values (time, day of the week, day, time, and the upper limit value of the number of backup data for the automatic backup set to the backup function setting) before restoration. (The backup function setting will not be re-set although the special register areas are overwritten by the restoration.)

To operate the system with the restored backup function setting, set the backup function setting again.

Data protected by security functions

■File password function

Unlock the file passwords of the files in the backup target CPU module. If any files to which file passwords have been set exist in the CPU module, the files are not restored.

■Security key authentication function

Locked programs can be restored regardless of whether security keys have been written or not. However, when the security key has not been written to the CPU module after the restoration processing, the program cannot be executed. Restore unlocked backup data or set the same security key.

When the SFC program is restarted from where the program was stopped

Specify the continue start. When the continue start has not been specified, the SFC program will be started from the block 0 and step 0 even though the bit 15 of SD955 is on (the continue start is executed).

When the IP address change function is used

If the backup processing is executed with the IP address stored in the IP address storage area (system memory), the IP address will be changed at the following timing in the restoration processing.

- Restoration processing triggered by turning on SM1354: When the CPU module is powered off and on or is reset after the restoration processing
- Automatic restoration using SD955: When the restoration processing is executed

Operations and functions that cannot be performed

While the following operations or functions are being executed, the restoration processing cannot be executed.
The following operations and functions cannot be executed during execution of the restoration processing.

Operation or function		
Operation from the engineering tool	Initializing the CPU built-in memory/SD memory card	
	Clearing values (Devices, labels, file registers, latches)	
	Reading data from the programmable controller	
	Writing data to the programmable controller (including online change of files)	
	Verifying data with the programmable controller	
	Deleting data in the programmable controller	
	User data operation	Reading user data
		Writing user data
		Deleting user data
		Creating a folder
		Deleting a folder
		Changing a folder name
	Online change (online change (ladder block))	
	Event history function (Updating event history data, clearing event history)	
	File password function	
	Security key authentication function (Writing/deleting a security key to/in the CPU module)	
	Predefined protocol support function (writing/reading/verifying protocol setting data)	
	Memory dump function (Memory dump setting/reading results, registering/clearing memory dump)	
	Firmware update function (Firmware update using the engineering tool)	
Operation using the CPU module logging configuration tool	Data logging function (Writing/reading/deleting a logging setting file, registering/clearing a logging setting, stopping a logging)	
	Deleting a logging file	

Operation or function		
Others	• SLMP • MC protocol	Remote latch clear (Remote Latch Clear)
		Creating a new file (New File)
		Writing data to a file (Write File)
		Deleting a file (Delete File)
		Copying a file (Copy File)
		Changing a file attribute (Change File State)
		Changing file creation date (Change File Date)
		Opening a file (Open File)
		Reading a file (Read File)
	File transfer from an Ethernet-equipped module (FTP server)	Reading a file (get, mget)
		Writing a file (put, mput, pm-write)
		Deleting a file (delete, mdelete)
		Changing a file name (rename)
		Changing a file attribute (change)
	File transfer function (FTP server) of the built-in Ethernet function	
	File transfer function (FTP client) of the built-in Ethernet function	
	Changing an IP address	
	iQ Sensor Solution data backup/restoration function	
	System operation setting with SD384	
	Transfer to the data memory with special relay	
	Data logging file transfer to data memory[*1]	

*1 It is executed when the trigger logging data collection is completed or data collection for the specified number of storage files is completed.

Operation of when the data logging function is used

If data is backed up during execution of the data logging function and the function has been set to be started automatically when the operating status of the CPU module is changed to RUN, the data logging function will be automatically executed when the status of the CPU module changes to RUN after the restoration processing. To restart the data logging function after the restoration processing without the above setting, use the CPU module logging configuration tool.
When the CPU built-in memory (function memory) is specified for the data storage destination memory, the function memory is not backed up. Therefore, when the data logging is restarted after the restoration processing, the logging files are deleted after the CPU module is powered off or the RESET state is cleared, and the logging file number starts from 1 at the storage file switching.

Restoration during execution of the restoration processing

The restoration processing triggered by turning on SM1354 cannot be executed during execution of the restoration processing. (The latter restoration processing is ignored.)

When the CPU module database access function is used

When the data is backed up with the CPU module data backup/restoration function and restored by turning on SM1354 while using the CPU module database access function, power off and on or reset the CPU module regardless of the completion status of the restoration. (For the automatic restoration by SD955, the CPU module is not required to be powered off and on or reset.)

Restoration of when the data allocation in the program file is different

The data allocation in the program file differs depending on the firmware version of the CPU module. (☞ Page 101 Data allocation and procedure of read/write operations)

When the data backed up using the CPU module with the conventional data allocation is restored to the CPU module with the new data allocation, the restoration processing is completed successfully. In this case, data allocation in the CPU module is different. Therefore, an error occurs when the CPU module is powered off and on or is reset, or the module operating status is changed from STOP to RUN after the restoration processing. When the data backed up using the CPU module with the new data allocation is restored to the CPU module with the conventional data allocation, the restoration processing may be completed with an error.

In this case, an error occurs when the CPU module is powered off and on or is reset, or the module operating status is changed from STOP to RUN after the restoration processing.

16 MULTIPLE CPU SYSTEM FUNCTION

RnCPU RnENCPU RnPCPU (Process) ~~RnPCPU (Redundant)~~ ~~RnPSFCPU (Standard)~~ ~~RnPSFCPU (Safety)~~ RnSFCPU (Standard) RnSFCPU (Safety)

- When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

With multiple CPU modules mounted on the base unit, each of the CPU modules controls their own assigned I/O modules and intelligent function modules. In addition, the CPU modules communicate with each other.

Point

- For details on the concept of the multiple CPU system configuration specification (System configuration specifications such as the mounting position of the CPU modules and assignment of CPU number/IO number), refer to the MELSEC iQ-R Module Configuration Manual.
- For the start-up (Setting, operating procedures, etc) of the Multiple CPU system, refer to the GX Works3 Operating Manual.

Restriction

- The startup time of the multiple CPU system may be slowed, depending on the configuration of installed modules, boot operation, functions performed before CPU modules enter into the RUN state (e.g. setting of initial device/label values), the configuration of system parameters and CPU parameters (e.g. the number of programs to run).
- When diagnostics of the SD memory card is performed due to operation such as power-off during access to the SD memory card, the startup time of the multiple CPU system may be slowed.
- Create a program so that only one CPU module accesses the Q series module, if possible. If multiple CPU modules access the Q series module simultaneously, the scan time (including the execution processing of interrupt programs) may be extended due to access waiting time.
- Before using the multiple CPU system function with the RnENCPU, check the versions of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

16.1 Out-of-group I/O Fetch

The access range to the controlled module is different from that to the non-controlled module. To fetch the data which cannot be accessed, use the out-of-group I/O fetch function.

Accessing controlled module

The access range to the controlled module of the CPU module is the same as that to the single CPU system, and I/O refresh for the controlled module and/or reading/writing to buffer memory of the intelligent function module are enabled.

Accessing non-controlled module

Access to the non-controlled module of the CPU module is limited to reading input (DX) and buffer memory of the intelligent function module. Note that On/Off data of input (X)/output (Y) of the non-controlled module can be fetched using the out-of-group I/O fetch function. (☞ Page 328 Out-of-group I/O fetch setting)

○: Accessible, ×: Not accessible

Communication method	Access target		Access range	
			When "Import" is selected in the settings for out-of-group I/O fetch	When "Not Imported" is selected in the settings for out-of-group I/O fetch
Communication through refresh	Input (X)	Read to another CPU module	○[*1]	× (Non-processing)
	Output (Y)	Write to another CPU module	× (Non-processing)	
		Read to another CPU module	○[*1]	
	The buffer memory of the intelligent function module	Read	× (Cannot be specified from engineering tools)	× (Cannot be specified from engineering tools)
		Write		
	Link direct device	Read	× (Cannot be specified from engineering tools)	× (Cannot be specified from engineering tools)
		Write		
Communication through direct access	Input (DX)	Read to another CPU module	○	○
	Output (DY)	Write to another CPU module	× (Non-processing)	× (Non-processing)
		Read to another CPU module	× (Cannot be specified)	× (Cannot be specified)
	The buffer memory of the intelligent function module	Read	○	○
		Write	× (An error occurs in the CPU module.)	× (An error occurs in the CPU module.)
	Link direct device	Read	× (An error occurs.)	× (An error occurs.)
		Write		

*1 Cannot read from CPU modules which are synchronized through the inter-module synchronization function.

Precautions

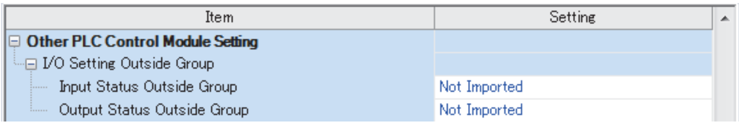
An error is not detected in reading input (DX) for the non-controlled module and buffer memory of the intelligent function module.

Out-of-group I/O fetch setting

In this menu item, whether or not out-of-group I/O status is fetched can be specified.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Other PLC Control Module Setting] ⇨ [I/O Setting Outside Group]

Window



Displayed items

Item	Description	Setting range	Default
Input Status Outside Group	Specifies whether or not out-of-group input status is fetched.	• Not Imported • Import	Not Imported
Output Status Outside Group	Specifies whether or not out-of-group output status is fetched.	• Not Imported • Import	Not Imported

Fetching input (X)

When "Import" is selected for "Input Status Outside Group" in "I/O Setting Outside Group", the input refresh before the start of the program operation fetches On/Off data from the input module and/or intelligent function module controlled by other CPU modules.

■Modules from which input (X) can be fetched

Input (X) can be fetched from the following modules mounted on the base/extension base unit.

- Input module
- I/O combined module*1
- Intelligent function module

*1 Data in the areas assigned to the output portion (area not used) are not fetched. On/Off state is maintained.

Restriction

Input (X) cannot be fetched from the Q series modules.

Fetching output (Y)

When "Import" is selected for "Output Status Outside Group" in "I/O Setting Outside Group", the output refresh before the start of the program operation fetches On/Off data output to the output module and/or intelligent function module controlled by other CPU modules into output (Y) for the host CPU module.

■Modules from which output (Y) can be fetched

Output (Y) can be fetched from the following modules mounted on the base/extension base unit.

- Output module
- I/O combined module
- Intelligent function module

Restriction

Output (Y) cannot be fetched from the Q series modules.

Output to output/intelligent function module

On/off data cannot be output to non-controlled modules. When turning on or off output of the output module and/or intelligent function module controlled by other CPU modules by the program or others, the output is turned on or off within the CPU module. However, it is not output to output/intelligent function module.

Accessing buffer memory of intelligent function module

■Reading data on buffer memory

The following instructions can be used to read data stored in the buffer memory of the intelligent function module.

- FROM instruction
- Instruction using the CPU module access device (Un\Gn)

■Writing data to buffer memory

The following instructions cannot be used to write data to the buffer memory of the intelligent function module.

- TO instruction
- Instruction using the CPU module access device (Un\Gn)

Accessing a module by using link direct devices

The CPU module can access only to modules under its control by using link direct devices. Non-controlled modules cannot be accessed.

16.2 Operation Settings

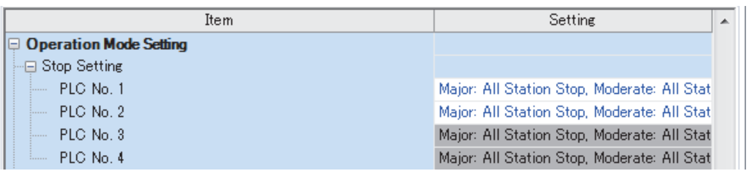
This section describes the operation settings of the multiple CPU system function.

Stop setting

An operating status, whether to stop the operation of all the CPU modules or not, if a major or moderate error occurs in any of the CPU modules is set.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Operation Mode Setting] ⇨ [Stop Setting]

Window



Displayed items

Item	Description	Setting range	Default
PLC No.1	Sets whether to stop the operation of all the CPU modules or not if a major or moderate error occurs in CPU No.1.	• Major: All Station Stop, Moderate: All Station Stop • Major: All Station Stop, Moderate: All Station Continue • Major: All Station Continue, Moderate: All Station Continue	Major: All Station Stop, Moderate: All Station Stop
PLC No.2	Sets whether to stop the operation of all the CPU modules or not if a major or moderate error occurs in CPU No.2.		
PLC No.3	Sets whether to stop the operation of all the CPU modules or not if a major or moderate error occurs in CPU No.3.		
PLC No.4	Sets whether to stop the operation of all the CPU modules or not if a major or moderate error occurs in CPU No.4.		

Applicable errors to the stop setting

The following table lists the applicable errors to the setting that specifies the operation of all the CPU modules of when a major or moderate error has occurred in any of the CPU modules.

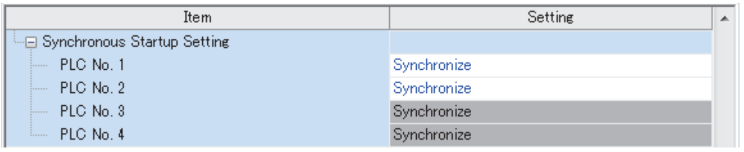
Error name	Error code
Another CPU module moderate error	1220H
Another CPU module major error	2461H, 2462H, 2470H

Synchronous startup setting

Startup time is synchronized among the CPU modules. This setting eliminates the need for an interlock program that monitors the startup time of another CPU module when accessing it. Note, however, that the startup of the entire system delays because the system starts up with the last CPU module.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Operation Mode Setting] ⇨ [Synchronous Startup Setting]

Window



Displayed items

Item	Description	Setting range	Default
PLC No.1	Sets the CPU modules whose startup time is synchronized in the multiple CPU system.	• Synchronize • Do not Synchronize	Synchronize
PLC No.2			
PLC No.3			
PLC No.4			

Point

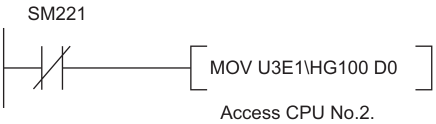
- Group setting for the synchronized start-up is available. For example, a setting in which only CPUs No.1 and No.2 start synchronously within a multiple CPU system with four CPU modules is possible.
- If a reserved (empty) CPU is specified to synchronize, it is skipped and the other CPUs in the group start synchronously.
- This setting is designed to access to each CPU module without interlock in the multiple CPU system. It is not intended to be used for starting operation processing at the same time across CPU modules after startup.

Program to check start-up of each CPU module

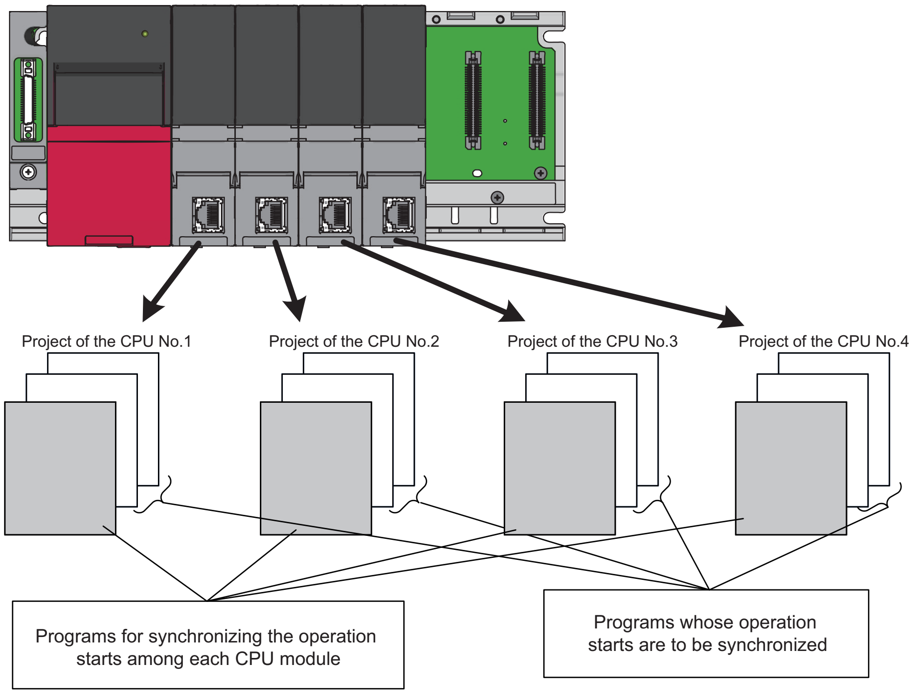
It is recommended to create a program that checks start-up of each CPU module using SM220 to SM223 (CPUs No.1 to No.4 preparation completed) when the multiple CPU synchronized startup is disabled. If a certain instruction is issued against a CPU module that has not started, the instruction executes no processing.

Ex.

Program to check start-up of CPU module of CPU No.2



To synchronize the start of operation processing across CPU modules, a program is required to check whether individual CPU modules are ready to start operation processing or not.



The following is an example of the configuration of a program which synchronizes the start of operation processing across CPU modules and programs which should start operation processing synchronously.

Program	Execution type	Description
Program to synchronize the start of operation processing	Scan	This is a program used to synchronize the start of operation processing across all CPU modules. This program must be specified as one for running at the beginning of the scan execution type of program. Also, when all the CPU modules are ready to start operation processing, the flag indicating an operation start turns on for only one scan.
Program to start operation processing synchronously	Event	This is a program which should be executed when all the CPU modules are ready to start operation processing. This program must be specified as an event execution type program which is triggered when the operation start flag is turned on.

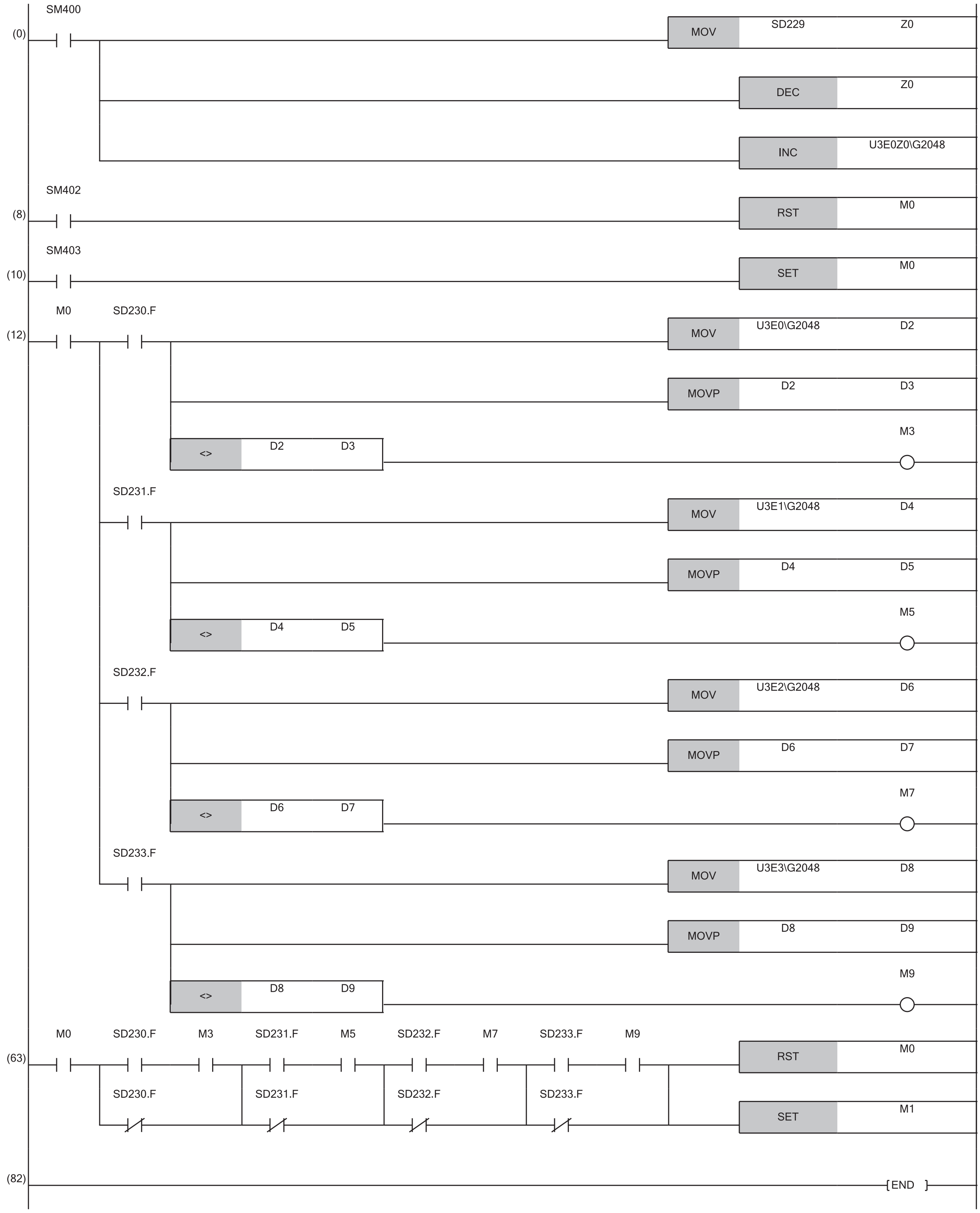
16

■Program example

- Devices to be used for programs to start operation processing synchronously

Device to be used	Application
M0	Flag that indicates the operation processing is ready to be started (after a flag that indicates the operation start turns on, this flag turns off.)
M1	Flag that indicates an operation start (this flag turns on for only one scan.)
U3En\G2048	The synchronization counter of each CPU module (n=0: CPU No.1, n=1: CPU No.2, n=2: CPU No.3, n=3: CPU No.4)
D2 to D9	The storage location for values of the synchronization counter of each CPU module
M2 to M9	The operation processing start wait state of each CPU module

- Example of a program to synchronize the start of operation processing



Clock data

CPUs No.2 to No.4 automatically synchronize their clock data to the one set for CPU No.1 (even if setting up clock data individually for each CPU, they will be overwritten). Therefore, simply setting up the clock data for CPU No.1 allows to manipulate a unified clock data across the entire multiple CPU system (☞ Page 118 Time Setting)

Point

As with clock data, the same settings as those for CPU No.1 apply to CPU Nos. 2 to 4.
- Time zone setting (☞ Page 120 Setting Time Zone)
- Daylight saving time setting (☞ Page 121 Daylight Saving Time Function)

16

16.3 Multiple CPU Parameter Checking

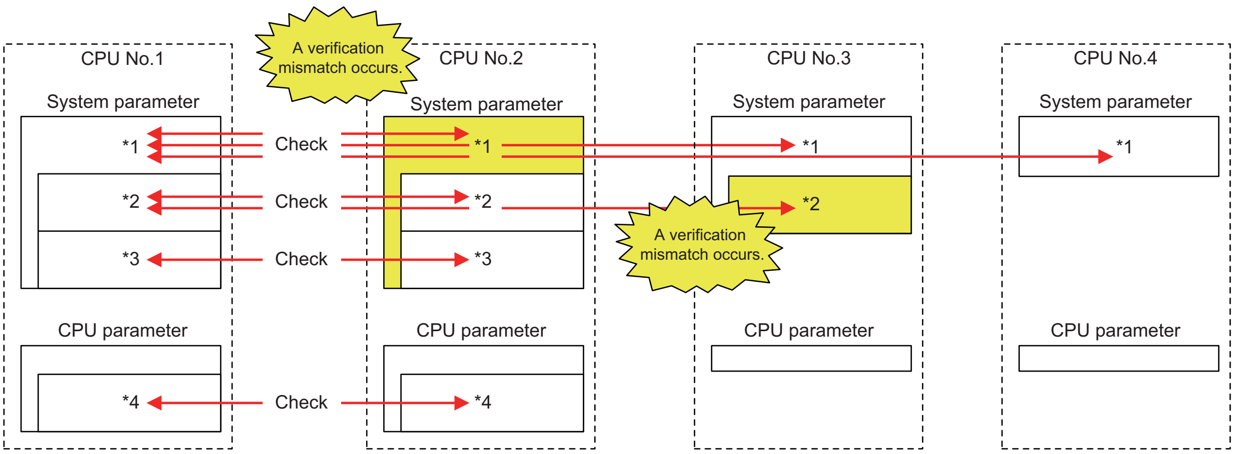
Whether the same setting is configured for between the system parameter of each CPU module and multiple CPU refresh number of points of CPU parameter is checked by the multiple CPU system at the timing shown below. However, as for the fixed scan communication setting and inter-module synchronization setting, checking is done only for the module using the functions.

- At power-on
- At reset of the CPU No.1
- At operating status change (STOP to RUN) after parameter change

Timing	Parameters to check	Checking conditions for CPU No.1	Checking conditions for CPU No.2 and later
At power-on or reset of the CPU No.1	System parameters (other than fixed scan communication setting and inter-module synchronous setting)	Checking is not conducted.	Compares with the parameters of the CPU of the lowest number.
	Settings of fixed scan communication	Checking is not conducted by the CPU module of the number for which the fixed scan communication setting is not configured. The CPU module of the number for which the fixed scan communication setting is configured will compare the parameters with those of the CPU of the lowest number.	
	Inter-module synchronization setting	Checking is not conducted by the CPU module of the number for which the Inter-module synchronization setting is not configured. The CPU module of the number for which the Inter-module synchronization setting is configured will compare the parameters with those of the CPU of the lowest number.	
	CPU parameters (number of points of refresh settings)	Checking is not conducted by the CPU module of the number for which the fixed scan communication setting is not configured. The CPU module of the number for which the fixed scan communication setting is configured will compare the parameters with those of the CPU of the lowest number.	
At operating status change (STOP to RUN) after parameter change	—	Compares with the parameters of the host CPU module before parameters are changed.	

Ex.

An error occurs in the module of each CPU No. 2 and 3, and the module of each CPU No. 1 and 4 starts up normally when parameter checking. (Operation in error)



○: there is setting (if the number immediately next is the same number, it refers to the same parameter) , ×: there is no setting

Item		Setting conditions for the CPUs of each number			
		PLC No.1	PLC No.2	PLC No.3	PLC No.4
With or without the setting for each parameter	System parameters (other than fixed scan communication setting and inter-module synchronous setting)*1	○1	○5	○1	○1
	Settings of fixed scan communication*2	○2	○2	○6	×
	Inter-module synchronization setting*3	○3	○3	×	×
	CPU parameters (number of points of refresh settings)*4	○4	○4	×	×

16.4 Data Communication Between CPU Modules

CPU modules within a multiple CPU system can send and transfer data to each other. The refresh communication and direct access communication enable data writing or reading between CPU modules. The following table lists the data communication method.

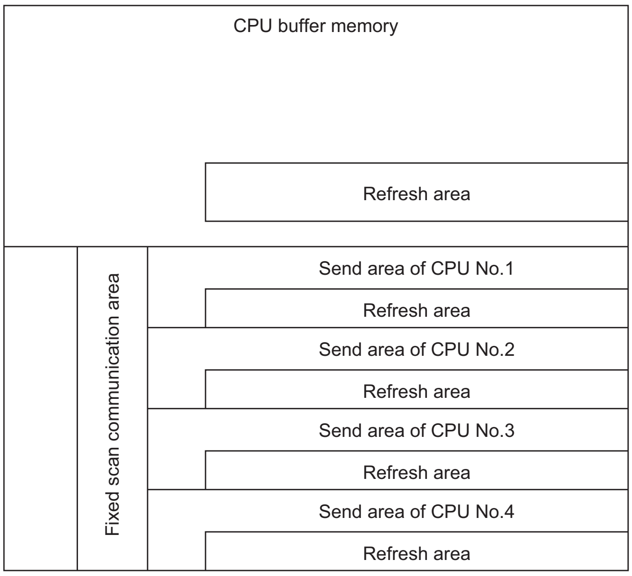
Communication method	Application	Description
Data communication with CPU buffer memory	This method is used when data is sent or received at the timing of each CPU module.	The CPU module for sending the data writes data into the CPU Buffer memory of the host CPU module. The CPU module for receiving data reads data from the CPU Buffer memory of the sender CPU module (other CPU modules).
Data communication with fixed scan communication area	This method is used when data is sent or received through adjusting the timing between CPU modules.	The CPU module for sending the data writes data into the fixed scan communication area (send area) of the host CPU module. The CPU module for receiving data reads data from the fixed scan communication area (receive area) of the host CPU module of the send source CPU module.

Memory to be used

CPU buffer memory is utilized for data communication between the CPU modules.

Memory configuration of CPU buffer memory

This section describes the memory configuration of CPU buffer memory.



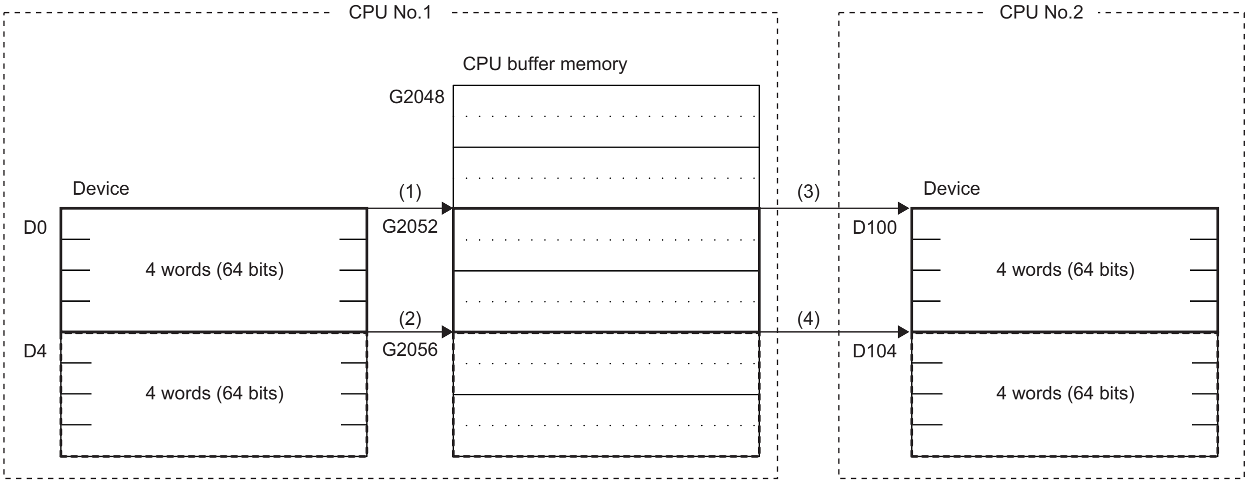
Memory	Communication method	Description	Area size
CPU buffer memory	Communication through direct access	This area is used to read/write data from/to the areas of the host CPU module or other CPU modules. *1 (☞ Page 346 Communication through direct access)	512K words fixed
Refresh area	Communication through refresh	Data communication is conducted through refresh at END processing. (☞ Page 342 Communication through refresh)	
Fixed scan communication area	Communication through direct access	This area communicates the data to the fixed scan communication area of the host CPU module, and is used when the host CPU module area and another CPU module area transfer the data at the fixed scan communication cycles. (☞ Page 346 Communication through direct access)	It is possible to change within the range of 0 to 24K words in total. The send area as per unit can be set within the range of 0 to 12K words. (☞ Page 337 Setting fixed scan communication area)
The Refresh Area in the Fixed Scan Communication Area	Communication through refresh	The refresh operation is conducted at the fixed scan communication cycle. (☞ Page 342 Communication through refresh)	

*1 Data cannot be written to the areas of other CPU modules.

16

■Avoidance of 64-bit data inconsistency

To avoid 64-bit data inconsistency, access the CPU buffer memory by specifying the start address as a multiple of four similarly to the device to be specified.



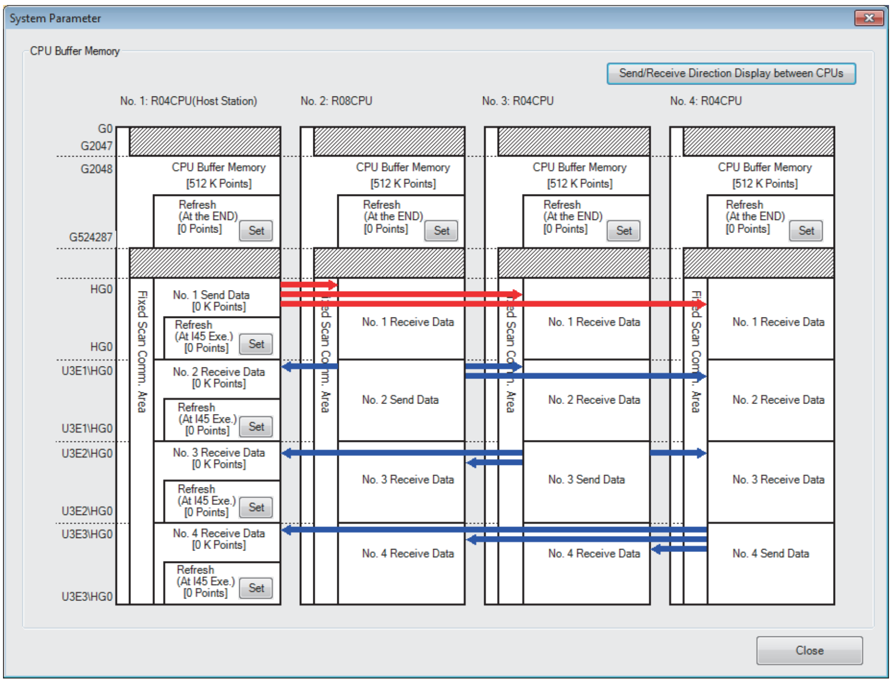
(1) The CPU module assures a 64-bit data and write the data to the CPU buffer memory. (TO U3E0 K2052 D0 K4)
(2) The CPU module assures a 64-bit data and write the data to the CPU buffer memory. (TO U3E0 K2056 D4 K4)
(3) The CPU module assures a 64-bit data and read the data from the CPU buffer memory. (FROM U3E0 K2052 D100 K4)
(4) The CPU module assures a 64-bit data and read the data from the CPU buffer memory. (FROM U3E0 K2056 D104 K4)

Checking for the memory configuration

This section describes the CPU buffer memory configuration of each CPU No. The refresh setting can be configured in both the CPU parameter and the window shown below. (☞ Page 345 Refresh settings)

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Communication Setting between CPUs] ⇨ [CPU Buffer Memory Setting] ⇨ [Detailed Setting]

Window



Displayed items

Item	Description	Setting range	Default
[Set] button of each refresh area	This menu item sets up the refresh settings used for data communication between CPU modules. (The settings are linked with the refresh settings of CPU parameter)	—	0 points
[Send/Receive Direction Display between CPUs] button	Click the button and arrows that indicate the send/receive direction appear.	—	—

Setting the data communication with fixed scan communication area

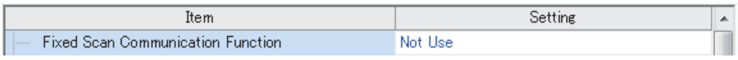
This section describes the setting for making the data communication with fixed scan communication area.

■Setting whether or not it should be used

To communicate data with the fixed scan communication area, "Enable" must be set to "Fixed Scan Communication Function".

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Communication Setting between CPUs] ⇨ [Fixed Scan Communication Function]

Window



Displayed items

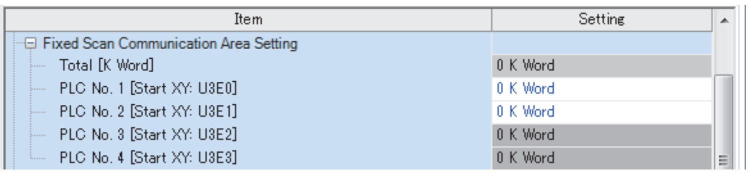
Item	Description	Setting range	Default
Fixed Scan Communication Function	Specifies whether or not the fixed scan communication function should be used.	• Not Use • Use	Not Use

■Setting fixed scan communication area

This menu item sets up the range of send area for each CPU in the fixed scan communication area (the refresh area and the total of areas used for direct access communication). Only the fixed scan communication area can be changed with the parameter settings. The other areas are not configurable.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Communication Setting between CPUs] ⇨ [Fixed Scan Communication Area Setting]

Window



Displayed items

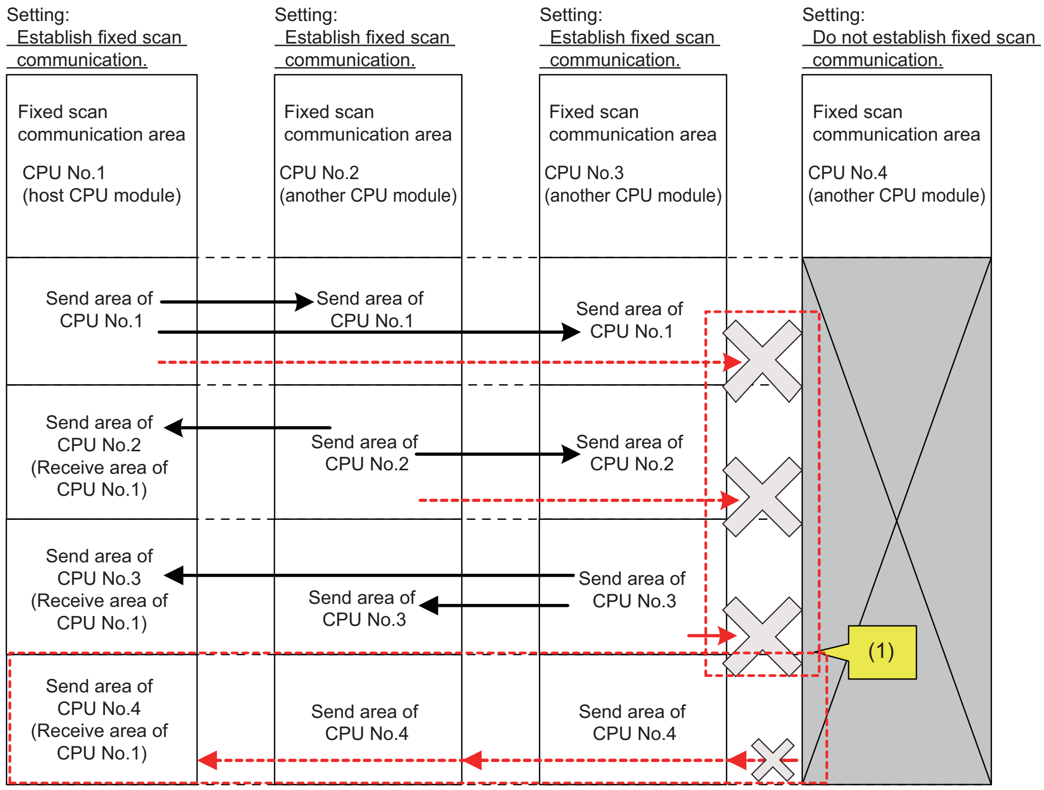
Item	Description	Setting range	Default
Total [K Word]	Displays the total value.	Entire system: 0 to 24K words*1	—
PLC No. 1 [Start XY: U3E0]	Sets the send area size for CPU No.1.	0 to 12K words	0K word
PLC No. 2 [Start XY: U3E1]	Sets the send area size for CPU No.2.	0 to 12K words	0K word
PLC No. 3 [Start XY: U3E2]	Sets the send area size for CPU No.3.	0 to 12K words	0K word
PLC No. 4 [Start XY: U3E3]	Sets the send area size for CPU No.4.	0 to 12K words	0K word

*1 The max area size that can be set through setting of the fixed scan communication area differs depending on the fixed scan communication cycle.

Fixed scan communication cycle	Max area size that can be set through setting of the fixed scan communication area
0.10ms	12K points
0.15ms	20K points
Except for shown above	24K points

Point

When there exists a CPU module for which "Disable" is set to "Fixed Scan Communication Function", if the send area of the fixed scan communication area is set to a CPU module for which "Disable" is set to "Fixed Scan Communication Function" (unspecified) in the parameter setting on the host CPU module, no error is generated because the unspecified CPU module is considered as a reserved one for future configuration.
Example: in four module configuration with the host CPU module is set to CPU No.1 and the fixed scan communication function of CPU No.4 set to "Disable"



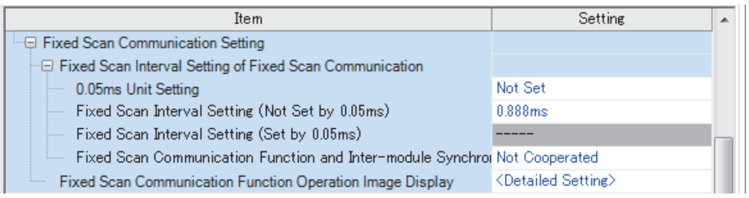
(1) Since the setting does not allow fixed scan communication, this CPU module does not send data to the other CPU modules although send areas have been reserved. This CPU module does not receive data from the other CPU modules.

Fixed scan communication setting

This menu item sets up the interval for data transfer between CPU modules. The data transfer interval can be synchronized with the timing for inter-module synchronization cycle. (MELSEC iQ-R Inter-Module Synchronization Function Reference Manual)

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Fixed Scan Communication Setting] ⇨ [Fixed Scan Interval Setting of Fixed Scan Communication]

Window



Displayed items

Item	Description	Setting range	Default
0.05ms Unit Setting	Specifies whether to set the fixed scan communication cycle in increments of 0.05ms.	• Not Set • Set	Not Set
Fixed Scan Interval Setting (Not Set by 0.05ms)	Selects the fixed scan communication cycle from the drop-down list. The same option should be specified only for CPU modules which will use the fixed scan communication function.	• 0.222ms*1*2 • 0.444ms*1*2 • 0.888ms*2 • 1.777ms • 3.555ms • 7.111ms	• Safety CPU: 7.111ms Other CPU modules: 0.888ms
Fixed Scan Interval Setting (Set by 0.05ms)	Specifies the fixed scan communication cycle freely in increments of 0.05ms.	• R00CPU, R01CPU, R02CPU: 0.50 to 10.00ms • Safety CPU: 1.00 to 10.00ms • Other CPU modules: 0.10 to 10.00ms	• R00CPU, R01CPU, R02CPU: 0.50ms • Safety CPU: 5.00ms • Other CPU modules: 0.10ms
Fixed Scan Communication Function and Inter-module Synchronization Function	Sets whether the fixed scan communication cycle cooperates with the inter-module synchronization cycle.	• Not Cooperated • Cooperate	Not Cooperated

*1 This item is not displayed on the R00CPU, R01CPU, and R02CPU.

*2 The items cannot be specified in the Safety CPU.

Point

The send image for the fixed scan communication can be reviewed through the "Fixed Scan Communication Function Operation Image Display".

Restriction

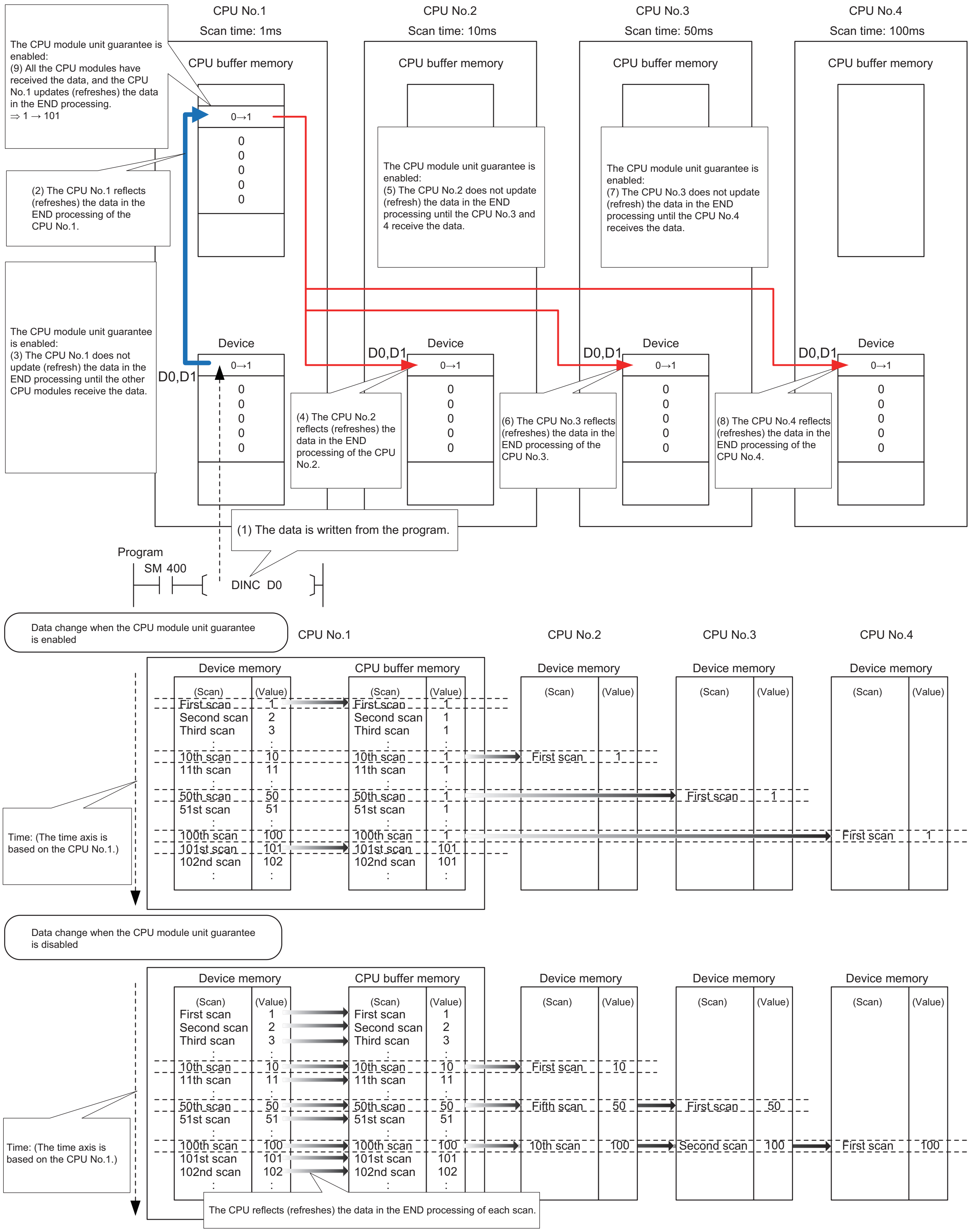
The minimum value for the fixed scan interval setting that can be set in the R00CPU, R01CPU, and R02CPU is 0.50ms.
Therefore, when building the multiple CPU system including the R00CPU, R01CPU, and R02CPU, set the interval to 0.50ms or more for the CPU module to be used with.

Error detection setting

When data is communicated among the CPU modules in a multiple CPU system, some data writing timing may lead to a failure of the complete data communication in the fixed scan communication cycle after the data writing and data inconsistency may occur. In this setting, set whether to detect a continuation error or not in this case. (Page 139 Error detection setting)

Module-by-module data guarantee

In data communication, data is handled in units of 64 bits. Therefore, when data larger than 64 bits is handled, old and new data may be mixed for each CPU module depending on the timing between data reading by the host CPU module and data writing by other CPU modules/data receiving from other CPU modules. (Data inconsistency) To avoid this problem, the system conducts the refresh operation by exchanging handshake signals between the CPU modules for guaranteeing refresh data on a module-by-module basis.



Prevention of data inconsistency by module-by-module data guarantee

The following table shows the preventive control against data inconsistency according to the presence or absence of module-by-module data guarantee.

○: With the preventive control against data inconsistency by the system, ×: Without the preventive control against data inconsistency by the system*1

Communication method	CPU buffer memory		Fixed scan communication area	
	Module-by-module data guarantee enabled	Module-by-module data guarantee disabled	Module-by-module data guarantee enabled	Module-by-module data guarantee disabled
Communication through refresh	○	×	○	×
Communication through direct access	×	×	○*2	×

*1 The countermeasure by the program is required. (☞ Page 349 Data assurance by program)

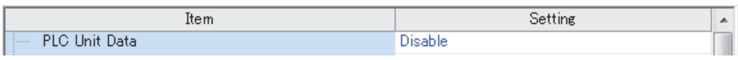
*2 Limited to the case of access within the multiple CPU synchronous interrupt program (I45). (☞ Page 351 When accessing fixed scan communication area)

Module-by-module data guarantee

Set up the module-by-module data guarantee.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Communication Setting between CPUs] ⇨ [PLC Unit Data]

Window



Displayed items

Item	Description	Setting range	Default
PLC Unit Data	Specifies whether data is guaranteed on a module-by-module basis.	• Disable • Enable	Disable

Point

- For communications by the direct access with the CPU buffer memory other than the fixed scan communication area, data is not guaranteed on a module-by-module basis even when this setting is enabled, because the data on the CPU buffer memory of other CPU modules is directly read after execution of the read instruction.
- Data send/receive timing coincides with an update interval of the CPU module with the slowest scan time, because the next send/receive starts only after all CPU modules have completed data reception.

Communication through refresh

The device data for each CPU module is written/read only by the parameter settings. Using refresh areas allows data communication between all or a part of the CPU modules in the multiple CPU system, thereby enabling devices of other CPU modules to be used by the host CPU module.

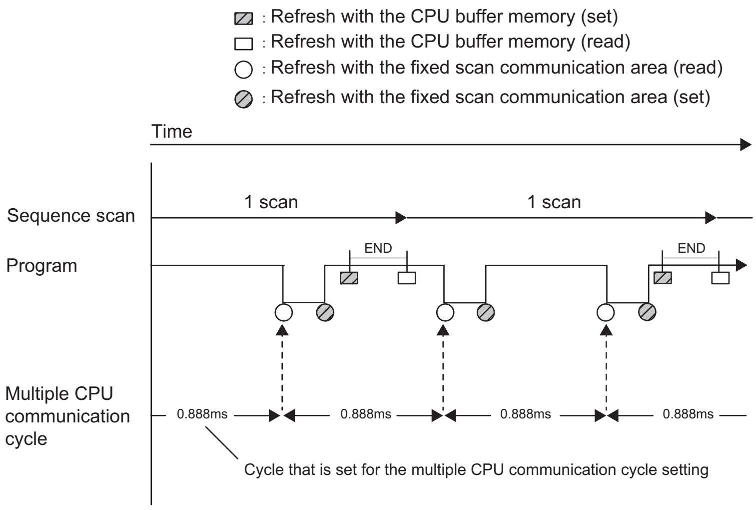
Types of refresh

The following table shows the types of refresh.

Refresh timing	Application	Memory to be used
At the END processing	For data communication of devices in synchronization with the scan	CPU buffer memory
At the execution of multiple CPU synchronous interrupt program (I45)*1*2	For data communication of devices at the period specified in the fixed scan communication setting	Fixed scan communication area

*1 The fixed scan communication area is refreshed on the cycle specified in the fixed scan communication setting.
*2 Refresh fails if the multiple CPU synchronous interrupt program (I45) does not exist on the program.

The following figure shows the timing of each refresh.

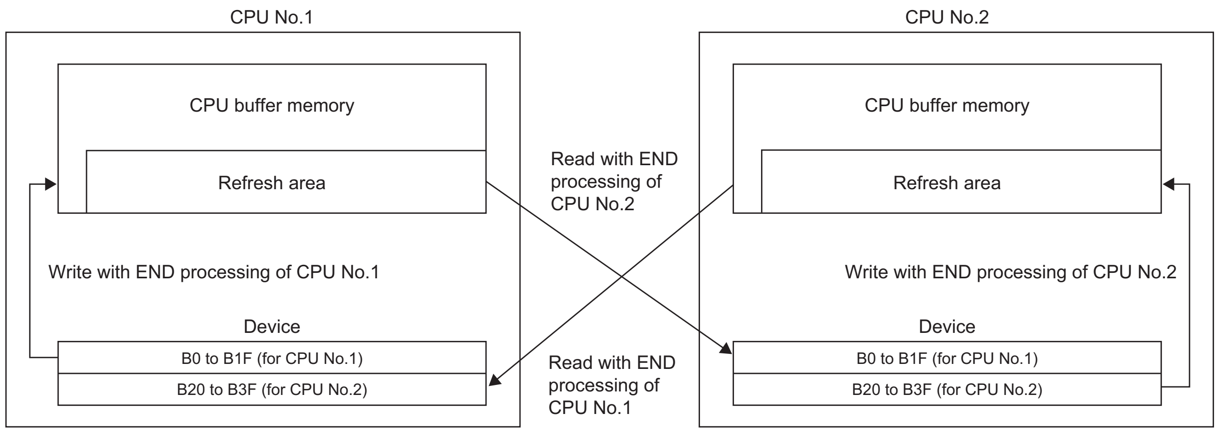


■Refresh using CPU buffer memory

At the END processing of the host CPU module, device data of the host CPU module is written to the refresh area within the CPU buffer memory on the host CPU module. The data written to the refresh area is transferred to the device of another CPU module at the END processing of another CPU module.

Ex.

When CPU No.1 refreshes 32 points (B0 to B1F) and CPU No.2 refreshes 32 points (B20 to B3F):

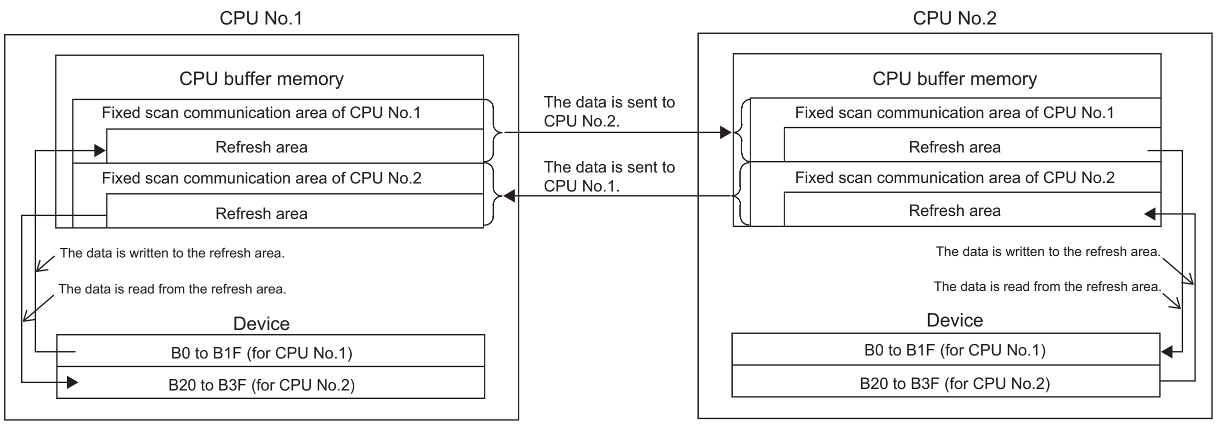


■Refresh using fixed scan communication areas

At the period specified in the fixed scan communication setting, the device data of the host CPU module is written to the refresh area within the fixed scan communication area of the host CPU module. The data written to the refresh area is sent to the refresh area within the fixed scan communication area of another CPU, which in turn reads the transferred data into the device.

Ex.

When CPU No.1 refreshes 32 points (B0 to B1F) and CPU No.2 refreshes 32 points (B20 to B3F):



16

Executing refresh

Refresh is executed when the CPU module is in RUN and/or STOP (PAUSE) state. For details on the behavior when the CPU module is in stop error state, refer to CPU module operation upon error detection setting. (☞ Page 141 CPU module operation upon error detection setting)

■Behavior during the multiple CPU synchronous interrupt program (I45) execution

If refresh is set to be performed during the multiple CPU synchronous interrupt program (I45) execution, the refresh behavior when the CPU module is in RUN state varies depending on either of the following conditions. (☞ Page 353 Multiple CPU Synchronous Interrupt)

- A program (I45 to IRET) which includes the multiple CPU synchronous interrupt program (I45) exists.
- Event execution type program exists.

The following table lists the refresh behaviors.

○: Execute refresh, ×: Not execute refresh

Execution type		Refresh behavior		Refresh behavior after the program control instruction is executed
		STOP state	RUN state	RUN state
Other than event execution type	The I45 interrupt pointer doesn't exist	○	×	×
	The I45 interrupt pointer exists	○	○	○
Event execution type (I45 interrupt specified)		○	○	× (stop after changing the execution type*1)

*1 The PSCAN/PSTOP instruction changes after the next scan, and the POFF instruction changes after the next two scans.

○: Execute refresh, ×: Not execute refresh, —: Execution disabled

Execution type		Refresh behavior triggered by operating status change of the CPU module							
		RUN state	Upon addition of the I45 interrupt pointer during online program change	Upon removal of the I45 interrupt pointer during online program change	RUN to STOP state	STOP state	After the refresh setting is changed; when parameters are written	STOP to RUN state	Power off and on/ Reset
Other than event execution type	The I45 interrupt pointer doesn't exist	×	○	×	○	○	○*3	×	×
	The I45 interrupt pointer exists	○	○	×	○*2	○	○*3	○*3	○*4
Event execution type (I45 interrupt specified)		○	—	—	○*2	○	○*3	○*3	○*4

*2 Not affect to the behavior.

*3 Operates with the pre-change parameters.

*4 Operates with the post-change parameters.

Configurable data

This section lists the configurable data with the refresh settings.

■Maximum number of settings

Maximum of 32 settings can be configured per CPU module for any refresh setting.

■Data number of points

Data number of points can be configured within the number of points assigned to the area (within the send range of the fixed scan communication area) in increments of two points. Device duplication between multiple setting numbers is not allowed.

■Data that can be specified

The device other than local device can be specified. However, when "Use File Register of Each Program" is enabled, file registers cannot be specified. Doing so may prevent the device from operating at file registers for each program depending on the CPU operating status.

Type	Devices that can be specified
Bit device*1	X, Y, M, L, B
Word Device	D, W, R, ZR, RD

*1 This device can be specified only in units of 16 points (one word).

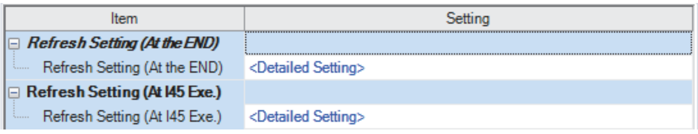
Refresh settings

The refresh can be set up with "Refresh Setting between Multiple CPUs" in "CPU Parameter".

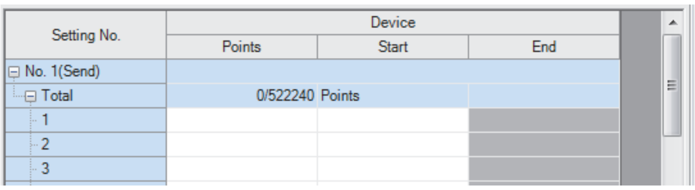
[CPU Parameter] ⇨ [Refresh Setting between Multiple CPUs]

Operating procedure

"Refresh Setting between Multiple CPUs" window



"Detailed Setting" window



1. Click "Detailed Setting" at the execution timing for each refresh.

2. Enter the start/end of the device.

Point

Set a blank column for "Head" and "End" and set only number of points so that the setting by which auto refresh is not conducted can be configured. (It is possible to perform setting which does not allow the refresh data of other CPU modules, which is unwanted for the host CPU module, to be obtained.)

■Enabling timing of refresh settings

The refresh settings are enabled at the following timing:

- When CPU module is powered off and on or is reset.

CPU module operating status and refresh behavior

When the refresh timing coincides with the period specified in the fixed scan communication setting, the refresh behavior depends on the operating status of the CPU module.

CPU module operating status	Refresh enabled/disabled	Refresh timing	
		CPU buffer memory	Fixed scan communication area
RUN (including a continuation error period)	Execute refresh	During the END processing	Before and after the multiple CPU synchronous interrupt program (I45) execution*1
STOP (including a stop error period due to a moderate error)	Execute refresh	During the END processing	During the END processing
STOP (major error period)	Not execute refresh	—	—
PAUSE	Execute refresh	During the END processing	During the END processing

*1 In the disabling interrupt state by the DI instruction, the refresh is not operational, because the multiple CPU synchronous interrupt program (I45) does not work.

16

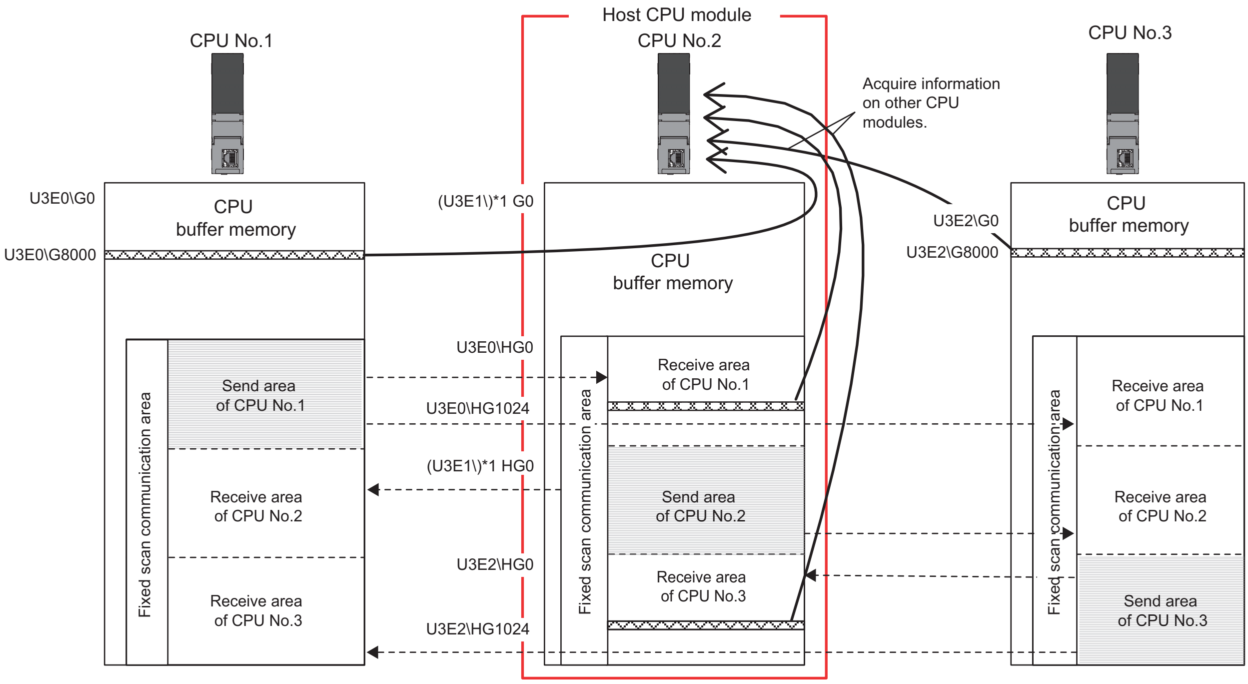
Communication through direct access

This method uses programs to communicate with other CPU modules. The following table lists the communications using the direct access method.

Communication method	Description	Instruction to be used
Communication using CPU buffer memory	Data between CPU modules are transferred using any area on the CPU buffer memory.	• FROM/TO instruction • DFROM/DTO instruction • Instruction using the CPU buffer memory access device (U3En\G□)
Communication using the fixed scan communication area	Data between CPU modules are transferred using the fixed scan communication area.	Instruction using the CPU buffer memory access device (U3En\HG□)
Communication by a dedicated instruction	Data between CPU modules in the multiple CPU system are transferred and/or control commands are sent to other CPU modules by using a dedicated instruction.	DDWR/DDRD instruction and others

Specification method thorough CPU buffer memory access device

Specify the CPU buffer memory as "U3En\Gn" or the fixed scan communication area as "U3En\HGn" when accessing the memory or area. (☞ Page 400 CPU Buffer Memory Access Device)



■Obtaining information stored on the CPU buffer memory

Access the CPU buffer memory.

■Obtaining information stored on the fixed scan communication area for other CPUs

Access the fixed scan communication area on the host CPU module. Obtain data sent to the fixed scan communication area on the host CPU module at the fixed scan communication cycle.

Communication using CPU buffer memory and fixed scan communication area

This section describes the communication using CPU buffer memory and fixed scan communication area.

■Available area for communication

The following area can be used for communication.

Area	Description
CPU buffer memory	All the CPU buffer memory area except for the refresh area is available. The start address of the available area for each CPU module varies depending on the refresh settings. The end address of the area is fixed by CPU module models.
Fixed scan communication area	All the fixed scan communication area except for the refresh area is available. The start address of the available area for each CPU module is HG0 and the end address varies depending on the refresh settings.

■Instructions to be used for communication

Communication with each CPU module is enabled by issuing the following read/write instructions to each area.

- Write instruction: the instructions using the CPU buffer memory access device*1 and the TO/DTO instruction
- Read instruction: the instructions using the CPU buffer memory access device*1 and the FROM/DFROM instruction

*1 Specify "U3En\G□" when accessing the CPU buffer memory and "U3En\HG□" when accessing the fixed scan communication area.

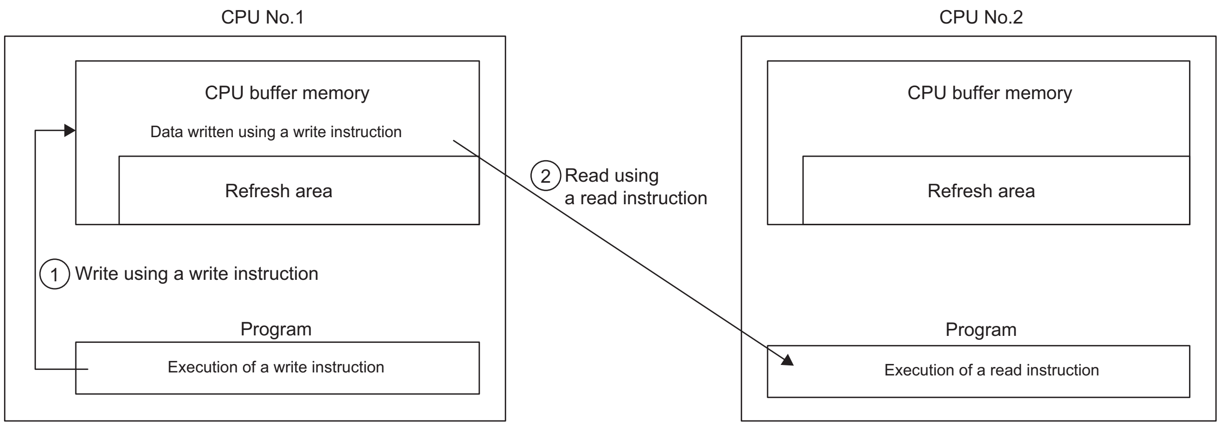
■Data communication behavior

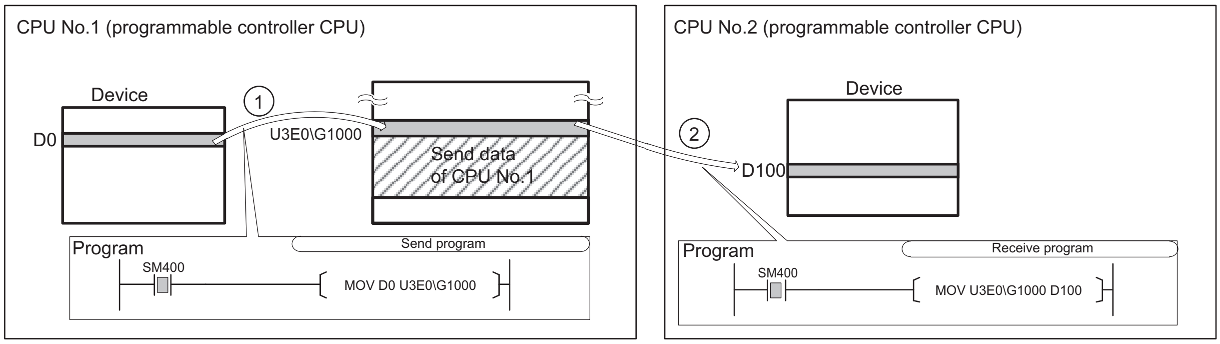
- When using an area within the CPU buffer memory

Data written to the area within the CPU buffer memory on the host CPU module using the write instruction can be read by other CPU modules using the read instruction. Unlike the refresh, data registered during the instruction execution can be directly read.

Ex.

When data written to the CPU buffer memory on the CPU No.1 using the write instruction is read by CPU No.2 using the read instruction:





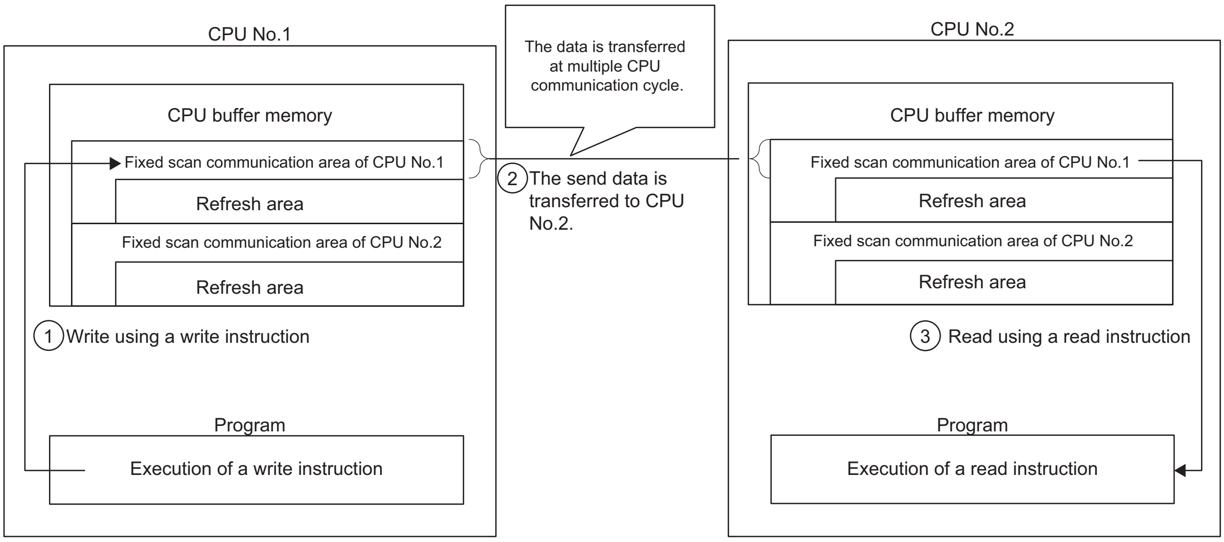
16

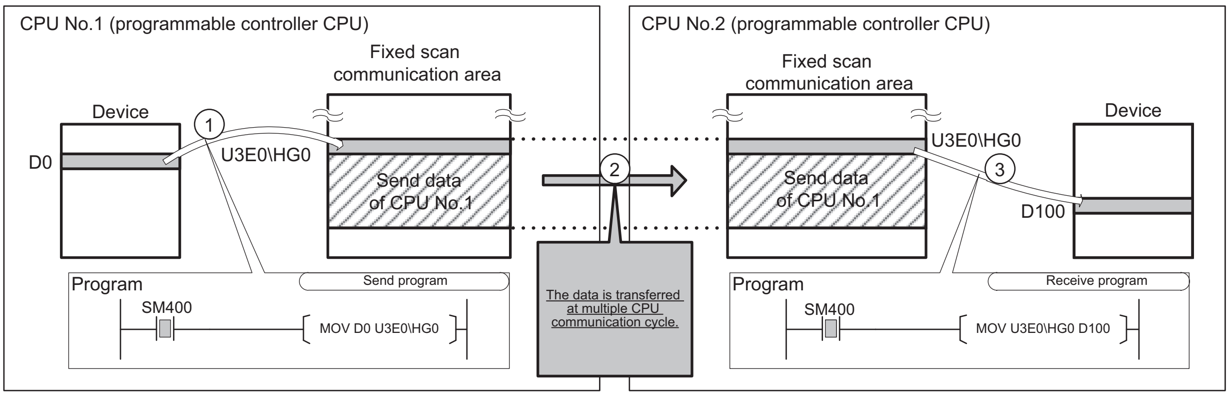
• When using an area within the fixed scan communication area

Data written to the area within the fixed scan communication area on the host CPU module using the write instruction is sent to other CPU modules at the period specified in the fixed scan communication setting. Other CPU modules read the received data using the read instruction. Unlike the refresh, data registered during the instruction execution can be directly read.

Ex.

When data written to the fixed scan communication area on the CPU No.1 using the write instruction is read by CPU No.2 using the read instruction:





Data assurance by program

This section describes how to avoid the inconsistency of data larger than 64 bits using the program. To set up the module-by-module data guarantee using the parameters, use the multiple CPU setting. (☞ Page 341 Module-by-module data guarantee)

Data assurance in communication through the refresh

Inconsistency of transferred data can be avoided by setting the interlock device to a transfer number lower than the one for the transferred data, because data is transferred in descending order from the highest setting number in the refresh settings.

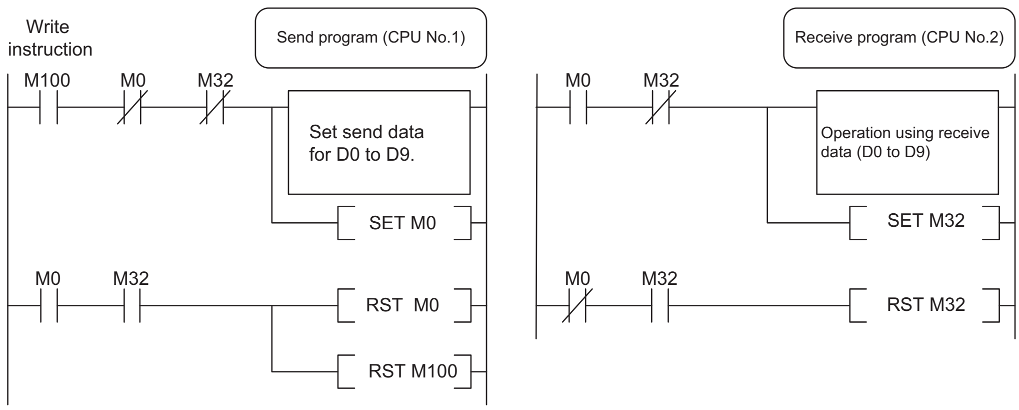
Ex.

Interlock program in communication by refresh

- Parameter settings

CPU No.1 refresh setting							Direction	CPU No.2 refresh setting						
CPU No.	Transfer No.	Send/receive range for each CPU module			Send/receive device setting			CPU No.	Transfer No.	Send/receive range for each CPU module			Send/receive device setting	
		Number of points	start	end	start	end				Number of points	start	end	start	end
CPU No.1	Transfer No.1	2	0	1	M0	M31	→	CPU No.1	Transfer No.1	2	0	1	M0	M31
	Transfer No.2	10	2	11	D0	D9			Transfer No.2	10	2	11	D100	D109
CPU No.2	Transfer No.1	2	0	1	M32	M63	←	CPU No.2	Transfer No.1	2	0	1	M32	M63

- Program example



Data assurance for communication through direct access

The behavior varies depending on the area to be accessed.

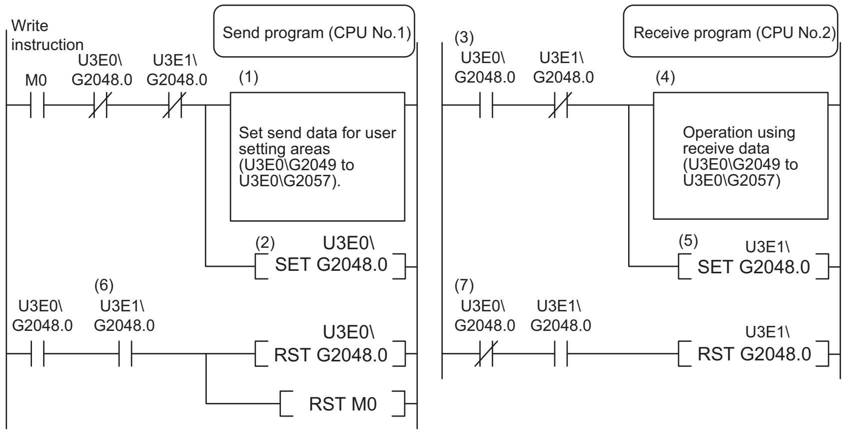
■When accessing CPU buffer memory:

The program reads data in ascending order from the start address of the CPU buffer memory other than the refresh area, and the write instruction writes send data in descending order from the end address of the CPU buffer memory other than the refresh area. Therefore data inconsistency can be avoided by setting an interlock device at the start position of data to be communicated.

Ex.

Interlock program in communication by direct access (when accessing CPU buffer memory)

- Program example



(1) CPU No.1 creates send data.

(2) CPU No.1 turns on the data setting complete bit.

[Data transfer with CPU No.2 END processing]

(3) CPU No.2 detects send data setting complete.

(4) CPU No.2 performs receive data processing.

(5) CPU No.2 turns on receive data processing complete.

[Data transfer with CPU No.1 END processing]

(6) CPU No.1 detects receive data processing complete, and turns off the data setting complete bit.

[Data transfer with CPU No.2 END processing]

(7) CPU No.2 detects that send data setting complete is turned off, and turns off receive data processing complete.

■When accessing fixed scan communication area

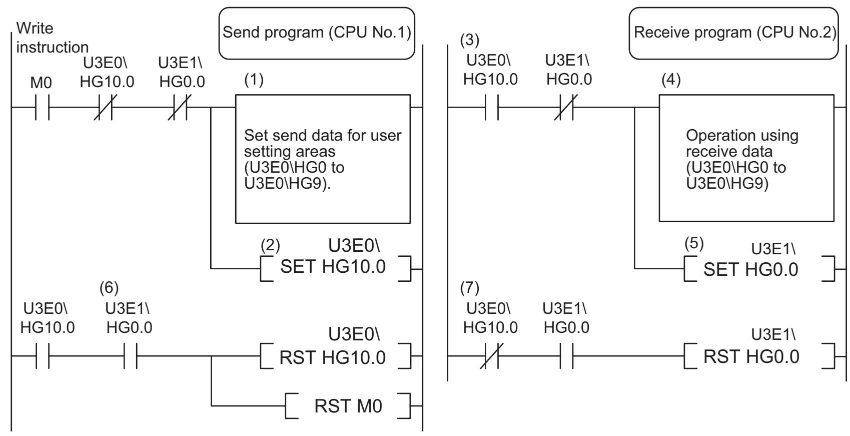
When accessing within the multiple CPU synchronous interrupt program (I45), enabling the setting of module-by-module data guarantee eliminates the need of an interlock circuit. When this refresh area is accessed within a program other than the above, or when the setting of module-by-module data guarantee is disabled, an interlock circuit is required, as with the access to the CPU buffer memory. (☞ Page 340 Module-by-module data guarantee)

The program reads data by transferring it in the order that it is written to the CPU buffer memory (fixed scan communication area). Data inconsistency can be prevented by using devices written after the transfer data for interlocks, regardless of the device type or address.

Ex.

Interlock program in communication by direct access (when accessing fixed scan communication area)

- Program example



(1) CPU No.1 creates send data.
(2) CPU No.1 turns on the data setting complete bit.
[Data transfer with multiple CPU communication cycle]
(3) CPU No.2 detects send data setting complete.
(4) CPU No.2 performs receive data processing.
(5) CPU No.2 turns on receive data processing complete.
[Data transfer with multiple CPU communication cycle]
(6) CPU No.1 detects receive data processing complete, and turns off the data setting complete bit.
[Data transfer with multiple CPU communication cycle]
(7) CPU No.2 detects that send data setting complete is turned off, and turns off receive data processing complete.

Also, with instructions such as BMOV instructions that involve writing data with two or more words to the CPU buffer memory, data is written from the end address to the start address. If combining and writing send data with interlock signals with a single instruction, data inconsistency can be prevented with an interlock signal at the start of the data.

16

Communication between CPU modules in error state

The following section describes communication between CPU modules in an error state.

Behavior in receive data error state

A CPU module receiving illegal data due to noise and/or failure discards the received data. If a received data is discarded, the receive-side CPU module keeps the last data received before discarding. When the CPU module receives the next correct data, it updates the received data.

Refresh execution in an error state

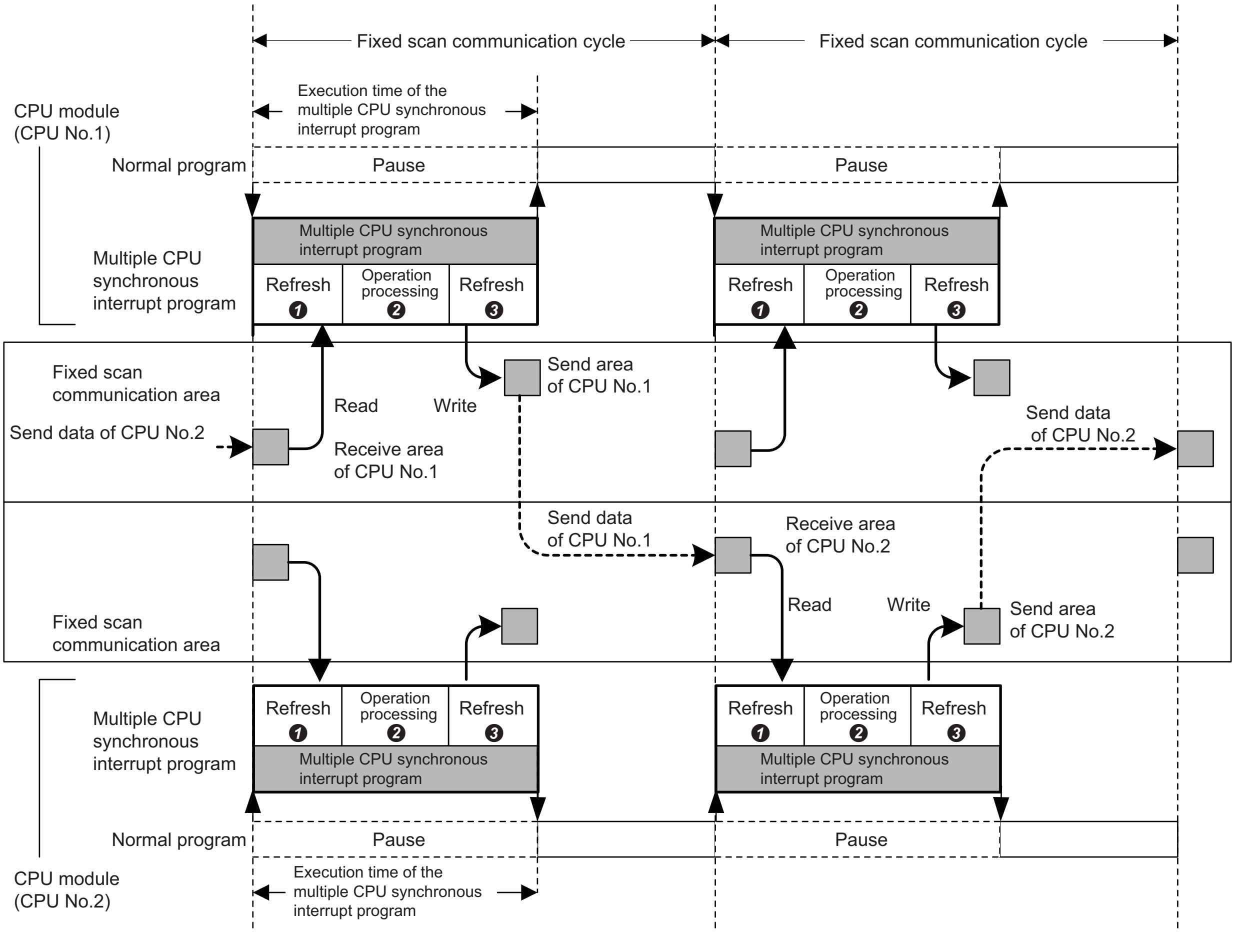
The following table lists the refresh and send/receive operation between CPU modules when the host CPU module detects a self diagnostic error. If one of the CPU modules enters into stop error state, the other CPU modules which are not in stop error state keep data stored before the stop error occurs.

Error		Refresh*1	Data communication between CPU modules*2
Minor error		○	○
Moderate error	Causes other than the following item	○	○
	Parameter error for fixed scan communication function (including the consistency check during start-up)	×*3*4	×*3*4
Major error		×*3	×*3

*1 This item indicates data transfer between user devices and the fixed scan communication area on the host CPU module.
*2 This item indicates data communication between the fixed scan communication areas on the host CPU module and other CPU modules.
*3 When an error occurs during normal operation, normal data generated immediately before the error occurs is continued to be sent between the fixed scan communication areas on the host CPU module and other CPU modules.
*4 If the consistency check fails due to a parameter change in normal operation, refresh and data send/receive between the CPU modules are continued.

16.5 Multiple CPU Synchronous Interrupt

This function triggers an interrupt program at the fixed scan communication cycle set in a parameter. An interrupt program executed at the fixed scan communication cycle is called a multiple CPU synchronous interrupt program. Using the multiple CPU synchronous interrupt enables synchronizing with the fixed scan communication cycle so that data communication between CPU modules can be made. (It allows synchronizing the data communication timing between CPU modules.)



❶ Data reading of other CPU modules (refresh): Data sent from other CPU modules is read to a device or a label. (Data is read from the receive area of the host CPU module.)

❷ Operation processing: The multiple CPU synchronous interrupt program is executed.

❸ Data sending of other CPU modules (refresh): Data to be sent to other CPU modules is written from a device or a label. (Data is written to the send area of the host CPU module.)

Point

The operation methods required when an interrupt factor occurs and the program creating methods are the same as those for normal interrupt program. (☞ Page 74 Interrupt Program)

16

Execution timing

The multiple CPU synchronous interrupt program (I45) is executed at the timing for the fixed scan communication cycle. The fixed scan communication cycle can be changed through the fixed scan communication setting. (☞ Page 339 Fixed scan communication setting)

Point

It is also possible to perform refresh during the multiple CPU synchronous interrupt program (I45) in execution. (☞ Page 342 Communication through refresh)

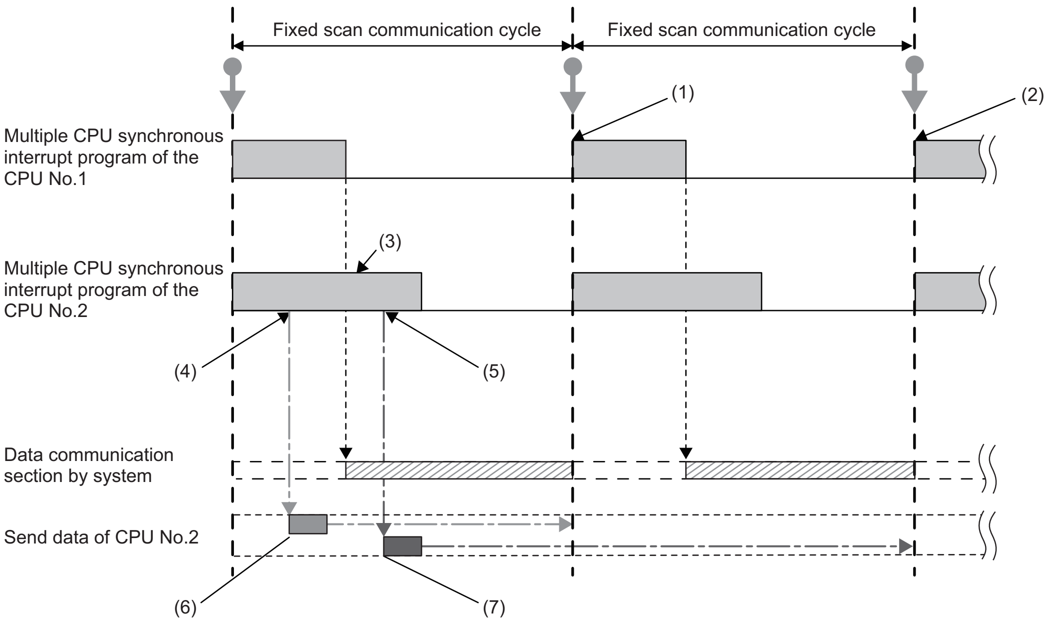
Multiple interrupt

For the multiple interrupt of the multiple CPU synchronous interrupt program, refer to the multiple interrupt function. (☞ Page 89 Multiple interrupt function)

Precautions

The precautions for the multiple CPU synchronous interrupt program are mentioned below.

- Create a multiple CPU synchronous interrupt program so that it has the execution processing time shorter than the fixed scan communication cycle. If the interrupt program has the execution processing time equal to or longer than the cycle, the multiple CPU synchronous interrupt interval cannot be guaranteed. (☞ Page 75 Operation upon occurrence of an interrupt factor) The execution time of the multiple CPU synchronous interrupt program (I45) can be monitored using the RAS setting of the CPU parameter. (☞ Page 139 Error detection setting)
- To send data successfully in the next fixed scan communication cycle, select "Detect" for "Program Execution Section Exceed (I45)" in "RAS Setting" of [CPU Parameter]. (☞ Page 139 Error detection setting) With this setting, when data is written after the host CPU module starts data transfer in the multiple CPU synchronous interrupt program, SM484 (Execution section excess error flag for multiple CPU synchronization interrupt program) is turned on and the number of data sending errors in the next cycle is stored in SD484 (Number of execution section excess errors for multiple CPU synchronization interrupt program). In addition, when the data is written while the data cannot be sent in the fixed scan communication cycle set in a parameter, an error can be detected. (The CPU module continues its operation.)



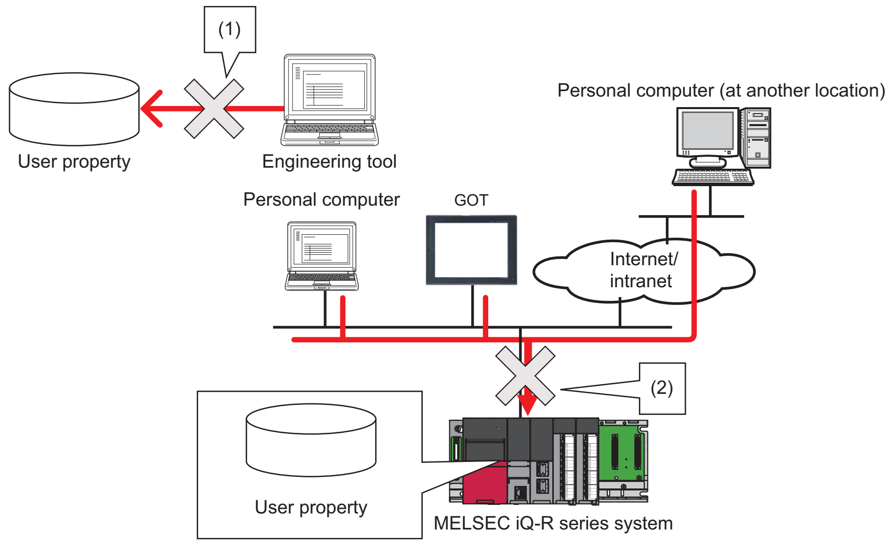
(1) CPU No.1 receives send data (A) of CPU No.2 in the next scan.
(2) CPU No.1 receives send data (B) of CPU No.2 after two scans.
(3) SM484 turns on and SD484 counts up because CPU No.2 continuously executes its program even after the start of data communication.
(4) Send data (A) of CPU No.2 is written.
(5) Send data (B) of CPU No.2 is written.
(6) This data is sent in the next scan because it has been written before data update.
(7) This data is sent after two scans because it has been written after data update.

17 SECURITY FUNCTION

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

• Only the SIL2 Process CPU and Safety CPU support the user authentication function.

This function serves to protect the user property stored in a personal computer and the user property inside modules in the MELSEC iQ-R series system against threats such as theft, tampering, faulty operation, and unauthorized execution due to the unauthorized access by an outsider. Use an appropriate security function according to the purpose as shown in the following table:



(1) Protection is provided against theft, tampering, and faulty operation resulting from unauthorized access by an outsider. (Protection for user property on a personal computer)

(2) Protection is provided against theft, tampering, faulty operation, and unauthorized execution resulting from unauthorized access by an outsider. (Protection for user property in CPU modules)

Data to be protected		Purpose	Function	Reference
GX Works3	Project	Prevents unauthorized access to programs (in units of POU). (A password is used.)	Block password function	GX Works3 Operating Manual
		Prevents unauthorized access to programs (in units of program file). (A security key is used.) Project data locked with a security key can only be viewed with an engineering tool for which the same security key has been registered.	Security key authentication function for a project	
		Restricts access to projects on the personal computer with a combination of user name and password. The security level can be set individually for each user.	User authentication function for a project	

Data to be protected		Purpose	Function	Reference
CPU module	Program	Prevents unauthorized execution of programs. (A security key is used.) Programs locked with a security key can only be executed at modules for which the same security key has been set.	Security key authentication function for a CPU module	Page 357 Security key authentication for a CPU module
	Program, parameter	Restricts access to programs or parameters in the CPU module with a combination of user name and password. The security level can be set individually for each user.	User authentication function of CPU modules*1	• Page 653 User Authentication Function of CPU Modules • Page 763 User Authentication Function of CPU Modules
	File	Prevents unauthorized read and write of files. (A password is used.)	File password function*1	• GX Works3 Operating Manual • MELSEC iQ-R Ethernet User's Manual (Application) • SLMP Reference Manual • MELSEC iQ-R Serial Communication Module User's Manual (Application)
	Access	Blocks access from an invalid IP address by identifying the IP address of an external device via Ethernet.	IP filter function	MELSEC iQ-R Ethernet User's Manual (Application)
		Restricts access via Ethernet that is taking a different route from specific communication routes. (A password is used.)	Remote password function*1	
	Device data	Protects arbitrary device data from being tampered with. (Writing to devices from engineering tools and GOTs is disabled.)	Write-protect function for device data (from outside the CPU module)	• Page 358 Write-Protect Function for Device Data (from Outside the CPU Module) • GX Works3 Operating Manual

*1 These functions disable the password authentication for a certain duration of time after a certain number of failed authentication attempts. For details, refer to the manuals for each function.

Point

If a personal computer with a security key registered is abused by an outsider, there is no way to prevent the outflow of the program property, and thus the user needs to take adequate measures as shown below:

- Preventive measures against the theft of a personal computer (for example, wire locking)
- Management of users of the personal computer (for example, deletion of unnecessary user accounts, strict control of login information, and implementation of fingerprint authentication)

Furthermore, if a personal computer with a security key registered has failed, the locked project data cannot be viewed and edited. We assume no responsibility whatsoever for any damage or loss to the user and any other individual or organization, resulting from such a situation. Therefore, the user needs to take adequate measures as shown below:

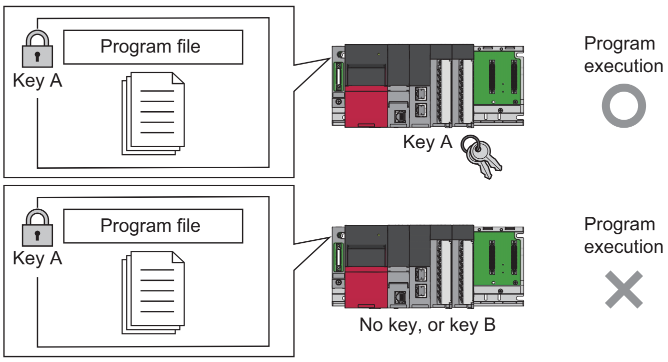
- Import the registered security key to another personal computer.
- Export the registered security key to a file and store the file in a safe place.

The following information in the CPU module can be deleted by selecting [Online] ⇨ [User Authentication] ⇨ [Initialization of all PLC Data].

- User management information set with user authentication function (user authentication function returns to disabled status)
- Security key
- All program memory, data memory files
- All files in file storage areas in device/label memory
- Safety CPU safety operation mode status (returns to TEST MODE)

Security key authentication for a CPU module

This function prevents an unauthorized execution of programs written in a CPU module.
The operations are restricted by detecting a mismatch between the security key of the program file written in a CPU module and the security key of a CPU module.



If the security key of only one program does not match the security key of the CPU module, all the other programs in the CPU module are not executed as well.
The security key written in the CPU module is retained even after power-off of the CPU module.
For details, refer to the following.

GX Works3 Operating Manual

17.1 Write-Protect Function for Device Data (from Outside the CPU Module)

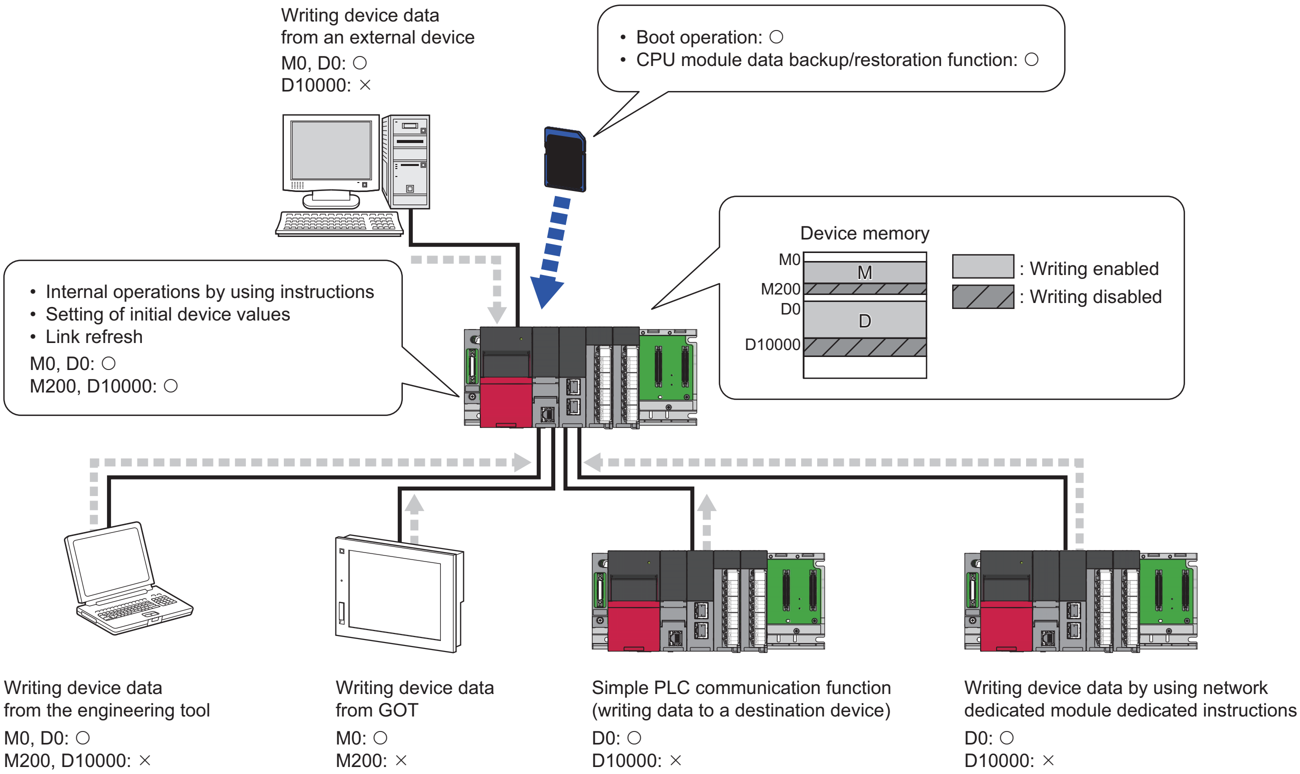
RnCPU RnENCPU ~~RnPCPU (Process)~~ ~~RnPCPU (Redundant)~~ ~~RnPSFCPU (Standard)~~ ~~RnPSFCPU (Safety)~~ RnSFCPU (Standard) ~~RnSFCPU (Safety)~~

- When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

This function disables writing to devices from engineering tools and GOTs. Protected devices and their ranges can be set by the user. Therefore, desired device data can be protected from tampering.

Even when the write-protect range is set, the operation of the set CPU module and device data writing by the execution of functions (internal device data writing such as device data writing by instructions) are not disabled.



○: Can be written, ×: Cannot be written

Point

This function applies to a global label to which a device is assigned. Therefore, when a device assigned to a global label is protected from writing, write processing cannot be performed.

Restriction

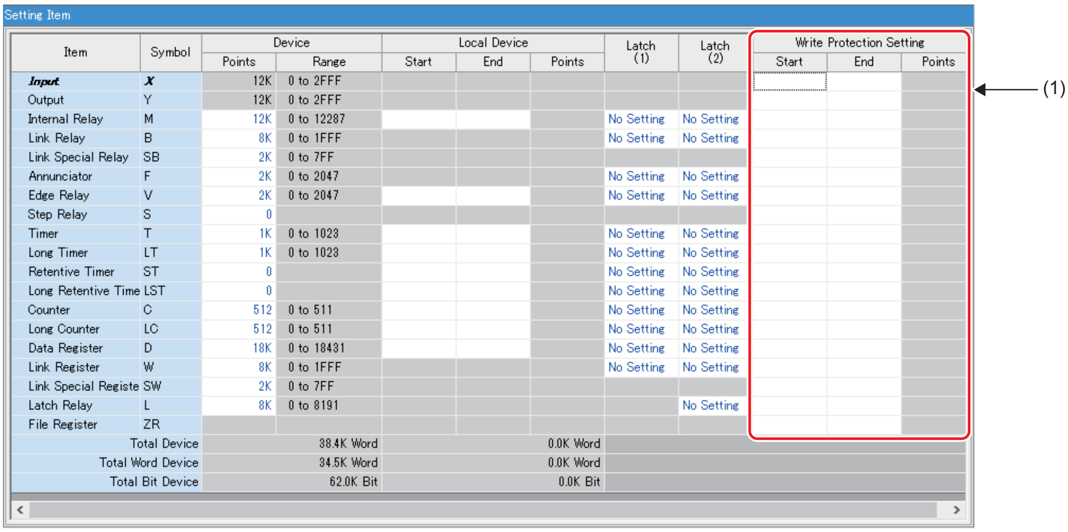
When using the write-protect function for device data (from outside the CPU module), check the versions of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

Setting method

Set the write-protect range in "Device Setting" of the CPU parameter.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Detailed Setting] ⇨ [Device Setting] ⇨ [Detailed Setting]

Window



(1) One range can be set per device by specifying "Start" and "End".*1

*1 The device assignment method can be changed to "Points/Start" by right-clicking in the setting window and selecting "Setting Method".

Target devices

The following table lists the devices to which write-protect ranges can be set with this function. Note that the following devices includes the digit-specified bit devices and the bit-specified word devices.

Type		Device
User device	Bit	X*1, Y*1, M, B, F, SB, V, S, L
	Bit/word	T, ST, C
	Bit/double word	LT, LST, LC
	Word	D, W, SW
File register	Word	R*2, ZR

*1 X and Y that are accessed in direct mode (DX and DY) are also included.

*2 Specify the write-protect range for the file register in ZR.

Operations and functions

The following table lists the operations and functions that cannot be executed for devices in the write-protect range.

Operation and function			
Operation from the engineering tool	Current value change in the watch window		
	Remote operation	Remote RUN[*1]	
	CPU memory operation	Device/label memory value clear	Device/label zero clear
			File register zero clear
			Device/label/file register latch clear
	Write to PLC	Device memory	
		Device initial value[*2]	
		File register	
Operation using a GOT	Device write		
	File write		
Operation using SLMP	Device write using SLMP		
	File write/change using SLMP[*3]		
Operation using FTP	File write using the file transfer function (FTP server)[*3]		
Operation using instructions	Device write from another CPU module		
	Device write from the programmable controller of another station		
	Remote RUN at device memory clear specification from the programmable controller of another station		
Other operations	Device write using the simple CPU communication function		
	Predefined protocol support function		
	Device test using the Web server function[*4]		

*1 This function cannot be executed when the device memory clear is specified.
*2 When this function is enabled, the files cannot be written.
*3 The file register file and initial device value file cannot be written or changed.
*4 In "Account Settings" of "Web Server Settings", set "Write Device" to "Enable".

■Operation using SLMP

Device write processing is disabled when the write-protect range is set in the device write using SLMP. In addition, device write processing is disabled when the following commands are executed in the operations such as the predefined protocol support function, SLMP frame send instruction, or access from an external device.

Type	Operation	Command	Subcommand
Device	Write	1401	00□0, 00□1
			00□2, 00□3
	Write Random	1402	00□0, 00□1
			00□2, 00□3
	Write Block	1406	00□0
			00□2
Label	Array Label Write	141A[*1]	0000
	Label Write Random	141B[*1]	0000
Remote Control	Remote Run	1001[*2]	0000
	Remote Latch Clear	1005	0000

*1 Device write processing is disabled only when a device is assigned to a label.
*2 Device write processing is disabled only when the device memory clear is specified.

Write processing is disabled when a file write using SLMP is performed to the device initial value file and file register file. In addition, write processing is disabled when the following commands are executed in the operations such as the SLMP frame send instruction or access from an external device.

Type	Operation	Command	Subcommand
File	New File	1820	0000, 0040
	Copy File	1824	0000, 0004, 0040
	Open File	1827[*3]	0000, 0004, 0040
	Write File	1829	0000

*3 Write processing is disabled only when the open mode is for writing (0100H).

■Operation using the file transfer function (FTP server)

Write processing is disabled when a file write using the file transfer function (FTP server) is performed to the initial device value file and the file register file.

Function	Command	Subcommand
Writing to file	put	—
	mput	—
File name change	rename*1	—
FTP server subcommand send (RUN operation)	quote*2	run

*1 Write processing is disabled when the file extension is changed to DID or QDR.
*2 Write processing is disabled only when the device clear is specified.

■Operation using instructions

Write processing is disabled when the device write is performed by a multiple CPU dedicated instruction or a module dedicated instruction (including a device clear instruction (dedicated instruction)) to devices in the write-protect range.

Major classification	Classification		Instruction name
Writing device data from another station	Module dedicated instruction	Writing data to the programmable controller on another station	JP.WRITE, GP.WRITE
		Writing data to the programmable controller on another station (with notification)*1	JP.SWRITE, GP.SWRITE
		Reading data from the programmable controller on another station (with notification)*1	JP.SREAD, GP.SREAD
		Writing data to the target station	J(P).RIWT, G(P).RIWT
Writing device data from another CPU module	Multiple CPU dedicated instruction	Writing device data to another CPU module	D(P).DDWR, M(P).DDWR
Clearing device data another station	Module dedicated instruction	Remote RUN*2	J(P).RRUN, G(P).RRUN, Z(P).RRUN, J(P).REQ, G(P).REQ

*1 Writing to notification devices is also disabled.
*2 Write processing is disabled only when the device clear is specified.

Precautions

- When this function is enabled, a device write operation with indirect specification cannot be performed. Perform a device write operation with normal device specification. (Do not use indirect specification.)
- When this function is enabled and writing to the file register is disabled, perform a device write operation specifying the ZR device. If the R device is specified, a device write operation may not be disabled.
- When this function is enabled, a device write operation with index modification cannot be performed when the write-protect range is set in the accessible area. When the write-protect range exists in the accessible area, perform a device write operation with normal device specification. (Do not use index modification.)
- When this function is enabled, CPU parameters that disables the function and the device memory where values are set to the write-protect device in the CPU module cannot be written to the programmable controller at the same time. Write CPU parameters to disable the function, and then write the device memory where the values are set to the programmable controller.

18 SEQUENCE SCAN SYNCHRONIZATION SAMPLING FUNCTION

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

This function allows a module to collect data of the CPU module in synchronization with the sequence scan of the CPU module.

This function is available to the following modules:

- MES interface module (high-speed access)
- High speed data logger module (high-speed collection)
- C intelligent function module (data sampling in sequence scan)
- High speed data communication module (high-speed collection)

The following table lists the number of modules that can execute this function on a single CPU module and the total number of collectable points.

Item	R00CPU, R01CPU, R02CPU	Other CPU module
Number of executable modules	Two	Four
Total number of collectable points	16K points	32K points

For details on the function using the sequence scan synchronization sampling function and collectable data for each module, refer to the manual for the module used.

Manual for the module used

Restriction

When using the sequence scan synchronization sampling function, check the firmware versions of the CPU module and the modules used. (Page 1139 Added and Enhanced Functions, Manual for the module used)

Precautions

Use points in 8K-point increments to use all of the total number of points that can be collected.

The points are assigned in 8K-point increments as follows:

- 6K points used → 8K points to be assigned
- 9K points used → 16K points to be assigned

MEMO

19 LABEL INITIALIZATION FUNCTION

~~RnCPU~~ ~~RnENCPU~~ RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) ~~RnSFCPU (Standard)~~ ~~RnSFCPU (Safety)~~

• When using a SIL2 Process CPU, refer to the following as well.

☞ Page 704 FUNCTIONS

In the Process CPU and SIL2 Process CPU, the labels assigned to label areas will be initialized (Initial values are set if the values have been set, or the labels are cleared to zero if not) when the CPU module is powered off and on or the operating status of the CPU module is switched from STOP to RUN after data is rebuilt (reassigned) and then written to the programmable controller.

19.1 Initialization of Labels After Rebuilt All (Reassignment)

In the Process CPU and SIL2 Process CPU, the labels assigned to label areas will be initialized (Initial values are set if the values have been set, or the labels are cleared to zero if not) when the CPU module is powered off and on or the operating status of the CPU module is switched from STOP to RUN after data is rebuilt (reassigned) with the engineering tool and then written to the programmable controller.

Point

Labels are initialized only when the data is written to the programmable controller for the first time after data is rebuilt (reassigned). Thus, to write the same project to the other CPU module, rebuild (reassign) the data again.

Restriction

This function is supported by the Process CPU and SIL2 Process CPU. Before using this function with the Process CPU, check the versions of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

Label initialization operation

The following table lists the label initialization operations.

Target label		Label initial value setting	Label initialization operation*3	
			When the CPU module is powered on	When the operating status of the CPU module is switched from STOP to RUN*2
Standard label	Labels outside the latch range	Set	Initial values are set.	Initial values are set.
		Not set	Zero clear*1	Zero clear*1
	Latch type labels	Set	Initial values are set.	Initial values are set.
		Not set	Zero clear*1	Zero clear*1
Safety label and standard/safety shared label		Set	— (No setting)	— (No setting)
		Not set	Zero clear*1	Zero clear*1

*1 The labels (whose attributes are other than CONSTANT) to be assigned to label areas are cleared to zero.
*2 This operation is also performed when the operating status of the CPU module is switched from PAUSE to RUN.
*3 Since the labels assigned to devices are assigned to device areas, the labels are not initialized by this function.

Operating procedure

The following describes the operating procedure for this function.

1. Rebuild (reassign) all data.

[Convert] ⇨ [Rebuild All]

2. Set the CPU module to the STOP state.

3. Write the new program files. When initial values have been set in the labels used in the program, write the label initial value file together with the program files.

[Online] ⇨ [Write to PLC]

4. Set the CPU module to the RUN state. (Labels are initialized only when the operating state is changed from STOP to RUN for the first time.)

5. Labels are not initialized at the second state change or later. Label initial values can be set at the second state change or later depending on the parameter settings. (☞ Page 366 Label Initial Value Reflection Setting)

Point

Since this function automatically initializes labels (zero clear), the reset operation after writing is unnecessary.

Restriction

Labels are initialized only when the data is written to the programmable controller for the first time after data is rebuilt (reassigned). To write the same project to other programmable controller, rebuild (reassign) the data again before writing.

Precautions

The following describes precautions on this function.

- Even though SM326 (SFC device/label clear mode) is on (device/label values are kept), only labels are initialized when data is rebuilt and then written to the programmable controller. (Devices are kept.)
- Before boot operation, clear the value of the latch label. Even if the boot file after rebuilt all (reassignment) is written to the SD memory card and boot operation is executed, label initialization function does not operate.
- Once all data is rebuilt (reassigned) and written, the label initialization cannot be canceled. For example, when all data is written to the programmable controller without being rebuilt (reassigned) again after the data is rebuilt (reassigned) and written to the programmable controller, labels will be initialized after the CPU module is powered on or the operating status of the CPU module is switched from STOP to RUN.

19.2 Label Initial Value Reflection Setting

With the default settings of the Process CPU and SIL2 Process CPU, initial label values are not set in labels when the operating status of CPU module is switched from STOP to RUN even though the label initial values have been set for the labels.
This function can set whether or not to set label initial values when the operating status of the CPU module is switched from STOP to RUN.

Restriction

This function is supported by the Process CPU and SIL2 Process CPU. Before using this function with the Process CPU, check the versions of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

Label initialization operation

The following table lists the label initialization operations.

Target label		Label initial value setting	Label initialization operation*4			
			When "Label Initial Value Reflection Setting" has been set to "Disable"		When "Label Initial Value Reflection Setting" has been set to "Enable"	
			When the CPU module is powered on	When the operating status of the CPU module is switched from STOP to RUN*3	When the CPU module is powered on	When the operating status of the CPU module is switched from STOP to RUN*3
Standard label	Labels outside the latch range	Set	Initial values are set.*1	Initial values are not set. (Values do not change.)	Initial values are set.	Initial values are set.
		Not set	Zero clear*2	Values do not change.	Zero clear*2	Values do not change.
	Latch type labels	Set	Initial values are not set. (Values do not change.)	Initial values are not set. (Values do not change.)	Initial values are set.	Initial values are set.
		Not set	Values do not change.	Values do not change.	Values do not change.	Values do not change.
Safety label and standard/safety shared label		Set	— (No setting)	— (No setting)	— (No setting)	— (No setting)
		Not set	Zero clear*2	Values do not change.	Zero clear*2	Values do not change.

*1 Since values in labels outside the latch range are cleared when the CPU module is powered off, initial values are set in the labels when the CPU module is powered on even though "Label Initial Value Reflection Setting" has been set to "Disable".
*2 The labels (whose attributes are other than CONSTANT) to be assigned to label areas are cleared to zero.
*3 This operation is also performed when the operating status of the CPU module is switched from PAUSE to RUN.
*4 Since the labels assigned to devices are assigned to device areas, the labels are not initialized by this function.

Setting procedure

The following describes the procedure of disabling label initial values to be set when the operating status of the CPU module is switched from STOP to RUN.

1. Check that "Label Initial Value Reflection Setting" has been set to "Disable". (☞ Page 367 Setting method)
2. Write data to the programmable controller.
3. Set the CPU module to the RUN state. (Label initial values are not set.*1)

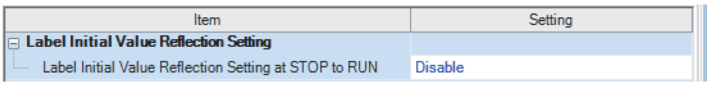
*1 Initial values are set when the CPU module is powered on or the operating status of the CPU module is switched from STOP to RUN for the first time after all data is rebuilt (reassigned) and written to the programmable controller.

Setting method

The following describes how to configure the label initial value reflection setting.

[CPU Parameter] ⇨ [File Setting] ⇨ [Label Initial Value Reflection Setting]

Window



Displayed items

Item	Description	Setting range	Default
Label Initial Value Reflection Setting at STOP to RUN	Sets whether or not to set label initial values when the operating status of the CPU module is switched from STOP to RUN.	• Disable • Enable	Disable

Point

When this setting is set to "Enable" (Label initial values are set when the operating status of the CPU module is switched from STOP to RUN), the Process CPU will perform the same operation as the one performed by the programmable controller CPU. In the default setting, this setting has been set to "Disable". Thus, set this setting to "Enable" to perform the same operation as the one performed by the programmable controller CPU.

Precautions

The following describes precautions on the label initial value reflection setting.

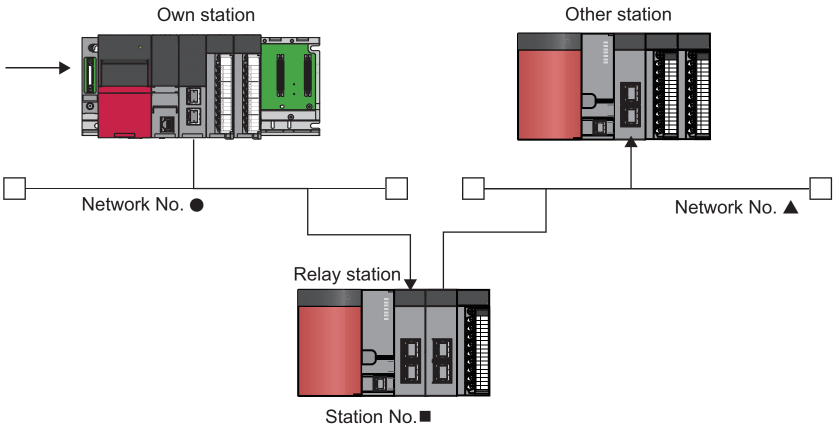
- If the battery runs out and latch type labels cannot be kept at a power failure, values in the labels are cleared when the CPU module is powered off. Thus, initial values are set when the CPU module is powered on even though "Label Initial Value Reflection Setting" has been set to "Disable".

20 ROUTING SETTING

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

The user can configure any communication route to perform transient transmission to stations in a different network. This setting can be used when the system has a network module which does not support dynamic routing or when it is necessary to clearly specify a communication route.

Relay station		Destination station
Network No.	Station No.	Network No.
●	■	▲

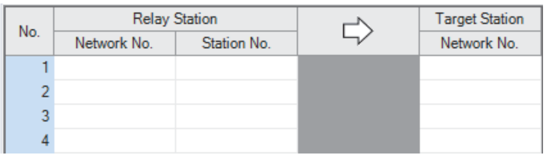


20.1 Setting Method

The user must specify the following: The network number and the station number of the own network (relay station) which will be pass through to another network, and the network number of the final arrival network (destination station). The maximum 238 routing settings can be specified.

[CPU Parameter] ⇨ [Routing Setting]

Window



Displayed items

Item		Description	Setting range	Default
Relay Station	Network No.	Sets the network number of the first relay station to pass through to the destination station network.	1 to 239	—
	Station No.	Sets the station number of the first relay station to pass through to the destination station network.	0 to 120	—
Target Station	Network No.	Sets the network number of the final arrival network.	1 to 239	—

Point

The S(P).RTWRITE instruction can be used to temporarily change or add a routing setting during operation (setting made by the S(P).RTWRITE instruction is cleared when the CPU module is powered off or reset). Also, the S(P).RTREAD instruction can be used to read setting details of the parameters. For details on these instructions, refer to the following.

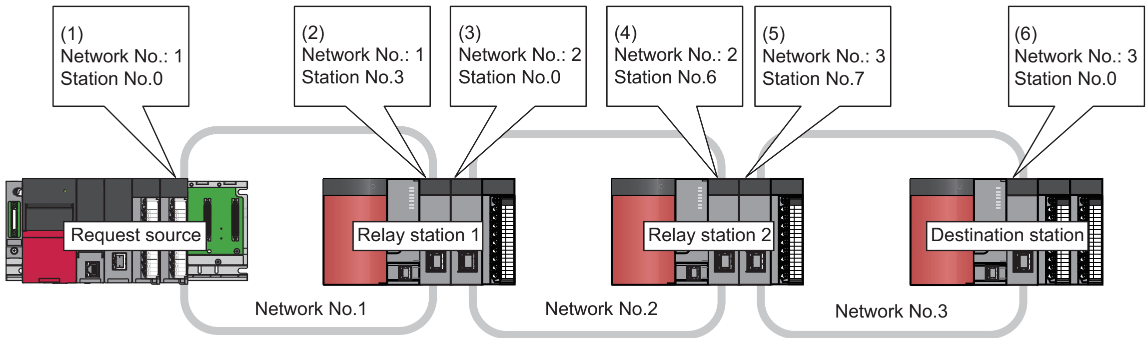
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

20.2 Setting Example

The following is an example of the routing setting.

Ex.

Transient transmission from the request source (Network No.1) to the target (Network No.3) via Network No.2.



<table>
<tr><th>Station</th><th>Request route</th><th>Response route</th></tr>
<tr><td>Request source</td><td>(1)
<table><tr><th colspan="2">Relay station</th><th>Destination station</th></tr><tr><th>Network No.</th><th>Station No.</th><th>Network No.</th></tr><tr><td>1</td><td>3</td><td>3</td></tr></table>The data is transferred to the relay station (2) of the own network to go to the network No.3.</td><td>(1)</td></tr>
<tr><td>Relay station 1</td><td>(2)
(3)
<table><tr><th colspan="2">Relay station</th><th>Destination station</th></tr><tr><th>Network No.</th><th>Station No.</th><th>Network No.</th></tr><tr><td>2</td><td>6</td><td>3</td></tr></table>The data is transferred to the relay station (4) of the own network to go to the network No.3.</td><td>(2)
(3)
The data has been transferred to the network No.1, and is to be transferred to (1).</td></tr>
<tr><td>Relay station 2</td><td>(4)
(5)
The data has been transferred to the network No.3, and is to be transferred to (6).</td><td>(4)
(5)
<table><tr><th colspan="2">Relay station</th><th>Destination station</th></tr><tr><th>Network No.</th><th>Station No.</th><th>Network No.</th></tr><tr><td>2</td><td>0</td><td>1</td></tr></table>The data is transferred to the relay station (3) of the own network to go to the network No.1.</td></tr>
<tr><td>Destination station</td><td>(6)</td><td><table><tr><th colspan="2">Relay station</th><th>Destination station</th></tr><tr><th>Network No.</th><th>Station No.</th><th>Network No.</th></tr><tr><td>3</td><td>7</td><td>1</td></tr></table>The data is transferred to the relay station (5) of the own network to go to the network No.1.</td></tr>
</table>

20.3 Precautions

The precautions on the routing setting are as follows:

- For the multiple CPU system configuration, the same routing setting must be used for all CPUs.

21 FIRMWARE UPDATE FUNCTION

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

This function enables users to update the firmware versions of modules by using firmware update files. (For the firmware update file, please consult your local Mitsubishi representative.)

The following two methods for updating firmware are available.

Update method	Description
Update using the engineering tool	Firmware versions of multiple modules can be changed at a time by using the engineering tool.
Update using an SD memory card	The firmware version of a module can be changed just using an SD memory card. Dedicated software is not required.

For the module models that support this function, precautions that are common to those modules, and update procedure, refer to the following.

MELSEC iQ-R Module Configuration Manual

Restriction

Before using the firmware update function, check the versions of the CPU module and engineering tool used.
(Page 1139 Added and Enhanced Functions)

Precautions

This section describes the precautions for the CPU module when using the firmware update function.

■Incompatible firmware and CPU modules

The following table shows incompatible combinations of firmware and CPU modules.

CPU module	3rd and 4th digits from left of production information	Incompatible firmware version
R04CPU	"18" or later	"33" or earlier
	"19" or later	"47" or earlier
R08CPU	"19" or later	"33" or earlier
	"20" or later	"47" or earlier
R16CPU	"19" or later	"33" or earlier
	"20" or later	"47" or earlier
R32CPU	"16" or later	"33" or earlier
	"17" or later	"47" or earlier
R120CPU	"16" or later	"33" or earlier
	"17" or later	"47" or earlier

CPU module	3rd and 4th digits from left of production information	Incompatible firmware version	
		CPU part	Network part
R04ENCPU	"32" or later	"47" or earlier	—
	"37"[*1], "38" or later		"51" or earlier
R08ENCPU	"30" or later		—
	"35"[*1], "36" or later		"51" or earlier
R16ENCPU	"27" or later		—
	"32"[*1], "33" or later		"51" or earlier
R32ENCPU	"30" or later		—
	"33"[*1], "34" or later		"51" or earlier
R120ENCPU	"22" or later		—
	"25"[*1], "26" or later		"51" or earlier

*1 A module whose first and second digits from the left of the production information are "43" or later

■When the RnENCPU is updated

To update the firmware version from "49" (CPU part)/"46" (network part) or earlier to "54" (CPU part)/"52" (network part) or later, follow the procedure below.

- Update using the engineering tool

1. First update the firmware version of the CPU part alone to "54" or later.
2. Then, update the firmware version of the network part to "52" or later. (Updating the CPU part and the network part simultaneously generates an error. Always update the CPU part before updating the network part.)

- Update using an SD memory card

1. First update the firmware version to "54" (CPU part)/"50" (network part).
2. Update the firmware version to the desired one.

MEMO

PART 3 DEVICES, LABELS, AND CONSTANTS

This part consists of the following chapters.

22 DEVICES

This chapter describes the devices.

22.1 Device List

This section lists the devices.

Classification	Type	Device name	Symbol	Number of points of Default	Parameter-set range	Notation
User device	Bit	Input	X	12K points[*5]	Unchangeable	Hexadecimal
	Bit	Output	Y	12K points[*5]		Hexadecimal
	Bit	Internal relay	M	12K points[*5]	Changeable (☞ Page 376 Device Setting)	Decimal
	Bit	Link relay	B	8K points		Hexadecimal
	Bit	Annunciator	F	2K points		Decimal
	Bit	Link special relay	SB	2K points		Hexadecimal
	Bit	Edge relay	V	2K points		Decimal
	Bit	Step relay[*3]	S	0 points		Decimal
	Bit/word	Timer	T	1K points[*5]		Decimal
	Bit/word	Retentive timer	ST	0 points		Decimal
	Bit/double word	Long timer	LT	1K points[*5]		Decimal
	Bit/double word	Long retentive timer	LST	0 points		Decimal
	Bit/word	Counter	C	512 points[*5]		Decimal
	Bit/double word	Long counter	LC	512 points[*5]		Decimal
	Word	Data register	D	18K points[*5]		Decimal
	Word	Link register	W	8K points		Hexadecimal
	Word	Link special register	SW	2K points		Hexadecimal
	Bit	Latch relay	L	8K points		Decimal
System Device	Bit	Function input	FX	16 points	Unchangeable	Hexadecimal
	Bit	Function output	FY	16 points		Hexadecimal
	Word	Function register	FD	5 points × 4 words		Decimal
	Bit	Special relay	SM	4K points		Decimal
	Word	Special register	SD	4K points		Decimal
Link Direct Device	Bit	Link input	Jn\X	160K points (Max.)[*1*6]	Unchangeable	Hexadecimal
	Bit	Link output	Jn\Y	160K points (Max.)[*1*6]		Hexadecimal
	Bit	Link relay	Jn\B	640K points (Max.)[*1*6]		Hexadecimal
	Bit	Link special relay	Jn\SB	5120 points (Max.)[*1*6]		Hexadecimal
	Word	Link register	Jn\W	2560K points (Max.)[*1*6]		Hexadecimal
	Word	Link special register	Jn\SW	5120 points (Max.)[*1*6]		Hexadecimal
Module access device	Word	Module access device	Un\G	268435456 points (Max.)[*1]	Unchangeable	Decimal
CPU buffer memory access device	Word	CPU buffer memory access device	U3En\G	268435456 points (Max.)[*1]	Unchangeable	Decimal
			U3En\HG	12288 points maximum	Changeable	Decimal
Index register	Word	Index register	Z	20 points	Changeable (☞ Page 402 Index register setting)	Decimal
	Double word	Long index register	LZ	2 points		Decimal
File register	Word	File register	R/ZR	0 points	Changeable	Decimal
Refresh data register	Word	Refresh data register	RD	512K points	Changeable	Decimal
Nesting	—	Nesting	N	15 points	Unchangeable	Decimal
Pointer	—	Pointer	P	8192 points[*2]	Changeable (☞ Page 411 Pointer setting)	Decimal
	—	Interrupt pointer	I	1024 points	Unchangeable	Decimal

Classification	Type	Device name	Symbol	Number of points of Default	Parameter-set range	Notation
Other devices	—	Network No. specification device	J	—	Unchangeable	Decimal
	—	I/O No. specification device	U	—		Hexadecimal
	—	SFC block device*3	BL	320 points*5		Decimal
	—	SFC transition device*3*4	TR	0 points		Decimal

*1 These are the maximum points that can be handled in the CPU module. The number of points actually used differs depending on the module used.

*2 The default number of points is 16384 for the R120CPU, R120ENCPU, R120PCPU, and R120SFCPU.

*3 The device can be collected by using the CPU module where the SFC function can be used. (☞ Page 1139 Added and Enhanced Functions)

*4 Can be used as a device comment in the SFC program.

*5 The default number of points for the R00CPU, R01CPU, and R02CPU are as follows.
· Input (X), Output (Y), Internal relay (M): 8K points
· Timer (T): 2K points
· Long timer (LT), Long counter (LC): 0 point
· Counter (C): 1K points
· Data register (D): 12K points
· SFC block device (BL): 128 points

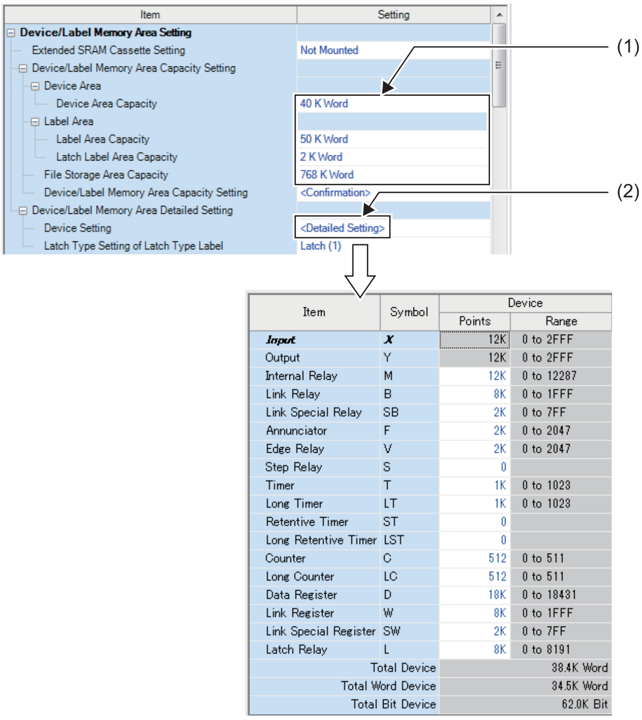
*6 The maximum number of points differs depending on the "Link Direct Device Setting" of the engineering tool. Before using the "Link Direct Device Setting", check the version of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

22.2 Device Setting

The number of points of each user device can be changed (☞ Page 378 User Device)

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Detailed Setting] ⇨ [Device Setting] ⇨ [Detail Setting]

Window



(1) The capacity of each area can be changed. (☞ Page 103 Device/label memory area setting)

(2) The number of points of user devices can be changed.

Point

Specify each item so that the total number of points for each user device does not exceed the capacity of the device area. (☞ Page 103 Device/label memory area setting)

Range of use of device points

The following table lists the range of use of device points to be set in the device setting.

Type	Device name	Symbol	Range of use*1	Setting unit
Bit	Input	X	X0 to X2FFF	—
Bit	Output	Y	Y0 to Y2FFF	—
Bit	Internal relay	M	M0 to M161882111	64 points
Bit	Link relay	B	B0 to B9A61FFF	64 points
Bit	Annunciator	F	F0 to F32767	64 points
Bit	Link special relay	SB	SB0 to SB9A61FFF	64 points
Bit	Edge relay	V	V0 to V32767	64 points
Bit	Step relay*2	S	S0 to S16383	1024 points
Word	Timer	T	T0 to T8993439	32 points
Word	Retentive timer	ST	ST0 to ST8993439	32 points
Word	Long timer	LT	LT0 to LT2529407	1 points
Word	Long retentive timer	LST	LST0 to LST2529407	1 points
Word	Counter	C	C0 to C8993439	32 points
Word	Long counter	LC	LC0 to LC4761215	32 points
Word	Data register	D	D0 to D10117631	4 points
Word	Link register	W	W0 to W9A61FF	4 points
Word	Link special register	SW	SW to SW9A61FF	4 points
Bit	Latch relay	L	L0 to L32767	64 points

*1 This is the maximum number of points for the R120CPU with an extended SRAM cassette (16MB) (NZ2MC-16MBS). The number of points varies depending on the model of the CPU module used, whether to use an extended SRAM cassette, and the type of its cassette.

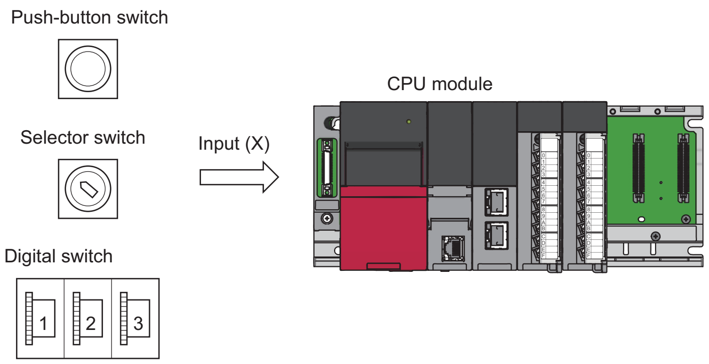
*2 The CPU module where the SFC function can be used supports this device. (☞ Page 1139 Added and Enhanced Functions)

22.3 User Device

This section describes the user device.

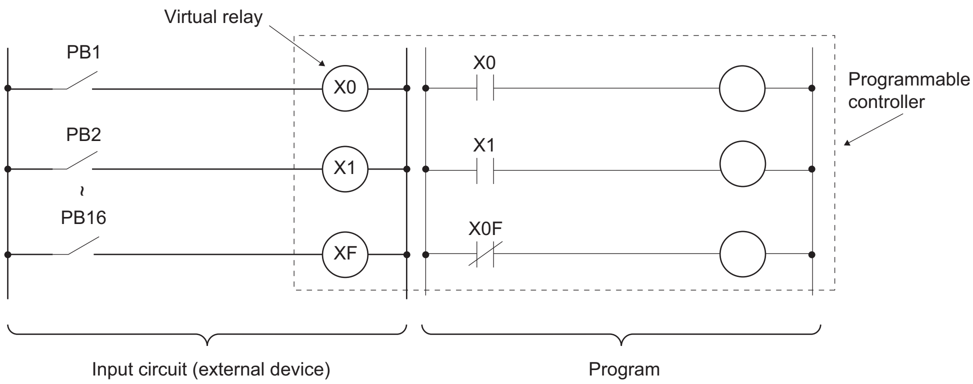
Input (X)

This device provides the CPU module with commands and/or data using an external device, such as pushbutton, transfer switch, limit switch, and digital switch.



Concept of input

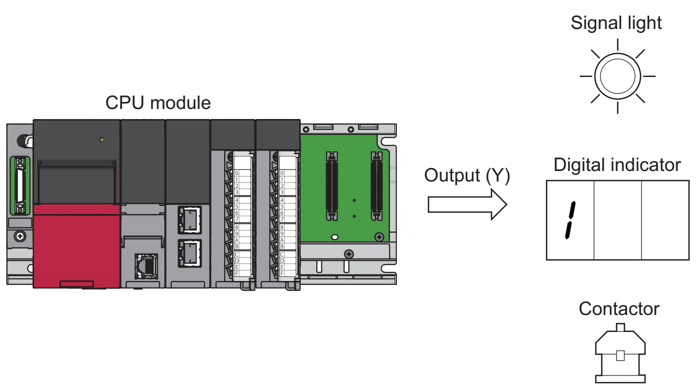
Assume that one virtual relay Xn is incorporated into the CPU module for each input point. In the program, a normally open contact and normally closed contact for the Xn are used.



Also, the input can be used as a target re-flesh (CPU module side) device of the remote input (RX), such as the CC-Link IE Field Network.

Output (Y)

This device outputs the control results of the program to various devices, such as external signal light/digital HMI/electromagnetic switch (contactor)/solenoid.



Internal relay (M)

This device is used as an auxiliary relay within the CPU module. The following operations turn off all the internal relays.

- Powering off and on the CPU module
- Reset
- Latch clear

Latch relay (L)

This device is an auxiliary relay which enables latching (data retention during power failure). This device is used within the CPU module. This device latches operation results (ON/OFF information) even after the following operations.

- Powering off and on the CPU module
- Reset

Link relay (B)

This device is used as a CPU side device when refreshing bit data between the network module, such as the CC-Link IE Controller Network module and the CPU module.

Refreshing network modules using link relay

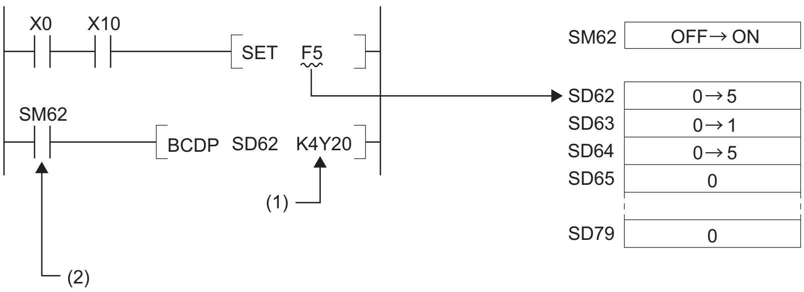
Data are transferred/received between the link relay (B) within the CPU module and the link relay (LB) of the network module, such as the CC-Link IE Controller Network module. The refresh range is specified using parameters on the network module. The part which is not used for refreshing can be used for other applications.

Annunciator (F)

This device is an internal relay used for a customer-created program which detects malfunction/failure of the equipment. When the annunciators are turned on, SM62 (Annunciator) is turned on, the number of activated annunciators and their device numbers are stored in SD62 (Annunciator number) to SD79 (Table of detected annunciator numbers).

Ex.

Failure detection program



(1) Outputs the annunciator number of the annunciator that turned on.
(2) ON detection of the annunciator

Also the number of the annunciator that turned on first (the number stored in SD62) is registered in the event history.

Point

Only one annunciator number is registered in the event history while power is turned on.

On/off method for annunciator

Annunciators are turned on by either the SET F□ instruction or the OUT F□ instruction. Annunciators are turned off by the RST F□ instruction or the LEDR instruction or the BKRST instruction.

Point

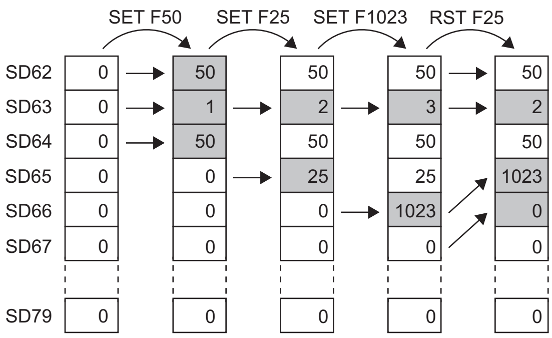
When the annunciators are turned on/off using any methods (e.g. the MOV instruction) other than shown above, the operation is the same as that of internal relays. As a result, SM62 is not turned on and annunciator numbers are not stored into SD62 and SD64 (Table of detected annunciator numbers) to SD79.

■Operations when annunciators are turned on

1. The annunciator numbers that turned on are stored sequentially into SD64 to SD79.
2. The annunciator number stored into SD64 is stored into SD62.
3. SD63 value (Number of annunciators) is incremented by one.

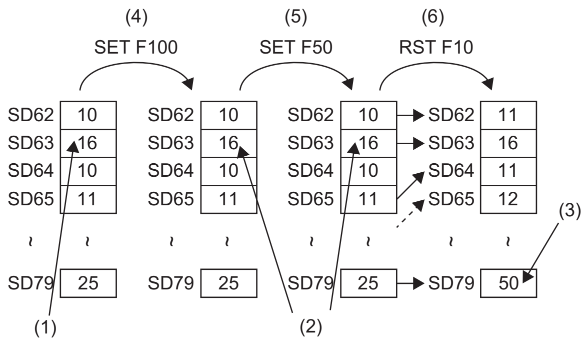
■Operations when annunciators are turned off

1. The number of the annunciator deactivated is removed, and the numbers of annunciators, which were lined up behind the removed one, move forward one by one.
2. When the annunciator number stored into SD64 is turned off, the annunciator number newly stored into SD64 is stored into SD62.
3. The value of SD63 is decremented by one. When the SD63 value gets decremented to 0, SM62 is turned off.



Point

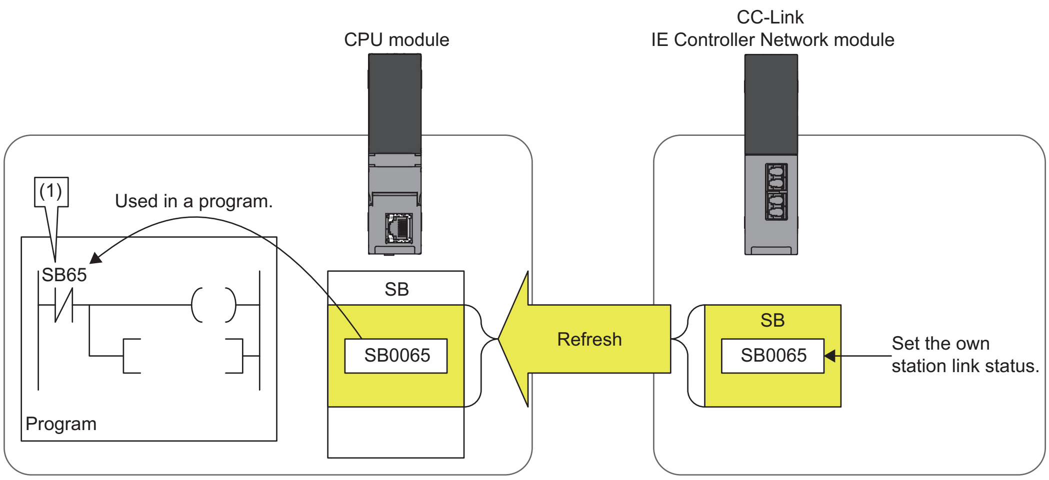
If more than 16 annunciators are turned on, the 17th annunciator onwards are not stored into SD64 to SD79. However, if the numbers of annunciators registered in SD64 to SD79 are turned off, the lowest numbers, which are not registered in SD62 to SD79, of the numbers of annunciators which were turned on for the 17th on and after, are stored into SD64 to SD79.



(1) Annunciators have been stored for the maximum number (16).
(2) Because annunciators have been stored for the maximum number, the value does not change.
(3) The smallest number is stored.
(4) Turn on the 17th device.
(5) Turn on the 18th device.
(6) Turn off the first device.

Link special relay (SB)

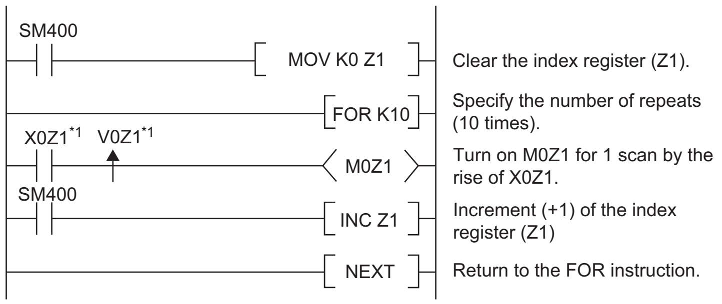
The communication status and error detection status of network modules, such as the CC-Link IE Controller Network module, are output to the link special relay (J□\SB□) on the network. The link special relay (SB) is a device for using as a refreshing target of link special relays in the network. The part which is not used for refreshing can be used for other applications.



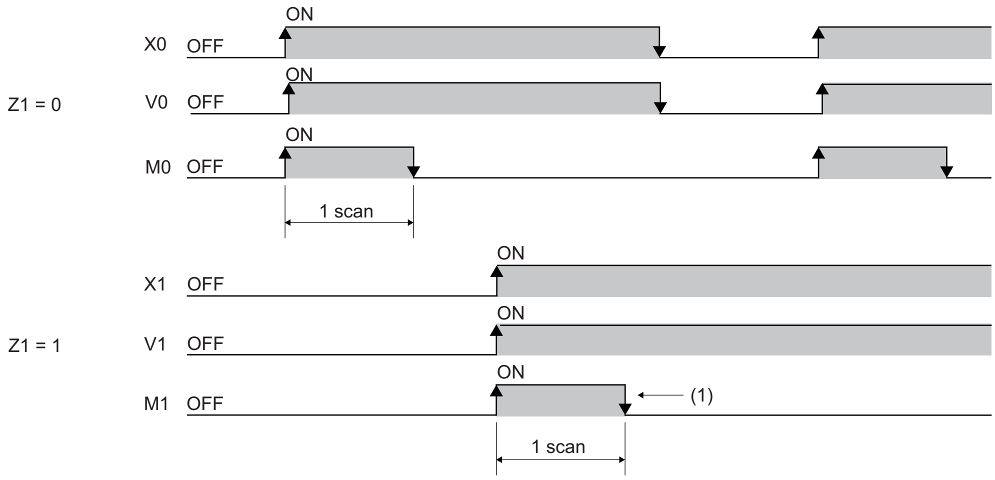
(1) The network status is checked.

Edge relay (V)

The edge relay is a device that memorizes operation results (on/off information) from the head of the ladder block, allowing its use only by the EGP/EGF instruction. This device is executed for various objectives such as the rising (from off to on) detection in the structured programs by the index modification.



*1 Edge relay V0Z1 memorizes on/off information of X0Z1.



(1) When X1 is rising, this device is turned on for one scan duration.

Step relay (S)

This device is used when specifying SFC program steps. This device is also used when specifying step No. through such methods as verifying (monitor, current value changes) SFC programs with SFC control instructions or the engineering tool. (📖 MELSEC iQ-R Programming Manual (Program Design))

Point

This device is designed only for SFC programs, and cannot be used as a substitute for an internal relay in a sequence program. If it is used, an error may occur, causing a system failure.

Timer

This device starts measurement when the timer coil is turned on. When the current value reaches a setting value, time is up and the contact is turned on. This timer is an up-timing type device and therefore the current value matches a setting value when the timer time is up.

Types of timers

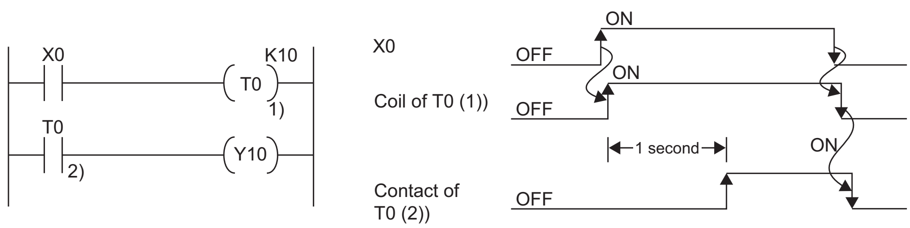
There are two types of timers: timer (T) which retains the current value in 16-bit units and long timer (LT) which retains it in 32-bit units. The timer (T) and the long timer (LT) are different devices and the number of device points can be set for each of them. In addition, there are the retentive timer (ST) and the long retentive timer (LST), both of which retain the current value even if the coil is turned off.[*1]

Types of timers			
Timer	Current value = 16 bits	Timer (T)	Low-speed timer
			High-speed timer
		Retentive timer (ST)	Low-speed retentive timer
			High-speed retentive timer
	Current value = 32 bits	Long timer (LT)	
		Long retentive timer (LST)	

*1 For the timer (T)/long timer (LT), the current value returns to 0 when the coil is turned off.

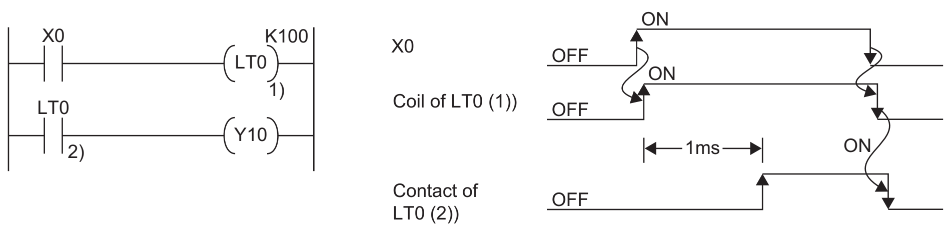
■Timer (T)

This device starts measurement when the coil of the timer is turned on. When the timer current value matches a setting value, time is up and the timer contact is turned on. When the timer coil is turned off, the current value returns to 0 and the timer contact is turned off.



■Long timer (LT)

This device can count from 0 to 4294967295 to measure the time. If the measuring unit is set to 0.01ms, the measurable time range of this device is from 0 to about 11.9 hours. The current value of the long timers is updated by adding a difference of the counter which is used in the system when the OUT LT□ instruction is executed. Proper time can be measured even when the coil instruction of the long timer cannot be executed once per scanning because the counters used in the system count asynchronously with scanning. When the long timer coil is turned on, measurement starts and when the long timer current value matches a setting value, time is up and the long timer contact is turned on. When the long timer coil is turned off, the current value returns to 0 and the long timer contact is turned off.



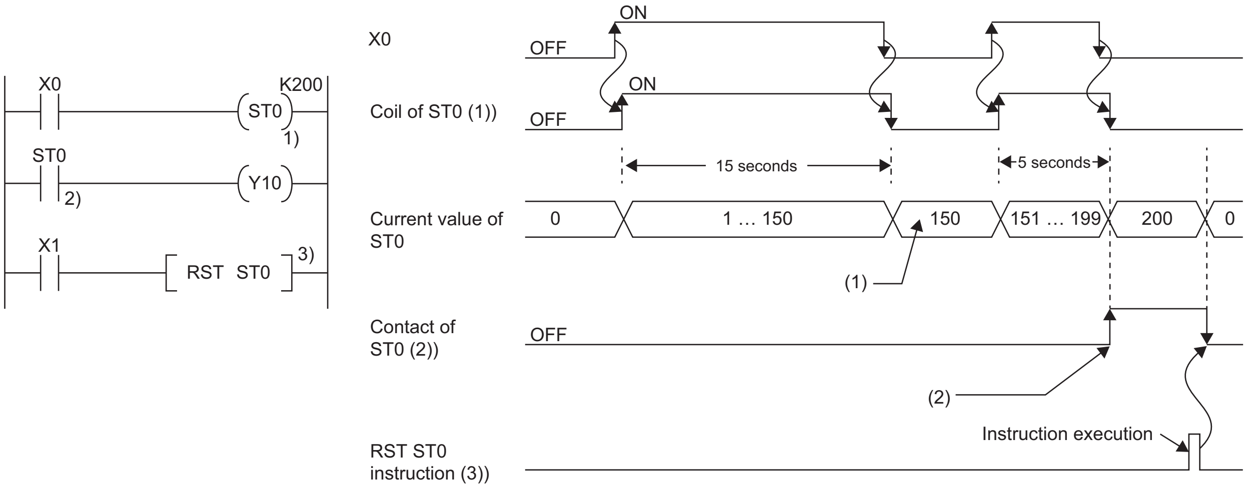
*1 This figure shows the example when the long timer time limit value setting is 0.01ms.

Point

- The long timer contact is turned on in the next or subsequent coil execution after the long timer coil is turned on. The long timer contact is not turned on simultaneously when the long timer coil is turned on.
- The long timer (LT) can be used in interrupt programs. (☞ Page 74 Interrupt Program)

■Retentive timer (ST)

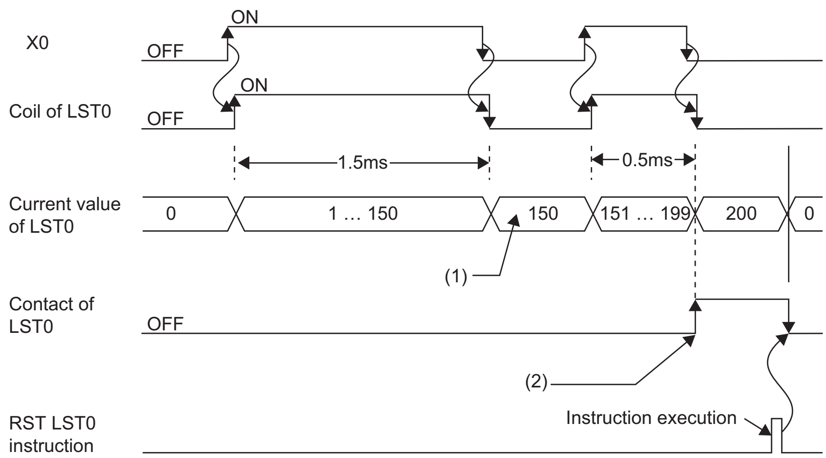
This device counts the sum of time duration in which the coil is turned on. When the retentive timer coil is turned on, measurement starts and when the timer current value matches a setting value (when time is up), the retentive timer contact is turned on. The current value and the contact on/off state is retained even when the retentive timer coil is turned off. When the coil is turned on again, measurement starts with the retained current value. To clear the retentive timer current value and turn off the contact, issue the RST ST□ instruction.



(1) Even though the coil (1)) turns off, the current value is held.
(2) Even though the coil (1)) turns off, the contact remains on.

■Long retentive timer (LST)

This device counts the sum of time duration in which the coil is turned on. When the long retentive timer coil is turned on, measurement starts and when the timer current value matches a setting value (when time is up), the contact is turned on. The current value and the contact on/off state is retained even when the long retentive timer coil is turned off. When the coil is turned on again, measurement starts with the retained current value. To clear the long retentive timer current value and turn off the contact, issue the RST LST□ instruction.



(1) Even though the coil turns off, the current value is held.
(2) Even though the coil turns off, the contact remains on.

Point

The long retentive timer (LST) can be used in interrupt programs. (☞ Page 74 Interrupt Program)

■Low-speed/high-speed timer (T/ST)

The low-speed timer and high-speed timer are the same device which is set to a low speed or high speed timer by writing the instruction accordingly to specify it on the timer. For example, specifying OUT T0 generates a low-speed timer and specifying OUTH T0 produces a high-speed timer even when using the same T0 device. This also applies to the retentive timer.

Timer time limit value

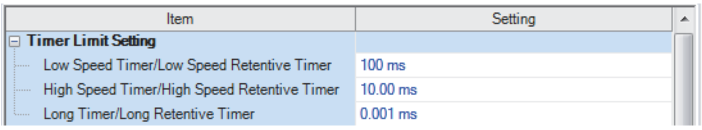
Although the low-speed timer and high-speed timer are the same device, timer limit value are different depending on how to specify the timer device (how to write the instruction). For example, specifying OUT T0 generates a low-speed timer and specifying OUT H T0 produces a high-speed timer even when using the same T0 device. This also applies to the retentive timer. The long timer cannot be set to a low-speed or high speed device. The time limit value for each timer is set in "Timer Limit Setting".

■Timer limit setting

The following window is to set the timer time limit values.

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Timer Limit Setting]

Window



Displayed items

Item	Description	Setting range	Default
Low Speed Timer/Low Speed Retentive Timer	Sets the timer time limit value of T and ST used for the low-speed timer and low-speed retentive timer.	1 to 1000ms (unit: 1ms)	100ms
High Speed Timer/High Speed Retentive Timer	Sets the timer time limit value of T and ST used for the high-speed timer and high-speed retentive timer.	0.01 to 100ms (unit: 0.01ms)	10.00ms
Long timer/Long retentive timer	Sets the timer time limit value of LT and LST used for the long timer and long retentive timer.	0.001 to 1000ms (unit: 0.001ms)	0.001ms

Timer current value and the measurable range

This sections describes the timer current value and the measurable range.

■Timer (T/ST)

The current value range is 0 to 32767. The measurable time range is from 0 to (timer time limit value × 32767).

■Long timer (LT/LST)

The setting range of the current value is 0 to 4294967295, which is the same as the range of unsigned 32-bit integers. The measurable time range is from long timer time limit value to (timer time limit value × 4294967295).

Handling timers

When executing the timer coil (the OUT T□ instruction), the timer coil is turned on/off, the current value is updated, and the contact is turned on/off.

Accuracy of timers

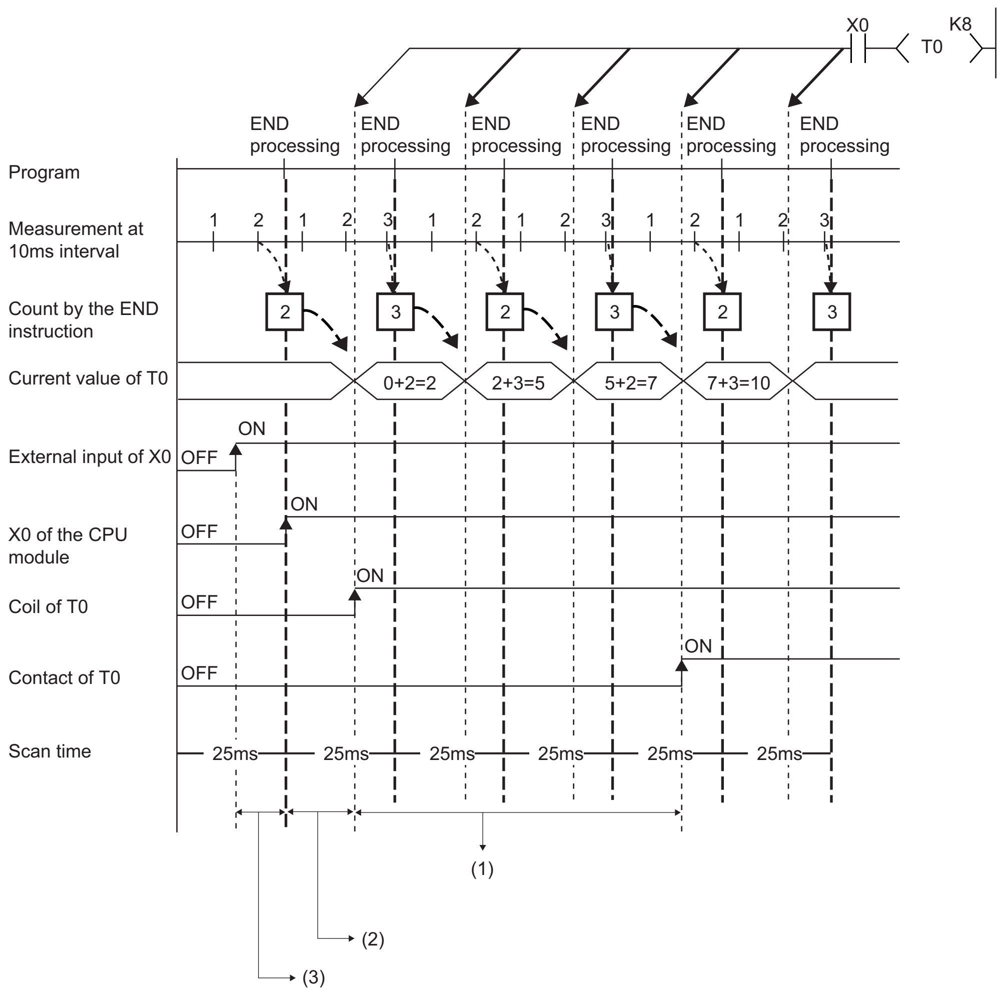
This sections describes the accuracy of timers.

■Timer (T/ST)

The scan time value measured by the END instruction is added to the current value when the OUT T□ instruction is executed. If the timer coil is turned off when the OUT T□ instruction is executed, the current value is not updated. The maximum response accuracy of the timer (the time duration from capture of an input (X) to output of it) is "2 scan time + timer time limit setting".

Ex.

Timer limit setting = 10ms, setting value of T0 = 8 (10ms × 8 = 80ms), scan time = 25ms



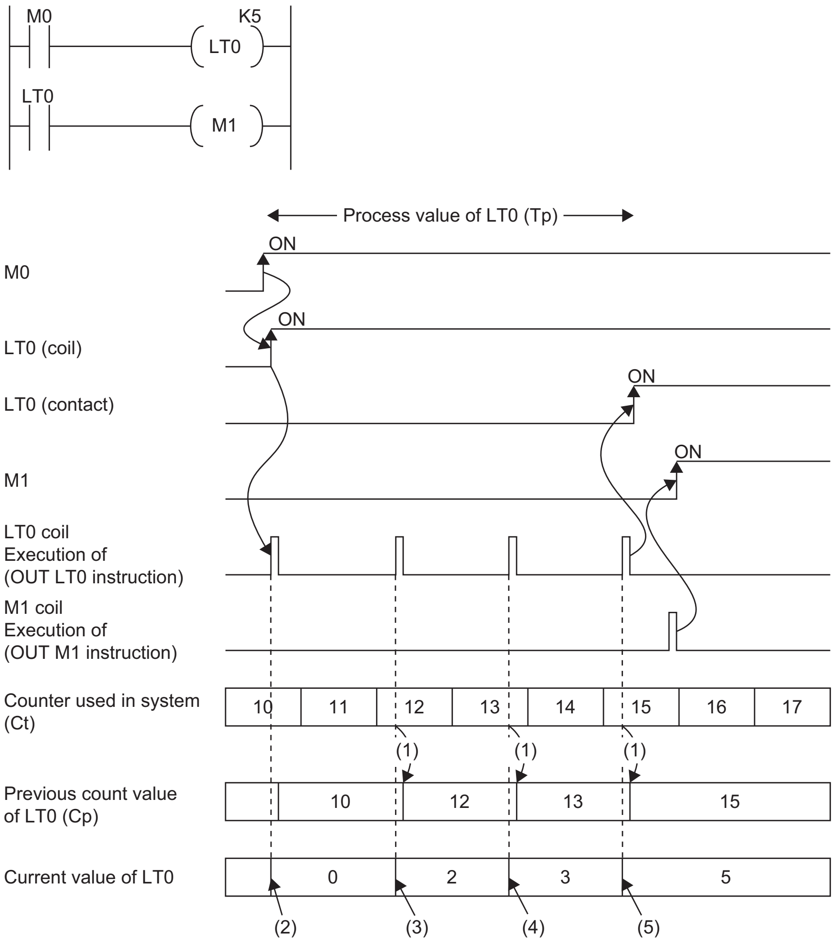
(1) Accuracy from when the coil of the timer turns on until when the contact of the timer turns on
- (1 scan time + timer limit setting) to (1 scan time)

(2) Timing when the coil of the timer turns on

(3) Input fetching timing

■Long timer (LT/LST)

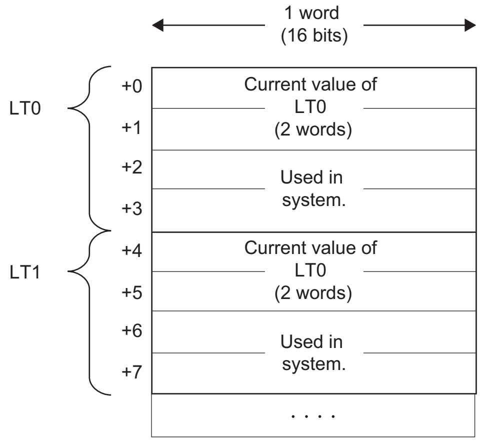
In the following program, the accuracy of Tp (the time duration from the long timer coil activation to long timer contact activation) is (Ts-Tu) ≤ Tp < (Ts + Tu).



(1) Cp is updated.
(2) LT0 is initialized to 0.
(3) The result of (Ct - Cp) is added.: 0 + (12 - 10) = 2
(4) The result of (Ct - Cp) is added.: 2 + (13 - 12) = 3
(5) The result of (Ct - Cp) is added.: 3 + (15 - 13) = 5
Tp: time duration from the long timer coil activation to long timer contact activation
Ts: setting value of the long timer
Tu: time limit value of the long timer

Data configuration of long timer (LT/LST)

The long timer (LT) and the long retentive timer (LST) use four words (64 bits) for each point. If the most significant two words are changed in a program, it is impossible to measure the time properly, because they are used by the system.



Point

The current value of the long timer (LT) and the long retentive timer (LST) is 32-bit data. It can be specified by the instructions which can specify signed or unsigned 32 bit data. (It cannot be specified with the BK + Instruction.)

Precautions

This section describes the precautions when using the timer and long timer.

■Precautions about timer usage

- Do not describe more than one coil (the OUT T□ instruction) on the same timer during a single scanning. Doing so results in improper measurement because the timer current value is updated when the coil for each timer is executed.
- When timer is not used for data collection for each scan: While the coil of a timer (e.g. T1) is turned on, the timer coil (the OUT T□ instruction) cannot be skipped by the instructions such as the CJ. When the timer coil is skipped, proper measurement is impossible because the timer current value is not updated. In addition, when the timer exists in a subroutine program, be sure to execute a subroutine call including T1 coil only once for each scanning operation while the coil of the timer (e.g. T1) is turned on. Otherwise proper measurement is impossible.
- The timer cannot be used in the initial execution type program, the fixed scan execution type program, or the event execution type program where the occurrence of an interrupt is set to be a trigger. The timer can be used in standby type programs if the coil of timer (OUT T□ instruction) is executed one time for one scan using a subroutine program.
- The timer cannot be used in interrupt programs. The timer can be used in subroutine programs or FB programs if the coil of timer (OUT T□ instruction) is executed one time for one scan.
- When setting value is 0: The contact is turned on when the OUT T□ instruction is executed.
- Even when the setting value is increased after the timer time is up, the timer status does not change (time continues to be up) and the timer does not operate.
- Do not set the timer setting value to 32768 or above. If used when set to 32768 or above, the timer contact may not turn on.

■Precautions about long timer usage

This section describes the precautions when using long timers (LT/LST).

- The long timer cannot be used in initial execution type programs.
- Even when the setting value is increased after the long timer time is up, the long timer status does not change (time continues to be up) and the long timer does not operate.

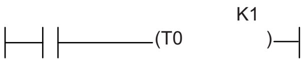
■Timer setting value and timer limit setting

When the condition is "Timer setting value < Scan time + Timer limit setting", the coil and the contact may be turned on at the same time depending on the timing when the coil turns on. When the condition is not satisfied, reduce the timer limit setting value to satisfy the condition.

Ex.

When changing the low-speed timer to high-speed timer and reducing the timer limit setting value (scan time: 20ms)

Before change (low-speed timer)

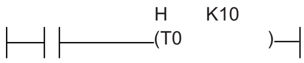


■Timer limit setting

Low Speed Timer/Low Speed Retentive Timer: 100ms

Timer setting value (100ms × 1 = 100ms) < Scan time (20ms) + Timer limit setting (100ms)

After change (high-speed timer)



■Timer limit setting

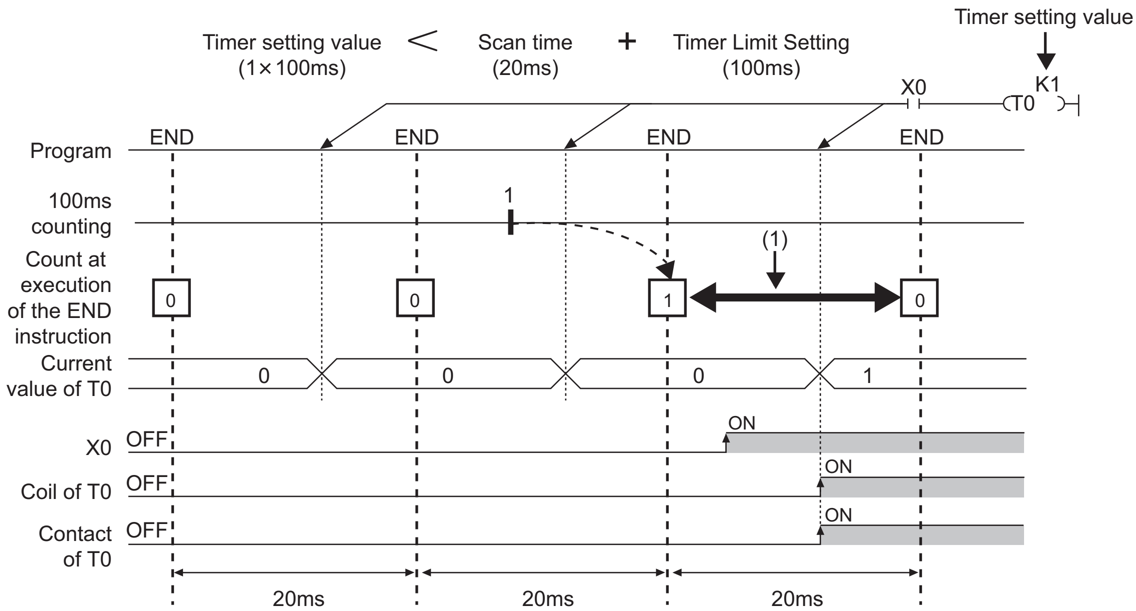
High Speed Timer/High Speed Retentive Timer: 10.00ms

Timer setting value (10.00ms × 10 = 100ms) < Scan time (20ms) + Timer limit setting (10ms)

The following shows an example of when the coil and the contact are turned on at the same time when the condition is "Timer setting value < Scan time + Timer limit setting".

Ex.

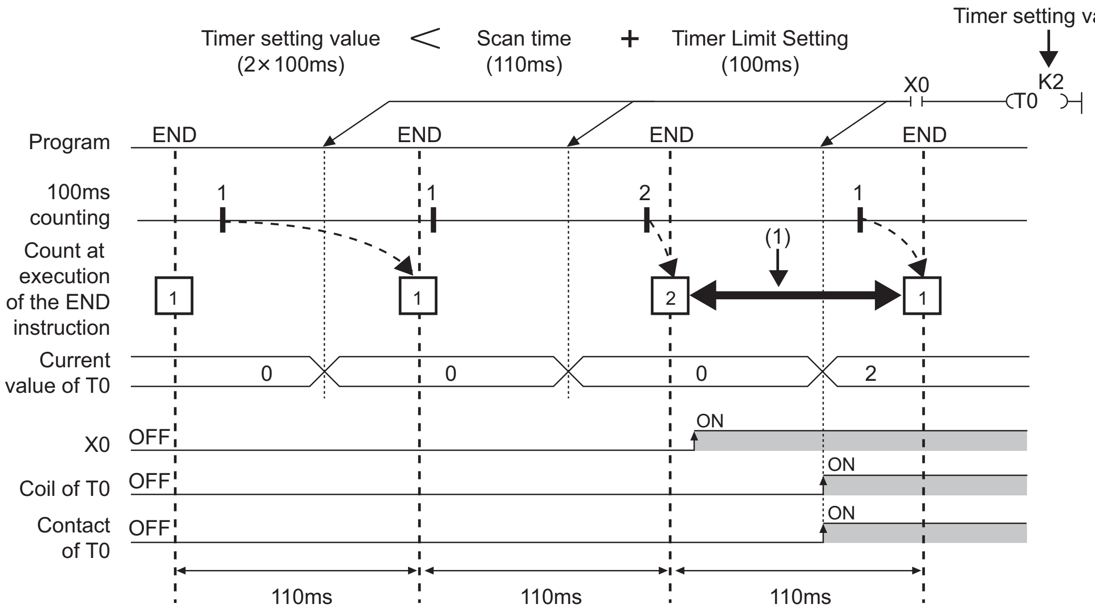
In the case where the values are set as follows (timer setting value: 1 (1 × 100ms), scan time: 20ms, and timer limit setting: 100ms), when the coil of the timer (T0) turns on in the next scan after the coefficient of the END instruction becomes equal to or higher than the timer setting value, the coil and the contact turn on at the same time because the timer current value becomes equal to the timer setting value at startup of the timer.



(1) The coefficient of the END instruction becomes equal to or higher than the timer setting value. Therefore, when the coil turns on during this period, the contact also turns on at the same time.

Ex.

In the case where the values are set as follows (timer setting value: 2 (2 × 100ms), scan time: 110ms, and timer limit setting: 100ms), when the coil of the timer (T0) turns on in the next scan after the coefficient of the END instruction becomes equal to or higher than the timer setting value, the coil and the contact turn on at the same time because the timer current value becomes equal to the timer setting value at startup of the timer.



(1) The coefficient of the END instruction becomes equal to or higher than the timer setting value. Therefore, when the coil turns on during this period, the contact also turns on at the same time.

Counter

This device counts the number of rising operation of the input condition in the program. The counter is an up-timing type device and therefore when the count value matches a setting value, the count reaches its upper limit and the contact is turned on.

Types of counters

There are two types of counters: counter (C) which retains the counter values in 16-bit units and long counter (LC) which retains them in 32-bit units. The counter (C) and the long counter (LC) are different devices and the number of device points can be set for each of them.

■Counter (C)

This device uses one word for each point. The measurable range is 0 to 65535.

■Long counter (LC)

This device uses two words for each point. The measurable range is 0 to 429467295.

Point

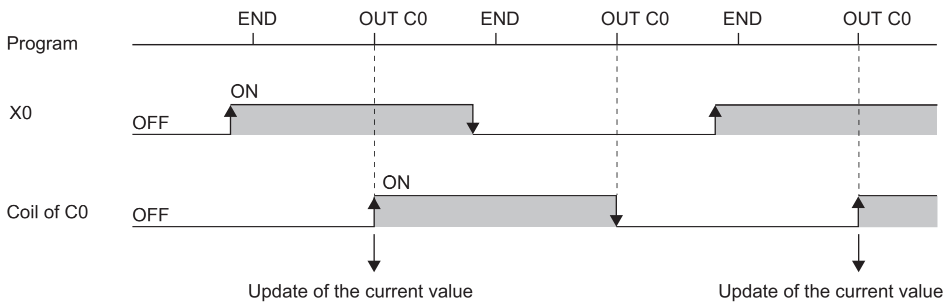
The long counter (LC) can be used in interrupt programs. (☞ Page 74 Interrupt Program)

Counting process

When executing the counter coil (OUT C□ instruction/OUT LC□ instruction), the counter coil is turned on/off, the current value is updated (count value +1), and the contact is turned on/off. The current value is updated (count value +1), when the counter coil input is rising (from off to on). The current value is not updated when the coil input is off, remains on, and is turned off.

[Ladder example]

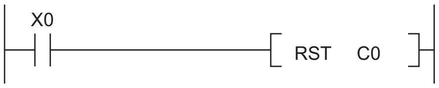
[Update timing of the current value]



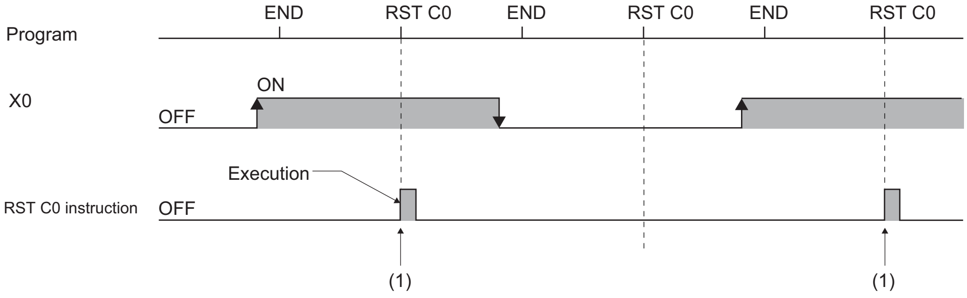
Resetting counters

The counter current value is not cleared even when the counter coil input is turned off. To clear the counter current value (resetting) and turn off the contact, issue the RST C□/RST LC□ instruction. When executing the RST C□ instruction, the counter value is cleared and the contact is turned off.

[Ladder example]



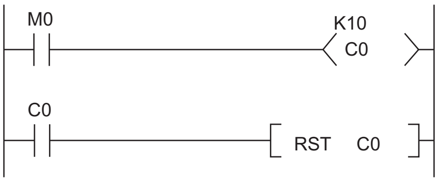
[Reset timing of the counter]



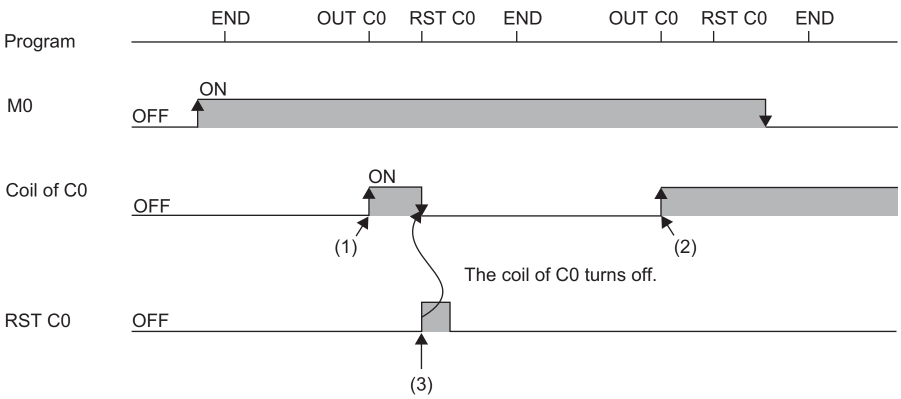
(1) The count value is cleared, and the contact turns off.

■Precautions about counter reset

When executing the RST C□ instruction, the coil for C□ is also turned off. If the execution condition for the OUT C□ instruction is turned on after the RST C□ instruction is executed, the coil of C□ is turned on and the current value is updated (count value +1) when the OUT C□ instruction is executed.

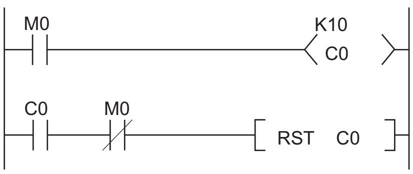


In the above ladder example, the coil of C0 is turned on by turning on M0 and as a result the current value is updated. When C0 count reaches its upper limit, C0 contact is turned on and C0 current value is cleared by execution of the RST C0 instruction. At this time C0 coil is also turned off. When M0 is turned on at the next scanning, the current value is updated because C0 coil is turned on during the OUT C0 instruction execution (the current value is changed to 1).



(1) The current value update contact turns on.
(2) The current value is updated because the coil of C0 turns on.
(3) The count value is cleared, and the contact turns off.

To address the above problem, insert a normally closed contact of the execution condition for the OUT C0 instruction into the execution condition for the RST C0 instruction to prevent C0 coil from being turned off while the execution condition (M0) of the OUT C0 instruction is turned on as shown in the following ladder example.

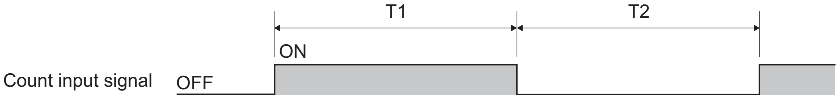


Maximum counting speed for counters

Counting is possible only when on/off time of the input condition is longer than the execution interval of the same OUT C☐ instruction. Maximum counting speed for counters can be obtained by the following equation:

$$\text{Maximum counting speed Cmax} = \frac{n}{100} \times \frac{1}{T} \text{ [times/s]}$$

*1 Duty (n) is a value which expresses the ratio of on/off time of the count input signal as a percent (%) value.



- When T1≥T2, $n = \frac{T2}{T1+T2} \times 100\%$
- When T1<T2, $n = \frac{T1}{T1+T2} \times 100\%$

Data register (D)

This device can store numerical values.

Link register (W)

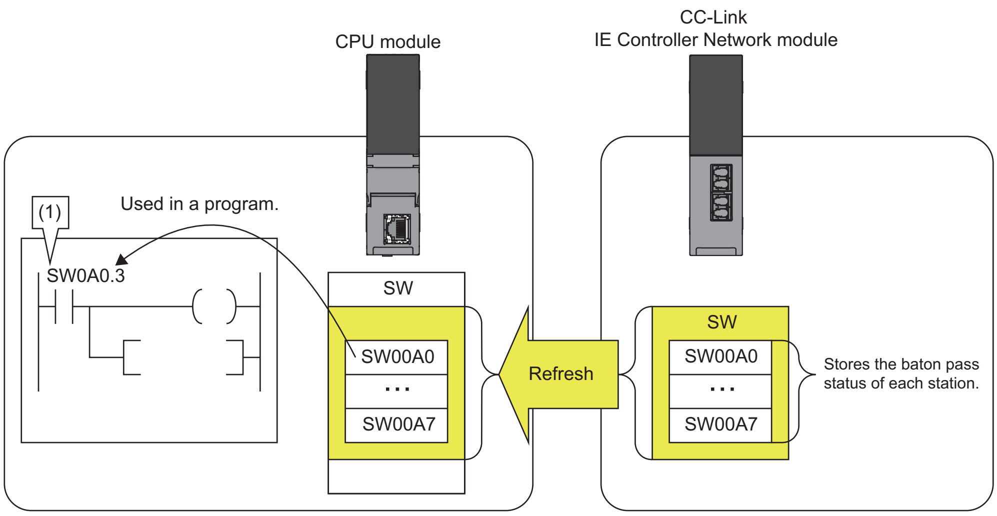
This device is used as a CPU module side device when refreshing word data between the network module, such as the CC-Link IE Controller Network module and the CPU module.

Refreshing network modules using link register

Data are transferred/received between the link register (W) within the CPU module and the link register (LW) of the network module, such as the CC-Link IE Controller Network module. The refresh range is specified using parameters on the network module. The part which is not used for refreshing can be used for other applications.

Link special register (SW)

Word data information on the communication status and error detection status of a network, such as CC-Link IE Controller Network, are output into the link special register (J☐\SW☐) on the network. The link special register (SW) is a device for using as a refreshing target of link special registers in the network. The part which is not used for refreshing can be used for other applications.



(1) The network status is checked.

22.4 System Device

The system device is used by the system. Assignment/capacity is fixed and cannot optionally be altered.

Function device (FX/FY/FD)

This device is used for the subroutine programs with argument passing. Data is written/read between the subroutine call sources with argument passing and the subroutine programs with argument passing. When using the function device in a subroutine program, the device used in each subroutine program call source can be determined. As a result, when using the same subroutine program, it can be used without being aware of the call source of other subroutine programs.

Function input (FX)

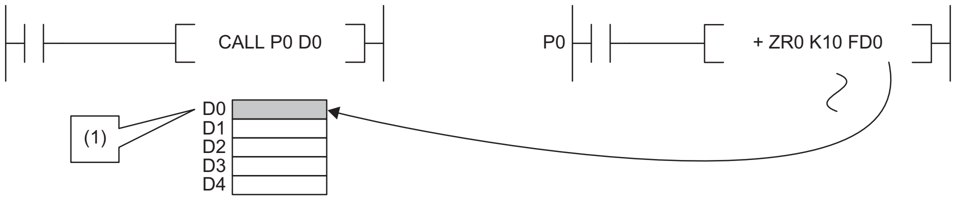
This device is used when passing on/off data to a subroutine program. In the subroutine program, bit data specified in a subroutine call instruction with argument passing are captured and used for operation. All the bit data specification devices of the CPU module are available.

Function output (FY)

This device is used when passing operation results (on/off data) in a subroutine program to a subroutine program call source. Operation results are stored into the device specified in the subroutine program with argument passing. The bit data specification device other than the CPU module input (X) are available.

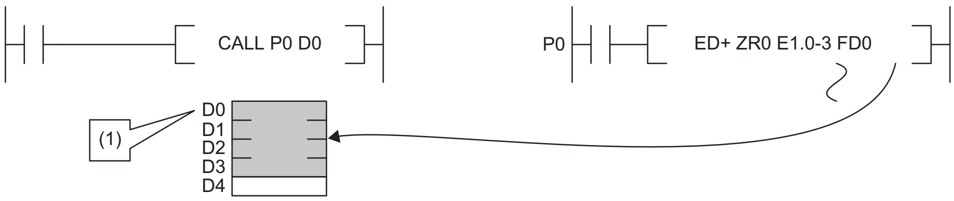
Function register (FD)

This device is used for writing/reading between the subroutine call source and the subroutine program. The input/output condition of the function register is automatically identified by the CPU module. "Source data" in a subroutine program means data inputted into the subroutine program. "Destination data" in a subroutine program means data outputted from the subroutine program. One point of a function register occupies maximum of four words and can store 16-bit data, 32-bit data, 64-bit data, single-precision real number, and double-precision real number. However the number of words to be used depends on the instruction in the subroutine program.
For example, for the destination of addition instruction (+instruction) of 16-bit signed integer, one word is used.



(1) The data is stored in one point of D0.

Besides, for the destination of addition instruction (ED+instruction) of double-precision real number, four words are used.



(1) The data is stored in four points of D0 to D3.

22

Special relay (SM)

This is the internal relay for which the specification is defined in the CPU module, where the status of the CPU module is stored. (☞ Page 930 List of Special Relay Areas)

Special register (SD)

This is the internal register for which the specification is defined in the CPU module, where the status (diagnostics information, system information, etc) of the CPU module is stored. (☞ Page 966 List of Special Register Areas)

22.5 Link Direct Device

This device directly accesses link relays and/or link registers of the network module in the CC-Link IE Controller Network and/or CC-Link IE Field Network.

Specification method

Specify a link direct device as shown below: The link register 10 (W10) of the network number 2 can be specified as "J2\W10".

Specification method: J□\□

Device No.

- Input • • • • • • • • • From X0
- Output • • • • • • • • From Y0
- Link relay • • • • • • • B0
- Link register • • • • • • From W0
- Link special relay • • • • From SB0
- Link special register • • • From SW0

Network No.1 to 239

Point

For bit devices, digit specification is allowed. (Example: J1\K1X0, J10\K4B0)

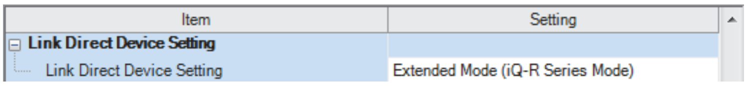
Specification range

All the link devices of the network module can be specified. The link devices which fall outside the range specified with "Refresh Setting" can also be specified.
For the following modules, specify the "Extended Mode (iQ-R Series Mode)" in the "Link Direct Device Setting" of the CPU parameter. (Default setting is "Q Series Compatible Mode".)*1

- CC-Link IE TSN master/local module
- Motion module
- CC-Link IE Controller Network-equipped module to which the link points extension is set.*2

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Link Direct Device Setting]

Window



Point

When the network module to be administered contains any of the following modules when specifying the link direct device, set the "Extended Mode (iQ-R Series Mode)" to the "Link Direct Device Setting".

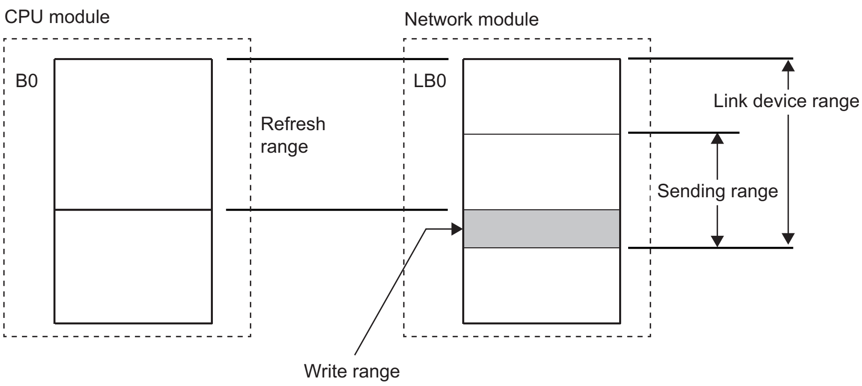
- CC-Link IE TSN master/local module
- Motion module
- CC-Link IE Controller Network-equipped module to which the link points extension is set.*2

Other modules operate in the both mode.

*1 Before using, check the versions of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)
*2 Applies when the "Link points extended setting" of the module parameter is set to "Extend" and the extended link device is specified. (📖 MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application))

Specification range for writing

Writing should be done in the range which is within the link device range specified as a sending range of network parameters, and outside the range specified as the refresh range for "Refresh Setting".*1
Note that writing in the range specified as the refresh range overwrites link device data of the network module during refresh process. When writing data to a writing range of another station using the link direct device, data is overwritten with received data during data reception from another station.



*1 There is only one network module to which the link direct device can write data for each network number. When more than one network module with the same network number is mounted, the network module with the lowest slot number is the target for writing by the link direct device.

Specification range for reading

Data can be read from the entire range of link devices of the network module.*1

*1 There is only one network module for which reading is allowed with the link direct device for each network number. When more than one network module with the same network number is mounted, the network module with the lowest slot number is the target for reading by the link direct device.

Difference from link refresh

The following table shows the difference between the link direct device and link refresh.

Item		Link direct device	Link refresh
Description method in program	Input	Jn\K4X0...	X0...
	Output	Jn\K4Y0...	Y0...
	Link relay	Jn\K4B0...	B0...
	Link register	Jn\W0...	W0...
	Link special relay	Jn\K4SB0...	SB0... Or module label
	Link special register	Jn\SW0...	SW0... Or module label
Access range in relation to network module		All the link devices for each network module	The range specified in "Refresh Setting"
Guarantee range of access data		In word (16-bit) units	In word (16-bit) units

22.6 Module Access Device

This device directly accesses from the CPU module to the buffer memory of the intelligent function module mounted on the main base unit and extension base unit.

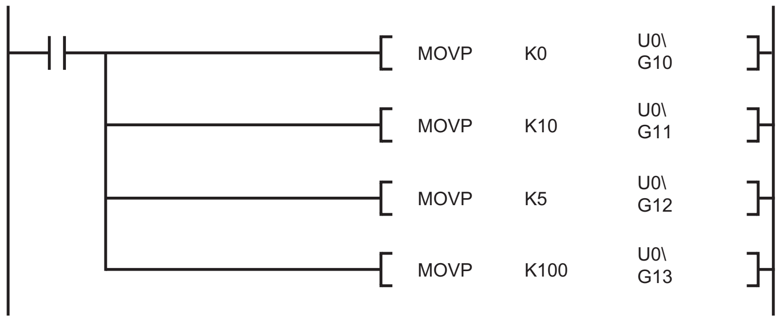
Specify this device with 'Un\Gn'. (Example: U5\G11)

Specified item		Value to be specified
Un	Start I/O number of intelligent function modules	Upper two digits when a start I/O number is described in three digits (00H to FFH) Example: 1F stands for X/Y1F0
Gn	Buffer memory address	0 to 268435455 (decimal)

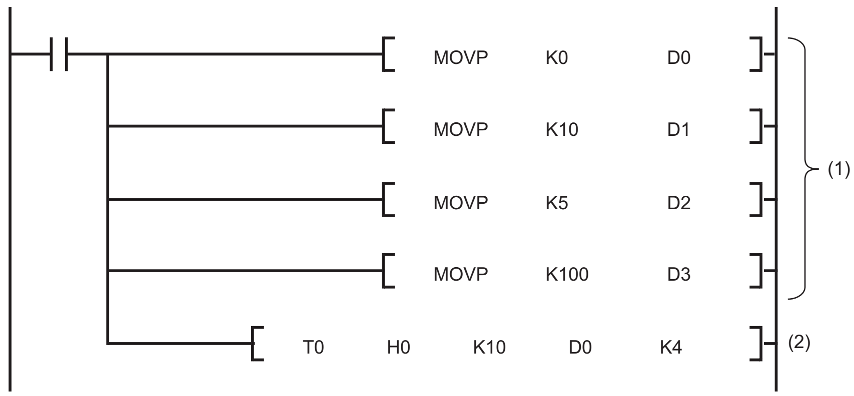
Point

When reading/writing the buffer memory data using the module access device more than twice within a program, the processing speed can be increased by conducting read/write operations at a single point of the program using the FROM/TO instruction.

- When data is written using the module access device more than twice:



- When data is written at a single point of the program using the TO instruction:



(1) Store data to devices such as the data register (D).

(2) Write data to only one point in the program.

Precautions

The following describes the precautions for when the module access device is used.

- If data is written to the refresh-target memory using a program while the refresh function is being used, the CPU module overwrites the data in the target memory at the execution of the refresh function. Thus, the expected operation may not be acquired. When the refresh function is used, do not write the data directly to the refresh-target memory but write it to the refresh-source memory.

*1 When the data is transferred from the CPU module to the module, the target memory is the buffer memory or link device. When the data is transferred from the module to the CPU module, the target memory is the specified device of the CPU module.

22.7 CPU Buffer Memory Access Device

This device accesses memory used by the built-in function of the CPU module, such as data writing/reading between CPU modules on the multiple CPU system and Ethernet function (☞ Page 346 Specification method thorough CPU buffer memory access device)

Specification method

Specify this device with 'Un\Gn'. (Example: U3E1\G4095, U3E2\HG1024)

Specified item		Value to be specified
Un (I/O number of the CPU module)	CPU No.1	3E0
	CPU No.2	3E1
	CPU No.3	3E2
	CPU No.4	3E3
G (CPU buffer memory area)	CPU buffer memory	G
	Fixed scan communication area	HG
n (CPU buffer memory address)		0 to 268435455 (decimal)

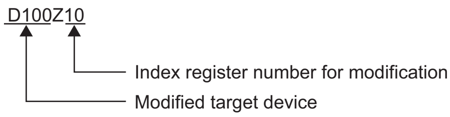
22.8 Index Register (Z/LZ)

This device is used for the index modification of the device. The index modification is the indirect specification using the index register.

Specify the device with the number obtained from "Device number of device targeted for modification" + "Contents of index register".

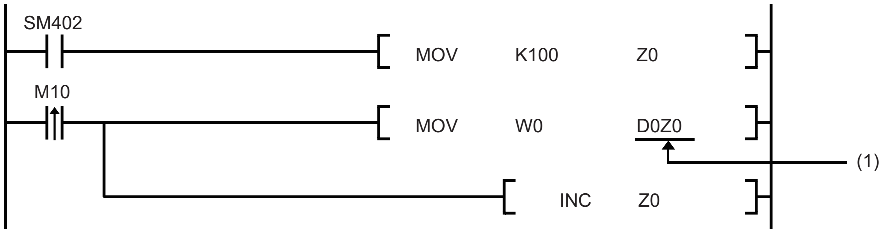
16-bit index modification

The device number is modified using the index register (Z). The modification range for the device in the case of the 16-bit index modification is -32768 to 32767.



Ex.

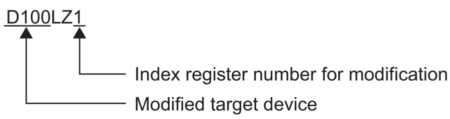
Modifying D0 with Z0



(1) Access D0Z0 = D100.

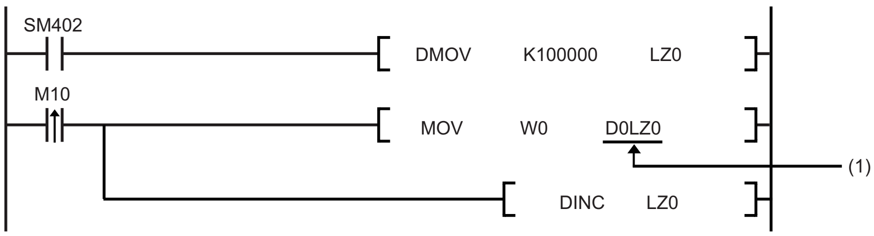
32-bit index modification

The device number is modified using the long index register (LZ). The modification range for the device in the case of the 32-bit index modification is -2147483648 to 2147483647.



Ex.

Modifying D0 with LZ0



(1) Access D0LZ0 = D100000.

In addition, 32-bit index modification with ZZ expression using two index registers is also available.

Device for which Index modification can be performed

The following table lists the devices that can be targeted for index modification.

Item	Description
16-bit index modification	X, DX, Y, DY, M, L, B, F, SB, V, S[*4], T[*1], LT[*1], ST[*1], LST[*1], C[*1], LC[*1], D, W, SW, SM, SD, Jn\X, Jn\Y, Jn\B, Jn\SB, Jn\W, Jn\SW, Un\G, U3En\G, U3En\HG, R, ZR, RD, P[*3], I[*3], BL[*4], BLn\S[*4], J, U, K, H
32-bit index modification	M, B, SB, T[*1], LT[*1], ST[*1], LST[*1], C[*1], LC[*1], D, W, SW, Jn\B[*2], Jn\W[*2], Un\G[*2], U3En\G[*2], U3En\HG[*2], R, ZR, RD, K, H

*1 The devices can be used for the contact, coil and current value.

*2 For network numbers and the specification source of I/O numbers, 32-bit index modification cannot be performed.

*3 When the devices are used as an interrupt pointer, index modification cannot be performed.

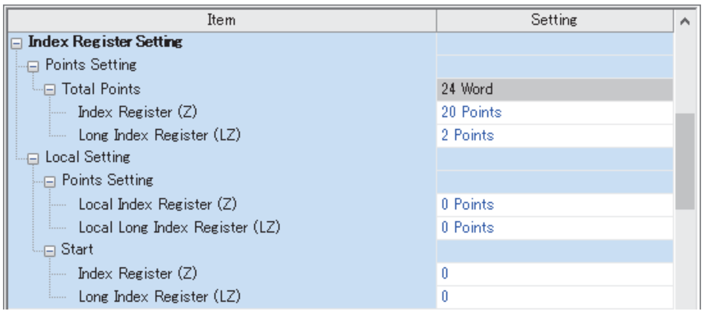
*4 If specifying the devices with the programmable controller CPU or Process CPU, check the versions of the CPU module and engineering tool used. (Page 1139 Added and Enhanced Functions)
Note that the devices cannot be specified with the SIL2 Process CPU or Safety CPU.

Index register setting

The following window allows to specify the number of points for the index register (Z) and long index register (LZ) and the range where they are used as a local device. The total number of points of the index register (Z) and the long index register (LZ) must be set to 24 words.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Index Register Setting]

Window



Displayed items

Item			Description	Setting range	Default
Points Setting	Total Points		Checks the total number of points for index register and long index register.	—	—
	Index Register (Z)		Sets the number of points for the index registers.	0 to 24 Points (in two-point increments)	20 points
	Long Index Register (LZ)		Sets the number of points for the long index registers.	0 to 12 Points (in one-point increments)	2 points
Local Setting	Points Setting	Local Index Register (Z)	Sets the number of points for index registers used as a local device. Set within the range of the index register.	0 to 24 Points (in one-point increments)	0 points
		Local Long Index Register (LZ)	Sets the number of points for long index registers used as a local device. Set within the range of the long index register.	0 to 12 Points (in one-point increments)	0 points
	Start	Index Register (Z)	Sets the start number for the local index registers. Set within the range of the index register.	0 to 23	0
		Long Index Register (LZ)	Sets the start number for the local long index registers. Set within the range of the long index register.	0 to 11	0

Combination of index modification

This section describes the combination of index modification.

Modification order for the device specification and index modification

According to the priority order shown below, the device specification (digit specification, bit specification, indirect specification) and index modification can be applied. However, some word devices may not follow the priority order shown below.

Order of priority	When the device targeted for the device specification and index modification is the bit device	When the device targeted for the device specification and index modification is the word device
High ↑ ↓ Low	1: Index modification 2: Digit specification	1: Index modification 2: Indirect specification 3: Bit specification

Specification method combined with device specification

The device targeted for specification is modified in order of: 1st modification, 2nd modification and then 3rd modification. Besides, the following contents can be used only for the device for which the 1st modification can be applied. (For example, index modification + digit specification is impossible for the function input (FX).)

Device targeted for specification	1st modification	2nd modification	3rd modification	Example
Bit device	Index modification	Digit specification	—	K4M100Z2
Word device	Index modification	Bit specification	—	D10Z2.0
	Index modification	Indirect specification	—	@D10Z2
	Bit specification	Index modification	—	D10.8Z2
	Indirect specification	Bit specification	—	@D10.8
	Index modification	Indirect specification	Bit specification	@D10Z2.8
	Indirect specification	Bit specification	Index modification	@D10.8Z2

Precautions

This section describes the precautions on using index modification.

Index modification between the FOR and NEXT instructions

Between the FOR instruction and the NEXT instruction, pulse output is provided through the edge relay (V). However, pulse output by the PLS, PLF, or pulse conversion (□P) instruction is not available (☞ Page 382 Edge relay (V))

Index modification by the CALL instruction

In the CALL instruction, pulse output is provided through the edge relay (V). However, pulse output by the PLS, PLF, or pulse conversion (□P) instruction is not available (☞ Page 382 Edge relay (V))

Device range check for index modification

For details on the device range check when index modification is performed, refer to the following.

📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

Change of the index modification range due to switching from 16-bit to 32-bit

To change the index modification range for switching from 16-bit to 32-bit, the user must:

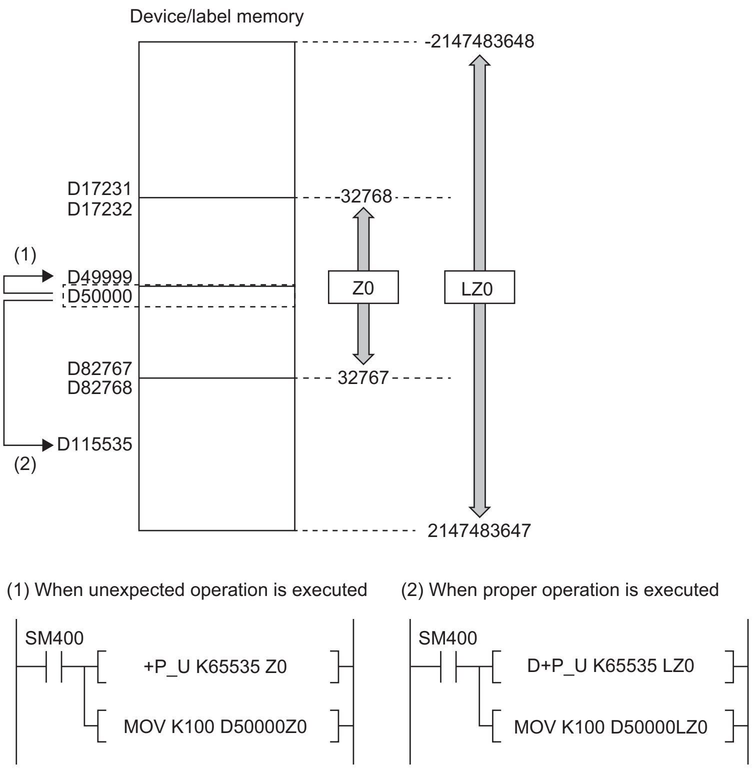
- Review the index modification block(s) within the program.
- To perform the 32-bit index modification specification with ZZ expression, review the range of the index register (Z). Note that the range within the LZ cannot be specified.
- For 32-bit index modification with ZZ expression, because the specified index register (Zn) and the immediately following index register (Zn+1) are used, caution must be taken to prevent duplicated index registers from being used.
- Review the number of points of the index register (Z) and that of the long index register (LZ), which are specified in "Index Register Setting" (☞ Page 402 Index register setting)

When values are stored in the index registers

For 16-bit index modification using the index register (Z), the range is -32768 to 32767. Therefore, when values within the range from 32768 to 65535 are stored in the index register (Z) for an instruction which processes unsigned data, the instruction does not work in design because the range of the index modification will be -32768 to 32767. For the range of values larger than or equal to 32768, the long index register (LZ) must be used so that 32-bit-based index modification can be applied.

Ex.

Operation for Index modification



(1) When the value 65535 is stored in the index register (Z), D50000(-1) to D49999 are accessed because the value is turned into -1 when an index modification is performed.

(2) When a value larger than or equal to 32768 is used for an index modification, the value must be stored in the long index register (LZ). In doing so, the value 65535 is used as such for an index modification using the long index register (LZ) and D50000 (65535) to D115535 become accessible.

22.9 File Register (R/ZR)

This device is a word device for extension. This device is specifically a file register file which exists in the file storage area on the device/label memory.

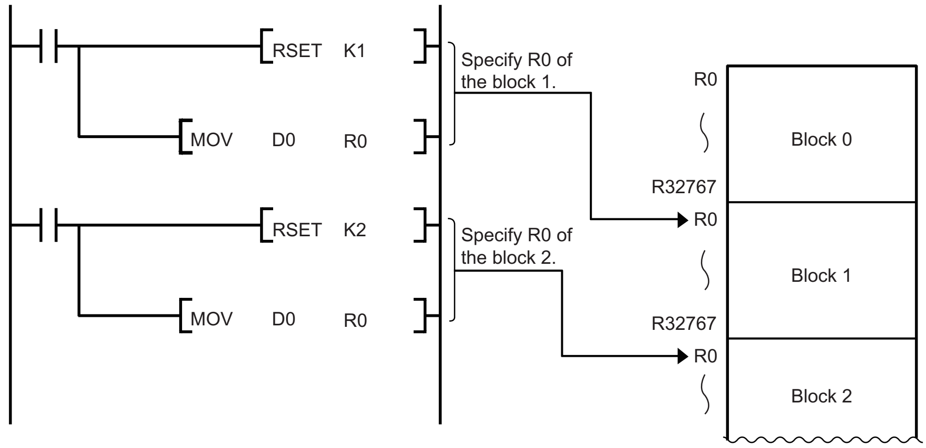
Specification method

There are two types of the specification methods for the file register: block switching and serial number methods.

Block switching method

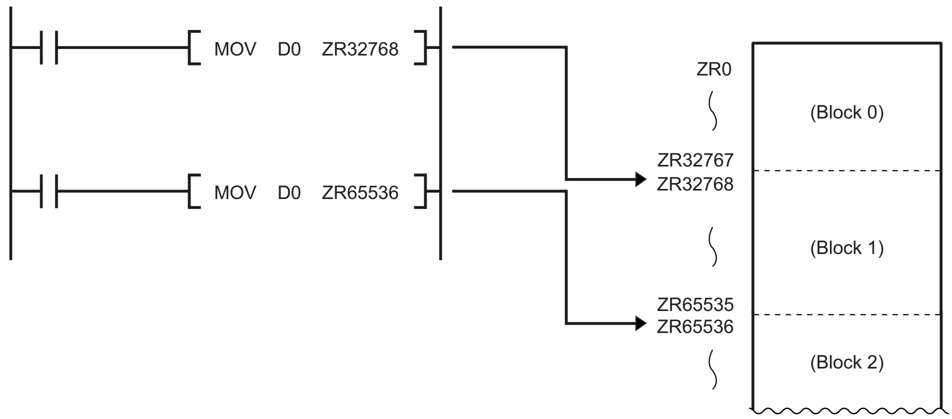
In this method the number of points of file register being used is specified by being divided in increments of 32K point (R0 to R32767). When using more than one block, specification is conducted by switching to the block number used by the RSET instruction. "R" is used as the device symbol. The range of "R" is from R0 to R32767. However in the following cases the upper limit of the device number is "block size (unit: word) -1".

- The file register file size is smaller than 64K bytes.
- The file register file size is not a multiple of 64K bytes and the end block is specified in the RSET instruction.



Serial number method

In this method file registers having more than 32K points are specified using serial device numbers. File registers of the multiple blocks can be used as consecutive file registers. "ZR" is used as the device symbol. The range of ZR is from ZR0 to (file register file size (unit: word) -1).



22

Setting file registers

This section describes the settings required to use the file registers.

Configuration procedure

This section describes the procedure to use the file registers.

1. Set the file register usage with [CPU Parameter].

2. To use the file register for each program, previously create the device memory which will become the file register file. (📖 GX Works3 Operating Manual)

3. When using the file registers, which are common for all programs, a file register file with the name and capacity set in the file register setting is created.*1

*1 If the capacity is not set, it must be set when creating a file register file and writing it to the programmable controller in the same manner as the procedure 2.

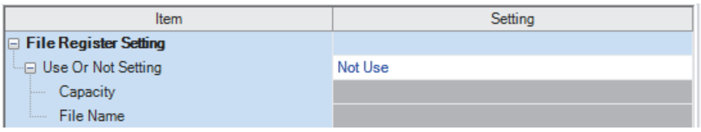
4. Write parameters and file register files into the CPU module.

File register setting

This setting must be completed before using the file registers.

[CPU Parameter] ⇨ [File Setting] ⇨ [File Register Setting]

Window



Displayed items

Item	Description	Setting range	Default
Use Or Not Setting	Specifies whether or not file registers should be used.	• Not Use • Use File Register of Each Program • Use Common File Register in All Programs	Not Use
Capacity	Specifies the capacity of the file register in increments of 1K words when "Use Common File Register in All Programs" is selected.	This value depends on whether or not the extended SRAM cassette is mounted and its capacity. (☞ Page 104 The setting range of the capacity of each area)	—
File Name	Assigns a file name to the file register when "Use Common File Register in All Programs" is selected.	1 to 60 characters	—

■Precautions

To access the CPU module from either of the following modules or engineering tools, the size of the file register needs to be 5696K words or less with an extended SRAM cassette (NZ2MC-16MBS) of 16MB.

- C Controller module
- MES interface module
- GOT2000 (when the programmable controller program monitor (R ladder) is used)
- Engineering tool version 1.015R or earlier
- CPU Module Logging Configuration Tool version 1.49B or earlier
- GX LogViewer version 1.49B or earlier
- MX Component version 4.10L or earlier

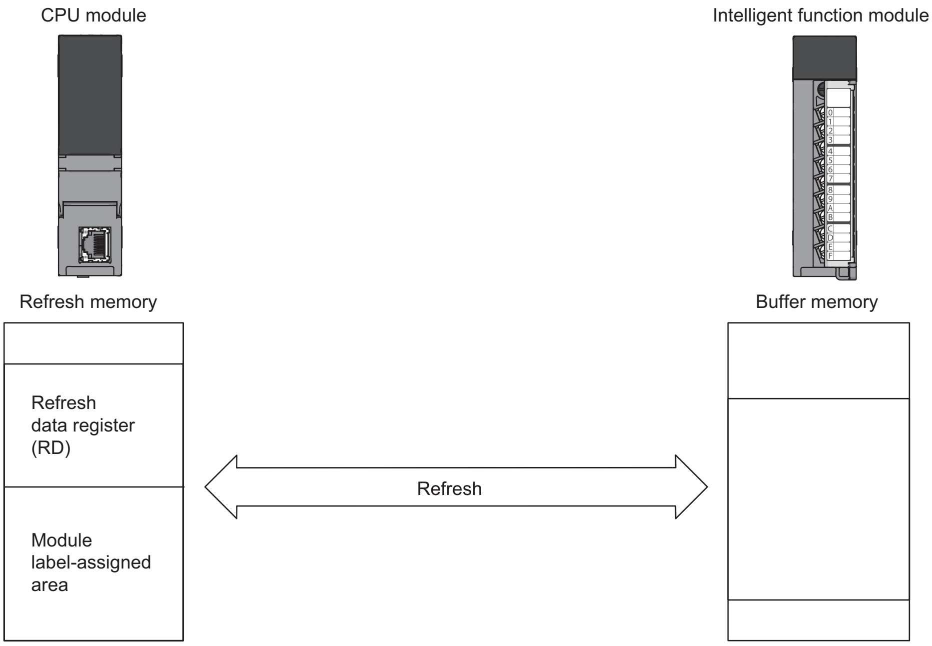
Clearing file registers

To clear the file registers, use the following methods (☞ Page 111 Memory Operation)

- Clearing in the program: write 0 into the file register range to be cleared.
- Clearing with engineering tool: clear them using engineering tool (📖 GX Works3 Operating Manual)

22.10 Refresh Data Register (RD)

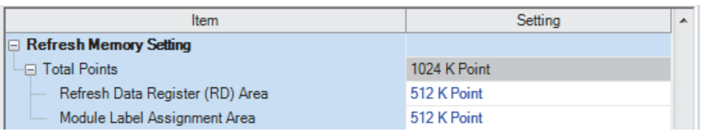
This device is provided for using as a refreshing target of buffer memory on the various devices, such as an intelligent function module. Refresh Data Register (RD) is assigned into the refresh memory area. (☞ Page 106 Refresh memory)



Refresh memory setting

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Refresh Memory Setting]

Window



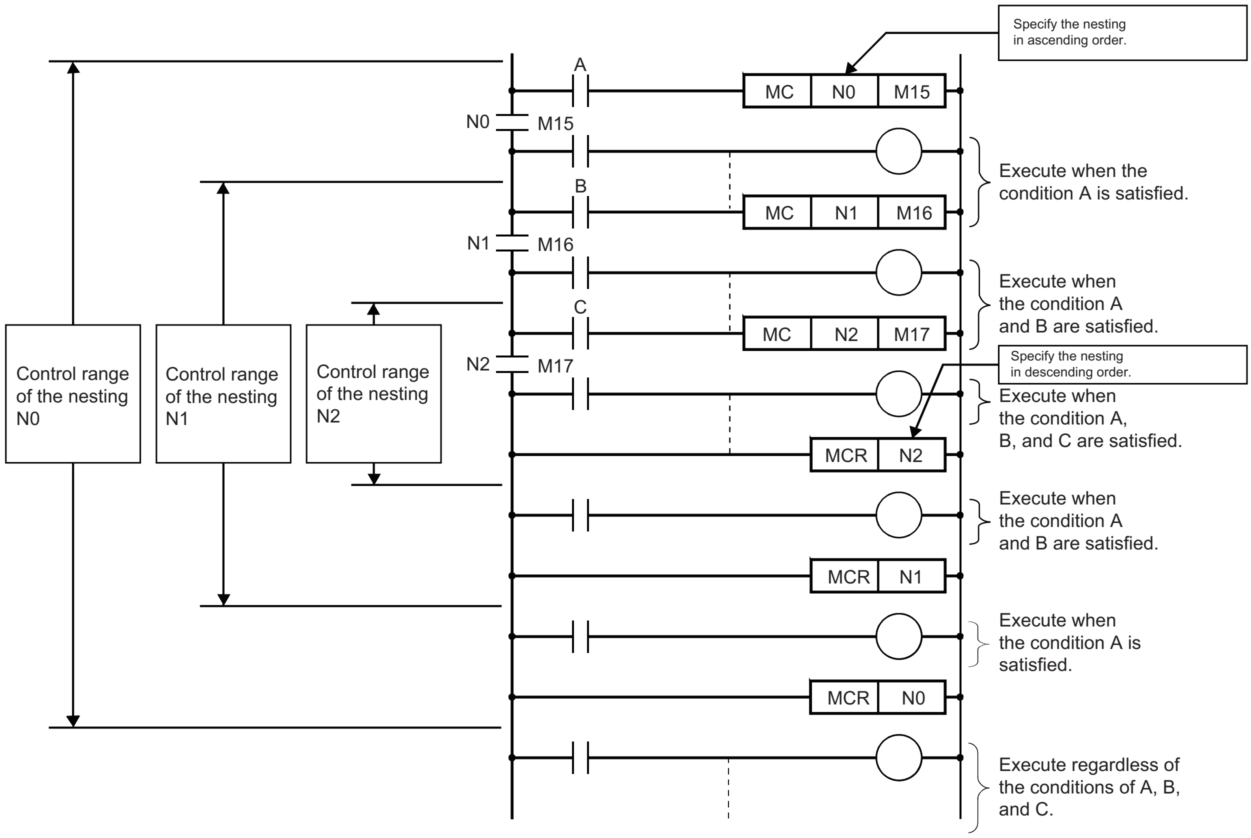
Item	Setting
Refresh Memory Setting	
Total Points	1024 K Point
Refresh Data Register (RD) Area	512 K Point
Module Label Assignment Area	512 K Point

Displayed items

Item	Description	Setting range	Default
Total Points	Checks the total number of points for the refresh data register and the assigned area.	—	1024K Point
Refresh Data Register (RD) area	Sets the number of points of refresh data register.	0 to 1024K points (in one-point increments)	512K Point
Module Label Assignment Area	Sets the number of points for the module label assigned area.	0 to 1024K points (in one-point increments)	512K Point

22.11 Nesting (N)

This device is used in the master control instructions (the MC/MCR instruction)*1 and enables the programming of operation conditions in a nesting structure. Specify this device from outside the nesting structure starting with the lowest number (in ascending order from N0 to N14).



*1 This instruction creates an effective ladder-switching program using opening/closing the common rail of the ladders.

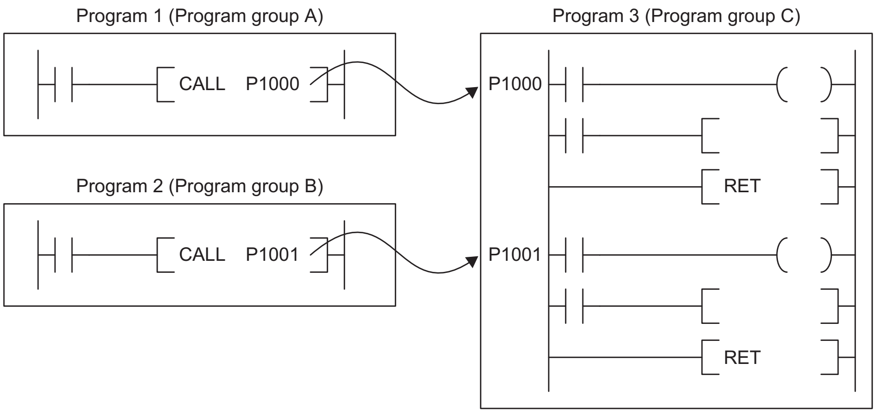
22.12 Pointer (P)

This device is used in the jump instructions (the CJ/SCJ/JMP instruction) and/or subroutine program call instructions (such as the CALL instruction). There are two types of pointer: the global pointer and the local pointer. Use the pointer when:

- Specifying the jump destination and label of the jump instructions (the CJ/SCJ/JMP instruction).
- Specifying the call destination and label (the head of subroutine program) of the subroutine call instructions (such as the CALL instruction).

Global pointer

This is the pointer which enables calling by the subroutine call instruction from all the program being executed.



Effective use of number of points

The number of points for global pointers can be obtained with the following formula: "pointer number of points specified by the parameter" - "the total point number of local pointer being used for each program".

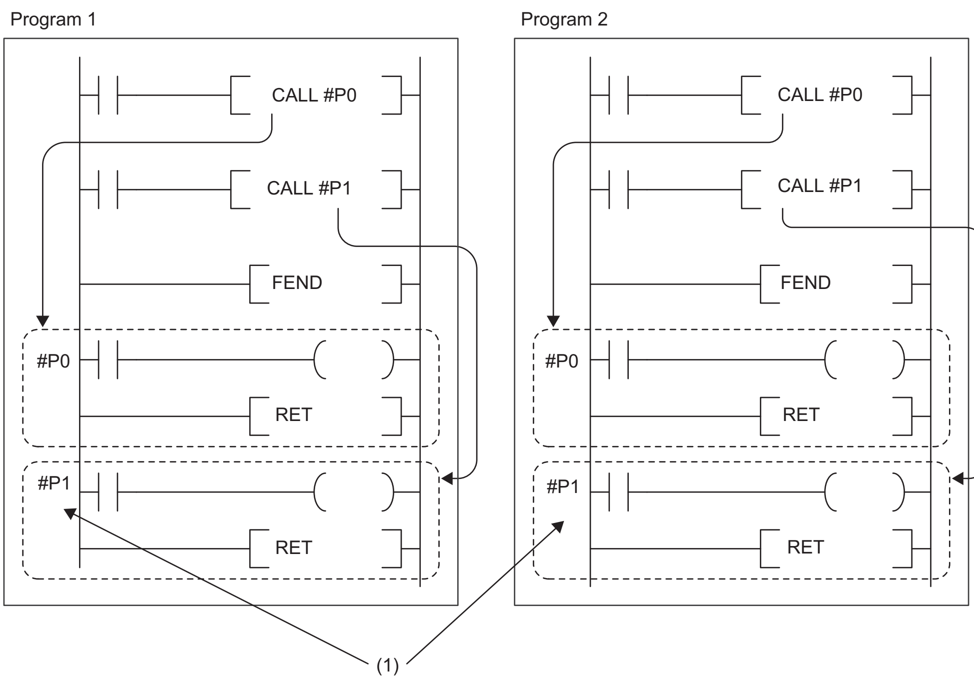
Precautions

This section describes the precautions when the global pointer is used.

- A global pointer with the same pointer number cannot be set as a label in multiple points.

Local pointer

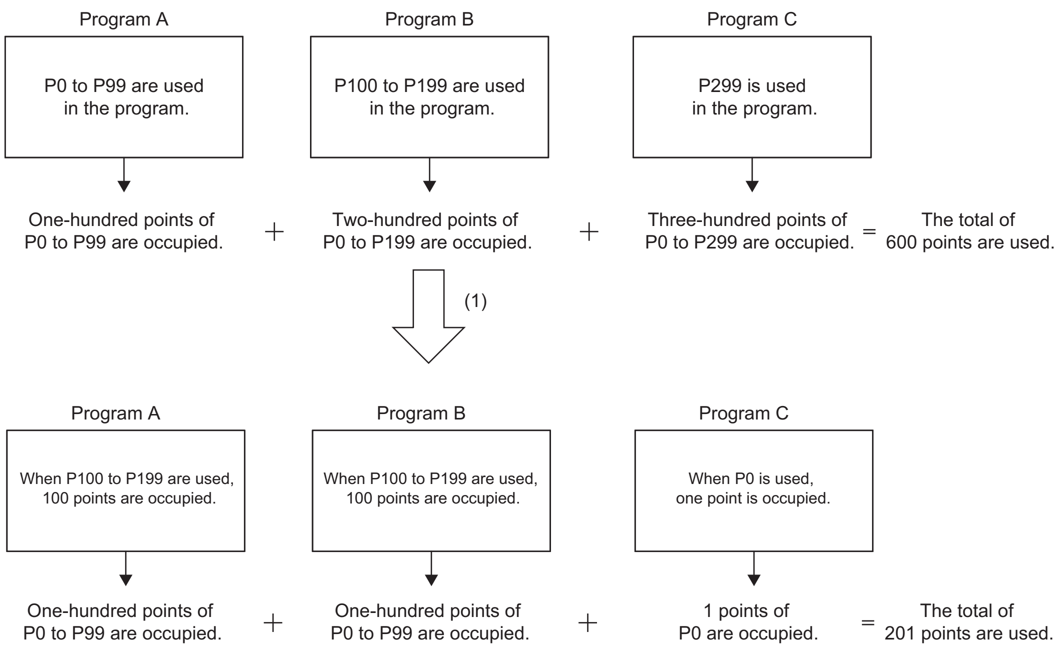
This is the pointer to be independently used in each program where the same pointer number can be used. This pointer is specified in the following format: # (pointer number) (Example: #P0) (☞ Page 418 Specification method for the local devices).



(1) The local pointers with the same number can be used between different programs.

Effective use of number of points

Local pointer number of points are shared among all the programs. The range of the local pointer number of points used by each program is from #P0 to the maximum value of the local pointer being used in that program. For example, even when a program actually uses only #P99, 100 points (#P0 to #P99) are considered to be used. When using local pointers in multiple programs, they can be effectively used by using them in ascending order from #P0 in each program group.



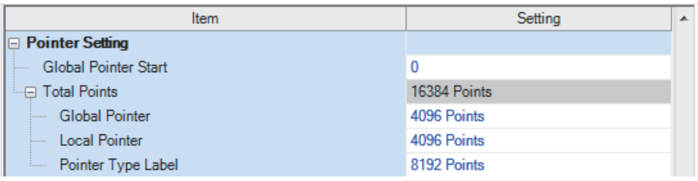
(1) By using pointers in ascending order from #P0 in each program group, a total of 201 points will be needed instead of a total of 600 points.

Pointer setting

The following menu item is to set pointers.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Pointer Setting]

Window



Displayed items

Item	Description	Setting range	Default
Global Pointer Start	Sets the start number of the global pointer.	P0 and over*1	0
Total Points	Checks the total number of points for the pointer.	—	16384 points*2
Global Pointer	Sets the number of points of the global pointer.	• R120CPU, R120ENCPU, R120PCPU, R120PSFCPU, R120SFCPU: 0 to 32768 points (in increments of 1 point) • Other CPU modules: 0 to 16384 points (in increments of 1 point)	4096 points*3
Local Pointer	Sets the number of points of the local pointer.		4096 points*3
Pointer Type Label	Sets the number of points for the pointer type label assignment area.		8192 points*4

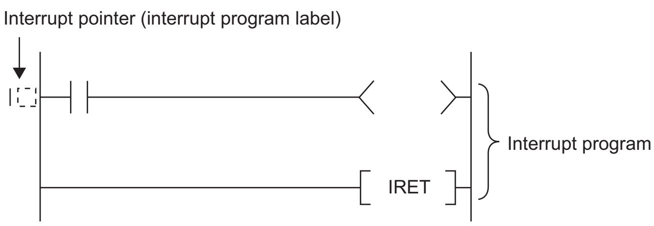
*1 Up to the number of "Total points of pointer device area" - "Number of points of global pointer".
*2 The default number of points is 32768 for the R120CPU, R120ENCPU, R120PCPU, R120PSFCPU, and R120SFCPU.
*3 The default number of points is 8192 for the R120CPU, R120ENCPU, R120PCPU, R120PSFCPU, and R120SFCPU.
*4 The default number of points is 16384 for the R120CPU, R120ENCPU, R120PCPU, R120PSFCPU, and R120SFCPU.

Point

Specify a pointer number which is equal to or lower than "(end number of the pointer range specified in the parameter) - (number of points of the global pointers)".

22.13 Interrupt Pointer (I)

This device is used as a label located at the head of the interrupt program. This pointer can be used in all the programs being executed.



Point

Setting the execution type of program to the event execution type eliminates the need to write (I□) the interrupt pointer. (☞ Page 65 Interrupt occurrence by the interrupt pointer (I))

Interrupt factors of the interrupt pointer numbers

The interrupt factors of the interrupt pointer numbers are indicated.

Interrupt factor	Interrupt pointer number	Description
Interrupt from module	I0 to I15	This is a pointer used for modules which have the interrupt function.
Interrupt by the internal timer	I28 to I31	This interrupt pointer is used in fixed scan interrupts by the internal timer.
Inter-module synchronous interrupt	I44	This fixed scan interrupt pointer is used in the inter-module synchronization function.
Multiple CPU synchronous interrupt	I45	This fixed scan interrupt pointer is used in the multiple CPU synchronization function.
High-speed internal timer interrupt 2	I48	This interrupt pointer is used in fixed scan interrupts by the internal timer and can be specified in a shorter interval than interrupt pointer numbers I28 to I31.
High-speed internal timer interrupt 1	I49	
Interrupt from module	I50 to I1023	This is a pointer used for modules which have the interrupt function.

The priority for the interrupt pointer numbers and interrupt factors

The priority for the interrupt pointer numbers and interrupt factors are indicated.

Interrupt pointer number	Interrupt factor		Interrupt priority	Interrupt priority order
I0	Interrupt from module	1st point	5 to 8	9
I1		2nd point		10
I2		3rd point		11
I3		4th point		12
I4		5th point		13
I5		6th point		14
I6		7th point		15
I7		8th point		16
I8		9th point		17
I9		10th point		18
I10		11th point		19
I11		12th point		20
I12		13th point		21
I13		14th point		22
I14		15th point		23
I15		16th point		24
I28	Interrupt by the internal timer		4	8
I29				7
I30				6
I31				5
I44	Inter-module synchronous interrupt		3	4
I45	Multiple CPU synchronous interrupt			3
I48	High-speed internal timer interrupt 2		2	2
I49	High-speed internal timer interrupt 1		1	1
I50 to I1023	Interrupt from module		5 to 8	25 to 998

Point

- The interrupt priority is the order which is executed at the time of the multiple interrupt. (☞ Page 89 Interrupt priority)
- The interrupt priority order is the order which is executed when the interrupt factor with the same interrupt priority is generated. (☞ Page 92 Multiple interrupt execution sequence)

22.14 Network No. Specification Device (J)

This device is used when specifying a network number with the Link dedicated instruction. (📖 MELSEC iQ-R Programming Manual (Module Dedicated Instructions))

22.15 I/O No. Specification Device (U)

This device is used when specifying an I/O number with the intelligent function module dedicated instruction. (📖 MELSEC iQ-R Programming Manual (Module Dedicated Instructions))

22.16 SFC Block Device (BL)

This device is used when specifying SFC program blocks. This device is also used when specifying step No. through such methods as verifying (monitor, current value changes) SFC programs with SFC control instructions or the engineering tool. (📖 MELSEC iQ-R Programming Manual (Program Design))

22.17 SFC Transition Device (TR)

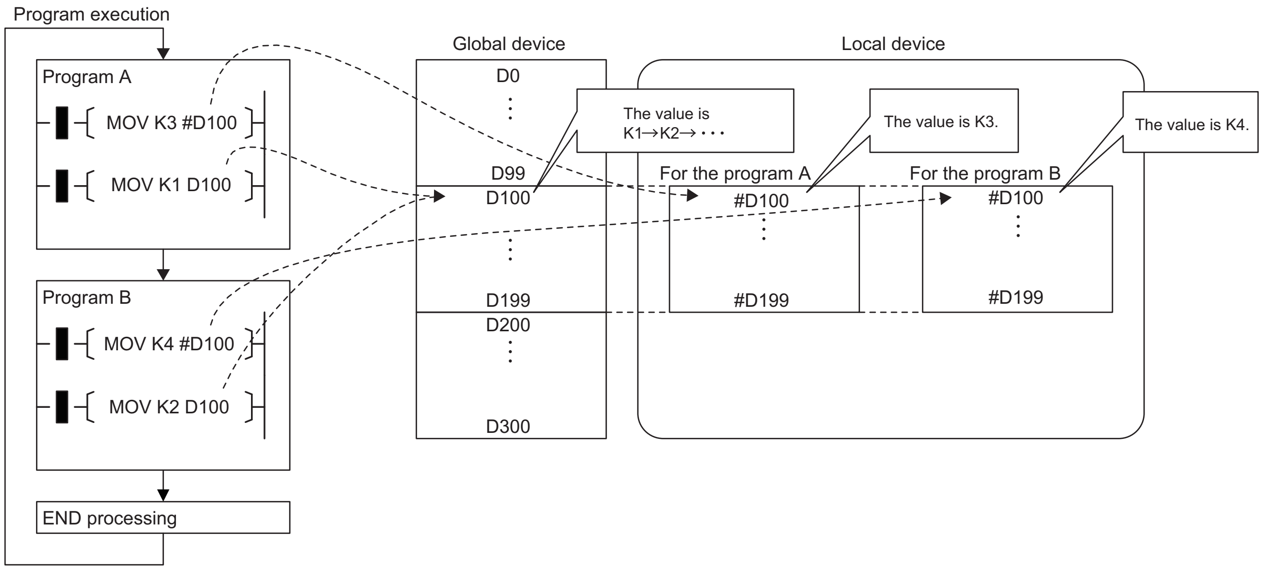
This device is used when specifying SFC program transition conditions. This device can only be used for device comments for transition conditions. (📖 MELSEC iQ-R Programming Manual (Program Design))

22.18 Global Device

This device can be shared by all the programs. All the devices that do not set as local device are handled as global device.

22.19 Local Device

This device can be used independently in each program. When creating multiple programs, programming can be completed without being aware of devices used in other programs.



Point

The local device area must be reserved independently from the global device area. Therefore a local device and global device with the same device number can exist.

Devices available as local device

The following devices are available as local devices.

- Internal relay (M)
- Edge relay (V)
- Timer (T, LT, ST, LST)
- Counter (C, LC)
- Data register (D)
- Pointer (P)

Point

Because the index register (Z, LZ) saves/returns during program execution, it should be regarded separately as the local index register unlike other local devices. (☞ Page 401 Index Register (Z/LZ))

Local device area

The CPU module reserves the local device area on the device/label memory based on the number of points setting of the local device when:

- CPU module is powered off and on or is reset.
- Operating status of the CPU module is changed from STOP to RUN.

When local device is used in subroutine program

Local devices to be used vary depending on whether SM776 (Local device setting at CALL) is turned on or off. Local index register to be used is also determined according to the SM776 setting.

SM776	Local device to be used
Off	Uses local devices of the program file from which subroutine program is called.
On	Uses local devices of the program file into which the subroutine program is stored.

Restriction

- In terms of on/off setting for SM776, the value (on or off) used when the subroutine call occurs is considered to be effective. Therefore when on/off setting for SM776 is switched in the subroutine program, the modified value (on or off) is not effective until the next subroutine call occurs.
- On/off setting for SM776 cannot be specified for each program file because it is enabled for each CPU module.

When local device is used in interrupt and other programs

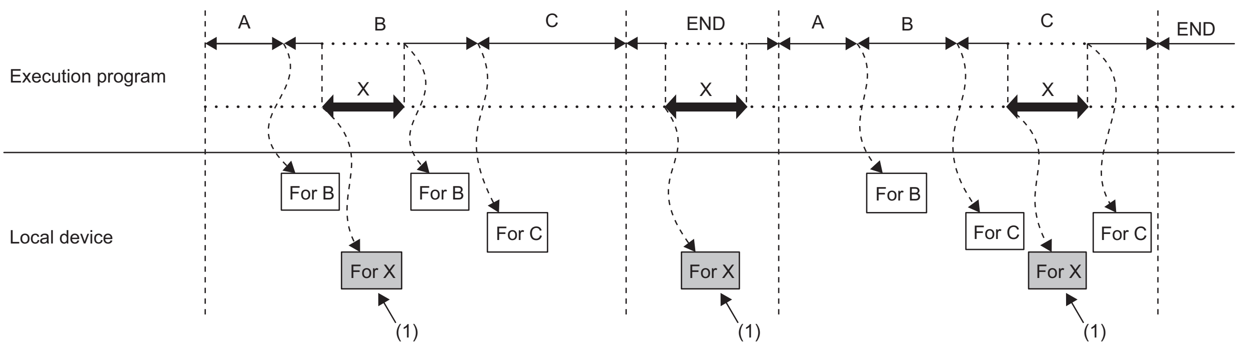
When using a local device for an interrupt program / a fixed scan execution type program / an event execution type program triggered by occurrence of an interrupt, turn on SM777 (Local device setting in interrupt programs). The programs will not function properly if SM777 is turned off.

For the SIL2 Process CPU and Safety CPU, however, the on/off setting of SM777 does not affect those programs. (☞ Page 782 SM777 setting, ☞ Page 658 SM777 setting)

Ex.

Operation if SM777 is turned on in following setting

Program name	Execution type	Local device use/not use
A	Scan	Not Use
B	Scan	Use
C	Scan	Use
X	Fixed scan	Use



(1) Uses the program X local device.

For local index register, the register of the program file which has been executed before these programs is used regardless of the SM777 setting.

Restriction

- For SM777, the value (on/off) set at the execution of an interrupt program / a fixed scan execution type program / an event execution type program triggered by occurrence of an interrupt is valid. For this reason, when the set value is changed while a program is being executed, the value changed does not become valid until the next time any of these programs is executed.
- On/off setting for SM777 cannot be specified for each program file because it is enabled for each CPU module.
- When the local device monitor is executed, the monitor switches to the applicable local device. Consequently, if SM777 is off, when an interrupt occurs immediately after switching, and a local device is accessed, the local device being monitored by the local device monitor is used. (The local device for the program being run prior to the interrupt (program immediately before END) is not accessed.)

Clearing local device

Local device can be cleared to 0 by the following operations:

- CPU module is powered off and on or is reset.
- CPU module status is changed from STOP to RUN.
- CPU module status is changed from PAUSE to RUN.

Setting method for the local devices

Set the range where each device will be used as a local device and also set whether or not it should be used.

■Range setting

The range setting for local devices is common to all the programs. Therefore the range for local devices cannot be set for each program.

Operating procedure

"Device/Label Memory Area Detailed Setting" window

Item	Setting
Device/Label Memory Area Detailed Setting	
Device Setting	<Detailed Setting>
Latch Type Setting of Latch Type Label	Latch (1)

"Device Setting" window

Item	Symbol	Local Device		
		Start	End	Points
Input	X			
Output	Y			
Internal Relay	M			
Link Relay	B			
Link Special Relay	SB			
Annunciator	F			
Edge Relay	V			
Step Relay	S			
Timer	T			
Long Timer	LT			
Retentive Timer	ST			
Long Retentive Time	LST			
Counter	C			
Long Counter	LC			
Data Register	D			
Link Register	W			
Link Special Registe	SW			
Latch Relay	L			
Total Device				0.0K Word
Total Word Device				0.0K Word
Total Bit Device				0.0K Bit

1. Click "Detailed Setting" on the "Device Setting" window.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Detailed Setting] ⇨ [Device Setting] ⇨ [Detail Setting]

2. Set the range where each device will be used as a local device.

Point

Configure the setting range of the local device within the range which has been set for the number of device points. The number of local devices used is calculated by the following calculation formula. Set the number of local devices used so that the number is equal to or less than the capacity of the local device area.

Total number of local devices used = $((A \div 16) + B + (C \times 2) + (D \times 4) + ((E \times 2) \div 16)) \times F$

- A: Number of points of the local devices M and V
- B: Number of points of the local devices D, T (current value), ST (current value), and C (current value)
- C: Number of points of the local device LC (current value)
- D: Number of points of the local devices LT and LST
- E: Number of points of the local devices T (contact/coil), ST (contact/coil), C (contact/coil), and LC (contact/coil)
- F: Number of programs using the local device

■Setting unit

The increment of setting of a local device is the same as that of a global device. (☞ Page 377 Range of use of device points)

Setting whether or not it should be used

Whether or not local devices should be used can be set for each program. Since the local device area of program for which "Do not use" has been set is not assured, it can suppress unnecessary consumption of device/label memory.

[CPU Parameter] ⇨ [Program Setting]

Operating procedure

"Program Setting" window

Item	Setting
Program Setting	
Program Setting	<Detailed Setting>

"Detailed Setting" window

Execute Order	Program Name	Execution Type		Refresh Group Setting	Device/File Use or not
		Type	Detailed Setting Information		
1	MAIN	Scan		(Do not Set)	<Detailed Setting>
2					

"Setting of Device/File Use Or Not" window

Execute Order	Program Name	Execution Type	Local Device Local Index Register Use or not	File Use Or Not	
				File Registers	Device Initial Value
1	MAIN	Scan	Use	Use File Setting	Use File Setting
2			Use	Use File Setting	Use File Setting
3			Use	Use File Setting	Use File Setting

1. Click "Detailed Setting" on the "Program Setting" window.
2. Click "Detailed Setting" of "Device/File Use or not".
3. Set whether or not it should be used for each program in the setting of "Local Device Local Index Register Use or not".

Restriction

Do not use local devices in a program which is configured not to use local devices.

Specification method for the local devices

To specify the local device in the program, add "#".

Ex.

For example, #D100, K4#M0, and @#D0 can be used.

Point

Local device is listed with a preceding # symbol in the program. This symbol is helpful to differentiate local devices from global devices.

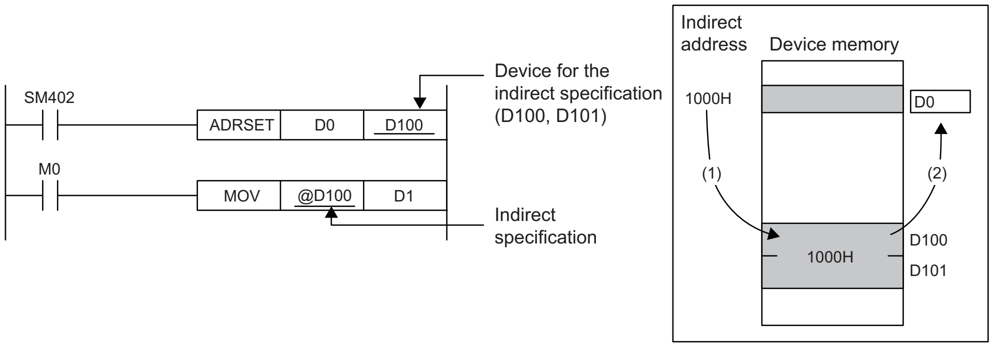
Precautions

This section describes the precautions when using local devices.

- Like global devices, the timer (T, LT, ST, LST) and counter (C, LC) specified as a local device cannot be checked for the device range. Therefore when operates the address in the index modification or indirect specification, be careful not to exceed the specified device range.
- Accessing the range including both global and local devices by the index modification is not allowed.
- When the range of the 32-bit index modification is across the setting ranges of local devices of the index register, proper index modification is impossible.
- Local devices are not latched.

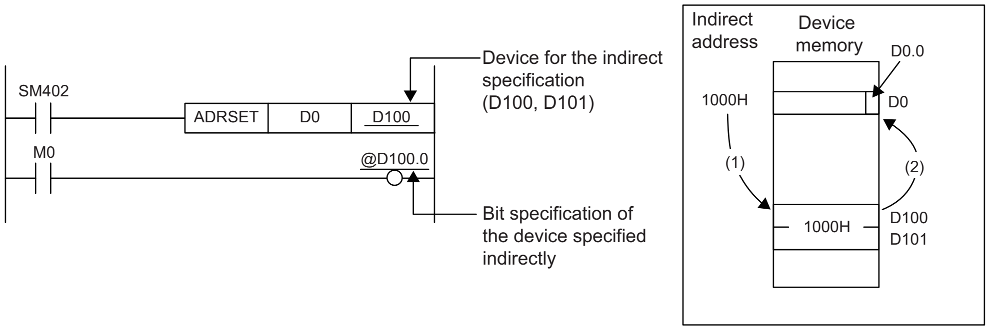
22.20 Indirect Specification

Specify the device using the indirect address of device. Store the indirect address of device to be specified into the device for indirect specification, and write as "@ + Device for indirect specification".



(1) The indirect address of D0 is read into D100, D101.
(2) The indirect address is used to indirectly specify D0.

Besides, specifying a bit of a word device allows the indirect specification for the instruction that specifies bits.



(1) The indirect address of D0 is read into D100, D101.
(2) The indirect address is used to output at the 0th bit of D0 which was indirectly specified using the indirect address.

The indirect specification can be used in the device/label memory or refresh memory.

Indirect address of device

To specify, use the 32-bit data, and to hold the value, use the device of two words. The indirect address of the device can be obtained with the ADRSET instruction. The ADRSET instruction specifies the indirect address of the device using instructions that handle 32-bit data. For the ADRSET instruction, refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

Point

When the block or the file of the file register is switched through the RSET or QDRSET instruction, the indirect address refers to the one of the block or the file before they are switched. To allow the indirect address in the device for indirect specification to specify the block or file after the file register is switched, specify the ADRSET instruction to obtain the indirect address again after block or file are switched.

Devices that can allow indirect specification

This section lists devices that can allow indirect specification.

Type	Device*3
Devices that can allow indirect specification where @ is added*1	T, ST, C, D, W, SW, FD, SD, Un\G, Jn\W, Jn\SW, U3En\G, U3En\HG, R, ZR, RD
Device that can acquire the indirect address through the ADRSET instruction*2	X, Y, M, L, B, F, SB, T, ST, C, D, W, SW, FX, FY, FD, SM, SD, R, ZR, RD

*1 Also can be used for the local device. (e.g.: @#D0)
*2 The indirect address of device can be obtained for the local device as well. (e.g.: ADRSET #D0 D100)
*3 Devices that cannot be used as operands of instructions cannot be used even when they are indirectly specified.

23 LABELS

A label is a variable consisting of a specified string used in I/O data or internal processing.
Programs can be created without considering the size of devices and buffer memory by using labels. For this reason, a program using labels can be reused easily even in a system having a different module configuration.
When labels are used, there are some precautions on programming and functions used. For details, refer to the following.
☞ Page 434 Precautions

Point

There are two types of labels described in this manual.
- Global labels
- Local labels

There are other types of labels available in addition to the global labels and local labels.
[System labels]
A system label is a label that provides the same data in all projects compatible with iQ Works. It can be referenced from the GOT and the CPU modules and Motion controllers on other stations, and used for monitoring and accessing data.
For details, refer to the following.
📖 Let's start iQ Works Version 2
[Module labels]
A module label is a label defined uniquely by each module. A module label is automatically generated by the engineering tool from the module used, and can be used as a global label.
For details, refer to the following.
📖 Function Block Reference for the module used

23.1 Global Labels

A global label is a label that provides the same data within a single project. It can be used in all programs in the project.
A global label can be used in program blocks and function blocks.
The settings of a global label include a label name, class, and data type.
By opening global labels, they can be referenced from GOT and other stations, and can be used for monitoring and accessing data.

Device assignment

Devices can be assigned to global labels.

Item	Description
Label to which no device is assigned	• Programming without being aware of devices is possible. • Defined labels are allocated to the label area or latch label area in the device/label memory.
Label to which a device is assigned	• If a device is to be programmed as a label against a device that is being used for input or output, the device can be assigned directly. • Defined labels are allocated to the device area in the device/label memory.

23.2 Local Labels

A local label is a label that can be used only in the declared POU. Local labels outside the declared POU cannot be used.
The settings of a local label include a label name, class, and data type.

23.3 Classes

The label class indicates from which POU and how a label can be used.
Different classes can be selected depending on the type of POU.

Global label				
Class	Description	Applicable POU		
		Program block	Function block (FB)	Function (FUN)
VAR_GLOBAL	A common label that can be used in both program blocks and function blocks	○	○	×
VAR_GLOBAL_CONSTANT	A common constant that can be used in both program blocks and function blocks	○	○	×
VAR_GLOBAL_RETAIN	A latch type label that can be used in both program blocks and function blocks	○	○	×

Local label				
Class	Description	Applicable POU		
		Program block	Function block (FB)	Function (FUN)
VAR	A label that can be used within the range of a declared POU. This label cannot be used in other POUs.	○	○	○
VAR_CONSTANT	A constant that can be used within the range of a declared POU. This label cannot be used in other POUs.	○	○	○
VAR_RETAIN	A latch type label that can be used within the range of a declared POU. This label cannot be used in other POUs.	○	○	×
VAR_INPUT	A label that inputs a value into a function or function block. This label receives a value, and the received value cannot be changed in a POU.	×	○	○
VAR_OUTPUT	A label that outputs a value from a function or function block	×	○	○
VAR_OUTPUT_RETAIN	A latch type label that outputs a value from a function block	×	○	×
VAR_IN_OUT	A local label that receives a value and outputs the value from a POU. The value can be changed in a POU.	×	○	×
VAR_PUBLIC	A label that can be accessed as the public variable from other POUs	×	○	×
VAR_PUBLIC_RETAIN	A latch type label that can be accessed as the public variable from other POUs	×	○	×

23.4 Data Types

The data types of a label are classified according to the bit length, processing method, and value range.

There are two data types.

- Primitive data type
- Generic data type (ANY type)

Primitive data type

The following table lists the data types included in the primitive data type.

Data type		Description	Value range	Bit length
Bit	BOOL	Represents the alternative status, such as on or off.	0 (FALSE), 1 (TRUE)	1 bit
Word [unsigned]/bit string [16 bits]	WORD	16-bit array	0 to 65535	16 bits
Double word [unsigned]/bit string [32 bits]	DWORD	32-bit array	0 to 4294967295	32 bits
Word [signed]	INT	Positive and negative integer values	-32768 to 32767	16 bits
Double word [signed]	DINT	Positive and negative double-precision integer values	-2147483648 to 2147483647	32 bits
Single-precision real number*1	REAL	Numerical values of decimal places (single-precision real number values)	-2^{128} to -2^{-126}, 0, 2^{-126} to 2^{128} E-3.402823+38 to E-1.175495-38, 0, E1.175495-38 to E3.402823+38	32 bits
Double-precision real number*1	LREAL	Numerical values of decimal places (double-precision real number values)	-2^{1024} to -2^{-1022}, 0, 2^{-1022} to 2^{1024} E-1.79769313486231+308 to E-2.22507385850721-308, 0, E2.22507385850721-308 to E1.79769313486231+308	64 bits
Time*2	TIME	Numerical values as day, hour, minute, second, and millisecond	T#-24d20h31m23s648ms to T#24d20h31m23s647ms*3	32 bits
String	STRING	Characters represented by ASCII code or Shift JIS code	255 one-byte characters maximum	Variable
String [Unicode]	WSTRING	Characters represented by Unicode	255 characters maximum	Variable
Timer	TIMER	Structure corresponding to the device, timer (T)	☞ Page 423 Timer and counter data types	
Retentive timer	RETENTIVETIMER	Structure corresponding to the device, retentive timer (ST)		
Long timer	LTIMER	Structure corresponding to the device, long timer (LT)		
Long retentive timer	LRETENTIVETIMER	Structure corresponding to the device, timer (LST)		
Counter	COUNTER	Structure corresponding to the device, counter (C)		
Long counter	LCOUNTER	Structure corresponding to the device, counter (LC)		
Pointer	POINTER	Type corresponding to the device, pointer (P) (☞ Page 409 Pointer (P))		

*1 For the number of significant digits and input range of real number data input by the engineering tool, refer to the following.
📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

*2 The time type is used in a time data type function of standard functions. For standard functions, refer to the following.
📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

*3 For the notation of time, refer to the following.
☞ Page 453 Notation of time

Point

- The bit data in the word type label can be used by specifying a bit number.
- The bit type array label can be used as 16-bit or 32-bit data by specifying the number of digits.

For the bit specification and digit specification methods, refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

■Timer and counter data types

The data types of the timer, counter, long counter, retentive timer, long retentive timer, and long timer are the structures having a contact, coil, or current value.

Data type		Member name	Member data type	Description	Value range
Timer	TIMER	S	Bit	Indicates a contact. The operation is the same as the contact (TS) of a timer device.	0 (FALSE), 1 (TRUE)
		C	Bit	Indicates a coil. The operation is the same as the coil (TC) of a timer device.	0 (FALSE), 1 (TRUE)
		N	Word [unsigned]/bit string [16 bits]	Indicates the current value. The operation is the same as the current value (TN) of a timer device.	0 to 65535*1
Retentive timer	RETENTIVETIMER	S	Bit	Indicates a contact. The operation is the same as the contact (STS) of a retentive timer device.	0 (FALSE), 1 (TRUE)
		C	Bit	Indicates a coil. The operation is the same as the coil (STC) of a retentive timer device.	0 (FALSE), 1 (TRUE)
		N	Word [unsigned]/bit string [16 bits]	Indicates the current value. The operation is the same as the current value (STN) of a retentive timer device.	0 to 65535*1
Long timer	LTIMER	S	Bit	Indicates a contact. The operation is the same as the contact (LTS) of a long timer device.	0 (FALSE), 1 (TRUE)
		C	Bit	Indicates a coil. The operation is the same as the coil (LTC) of a long timer device.	0 (FALSE), 1 (TRUE)
		N	Double word [unsigned]/ bit string [32 bits]	Indicates the current value. The operation is the same as the current value (LTN) of a long timer device.	0 to 4294967295*1
Long retentive timer	LRETENTIVETIMER	S	Bit	Indicates a contact. The operation is the same as the contact (LSTS) of a long retentive timer device.	0 (FALSE), 1 (TRUE)
		C	Bit	Indicates a coil. The operation is the same as the coil (LSTC) of a long retentive timer device.	0 (FALSE), 1 (TRUE)
		N	Double word [unsigned]/ bit string [32 bits]	Indicates the current value. The operation is the same as the current value (LSTN) of a long retentive timer device.	0 to 4294967295*1
Counter	COUNTER	S	Bit	Indicates a contact. The operation is the same as the contact (CS) of a counter device.	0 (FALSE), 1 (TRUE)
		C	Bit	Indicates a coil. The operation is the same as the coil (CC) of a counter device.	0 (FALSE), 1 (TRUE)
		N	Word [unsigned]/bit string [16 bits]	Indicates the current value. The operation is the same as the current value (CN) of a counter device.	0 to 65535
Long counter	LCOUNTER	S	Bit	Indicates a contact. The operation is the same as the contact (LCS) of a long counter device.	0 (FALSE), 1 (TRUE)
		C	Bit	Indicates a coil. The operation is the same as the coil (LCC) of a long counter device.	0 (FALSE), 1 (TRUE)
		N	Double word [unsigned]/ bit string [32 bits]	Indicates the current value. The operation is the same as the current value (LCN) of a long counter device.	0 to 4294967295

*1 The unit of the current value is set in CPU parameters ("Timer Limit Setting").

For details on the operation of each device, refer to the following.

Page 374 DEVICES

The specification method of each member is the same as that of the structure data type. (Page 429 Structures)

Generic data type (ANY type)

The generic data type is the data type of the labels which summarize several primitive data types.

Generic data types are used when multiple data types are allowed for function and function block arguments and return values.

Labels defined in generic data types can be used in any sub-level data type.

For the types of generic data types and the primitive data types, refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

Definable data types and initial values

The following tables list the definable data types and initial value setting possibilities for each label class.

Global label		
Class	**Definable data type**	**Initial value setting possibility**
VAR_GLOBAL	Primitive data type, array, structure, function block	○
VAR_GLOBAL_CONSTANT	Primitive data type*1*2	×
VAR_GLOBAL_RETAIN	Primitive data type*1, array, structure	○

Local label (program block)		
Class	**Definable data type**	**Initial value setting possibility**
VAR	Primitive data type, array, structure, function block	○
VAR_CONSTANT	Primitive data type*1*2	×
VAR_RETAIN	Primitive data type*1, array, structure	○

Local label (function)		
Class	**Definable data type**	**Initial value setting possibility**
VAR	Primitive data type*2, array, structure	×
VAR_CONSTANT	Primitive data type*1*2	×
VAR_INPUT	Primitive data type*1*2, array, structure	×
VAR_OUTPUT		×
Return value		×

Local label (function block)		
Class	**Definable data type**	**Initial value setting possibility**
VAR	Primitive data type, array, structure, function block	○
VAR_CONSTANT	Primitive data type*1*2	×
VAR_RETAIN	Primitive data type*1, array, structure	○
VAR_INPUT		○
VAR_OUTPUT		○
VAR_OUTPUT_RETAIN		○
VAR_IN_OUT		×
VAR_PUBLIC		○
VAR_PUBLIC_RETAIN		○

*1 The pointer type cannot be defined.

*2 None of the timer, retentive timer, long timer, long retentive timer, counter, and long counter types can be defined.

Point

- The initial value of the global label where the device has been assigned follows that in the device.
- The initial value of the function block follows the local label setting within the function block.
- The initial value of the structure type follows that of the structure definition.

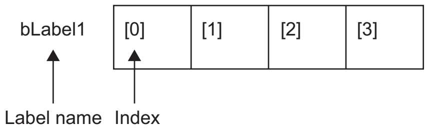
23.5 Arrays

An array represents a consecutive aggregation of same data type labels as a single name.
Primitive data types and structures can be defined as arrays.

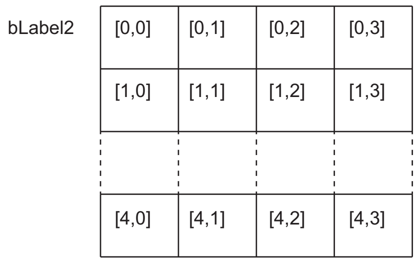
Ex.

Array image and setting in engineering tool

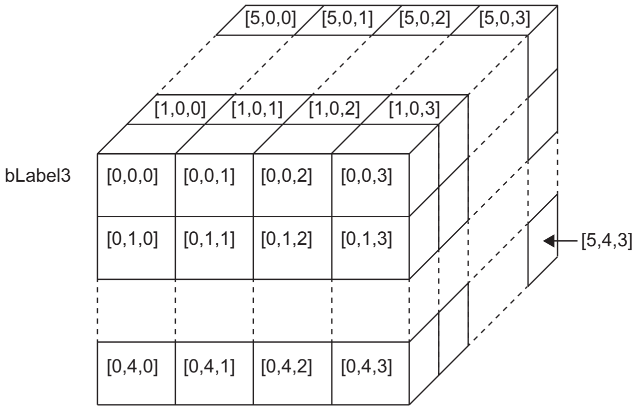
- One-dimensional array (The number of elements is 4.)



- Two-dimensional array (The number of elements is 5 × 4.)



- Three-dimensional array (The number of elements is 6 × 5 × 4.)



- Settings in the engineering tool

	Label Name	Data Type		Class
1	bLabel1	Bit(0..3)	...	VAR
2	bLabel2	Bit(0..4,0..3)	...	VAR
3	bLabel3	Bit(0..5,0..4,0..3)	...	VAR

Defining arrays

■Array elements

When an array is defined, the number of elements, or the length of array, must be determined. For the range of the number of elements, refer to the following.

☞ Page 428 Range of the number of array elements

■Dimension number of multidimensional array

Up to three-dimensional array can be defined.

■Definition format

The following table lists definition format.

The range from the array start value to the array end value is the number of elements.

Number of array dimensions	Format	Remarks
One dimension	Array of primitive data type/structure name (array start value..array end value)	• For the primitive data type: ☞ Page 422 Primitive data type • For the structure name: ☞ Page 429 Structures
	[Definition example] Bit (0..15)	
Two dimensions	Array of primitive data type/structure name (array start value..array end value, array start value..array end value)	
	[Definition example] Bit (0..1, 0..15)	
Three dimensions	Array of primitive data type/structure name (array start value..array end value, array start value..array end value, array start value..array end value)	
	[Definition example] Bit (0..2, 0..1, 0..15)	

■Initial value

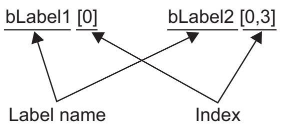
One initial value can be set for a single array definition. (Different initial values cannot be set for each element.)

The same initial value is stored in all the array elements.

How to use arrays

To use an array, add an index enclosed by '[ ]' after each label name to identify individual labels.

An array with two or more dimensions should be represented with indexes delimited by a comma (,) in '[ ]'.

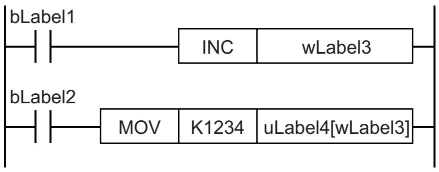


The following table lists the types of indexes that can be specified for arrays.

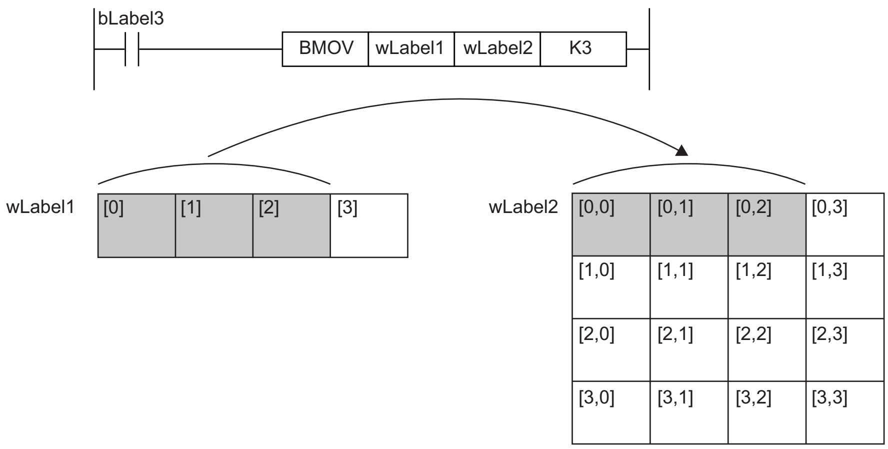
Type	Specification example	Remarks
Constant	bLabel1[0]	An integer can be specified.
Device	bLabel1[D0]	A word device, double-word device, decimal constant, or hexadecimal constant can be specified. (ST, LST, G, and HG cannot be specified.)
Label	bLabel1[uLabel2]	The following data types can be specified. • Word [unsigned]/bit string [16 bits] • Double word [unsigned]/bit string [32 bits] • Word [signed] • Double word [signed]
Expression	bLabel1[5+4]	Expressions can be specified only in ST language.

Point

- The data storage location becomes dynamic by specifying a label for the array index. This enables arrays to be used in a program that executes loop processing. The following is a program example that consecutively stores "1234" in the "uLabel4" array.



- The element number of the array can be omitted in ladder diagram. If the element number is omitted, it is regarded as the start number and converted. For example, when the defined label name is "boolAry" and the data type is "Bit (0..2, 0..2)", the operation of "boolAry[0,0]" is the same as that of "boolAry".
- When a multidimensional array is specified as setting data of instructions, functions, and function blocks that use arrays, the rightmost element is regarded as a one-dimensional array.



Range of the number of array elements

The maxim number of array elements varies depending on the data type.

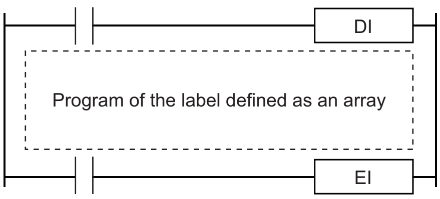
Data type	Setting range
Bit Word [unsigned]/bit string [16 bits] Word [signed]	1 to 2147483648
Double word [unsigned]/bit string [32 bits] Double word [signed] Single-precision real number Time	1 to 1073741824
Timer Counter Retentive timer Long counter Long retentive timer Long timer	1 to 32768
Double-precision real number	1 to 536870912
String	1 to 67108864
String [Unicode]	1 to 33554432
Structure type Function block	1 to 32768

Precautions

■When an interrupt program is used

When a label or device is specified for the array index, the operation is performed with a combination of multiple instructions. For this reason, if an interrupt occurs during operation of the label defined as an array, data inconsistency may occur producing an unintended operation result.

To prevent data inconsistency, create a program using the DI/EI instructions that disables/enables interrupt programs as shown below.



For the DI/EI instructions, refer to the following.

📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

■Array elements

When accessing the element defined in an array, access it within the range of the number of elements.

If a constant out of the range defined for the array index is specified, a compile error will occur.

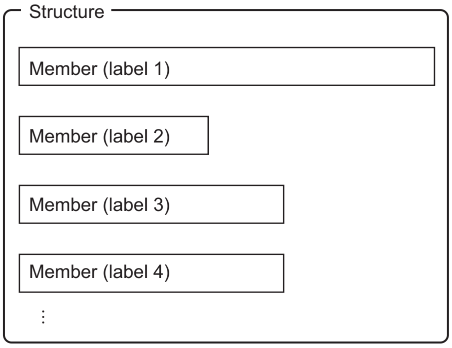
If the array index is specified with data other than a constant, a compile error will not occur. The processing will be performed by accessing another label area or latch label area.

23.6 Structures

A structure is a data type containing one or more labels and can be used in all POUs.
Members (labels) included in a structure can be defined even when their data types are different.

Creating structures

To create a structure, first define the structure, and then define members in the structure.



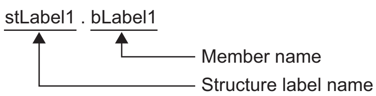
How to use structures

To use a structure, register a label using the defined structure as the data type.
To specify each member in a structure, add the member name after the structure label name with a period '.' as a delimiter in between.

23

Ex.

Specifying a member in the structure



Point

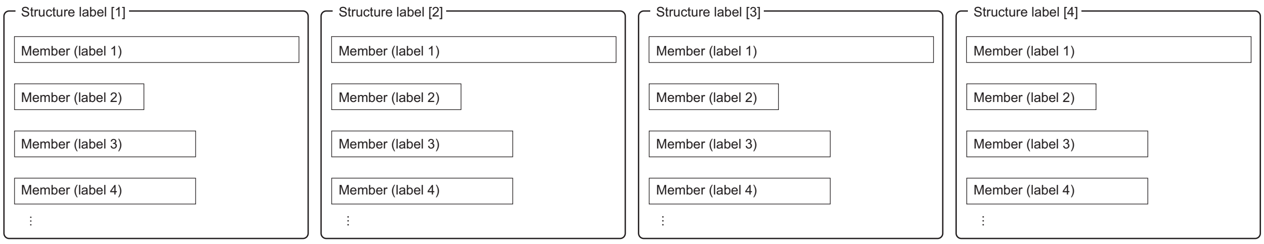
- When labels are registered by defining multiple data types in a structure and used in a program, the order the data is stored after operation is not the order the data types were defined. When programs are converted using the engineering tool, labels are classified into type and data type, and then assigned to the memory (memory assignment by packing blocks).

GX Works3 Operating Manual

- If the label of a structure is specified for an instruction that uses control data (a group of operands that determines operation of the instruction), the labels are not assigned in the order defined by packing blocks.

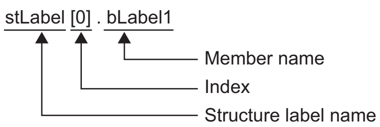
Structure arrays

A structure can also be used as an array.



When a structure is declared as an array, add an index enclosed by '[ ]' after the structure label name.
A structure array can also be specified as an argument of a function or function block.

Ex.
Specifying an element of a structure declared as an array



Data types that can be specified

The following data types can be specified as structure members.
- Primitive data type
- Pointer type
- Array
- Other structures

Types of structures

Each of the following labels is predefined as a structure.

Type	Reference
Module label	📖 Function Block Reference for the module used
Timer type	☞ Page 422 Data Types
Retentive timer type	
Counter type	
Long timer type	
Long retentive timer type	
Long counter type	

23.7 Label Access Setting from External Device

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- This function cannot be used in the R00CPU, R01CPU, and R02CPU.
- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 492 FUNCTIONS

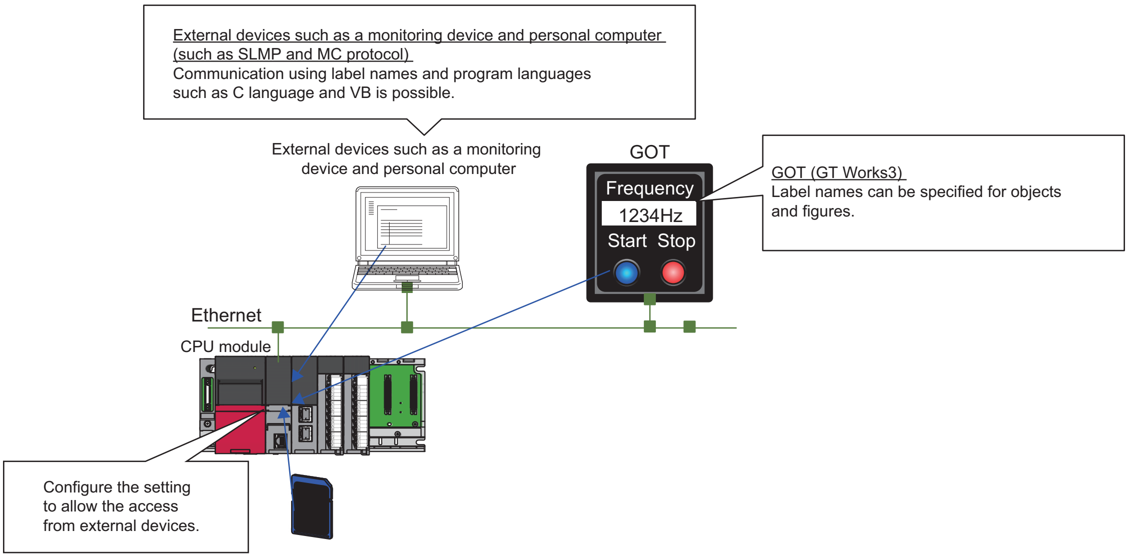
- When using the SIL2 Process CPU, refer to the following as well.

☞ Page 704 FUNCTIONS

- When using the Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

Set the parameters to enable external devices, such as GOT and SLMP-compatible devices, to specify global label names.

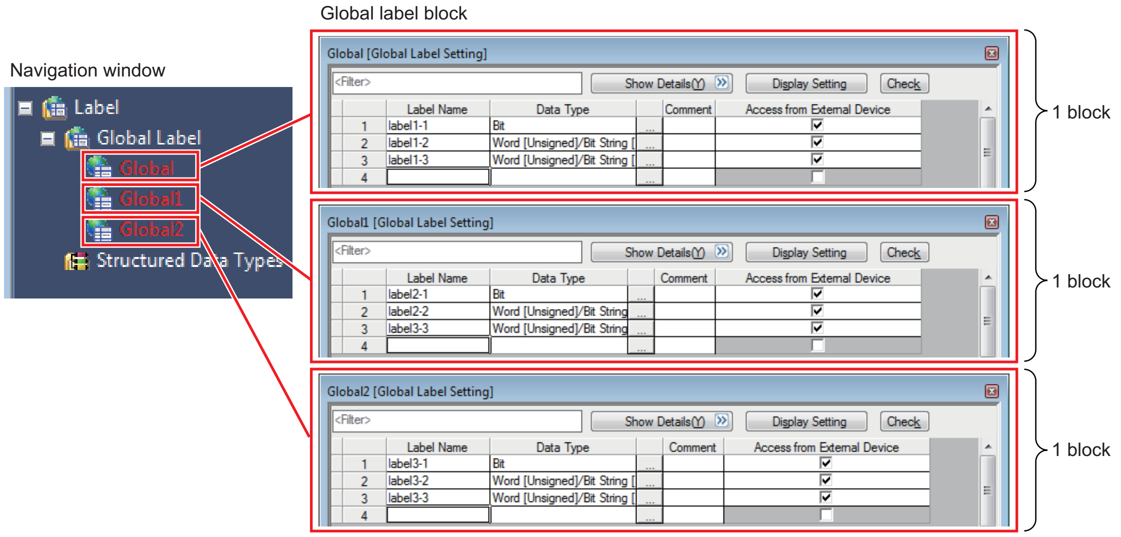


Point

The following CPU modules control global labels in the global label settings by block.

- Programmable controller CPU: Firmware version "40" or later
- Process CPU: Firmware version "28" or later
- Safety CPU: Firmware version "16" or later

One global label setting is shown as one block as below.



Set the global label setting for each label used with GOT in units of blocks.

When "Perform the label name resolution in global label block unit" is selected in the GT Designer3 setting, the label name resolution after online change is processed efficiently. However, during online change that involves addition of blocks, label names are resolved collectively, not in units of blocks.

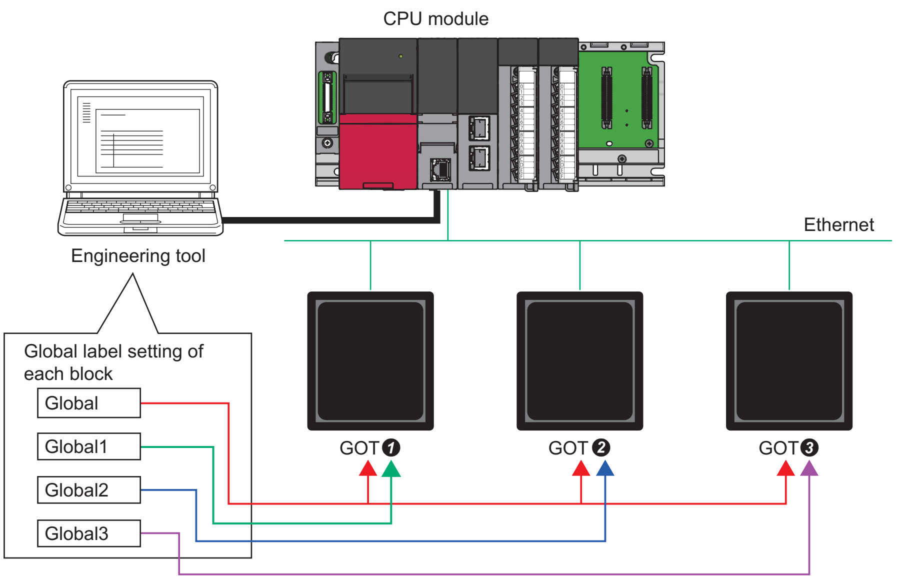
For the GT Designer3 setting, refer to the following.

📖 GT Designer3 (GOT2000) Screen Design Manual

The following shows an example of access to three GOT units.

- "Global": Global label setting that compiles labels accessed from multiple GOTs (GOT❶ to GOT❸)
- "Global1": Global label setting that compiles labels accessed from GOT❶
- "Global2": Global label setting that compiles labels accessed from GOT❷
- "Global3": Global label setting that compiles labels accessed from GOT❸

In this case, because global labels in each global label setting are accessed in units of blocks, the number of times of updating label communication data from GOT decreases.

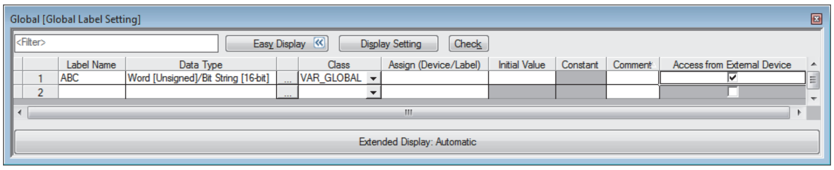


Configuration procedure

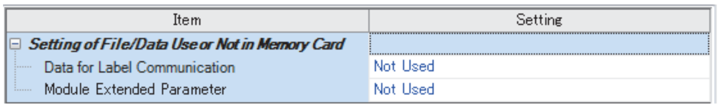
This section describes the configuration procedure to enable access by specifying the global label from external devices.

Operating procedure

"Global Label Setting" window



"Setting of File/Data Use or Not in Memory Card windows



1. Set the label in "Global Label Setting" and select the "Access from External Device" checkbox.

2. Check the capacity of the label communication data.

3. Enable "Data for Label Communication" to use the label communication data stored in the SD memory card (or store the label communication data in the SD memory card).

[Memory Card Parameters] ⇨ [Setting of File/Data Use or Not in Memory Card] ⇨ [Data for Label Communication]

4. Write parameters, global label setting, and global label assignment information files to the CPU module.

Point

Data storage to the data memory, not using the SD memory card, does not need the setting of procedure 3.

Label communication data storage location and operation at the time of each setting operation

This section describes the storage location for label communication data and the accessibility according to the setting of whether to use the file/data in a memory card.

○: Access from external device possible (Communication memory indicated in parentheses) ×: Access from external device not possible (Generates communication error)

Label communication data storage location	When "Enable" is set in the setting for the usage status of the file/data in a memory card	When "Disable" is set in the setting for the usage status of the file/data in a memory card
Data memory	×*1	○ (Data memory)
SD memory card	○ (SD memory card)	×
Data memory and SD memory card	○ (SD memory card)	○ (Data memory)

*1 An error occurs in the CPU module.

File operation

The label communication data can be written from an engineering tool. Reading and deleting*1 are not possible.

*1 Deleting an entire folder is possible.

23.8 Precautions

Functions with restrictions

The following functions have restrictions on the use of labels.

Item		Description
CPU parameter	• Trigger of an event execution type program • Refresh setting among multiple CPU modules	Use devices because global labels nor local labels cannot be specified for these functions.*1
Module parameter	• Predefined protocol support function	
	• Refresh setting of intelligent function module • Refresh setting of network module (SB/SW only)	Use module labels for these functions. Use devices if module labels are not used.*1
	• Refresh setting of network module (other than SB/SW)	Use devices because global labels nor local labels cannot be specified for these functions.*1
Data logging function Memory dump function Real-time monitor function		Use devices if there is a possibility for using these functions because global labels nor local labels cannot be specified for these functions.*1*2 In addition, if a device cannot be assigned to a global label, add the transfer instruction (copying the global label to a different device) on the scan program so that the instruction is executed every scan, and use that device.

*1 Global labels can be used as devices by assigning a device.

*2 Global labels or local labels can be specified depending on the CPU module for the data logging function. For the CPU modules which support the label specification, refer to the following.

☞ Page 1139 Added and Enhanced Functions

■Defining and using a global label with a device assigned

Define a global label following the procedure below, and use it when the functions having restriction on the use of labels are executed.

Since the device area in the device/label memory is used, secure the device area capacity.

1. Secure the device area to be used.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Setting]

2. Define a label as a global label, and assign a device manually.

3. Use the label defined in step 2 for the functions having no restrictions on the use of labels. Use the device assigned to the label for the function having restrictions on the use of labels.

■Copying the value of a label used to a different device

If a device cannot be assigned to a global label, copy the value of the label to a different device and use the device for the function having restrictions on the use of labels.

Since the device area in the device/label memory is used, secure the device area capacity.

1. Secure the device area to be used.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Setting]

2. Create a program using the label. The following is the program example for copying the data. (The data logging function uses the data in udLabel1.)

3. Use the device where the data has been transferred in step 2 for the function having restrictions on the use of labels. (In the program example in step 2, use D0.)

Point

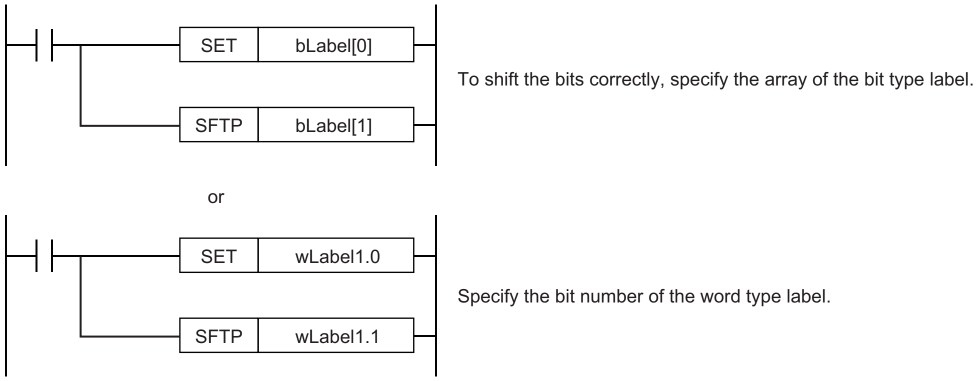
- The number of steps increases because of the transfer instruction. (The scan time increases.)
- Decide the transfer instruction position considering the timing of writing data to the label and executing the function.

Precautions for creating programs

When specifying a label as an operand used in instructions, match the data type of the label with that of the operand. In addition, when specifying a label as an operand used in instructions that control continuous data, specify the data range used in instructions within the data range of the label.

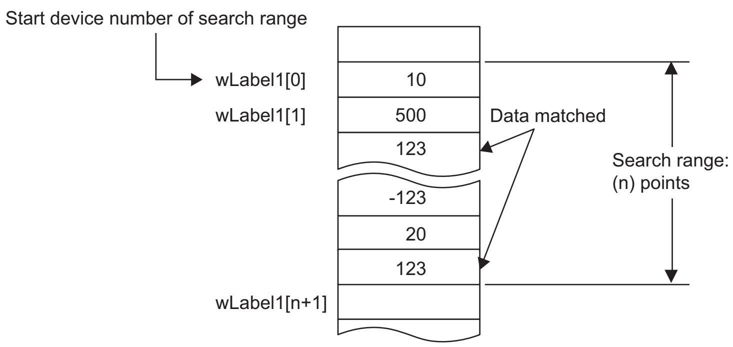
Ex.

SFT(P) instruction



Ex.

SER(P) instruction



Specify a label which has a larger data range than the search range (n) points.

Restrictions on naming labels

The following restrictions apply when naming labels.

- Start the name with a character or underline (_). Numbers cannot be used at the beginning of label names.
- Reserved words cannot be used.

For details on the reserved words, refer to the following.

GX Works3 Operating Manual

Program of external devices

When the global label which is set to the "Access from External Device" is deleted or the online change including label name change is executed, change the program of external devices that refer to the global label deleted/changed.

24 LATCH FUNCTION

24.1 Latch with Battery

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

• When using the Process CPU (redundant mode), refer to the following as well.
☞ Page 492 FUNCTIONS
• When using the SIL2 Process CPU, refer to the following as well.
☞ Page 704 FUNCTIONS
• When using the Safety CPU, refer to the following as well.
☞ Page 621 FUNCTIONS

The data in each device/label of the CPU module is cleared to its default value in the following cases:

- When the CPU module is powered off and on
- When the CPU module is reset
- When a power failure exceeding the allowable momentary power failure time occurs

The data in each device/label with latch setting is retained by using the battery on the CPU module unit even under the above described conditions. For this reason, control can be continued with the retained data even if the power of the CPU module is turned off or a power failure exceeding the allowable momentary power failure time occurs while the data is managed by a sequential control.

Point

The R00CPU, R01CPU, and R02CPU retain the data in each device/label with latch setting without a battery during power failure.
The other programmable controller CPUs retain the data in each device/label with latch setting without a battery during power failure by inserting a battery-less option cassette to the CPU module. (☞ Page 443 Latch with Battery-less Option Cassette)

Types of latch

The following types of latch are available:

- Latch (1): Data in this range can be cleared by a latch clear operation
- Latch (2): Data in this range cannot be cleared by a latch clear operation

Applicable devices and labels

This section describes the devices/labels that can be latched.

Applicable devices

The devices below can be latched. However, local devices cannot.

Device	Specification method	Applicable latch type
Internal relay (M)	Specify the latch range.	Latch (1) or Latch (2)
Link relay (B)	Specify the latch range.	Latch (1) or Latch (2)
Annunciator (F)	Specify the latch range.	Latch (1) or Latch (2)
Edge relay (V)	Specify the latch range.	Latch (1) or Latch (2)
Timer (T)/Long timer (LT)/Retentive timer (ST)/Long retentive timer (LST)	Specify the latch range.	Latch (1) or Latch (2)
Counter (C) and Long counter (LC)	Specify the latch range.	Latch (1) or Latch (2)
Data register (D)	Specify the latch range.	Latch (1) or Latch (2)
Link register (W)	Specify the latch range.	Latch (1) or Latch (2)
Latch relay (L)	Specify the number of points (latch is performed the same number of times as the specified number of points).	Latch (2) only
File register (R, ZR)	Specify the number of points (latch is performed the same number of times as the specified number of points). Specify the latch range.*1	Latch (2) only

*1 When "Use Common File Register in All Programs" is enabled, any of the number of points is registered within the range for Latch (2). Also, the range can be specified in "Latch (2)" of "Latch Range Setting" (values out of the specified range are cleared to 0).

Point

- When "Use File Register of Each Program" is enabled, the latch range for file registers cannot be specified (the whole range for file registers is retained).
- When the file register file which should be used is changed by the QDRSET instruction, the latch range setting for file registers is disabled. After the change, the whole range for file registers is retained regardless of the latch range setting.

Applicable labels

The labels below can be latched.

Label type	Class
Global label	VAR_GLOBAL_RETAIN
Local label of a program block	VAR_RETAIN
Local label of a function block	VAR_RETAIN
	VAR_OUTPUT_RETAIN
	VAR_PUBLIC_RETAIN

Setting latch on devices

Multiple latch ranges can be set for a device type. A total of 32 latch ranges between latch (1) and latch (2) can be set. However, the ranges of latch (1) and latch (2) must not overlap.

Setting a latch range

Set the device to latch, its range, and the latch type.

Operating procedure

"Device Setting" window

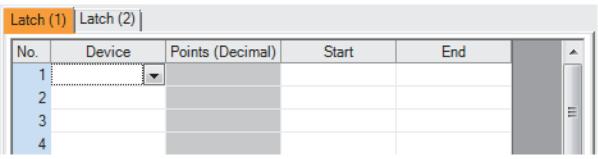
Item	Symbol	Latch (1)	Latch (2)
Input	*X*		
Output	Y		
Internal Relay	M	No Setting	No Setting
Link Relay	B	No Setting	No Setting
Link Special Relay	SB		
Annunciator	F	No Setting	No Setting
Edge Relay	V	No Setting	No Setting
Step Relay	S		
Timer	T	No Setting	No Setting
Long Timer	LT	No Setting	No Setting
Retentive Timer	ST	No Setting	No Setting
Long Retentive Time	LST	No Setting	No Setting
Counter	C	No Setting	No Setting
Long Counter	LC	No Setting	No Setting
Data Register	D	No Setting	No Setting
Link Register	W	No Setting	No Setting
Link Special Registe	SW		
Latch Relay	L		No Setting
	Total Device		
	Total Word Device		
	Total Bit Device		

1. Click "Detailed Setting" on the "Device Setting" window.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Detailed Setting] ⇨ [Device Setting] ⇨ [Detailed Setting]

2. In the "Device Setting" window, select the type of latch for the target device. The "Latch Range Setting" window is displayed.

"Latch Range Setting" window



No.	Device	Points (Decimal)	Start	End
1				
2				
3				
4				

3. Check the tab for the latch type, select the device to set and set the latch range (Start, End).

Setting the latch interval

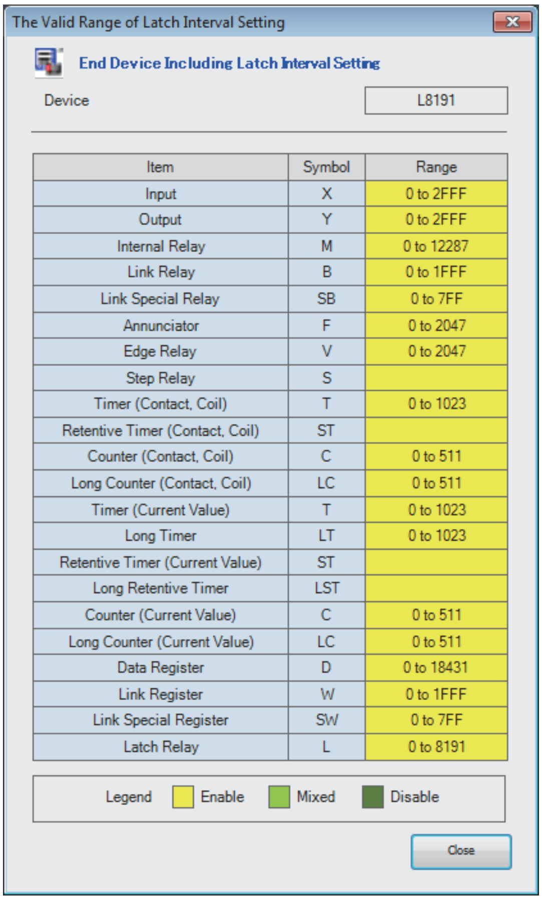
The user can specify the operation which should be performed at a latch interval (☞ Page 441 Timing of the latch processing) within the effective range of the latch interval setting*1.

*1 The effective range of the latch interval means the range of devices which is enabled on the "The Valid Range of Latch Interval Setting" window. For other ranges (ranges of devices which are not enabled on the "The Valid Range of Latch Interval Setting" window), this setting is not applied, the END processing is accelerated, and latching is performed in real time.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device Latch Interval Setting]

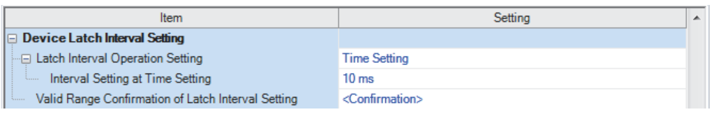
Operating procedure

"The Valid Range of Latch Interval Setting" window



1. On the "Valid Range Confirmation of Latch Interval Setting" window, check the effective range for the latch timing setting. Specify the latch interval operation within the effective range.

"Device Latch Interval Setting" window



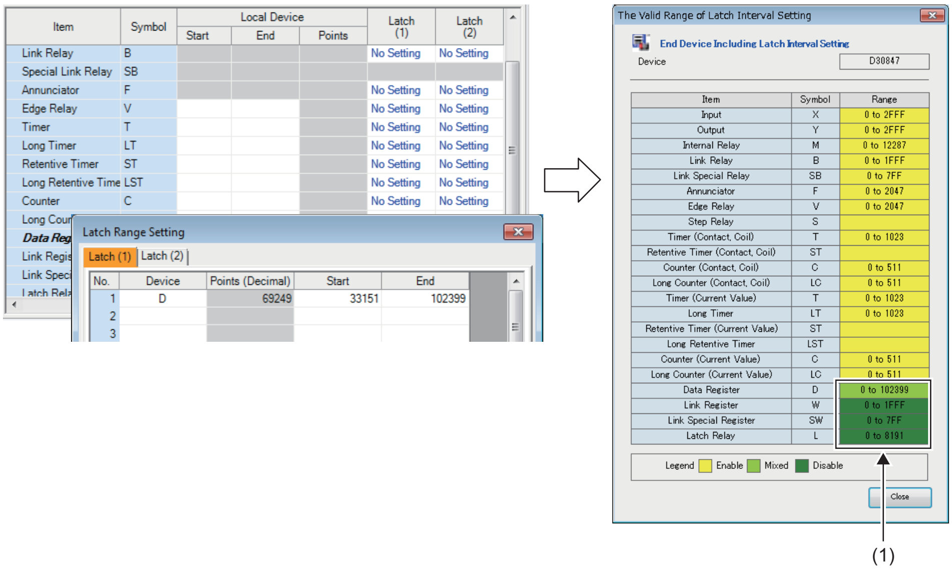
2. In "Latch Interval Operation Setting", select the timing of the latch processing. When "Time Setting" is selected, specify the time interval of latching.

Displayed items

Item	Description	Setting range	Default
Latch Interval Operation Setting	Sets the timing of the latch processing.	• Per Scan • Time Setting	Time Setting
Interval Setting at Time Setting	Sets the time interval of latching (when "Time Setting" is selected).	1 to 2000ms (in units of 1ms)	10ms
Valid Range Confirmation of Latch Interval Setting	Shows the effective range for the latch interval setting.	—	—

Point

For device latching, increasing the device range in the device setting of CPU parameters eliminates the latch processing from the END processing for the devices and enables real-time latching. For example, assume the following configuration for R04CPU: (1) 0K word is specified for the file storage area and 168K words for the device area in "Device/Label Memory Area Capacity Setting", (2) 100K points is specified for the data register (D) on the "Device Setting" window, and (3) D33151 and later are specified as the latch target devices. In this configuration, D30848 and later data registers (D) can be latched in real time as well as the devices (W, SW, and L) which are out of the effective range.



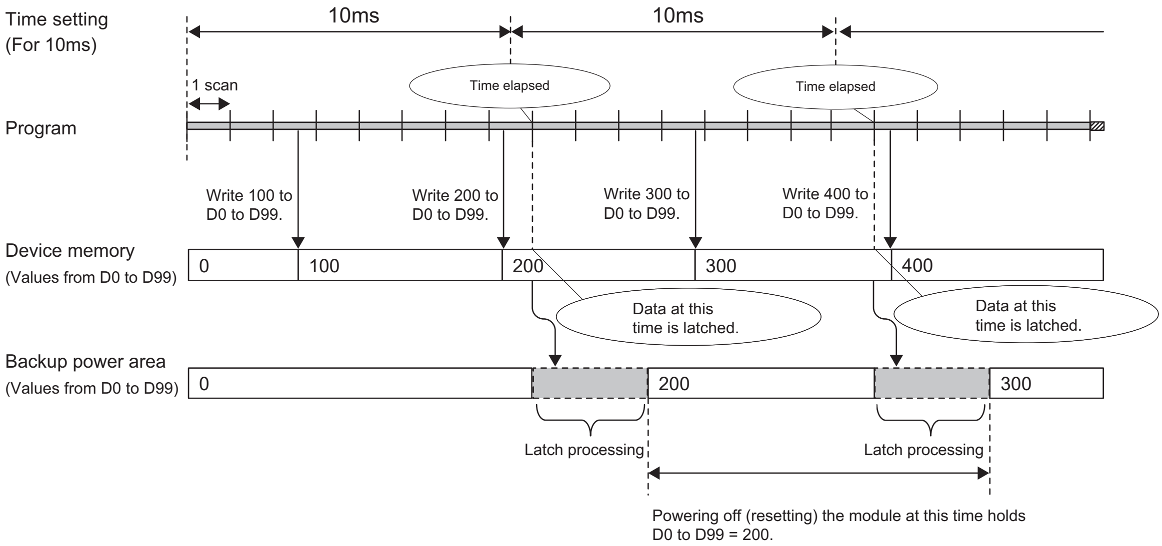
(1) D30848 or later areas of the data register (D) and areas of the devices (W, SW, and L) in disabled ranges can be latched in real time.

■Timing of the latch processing

The timing of the latch processing is determined based on the effective range of the latch interval setting and the operation setting for the specified latch interval (☞ Page 439 Setting the latch interval)

- When set to "Time Setting"

Latch processing is started in the END processing executed after the set time.

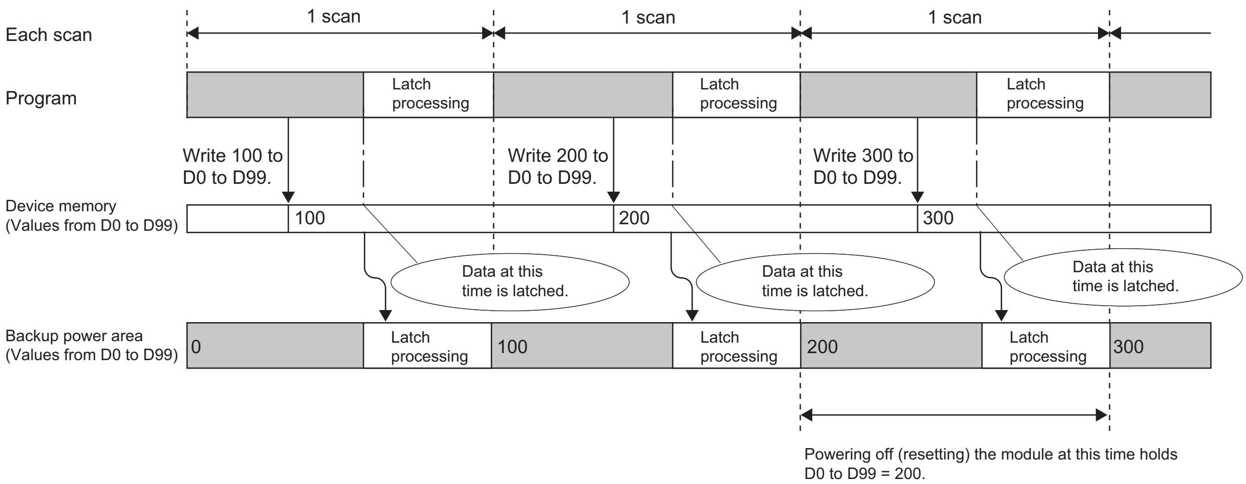


Point

- The more the latch points, the more it takes to complete the latch processing. If the set time elapses while latch processing is still executing, the next one is executed in the END processing that follows the completion of the previous latch processing. Set a latch interval longer than the latch processing time so that the specified time does not elapse during latch processing.
- If the latch interval is shorter than the scan time, latch timing occurs more than once within one scan. However, latch processing is executed once during the END processing.

- When set to "Per Scan"

Latch processing is executed in the END processing of every scan. This setting allows to always retain the device of the previous scan because the devices are latched at every scan. However, it may increase the scan time.



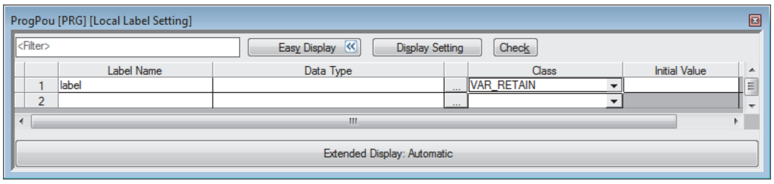
Setting latch on labels

This section describes latch setting on labels.

Operating procedure

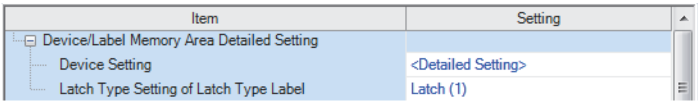
1. In the label edit window, specify "RETAIN" for label attribute.

Label edit window



2. There are two types of latch for labels: latch (1) and latch (2). Select one. The selected latch type is applied to labels of all latch attributes.

"Device/Label Memory Area Detailed Setting" window



[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Detailed Setting] ⇨ [Latch Type Setting of Latch Type Label]

Clearing latch range data

Latch range data is cleared by either of the operations below. (☞ Page 111 Memory Operation)

- Latch clear: Performed from the engineering tool. (📖 GX Works3 Operating Manual)
- Latch clear by program: Execute the RST instruction for latched devices, or clear by transferring K0 by using the MOV or FMOV instruction.

Precautions

This section describes the precautions when using the latch function.

- When using a CPU module other than the R00CPU, R01CPU, and R02CPU, the data in a device/label within the latch range is retained by using the battery on the CPU module unit. Therefore, data cannot be retained during power failure if this battery runs out. Use the battery-less option cassette for retaining the data without a battery during power failure. (☞ Page 443 Latch with Battery-less Option Cassette)
- If the latch range and number of device points are modified by using parameters, latch is performed on the modified latch range. However, if the value of the parameter setting the latch range is different between the previous and current operations because the CPU module was powered off and on or is reset, and the latch range has been increased, the device range in the increased portion is not latched.

24.2 Latch with Battery-less Option Cassette

RnCPU RnENCPU ~~RnPCPU (Process)~~ ~~RnPCPU (Redundant)~~ ~~RnPSFCPU (Standard)~~ ~~RnPSFCPU (Safety)~~ ~~RnSFCPU (Standard)~~ ~~RnSFCPU (Safety)~~

- This function cannot be used in the R00CPU, R01CPU, and R02CPU.

Programmable controller CPUs retain the data in each device/label with latch setting without a battery during power failure by inserting a battery-less option cassette to the CPU module.

The applicable devices and labels and the latch setting are the same as the ones for the latch with the battery. (☞ Page 436 Latch with Battery)

Point

The R00CPU, R01CPU, and R02CPU do not need a battery-less option cassette to retain the data in each device/label with latch setting without a battery during power failure.

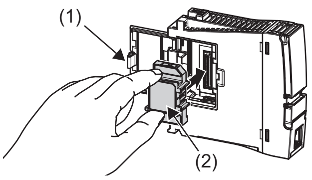
Usage procedure

This section describes how to use the latch with the battery-less option cassette.

Setup procedure

Insert/remove a battery-less option cassette / a battery while the programmable controller is powered off.

1. Insert a battery-less option cassette to the CPU module.



Open the cassette cover (1) located on the side of the CPU module.
Hold the top and bottom of the protruded part (2) of the battery-less option cassette (with the notched edge facing to the right). Insert the cassette straight to the connector, taking care not to tilt it. Check the connection for looseness.
Close the cassette cover (1).

2. Remove the battery from the CPU module. When disconnecting the connector, pull out the connector part so as not to damage the battery cable.

3. Mount the CPU module on the base unit.

4. Power on the programmable controller.

5. Monitor SM624 (Battery-less option cassette insertion flag) and check that it is on.

Point

Insertion of the battery-less option cassette can be checked from the following items in the engineering tool.
- "Product Information List" in "System Monitor"
- "Module Information List" in "Module Diagnostics"

6. Set "Mounted" for "Battery-less Option Cassette Setting" in "CPU Parameter" with the engineering tool.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Setting] ⇨ [Cassette Setting] ⇨ [Battery-less Option Cassette Setting]

7. Set "Battery Error" of "CPU Parameter" as "Not Detected".

[CPU Parameter] ⇨ [RAS Setting] ⇨ [Error Detections Setting] ⇨ [Battery Error]

8. Set "Time Setting" in "Module Parameter".

[Module Parameter] ⇨ [Application Settings] ⇨ [Time Setting]

The clock data is initialized when the battery-less option cassette is initialized. Clock data cannot be retained during power failure when no battery is mounted. Therefore, set "Time Setting (SNTP Client)". (MELSEC iQ-R Ethernet User's Manual (Application))

9. Write the set parameter to the CPU module, and then power off and on, or reset the CPU module.

10. When the uninitialized error is displayed, use the engineering tool to initialize the battery-less option cassette. After the initialization, power off and on, or reset the CPU module.

[Online] ⇨ [CPU Memory Operation]
Select "Device/Label Memory" on the "Memory Management" window, and click the [Initialization] button.

Point

If the battery-less option cassette is initialized during execution of the latch with the battery-less option cassette, only the files in the file storage area in the device/label memory are deleted.

11. Monitor SM625 (Battery-less function execution flag) and check that it is on.

Clearing procedure

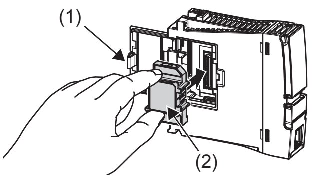
Remove/insert a battery-less option cassette / a battery while the programmable controller is powered off.

1. Read the data on the device/label memory from the CPU module, and save it in advance using the engineering tool in advance. (When the battery-less option cassette is removed, all of the data on the device/label memory are erased.)

2. Power off the programmable controller.

3. Remove the CPU module from the base unit.

4. Remove the battery-less option cassette from the CPU module.



Open the cassette cover (1) located on the side of the CPU module.
Hold the top and bottom of protruded part (2) of the battery-less option cassette and pull it straight, taking care not to tilt it.
Close the cassette cover (1).

5. Install a battery to the CPU module.

6. Mount the CPU module on the base unit.

7. Power on the programmable controller.

8. Set "Not Mounted" for "Battery-less Option Cassette Setting" in "CPU Parameter" with the engineering tool.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Setting] ⇨ [Cassette Setting] ⇨ [Battery-less Option Cassette Setting]

9. Set "Battery Error" of "CPU Parameter" to "Detect".

[CPU Parameter] ⇨ [RAS Setting] ⇨ [Error Detections Setting] ⇨ [Battery Error]

10. Write the CPU parameter to the CPU module, and then power off and on, or reset the CPU module.

Precautions

This section describes precautions on using the latch with the battery-less option cassette.

- When a battery-less option cassette is inserted, the current consumption of the CPU module increases by 0.15A at maximum.
- Insert or remove a battery-less option cassette while the programmable controller is powered off. If it is inserted or removed while the programmable controller is powered on, a stop error occurs in the CPU module and data may not be retained.
- Clock data cannot be retained during power failure when no battery is mounted. Therefore, it is recommended to set "Time Setting (SNTP Client)" in the module parameter. (MELSEC iQ-R Ethernet User's Manual (Application))
- The scan time during the execution of the latch with a battery-less option cassette is longer than the scan time of when the latch with a battery-less option cassette is not executed. As for the instructions specifying the data retained during power failure such as the file register for the arguments, the instruction processing time will be longer. (Page 1040 Processing Time, MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))
- When the latch with a battery-less option cassette is canceled, the data retained during power failure by using the battery-less option cassette with latch setting is not taken over.
- The extended SRAM cassette and battery-less option cassette cannot be used together.

25 DEVICE/LABEL INITIAL VALUE SETTINGS

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When using a Process CPU, refer to the following as well.

☞ Page 364 LABEL INITIALIZATION FUNCTION

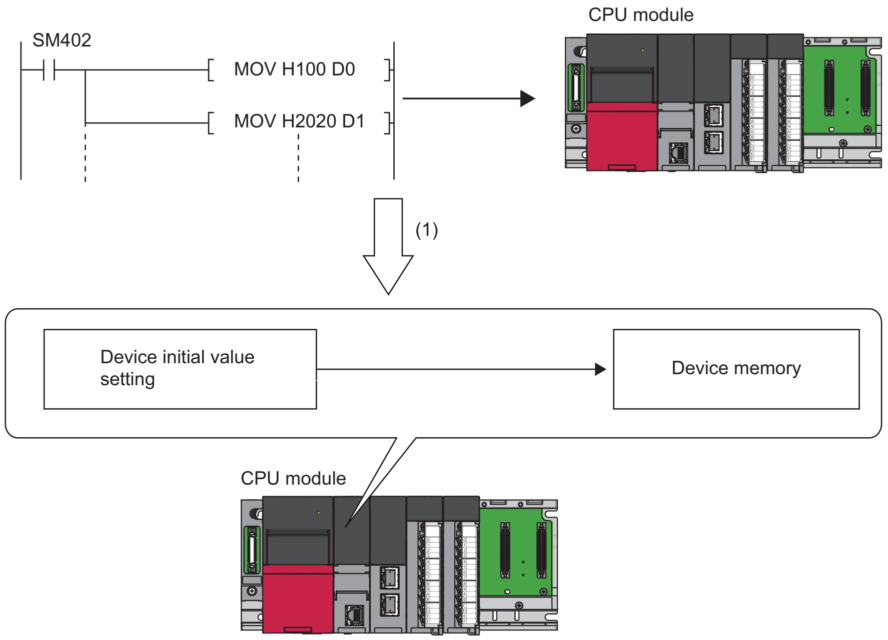
- When using a SIL2 Process CPU, refer to the following as well.

☞ Page 704 FUNCTIONS

- When using a Safety CPU, refer to the following as well.

☞ Page 621 FUNCTIONS

This function sets the initial values of devices and labels used in the program directly (not via the program) to the devices, labels, and buffer memory areas of intelligent function modules.



(1) If initial device values are used, a program to set data to the devices becomes unnecessary.

25.1 Setting Initial Device/Label Values

This section describes the settings required to use initial device/label values.

Setting initial device values

This section describes the settings of initial device values.

Setting procedure

The procedure for using initial device values is as follows.

1. First, the user must create an initial device value file. To set initial values to a local device, create an initial device value file with the same name as the program which sets these values, and specify the range of the values. To set initial values to a global device, create an initial device value file (with any name) which sets these initial values, and specify the range of the values.

2. On the device memory, set up initial device value data within the range specified in the initial device value file.

(GX Works3 Operating Manual)

3. In the "Device Memory Register Diversion", select the device memory which was set up in Step 2. Setting "Device Memory Register Diversion" enables data set up on the device memory to be used as initial device values for the device which is specified in the initial device value file.

(GX Works3 Operating Manual)

4. Configure CPU parameters. (Page 447 Initial value setting)

5. Write the set initial device value file and the CPU parameters to the CPU module.

(GX Works3 Operating Manual)

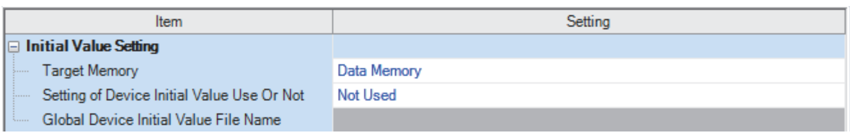
6. The data in the specified initial device value file is automatically set to the specified device or buffer memory of the intelligent function module when the CPU module is powered off and on, reset, or the status changes from STOP to RUN.

Initial value setting

Configure the initial value setting.

[CPU Parameter] ⇨ [File Setting] ⇨ [Initial Value Setting]

Window



Displayed items

Item	Description	Setting range	Default
Target Memory	Sets the storage memory for the initial device value file, initial global label value file, and initial local label value file.	• Data Memory • Memory Card*2	Data Memory
Setting of Device Initial Value Use Or Not	Selects whether or not to use initial device values.	• Not Used • Use	Not Used
Global Device Initial Value File Name	Sets the name of the initial global device value file.*1	60 characters or less	—

*1 If nothing is specified, initial global device values are not applied.

*2 It cannot be set in the R00CPU.

Applicable range of initial device value files

The applicable range of initial device value files is as follows.

Target device	Applicable range
Global device	Initial device values set up in the initial value file of the global device are used.
Buffer memory	
Local device	Initial device values set up in the initial value file of the local device (Program Name.DID) are used.
File register	• When "Use File Register of Each Program" is set, initial device values set up in the initial value file of the local device (Program Name.DID) are used. If there is a program which doesn't have its file register (a file register with the same name as the program), it is not assumed as an error but the initial device values are not used. • When a common file register is used across all programs, initial device values set up in the initial value file of the global device are used.

Number of initial device value settings and maximum range of one range

Up to 1000 ranges can be set in one initial device value file. Up to 8000 data points can be set in one range.

Setting initial label values

This section describes the settings of initial label values. For labels assigned to a device, the initial value setting follows the initial device value.

Setting procedure

The procedure for using initial label values is as follows.

1. Set up initial label values using a global label and local label.

2. Configure CPU parameters (☞ Page 447 Initial value setting).

3. The CPU parameters, programs, initial global label value file, and initial local label value file are written to the CPU module.

(📖 GX Works3 Operating Manual)

4. The data in the specified initial global label value file and initial local label value file is automatically set to the specified label when the CPU module is powered off and on, reset, or the status changes from STOP to RUN.

25.2 Applicable Devices/Labels

For details on devices/labels to which initial device/label values can be set, refer to the following.

GX Works3 Operating Manual

25.3 Precautions

This section describes the precautions when using the initial device/label value setting.

- When initial device values or initial label values are overlapped with the latch range, these initial values take precedence over the latch range. Therefore, even if a device or label is already latched, its values are changed to the specified initial values.
- The initial device value and initial label value are also set when the CPU module status changes from STOP to RUN. *1 Therefore, the initial device value and initial label value cannot be used for areas that are not desirable to be set when the CPU module status changes from STOP to RUN (because they should be set when the power is turned off and on, and changed by the program). By using the MOV instruction, for example, create a program to set an initial value to the specified device or label. In addition, if it is a module access device, make sure to write the initial value in the buffer memory by using the TO instruction.

*1 For details on the initial label value setting for the Process CPU and SIL2 Process CPU, refer to the following.
Page 364 LABEL INITIALIZATION FUNCTION

- To make the CPU module hold a device only when its values fall into the range of device initial setting (such as module access device and link direct device), the start-up time between the device and the CPU module must be synchronized using the CPU parameter, "Module Synchronous Setting".

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Module Synchronous Setting]

Window

Item	Setting
Module Synchronous Setting	
Module Rising	Synchronize

- If "Setting of Device Initial Value Use Or Not" is set to "Use", write the specified initial global device value file to the CPU module. If the specified initial global device value file does not exist, an error occurs. In addition, if the other file (initial local device value file, initial global label value file, or initial local label value file) does not exist or if the global device initial value file name is not specified, an initial value is not applied, but an error does not occur.
- Set "Memory Card" in the target memory, and when using the initial global label value file or initial local label value file, write it simultaneously with the global label setting file, program file, and FB File. If the total capacity of all files is within the capacity of the data memory, specify all files for boot operation. If the total exceeds the capacity of the data memory, remove the initial global label value file or initial local label value file from the boot targets. If only the device initial value file is used, it can be written independently. (Page 165 BOOT OPERATION)

26 CONSTANTS

This section describes constants.

26.1 Decimal Constant (K)

Use this type of constants when specifying decimal data in a program. Specify the decimal constant using K□ character (e.g. K1234). The specification range depends on the argument data type of the instruction using the decimal constant as shown in the following table:

Argument data type of the instruction		Specification range for decimal constant
Data size	Data type name	
16 bit	Word (signed)	K-32768 to K32767
	Word (unsigned)/bit string (16 bit)	K0 to K65535
	16-bit data	K-32768 to K65535
32 bit	Double word (signed)	K-2147483648 to K2147483647
	Double word (unsigned)/bit string (32 bit)	K0 to K4294967295
	32-bit data	K-2147483648 to K4294967295

26.2 Hexadecimal Constant (H)

Use this type of constants when specifying hexadecimal data in a program. Specify the hexadecimal constant using H□ character (e.g. H1234). To specify data in BCD, specify each digit of hexadecimal value with numerical characters 0 to 9. The specification range depends on the argument data type of the instruction using the hexadecimal constant. The range is H0 to HFFFF for 16-bit data size, and H0 to HFFFFFFFF for 32-bit data size.

26.3 Real Constant (E)

Use this type of constants when specifying a real number in a program. There are two types of real numbers: single-precision real number and double-precision real number. Specify it using E□ character. (e.g. E1.234).

Setting range for real numbers

The setting ranges are different between the single-precision real number and double-precision real number.

Specification range for single-precision real numbers

The following is the specification range for single-precision real numbers.

$-2^{128} < \text{device} \leq -2^{-126}$, 0, $2^{-126} \leq \text{device} < 2^{128}$

(E-3.40282347+38 to E-1.17549435-38, 0, E1.17549435-38 to E3.40282347+38)

Specification range for double-precision real number

The following is the specification range for single-precision real numbers.

$-2^{1024} < \text{device} \leq -2^{-1022}$, 0, $2^{-1022} \leq \text{device} < 2^{1024}$

(E-1.7976931348623157+308 to E-2.2250738585072014-308, 0, E2.2250738585072014-308 to E1.7976931348623157+308)

Point

For the number of significant digits and input range of real number data input by the engineering tool, refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

Processing when operation is in progress

This section describes the processing when operation is in progress.

When overflow and/or underflow occurs:

If an overflow and/or underflow occurs when the operation is in progress, the following process is performed:

- When overflow occurs: An error is returned.
- When underflow occurs: 0 is returned (no error occurs).

When a special value is input:

When a special value*1 is used for the input data operation, an error is returned. Note that when "-0" is generated during the operation is in progress, it is handled as "+0", and therefore "-0" is not returned as an operation result.

*1 The special values include -0, denormalized number, nonnumeric number, and ±∞.

26.4 Character String Constant

The character string can be specified by enclosing it with single quotation marks (' ') or double quotation marks (" "). (Example: "ABCDE") Note that the NULL character (00H) becomes the termination character.

Point

For details on character string data, refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

26.5 Notation of Constants

This section describes the notation of constants.

Type		Notation	Example	Applicable data type
Boolean value		Set "FALSE" or "TRUE".	TRUE, FALSE	Bit
		Add "K" or "H" before "0" or "1".	K0, K1, H0, H1	
Integral	Binary	Add "2#" before a binary number.	2#0010, 2#01101010, 2#1111_1111	• Word [unsigned]/bit string [16 bits] • Double word [unsigned]/bit string [32 bits] • Word [signed] • Double word [signed]
	Octal	Add "8#" before an octal number.	8#0, 8#337, 8#1_1	
	Decimal	Directly enter a signed or unsigned decimal number.	123, +123, -123, 12_3	
		Add "K" before a signed or unsigned decimal number.	K123, K-123	
	Hexadecimal	Add "16#" before a hexadecimal number.	16#FF, 16#1_1	
		Add "H" before a hexadecimal number.	HFF, HEAD	
Real number	Decimal notation	Directly enter a signed or unsigned real decimal number with a decimal point.	2.34, +2.34, -2.34, 3.14_15	• Single-precision real number • Double-precision real number
		Add "E" before a signed or unsigned real decimal number with a decimal point.	E2.34, E-2.34	
	Exponential notation	Add "E" and a signed or unsigned exponent (decimal) at the end of the real number.	1.0E6, 1.0E-6	
		Add a signed exponent (decimal) at the end of the real number with "E".	E1.001+5, E1.001-6	
String	String	Enclose a string in single quotation marks (' ').	'ABC'	String
	String [Unicode]	Enclose a string in double quotation marks (" ").	"ABC"	String [Unicode]
Time		Add "T#" or "TIME#" at the beginning. (☞ Page 453 Notation of time)	T#1h, T#1d2h3m4s5ms, TIME#1h	Time

Point

In the notation of binary, octal, decimal, hexadecimal, and real numbers, the numbers can be separated using an underscore (_) to make programs easy to see. For example, the double word [unsigned] in the notation of binary is as follows.

2#1100_1100_1100_1100

The separations by underscores (_) are ignored in program processing.

Notation of time

In the notation of time, add "T#" or "TIME#" at the beginning of the value specified in units of time; d (day), h (hour), m (minute), s (second), and ms (millisecond). The following table lists the effective range for each unit of time.

Item	Effective range
d (day)	0 to 24
h (hour)	0 to 23
m (minute)	0 to 59
s (second)	0 to 59
ms (millisecond)	0 to 999

- Each unit of time can be omitted sequentially from the beginning to the end or in reverse order.
  T#31m23s, T#31m0s648ms
- A sign can be added only before the first unit of time.
  T#-31m23s
- An unsigned real decimal number with a decimal point can be used for the last unit of time. The values after the decimal point of ms (millisecond) are rounded down.
  T#-24d20h31m23s648.123 ms (The time is treated as T#-24d20h31m23s648 ms.)
  T#1.2345 ms (The time is treated as T#1 ms.)
- Only for the first unit of time other than d, the following values within the effective range can be input.
  h: 0 to 596
  m: 0 to 35791
  s: 0 to 2147483
  ms: 0 to 2147483647

Special character (escape sequence)

"$" is used as an escape sequence in a string. The following characters can be input as escape sequences with "$".

Symbol used in a string or printer code	Escape sequence
$	$$
'	$'
"	$"
Line feed	$L or $l
Line break	$N or $n
Page feed	$P or $p
Return	$R or $r
Tab	$T or $t
Two hexadecimal digits and characters that correspond to the ASCII code	Two hexadecimal digits following $

If the two hexadecimal digits following "$" do not correspond to the ASCII code, a conversion error occurs.

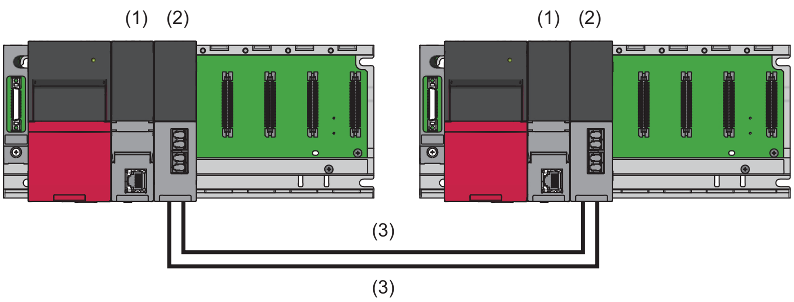
MEMO

PART 4 WHEN USING THE PROCESS CPU (REDUNDANT MODE)

This part consists of the following chapters. Please read these chapters when building the redundant system using the Process CPU (redundant mode). Since information same as that of the Process CPU (process mode) is not described in these chapters, refer to Part 1 to Part 3.

27 BASIC CONCEPT

This system consists of two basic systems that have a CPU module, a power supply module, a network module, or other modules for each of them. Even if an error occurs in one system, control is continued with the other system.
A redundant configuration of the systems of main base units is available when redundant function modules are used and Process CPUs are operated in the redundant mode. To build a redundant system, build the same system with modules on two main base units, and connect the redundant function modules of each system by using tracking cables. Connect the redundant function modules with two tracking cables to configure a redundant system of tracking cables.



(1) Process CPU (redundant mode)
(2) Redundant function module
(3) Tracking cable

Point

In a redundant system, match the both system configurations. When using functions added by the upgrade, use a CPU module with a firmware version that supports the functions for both systems. (☞ Page 1139 Added and Enhanced Functions)

27.1 System

System A and system B

In a redundant system, one system is specified as a system A and the other is specified as a system B to distinguish between two systems connected with tracking cables. Set the system A or B with the engineering tool. (☞ Page 488 System A/B Settings)

Point

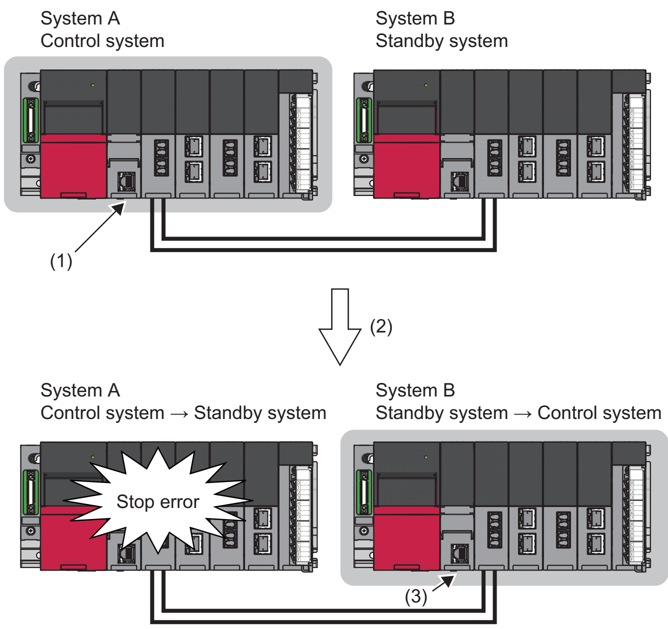
When one system is set to system A, the other system is automatically set to system B, and vice versa.

Control system and standby system

In a redundant system, the CPU module in one system executes programs to perform controls. The other system is in the standby state and does not perform controls. The system that performs controls is called a control system and the system in the standby state is called a standby system.
A control system or standby system is determined when both the systems are started and ready for tracking communications. (☞ Page 460 Determination of Control System/Standby System)

27.2 System Switching Between the Control System and Standby System

In a redundant system, data link is performed between the redundant function modules connected with tracking cables and data required for operation is transferred (tracking transfer) at every scan from the control system to the standby system. If an error occurs in the control system, the standby system will function as the new control system and continue the control using the data that the system has received.



(1) The control system is operating without errors. (Data is transferred to the standby system from the control system at every scan.)
(2) A stop error occurs on the control system.
(3) The standby system now functions as the new control system and continues the control using the data that the system has received.

For details on the system switching, refer to the following.
☞ Page 499 System Switching

27.3 System Consistency Check

In a redundant system, whether both systems have the same configuration is checked to switch the system and continue the operation without causing a system failure. (☞ Page 539 System Consistency Check)

27.4 Operation Mode of the Process CPU

A Process CPU operates in one of the following two modes.

Process mode

Use this mode when a redundant system is not built. When "Mode" has been set to "Process" at project creation with the engineering tool, the Process CPU is started up in the process mode.

Redundant mode

Use this mode when a redundant system is built. When "Mode" has been set to "Redundant" at project creation with the engineering tool, the Process CPU is started up in the redundant mode. (☞ Page 487 Creating a Project)

Point

Set the redundant mode for both systems in a redundant system.

27.5 Operation Modes of the Redundant System

A redundant system operates in one of the following two modes.

Operation mode	Description
Backup mode	A mode used to normally operate the redundant system. When an error or failure has occurred in the control system, the standby system is switched to the control system to continue the operation. Even if a system failure has occurred in the control system, the standby system of the redundant system continues the operation because data is transferred from the control system to the standby system at every scan.
Separate mode	A mode used to maintain the redundant system without stopping controls. In separate mode, the control system CPU module and the standby system CPU module can execute different programs. The programs and CPU parameters of the standby system can be modified*1 and the operation of the program can be checked with the data tracked from the control system while the system is being controlled in the control system.

*1 When the program or CPU parameters are modified while labels (except those with a device assigned) are being used, perform the following operations in the backup mode.
When the program is modified: Execute online change for both systems.
When the CPU parameters are modified: Write data to the programmable controller in the STOP state.
When the program or CPU parameters are modified only in the standby system, tracking transfer of labels is stopped. (☞ Page 529 When data is different between the control system and the standby system)

A redundant system is started in the backup mode. To switch the operation mode, use the engineering tool. (☞ Page 496 Operation Mode Change)

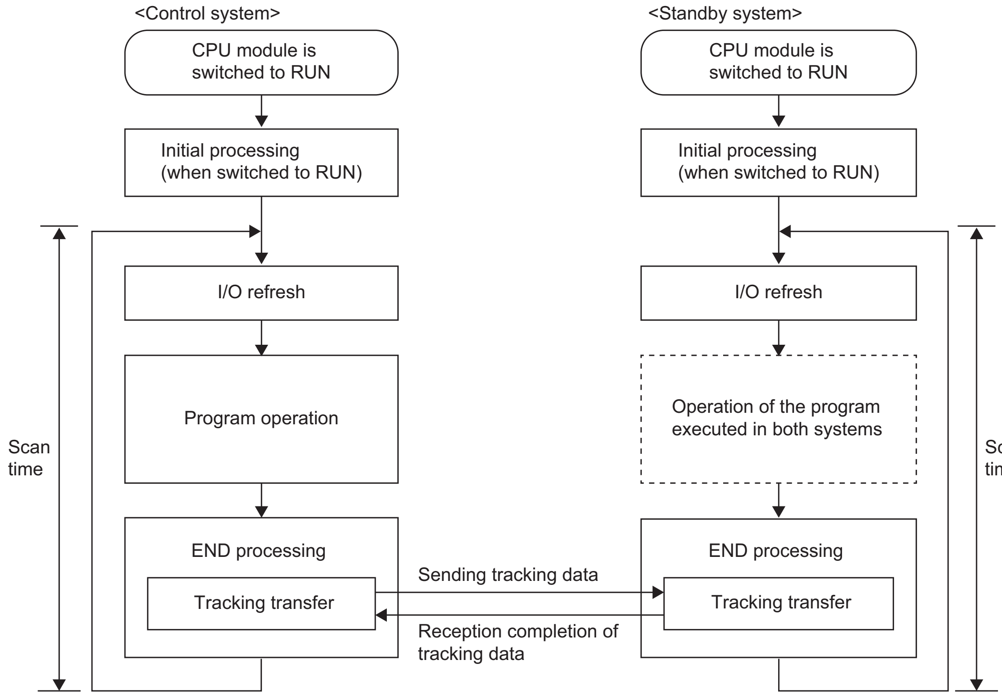
The following table describes differences between the backup mode and the separate mode.

Item	Backup mode	Separate mode
Program operation	The control system executes programs. The standby system CPU module executes programs according to "Both Systems Program Executions Setting" in "Program Setting" of "CPU Parameter". With the default setting, the standby system does not execute programs. To switch the mode to the backup mode, refer to the following. ☞ Page 497 Mode switching to the backup mode	Both of the control system and the standby system execute programs. To switch the mode to the separate mode, refer to the following. ☞ Page 496 Mode switching to the separate mode
System switching	Automatic system switching and manual system switching are supported.	Only manual system switching is supported.
Tracking transfer	Data is transferred according to "Tracking Setting" of "CPU Parameter".	Data is transferred according to "Tracking Setting" of "CPU Parameter". However, the following data causes tracking transfer to stop. • Local devices, local labels • Special relay, special register • PID control instruction information • Signal flow memory
Memory copy from control system to standby system	The following copy operations can be performed. • Automatic memory copy • Memory copy with the engineering tool • Memory copy with the special relay and special register	The following copy operations can be performed. • Memory copy with the engineering tool • Memory copy with the special relay and special register
System consistency check	The system consistency check is performed.	The system consistency check is not performed.
Program execution type	■When the operation mode is changed to the backup mode The control system and the standby system take over their execution types before the operation mode change.	■When the operation mode is changed to the separate mode The control system and the standby system operate differently. • The control system takes over its execution type before the operation mode change. • The standby system operates in the execution type specified by the CPU parameter.
Interrupt program execution enable/disable status	■When the operation mode is changed to the backup mode The control system and the standby system take over their execution enable status before the operation mode change.	■When the operation mode is changed to the separate mode The control system and the standby system operate differently. • The control system takes over its execution enable status before the operation mode change. • The standby system disables execution of interrupt programs.
I/O refresh	Whether or not to refresh the output (Y) of the standby system depends on "Standby System Output Setting" of "CPU Parameter". With the default setting, the output (Y) of the standby system is not refreshed.	Even though "Standby System Output Setting" has been set to "Disable" in "CPU Parameter", the output (Y) of the standby system is enabled.
Link refresh	Data is not transferred from devices of the CPU module in the standby system to link devices.	Data is transferred from devices of the CPU module in the standby system to only the link special relay (SB) and link special register (SW). (Except for CC-Link)

Item	Backup mode	Separate mode
Writing data to the programmable controller	Data can be written to both systems. Or, data can be written to only a system specified in the connection destination setting.	Data is written to only a system specified in the connection destination setting.
Remote operation function	Remote operations can be performed on both systems.	Remote operation with the engineering tool can be performed only on a system specified in the connection destination setting.
Time synchronization	Time of the standby system is synchronized with that of the control system.	Time of both systems is not synchronized.

27.6 Scan Configuration

This section describes scan configurations of the CPU modules in a redundant system.
In a redundant system, tracking is performed during the END processing. (☞ Page 513 Tracking Transfer)
However, in the redundant extension base unit configuration, tracking is performed before the program operation. (The order of processing is reversed.)
For details on other than tracking transfer, refer to Chapter 1. (☞ Page 40 Scan Configuration)
The following figure shows scan configurations of when both systems are started up simultaneously in the backup mode.



Both systems perform different processing depending on the set operation mode.
○: Performed, ×: Not performed

Processing	Backup mode		Separate mode	
	Control system	Standby system	Control system	Standby system
Initial processing/initial processing (when switched to RUN)	○	○	○*3	○*3
I/O refresh	○	○*1	○	○
Program operation	○	×*2	○	○
Tracking transfer	○	○	○	○
END processing	○	○	○	○

*1 Only the input refresh is performed. Whether or not to perform the output refresh depends on "Standby System Output Setting" of "CPU Parameter". (☞ Page 551 Redundant System Operation Setting)
*2 The program operation is not performed with the default setting. Programs are executed according to "Both Systems Program Executions Setting" in "Program Setting" of "CPU Parameter". (☞ Page 544 Program Execution in Both Systems)
*3 This processing will not be performed when the operation mode is changed from the backup mode to the separate mode.

27.7 Operation Processing at Momentary Power Failure

When a momentary power failure occurs in the power supply module mounted on the extension base unit in the redundant extension base unit configuration, operation processing for the CPU modules in both systems is interrupted.
However, when the power supply module is redundant, operation processing is not interrupted even if a momentary power failure occurs in either of the systems. But, if a momentary power failure occurs when power is being supplied to the power module of only one of the systems, operation processing for the CPU modules in both systems is interrupted.

27.8 Determination of Control System/Standby System

This section describes how to determine which system is the control system and the other is the standby system.

When starting up both systems simultaneously

The following describes the method of determining the system types of when both systems are started up simultaneously.

How to determine the system types

A control system or standby system is determined when both the systems are powered off and on or the CPU module is reset and then ready for tracking communications.

■When both systems are started up simultaneously

When both systems are started up simultaneously, the system A operates as the control system and the system B as the standby system.
"Simultaneously" here means that one CPU module is started up within three seconds after the other CPU module is started.

■When one of the system A and system B is started up first

- When either system A or system B is started up first in the configuration with the main base unit only, the system will wait for the start-up of the other system in three seconds. (☞ Page 462 The system waiting for the start-up of the other system)
- When the power is turned on or reset one system at a time in the redundant extension base unit configuration, the system that starts up earlier becomes the control system, and the system that starts up later becomes the standby system. The CPU parameter setting of waiting for the start-up of the other system (control/standby system start-up setting) is invalid.

Check method

Check the LED of each redundant function module to check the status of the control system/standby system.

System status	LED of the redundant function module
Control system	
Standby system	

Point

Users can check the status of the control system/standby system with the engineering tool. (📖 GX Works3 Operating Manual)
- System monitor
- Monitor status bar

Precautions

■If the system A/B settings have not been set for both systems or the same system setting has been set for both systems

If the system A/B settings have not been set for both systems or the same system setting has been set for both systems, a stop error occurs. To determine the control system and standby system, properly set the system A/B settings for both systems.

■If both systems are restarted when a stop error has occurred on the standby system (system A)

If both systems are restarted simultaneously in a system where the system A operates as the standby system due to a stop error and the system B normally operates as the control system, a stop error may occur on both systems.
If the stop error cause of the system A is a program error, a stop error occurs on the system A again after the systems are restarted. Even if the system B normally operates as the control system before the restart, a stop error occurs on the system B if a mismatch between files is detected in the system consistency check after the restart. As a result, a stop error occurs on both systems.
In this case, eliminate the cause of the error that has occurred on the system A and reset and restore both systems.

■When the READY LED of the CPU module in one of the systems is flashing

Do not power off the CPU module in the other system.
The system may start up without checking the system consistency even when the conditions between the both systems do not match. (☞ Page 540 Execution timing)

■When the configuration between the both systems do not match

When the configuration (such as insertion of an SD memory card) between the both systems do not match, it may take time to start communications between the both systems. Check that the both system configurations match in the process of starting up both systems at a time, and then start up a redundant system. (☞ Page 477 Starting up both systems at a time)

■Using an SD memory card

- Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card. A stop error may occur on both systems and the systems may not be started as a redundant system.
- If the CPU module is powered off or reset or an SD memory card is removed during an access to the SD memory card, the data in the SD memory card may be corrupted. In this case, diagnostics of the SD memory card (such as a file system check or restore processing) is performed when the CPU module is powered off and on or is reset.
- During diagnostics of the SD memory card, tracking communications are disabled. If diagnostics of the SD memory card on the other system are not completed in the timeout period set in "Other system Start-up Timeout Setting", a stop error occurs on the own system. If diagnostics of the SD memory card takes time on the other system, a stop error occurs on the other system as well. In this case, a stop error occurs on both systems and the systems cannot be started as a redundant system. Reset and restore both systems.
- The control/standby system start-up setting is invalid in the redundant extension base unit configuration, so if the own system is started up while the other system is executing SD memory card diagnostics, the own system may start as a control system. Even when both systems are started up simultaneously, if the SD memory card diagnosis of the other system is executed, the own system may start up first as the control system, as if the systems were started up one at a time. In this case, when the SD memory card diagnosis of the other system finishes during initialization of the own system, a stop error is detected in the other system.

Point

Even though diagnostics of the SD memory card takes time on the other system, only the own system can be started. (☞ Page 464 When one system is started automatically even though a tracking communication error has occurred)

When starting up one system first

- In the configuration with the main base unit only, only one of the system A and system B can be started up first by the following starting methods.
- When the power is turned on or reset one system at a time in the redundant extension base unit configuration, the system that starts up earlier becomes the control system, and the system that starts up later becomes the standby system. The CPU parameter setting of waiting for the start-up of the other system (control/standby system start-up setting) is invalid.

How to start one system first

Perform one of the following operations to the CPU module that is waiting for the start-up of the other system. The system of the CPU module will be started as the control system.

■Online operation

Perform the following operation on the engineering tool.

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

Select "Forced Start of Control System while Waiting for Other System to Start" and click the [Execute] button.

■Switch operation

Set the RUN/STOP/RESET switch of the CPU module to the RUN, STOP, and then the RUN position again.

Set "Control System Start-up Setting (Switch Operation)" to "Enable" in the CPU parameter in advance. (☞ Page 551 Redundant System Operation Setting)

■Operation with the input (X)

Turn on the input (X) set with the parameters.

Set "Control System Start-up Setting (Input (X))" to "Enable" in the CPU parameter in advance. (☞ Page 551 Redundant System Operation Setting)

The system waiting for the start-up of the other system

In the configuration with the main base unit only, when the CPU module is started up and tracking communications cannot be established with the other system for three seconds or more, the CPU module will start waiting for the start-up of the other system.

Both of the CTRL LED and SBY LED of the redundant function module turn off because the system of the CPU module is not the control system or standby system yet. The BACKUP LED flashes because a system switching disable cause has been generated.

After tracking communications are established with the other system, the control and standby systems are determined and the system switching disable cause is eliminated.

■Operation while the system is waiting for the start-up of the other system

Even though the RUN/STOP/RESET switch of a CPU module is set to RUN while the system is waiting for the start-up of the other system, the CPU module is in the STOP state because the system has not yet been set to control system or standby system. Thus, the CPU module does not execute programs.

The following table lists the refresh operations to be performed while the system is waiting for the start-up of the other system.

Type	Operation
I/O refresh	Only the input refresh is performed. The output refresh is not performed.
Network module link refresh	Data in SB/SW is transferred and data in devices other than SB/SW is not transferred from the network module to the CPU module, and from the CPU module to the network module.
Intelligent function module refresh	Data is transferred between the intelligent function module and the CPU module.

■Timeout of waiting for the start-up of the other system

Measuring the length of the waiting time for the start-up of the other system is started upon completion of the initial processing.

Set the waiting time for the start-up of the other system in "Other system Start-up Timeout Setting" of "CPU Parameter". (☞ Page 551 Redundant System Operation Setting)

When a timeout occurs, a stop error will occur. To prevent occurrence of a stop error, set "Other system Start-up Timeout Setting" of "CPU Parameter" to "Not Set".

Operation when the control/standby system has not been determined

In the configuration with the main base unit only, the operation is the same as while waiting for the start-up of the other system (each refresh operation while waiting for the start-up of the other system). (☞ Page 462 Operation while the system is waiting for the start-up of the other system)

The following tables describe each refresh operation in the redundant extension base unit configuration.

■Refresh for the modules mounted on the main base unit

Type	Operation
I/O refresh	Only the input refresh is performed. The output refresh is not performed.
Network module link refresh	Data in SB/SW is transferred and data in devices other than SB/SW is not transferred from the network module to the CPU module, and from the CPU module to the network module.
Intelligent function module refresh	Data is transferred between the intelligent function module and the CPU module.

■Refresh for the modules mounted on the extension base unit

Type	Operation
I/O refresh	Neither the input nor output refresh is performed.
Network module link refresh	Data is not transferred between the network module and the CPU module.
Intelligent function module refresh	Data is not transferred between the intelligent function module and the CPU module.

Precautions

- To start up the system that is waiting for the start-up of the other system as the control system, check that the other system is not operating as the control system.
- When tracking communications are established by connecting the other system with a tracking cable or powering on or resetting the CPU module in the other system, the CPU module that has been waiting for the start-up of the other system enters the RUN state and executes programs. Thus, always check that program execution by the CPU module does not cause any problems before powering on the other system or connecting the other system with a tracking cable.
- When the CPU module that has been waiting for the start-up of the other system becomes ready for tracking communications and a stop error has occurred in the other system, a stop error occurs in that CPU module as well. In this case, take a corrective action against the error of the other system, and turn off and on or reset the CPU module in both systems.
- When the system waits for the start-up of the other system, tracking communication cannot be made because the other system is off or either of tracking cables has problems. Check that the other system is on or tracking cables have no problems. (☞ Page 925 When the L ERR LED turns on)
- Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card. (☞ Page 461 Using an SD memory card)
- When starting up the systems one by one in the redundant extension base unit configuration, start up the other system only after the system that was started first has finished starting up. (☞ Page 480 Starting up the systems one by one) Because tracking communication cannot be performed by a system that is starting up, a stop error may occur in the CPU module of the system that was started up later. In this case, restart and restore the system with the CPU module with a stop error. (When automatic recovery is set, automatic restart can be executed without manual operation.) (☞ Page 557 Automatic recovery of the CPU module of the standby system)

When one system is started automatically even though a tracking communication error has occurred

In a redundant system without extension base units[*1], when the other system is powered off[*2] or there is an error with tracking cables at a system start, the CPU module enters the state that waits for the other system to start. The following shows examples, such as a system configuration and a program, to start up either of two systems using external signals without waiting for the other system to start, and prevent both systems from operating as control systems.

*1 In a redundant system with redundant extension base unit, even when the other system is powered off or there is an error with tracking cables at a system start, either of two systems starts up without waiting for the other system to start.

*2 Instead of configuring the system described below, configuring a redundant power supply system is recommended when taking a measure against failure of the power supply module in the other system.

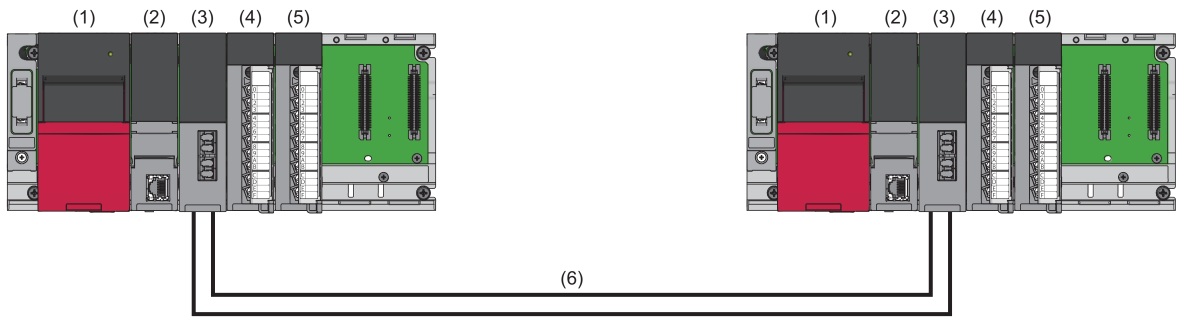
In the redundant extension base unit configuration, the Control System Start-up Setting (Input (X)) is disabled.

Restriction

When using this program example, do not power off and on or reset the CPU module on each system in separate mode. Change the mode to the backup mode before powering off and on or resetting the CPU module.

System configuration

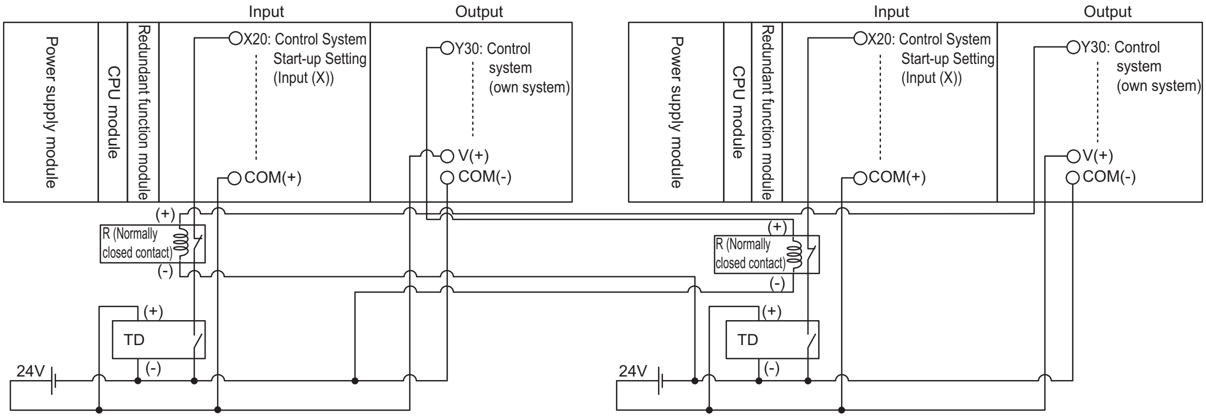
The following figure shows a configuration example of the system.



(1) Power supply module (R62P)
(2) CPU module (RnPCPU)
(3) Redundant function module (R6RFM)
(4) Input module (RX40C7)
(5) Output module (RY40NT5P)
(6) Tracking cable

Wiring example

The following figure shows a wiring example.



- Supply the 24V power using the power supply module (R62P) or the 24V external power supply. When using the R62P, ensure that the power capacity does not exceed the limit. When using the 24V external power supply, use the same power source as the one that supplies power to the power supply module in each system.
- TD is an on delay timer wired externally. Connect the output signal wire of the on delay timer to a relay (normally closed contact). To prevent both systems from starting up simultaneously, configure different timer settings for system A and system B.
- R is a relay (normally closed contact) wired externally. This relay connects the output signal wire of the on delay timer and the output device (Y30: Control system (own system)). The output signal wire of the relay is input to X20.

■I/O signals

The following table lists the details on the I/O signals.

Device No.	Signal name
X20	Control System Start-up Setting (Input (X)) With the timer wired externally, this bit turns on after a certain time. When the output Y of the other system is off (control system (own system)) at that time, the system starts as the control system.
Y30	Control system (own system)

■Setting time of the external on delay timer

For the external timer, with the following equation as a guide, set a longer time than the time until both systems start up so that this function (Automatic start-up at tracking communication error) is not executed when tracking communications are normally performed. In addition, set different times to system A and system B so that the times of both systems are not up simultaneously.

- Time set for the external timer*1 = ❶ Start-up time of the CPU module (Time from power-on until RUN) + ❷ One scan time + ❸ Time lag of power-on + ❹ α + ❺ β

❶ Start-up time of the CPU module: Time from when the CPU module is powered on until when the CPU module enters to the RUN state
❷ One scan time: Time until when Y30 (Control system (own system)) is refreshed
❸ Time lag of power-on: Time to add to the external timer of the system that has started up first when two systems start up one by one. It adjusts the activation timing of the external timer.
❹ α: Margin for variation in start-up time of the CPU module. Set a sufficient margin to accommodate the variation.
❺ β: Time to add to the timer of either of two systems so that the times of both systems are not up simultaneously

*1 If the time set for the external timer is shorter than the time determined by the above equation or an identical time is set to system A and system B, one system cannot recognize whether the other system has started up as the control system and both systems may start up as control systems.

Parameter settings

The following shows parameter settings.

■System parameter

Set the system parameter according to the system configuration in "I/O Assignment Setting".

Slot	Module Name	Module Status Setting	Points	Start XY
Main				
CPU	R32PCPU(Host Station)			3E00
0(0-0)	R6RFM	No Setting	32 Points	0000
1(0-1)	RX40C7	No Setting	16 Points	0020
2(0-2)	RY40NT5P	No Setting	16 Points	0030

■CPU parameter (program settings)

Set this program example (MAIN in this example) in "Program Setting" as follows.

Execute Order	Program Name	Execution Type: Type	Execution Type: Detailed Setting Information	Refresh Group Setting	Device/File Use or not	Both Systems Program Executions Setting
1	MAIN	Scan		(Do not Set)	<Detailed Setting>	Both Systems Executions

- Set "Execution Type" to "Scan".
- Set "Both Systems Program Executions Setting" to "Both Systems Executions".

■CPU parameter (redundant settings)

Set the CPU parameter in "Redundant System Settings" as follows.

Item	Setting	
Redundant Behavior Setting		
Watching Standby System Setting	Enable	
Standby System Output Setting	Disable	
Backup Mode Setting	Check Operating Status Consistency	
Auto Memory Copy Setting	Disable	
Control/Standby System Start-up Setting		
Other system Start-up Timeout Setting	Not Set	(1)
Time-out Period	60 Second	
Control System Start-up Setting(Switch Operation)	Disable	
Control System Start-up Setting(Input(X))	Enable	(2)
Input(X)	X20	(3)
Tracking Setting		
Signal Flow Memory Tracking Setting	Transfer	
Tracking Device/Label Setting	Detailed setting	(4)
Tracking Block No.1 Auto Transfer Setting	Transfer Automatically	
Device/Label Detailed Setting	<Detailed Setting>	

(1) Set "Not Set" in "Other system Start-up Timeout Setting".
(2) Set "Enable" in "Control System Start-up Setting (Input (X))".
(3) Set "X20" in "Input (X)".
(4) Set "Detailed setting" in "Tracking Device/Label Setting".

Set "Device/Label Detailed Setting" in "Global Device Setting" as follows.

Tracking Block No. 1
Total Global Device 36.2K Word
File Register Setting
File Name

No.	Device	Points (Decimal)	Start	End	
1	X	12240	30	2FFF	(5)
2	Y	12224	40	2FFF	(5)

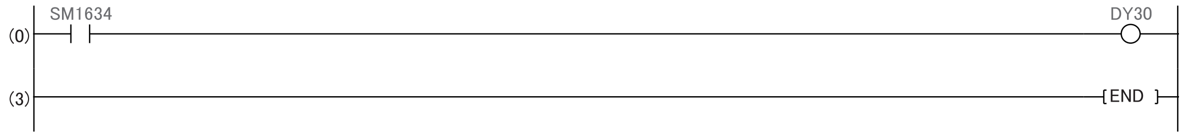
(5) Do not include X20 to X2F and Y30 to Y3F, which are used in the program example, in the tracking transfer range.

Point

- To switch enabling and disabling "Control System Start-up Setting (Input (X))", set a switch for the input to X20 so that "Control System Start-up Setting (Input (X))" is enabled only when the switch is on.
- When using this system, set the RUN/STOP/RESET switch to RUN to operate the ladder program.

Program example

The following shows a program example and the overview of the operation.



■Output of the control system (own system)

(0) The other system is notified of the start-up of the own system as the control system by turning on Y30 (Control system (own system)) using the direct access output when the own system operates as the control system (SM1634 is on), or by turning off Y30 when the own system does not operate as the control system.

Point

After the automatic start-up, to enable the system switching, eliminate the cause of tracking communication error and restart the CPU module in the control system or in the state that waits for the other system to start.

When starting up the previous control system as the control system

In a redundant system, the system A is always specified as the control system when both systems are started up simultaneously.

Even though both systems are temporarily powered off due to a power failure or other causes while the system B is operating as the control system, the system A is started up as the control system when both systems are powered on again.

To start up the system B that was the previous control system as the control system again, create a program that uses SM1636 (Previous control system identification flag).

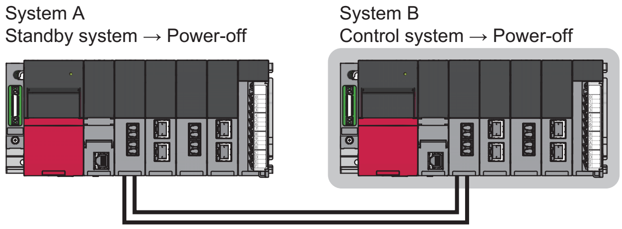
For a system having network modules, wait until the network modules of the other system start up and then execute the SP.CONTSW instruction.

When network modules are not mounted on main base units

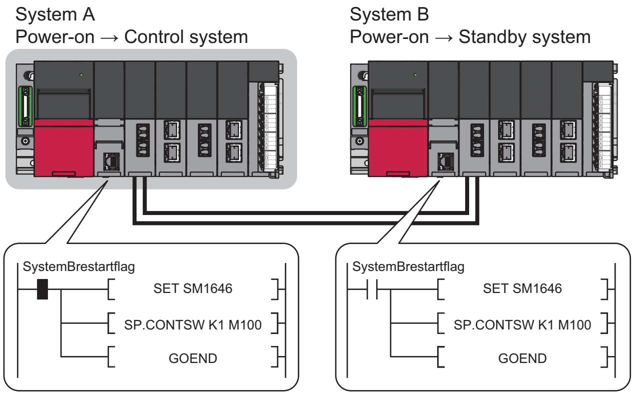
■Program example

Executing a system switching instruction when the CPU module in the system A is set to the RUN state for the first time switches the system B to the control system. Turn on SM1646 (System switching by a user) in advance.

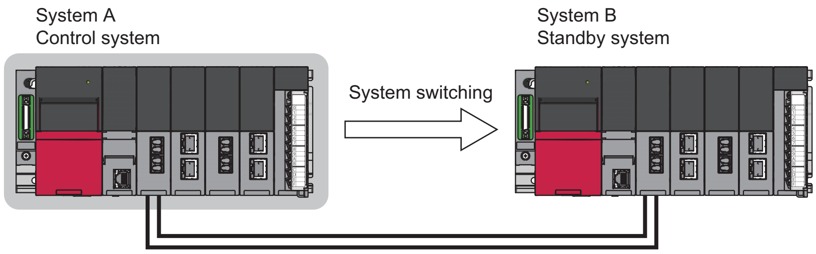
■Operation example



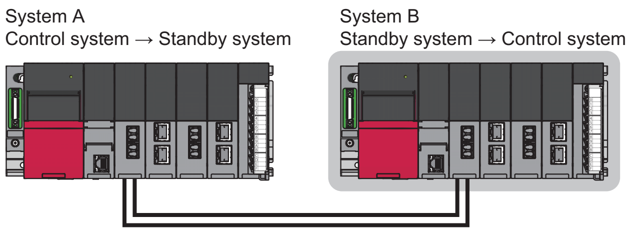
1. Both systems are temporarily powered off due to a power failure or other causes while the system B is operating as the control system.



2. The system A is started as the control system when both systems are powered on. SM1636 turns on for only one scan after the CPU module in the system A is set to the RUN state.



3. The system switching is performed by the SP.CONTSW instruction.



4. The system B is switched from a standby system to a control system.

When network modules are mounted on main base units

Wait until the start-up of the network module in the system B. Then, execute the system switching instruction.

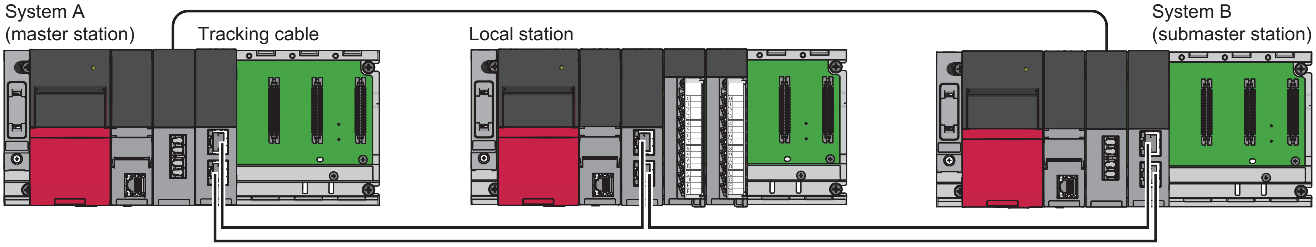
Point

When CC-Link modules are mounted on main base units, the system B cannot be started up as the control system. (The system A with the CC-Link module always starts up as the control system.)

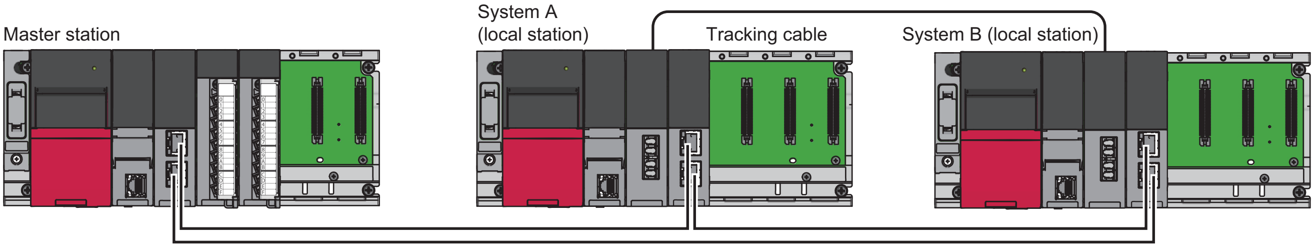
■In a redundant master station configuration or a redundant device station configuration

- System configuration

(Redundant master station)



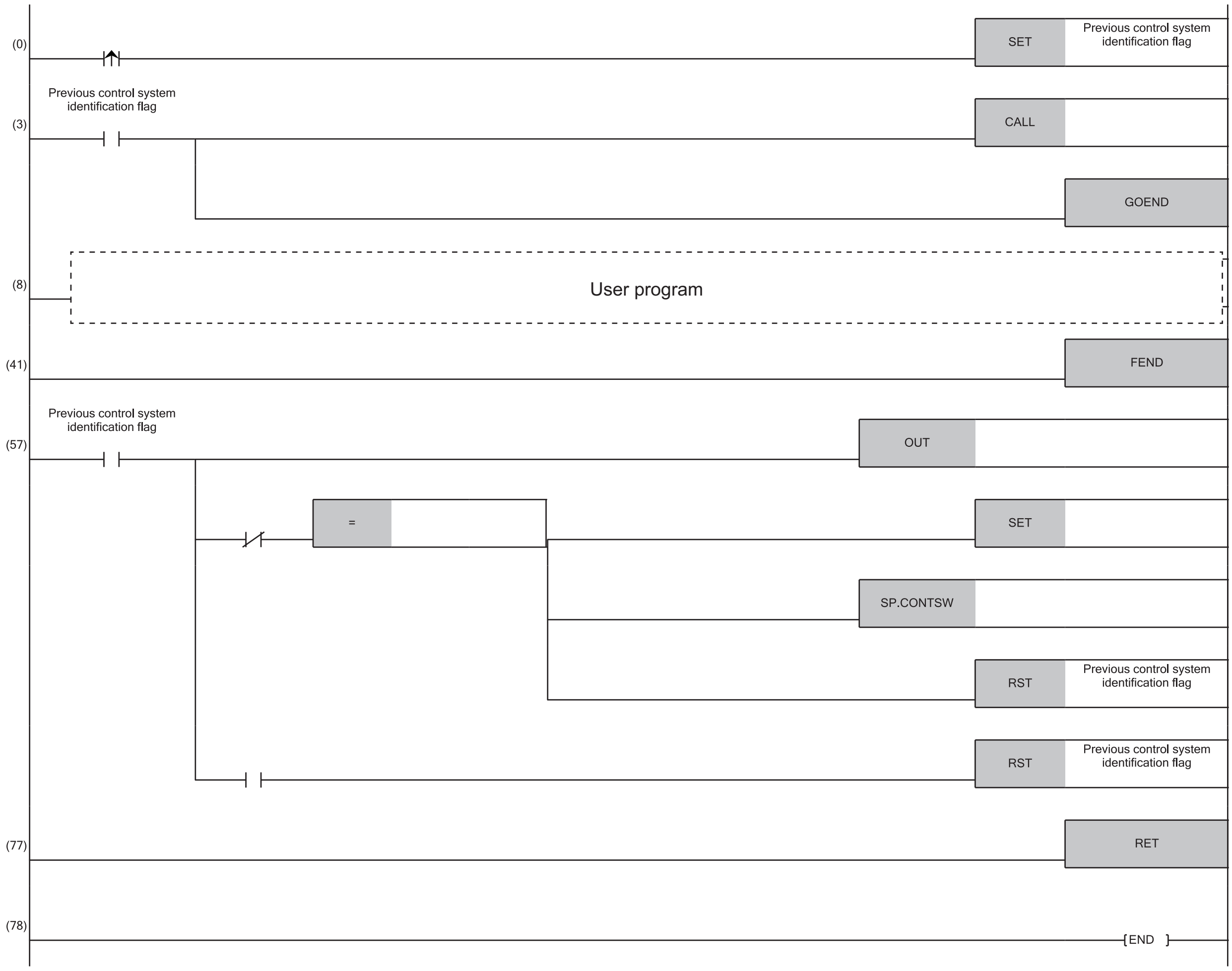
(Redundant device station)



Point

To make the system B start as the control system, connect network modules in a loop. (In a redundant master station configuration or a redundant device station configuration, if the network is configured in the line topology and link-down occurs in the network modules of the system A, link-up between the network modules of the system B cannot be detected. Therefore, the system switching instruction is not executed and the system A starts up as the control system.)

• Program example



(0) Save the ON/OFF status of SM1636 (Previous control system identification flag) to 'System B restart flag' since SM1636 is on during only one scan after the operating status of the CPU module has changed to RUN.

(3) Execute the subroutine program (P100) while 'System B restart flag' is on.
Jump to the END instruction not to execute the user program while 'System B restart flag' is on.

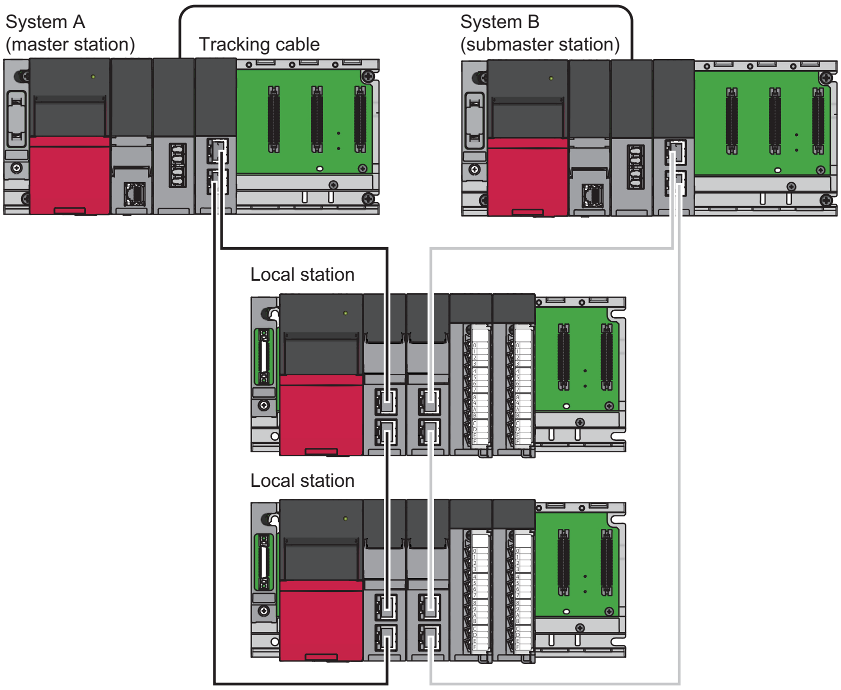
(41) Terminate the main routine program.

(57) Start measuring the timeout time (ten seconds (Timer limit setting: 100ms)) for the startup of previous control system.
Execute the SP.CONTSW instruction when the data link status of the network modules in the system B is normal and no system switching requests are received from them. Check the one from SW1B0 to SW1B7 for the data link status of the network modules in the system B. (Device and bit vary depending on the network modules and the station number in the system B.) Check SD1646 (System switching request status from a network module of the other system) whether system switching requests are received from the network modules in the system B.
Enable the manual system switching after the network module communications in the system B have been established successfully.
Execute the system switching instruction.
Turn off 'System B restart flag'.
Turn off 'System B restart flag' as well when the timeout time (ten seconds) has elapsed, because system switching is not performed and the CPU module proceeds to the next scan to execute the user program while the system A remains as the control system. (The timeout time (ten seconds) needs to be adjusted depending on the system configuration and environment.)

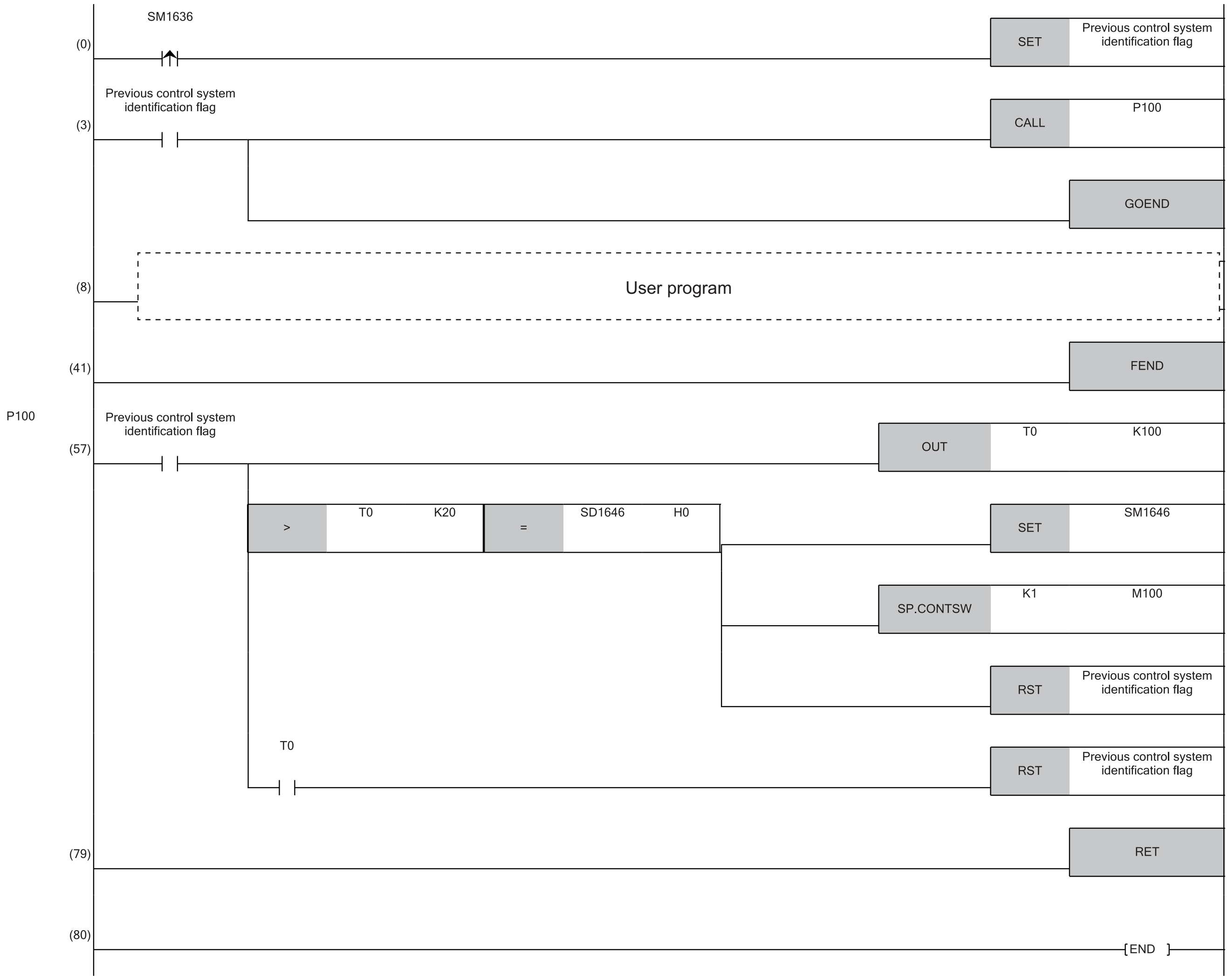
(77) Terminate the subroutine program.

■In a redundant line configuration

- System configuration



• Program example



(0) Save the ON/OFF status of SM1636 (Previous control system identification flag) to 'System B restart flag' since SM1636 is on during only one scan after the operating status of the CPU module has changed to RUN.

(3) Execute the subroutine program (P100) while 'System B restart flag' is on.
Jump to the END instruction not to execute the user program while 'System B restart flag' is on.

(41) Terminate the main routine program.

(57) Start measuring the timeout time (ten seconds (Timer limit setting: 100ms)) for the startup of previous control system.
Execute the SP.CONTSW instruction when the data link status of the network modules in the system B is normal and no system switching requests are received from them. Wait until the information in SD1646 (System switching request status from a network module of the other system) is updated (Two seconds). Check SD1646 for the system switching request status from the network modules in the system B.
Enable the manual system switching after the network module communications in the system B have been established successfully.
Execute the system switching instruction.
Turn off 'System B restart flag'.
Turn off 'System B restart flag' as well when the timeout time (ten seconds) has elapsed, because system switching is not performed and the CPU module proceeds to the next scan to execute the user program while the system A remains as the control system. (The timeout time (ten seconds) needs to be adjusted depending on the system configuration and environment.)

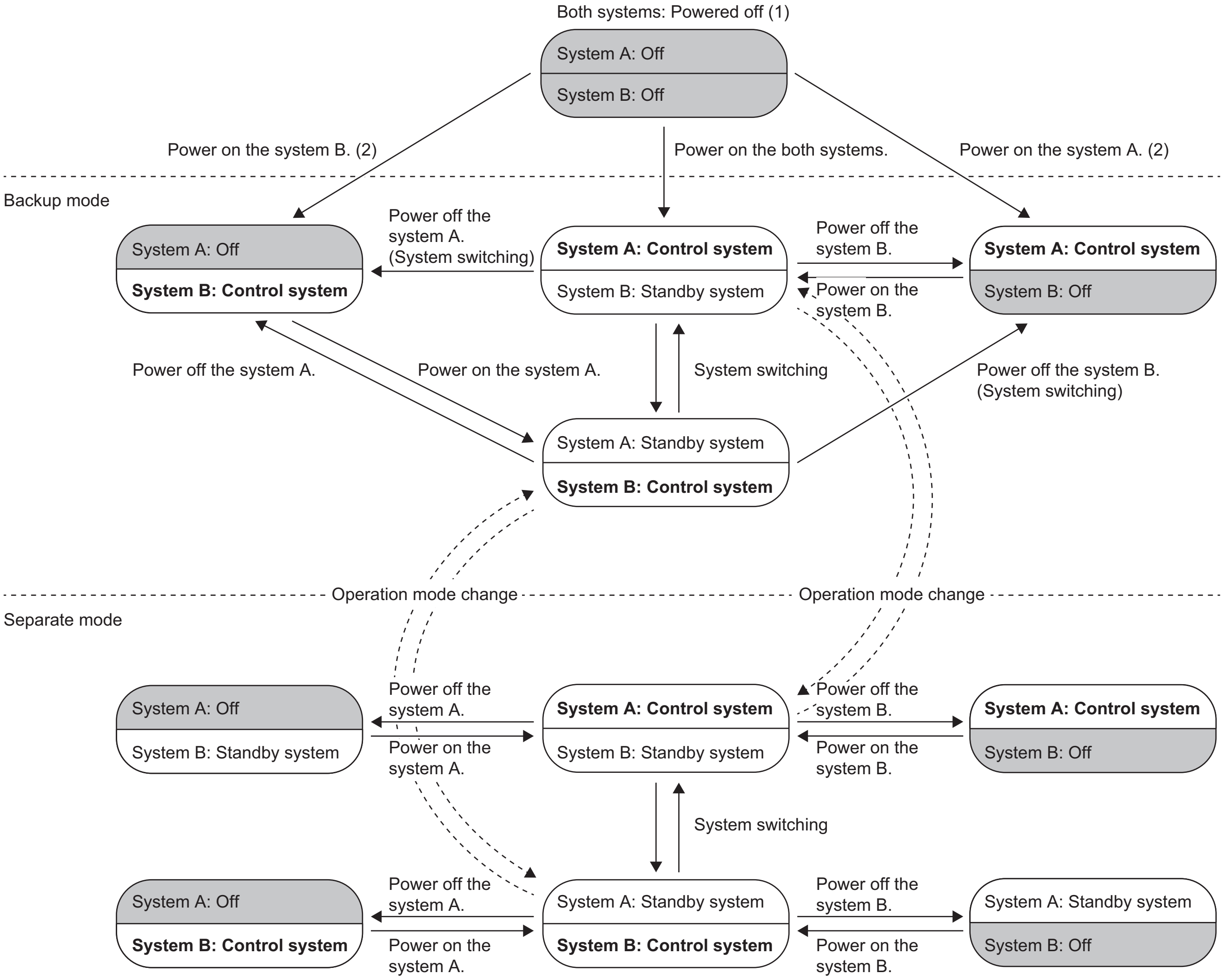
(79) Terminate the subroutine program.

■Precautions

• In the user program, SM402 (After RUN, ON for 1 scan only) and SM403 (After RUN, OFF for 1 scan only) cannot be used because the system switching instruction (SP.CONTSW) takes several scans to complete the processing. Use other devices instead. For example, use an alternative device to SM402 to turn it ON on the rising edge of SM1636 (Previous control system identification flag) and make it turn off after the first execution of the user program. Make an alternative device to SM403 to turn on at the head of the next scan following the falling edge of the alternative device to SM402.

27.9 State Transition of the Redundant System

The following figure shows the state transition due to the operation mode change and system switching for the redundant system after start-up.



(1) When both systems are powered off, both systems enter the power-off state regardless of the operation mode or system type.

(2) When a system that has been powered is started as the control system. (☞ Page 462 When starting up one system first)

27.10 Access in the Redundant Extension Base Unit Configuration

Access to the extension base unit in the redundant extension base unit configuration is limited to the control system. When systems are switched, access from the new control system (old standby system) to the extension base unit becomes enabled.

Connection status when the extension cable is redundant

The redundant extension cables between the extension base units are divided into active routes that have access to the modules and inactive routes that do not have access to the modules. The connection status of the extension cables can be checked in the following ways.

- SD1760 (extension cable connection status) and SD1761 (extension cable route information)
- CONNECT/ACTIVE LED of the extension base unit for the redundant system

Extension cable error between control system main base unit and extension level 1

When an extension cable error occurs between the main base unit of the control system and the extension base unit, a stop error occurs in the CPU module of the control system, and system switching occurs. In this case, communication with the extension base unit is performed by the new control system, but the communication route of extension level 1 and later is not switched. In addition, the new control system detects a continuation error by the stop error of the new standby system.

Extension cable error between standby system main base unit and extension level 1

When an extension cable error occurs between the main base unit of the standby system and the extension base unit, a continuation error is detected in the CPU module of the control system, and a stop error is detected in the CPU module of the standby system.

Extension cable errors between extension base units

■When the extension cable is redundant

When an error occurs in the extension cable on the active side between the extension base units in a redundant system, a stop error occurs in the CPU module of the control system, and system switching occurs. In this case, communication with the extension base unit is performed by the new control system, and the communication route in the section where the extension cable error occurred is switched from inactive to active. In addition, the new control system detects a continuation error by the extension cable error.

■When the extension cable is not redundant

If an extension cable error occurs in the following sections, the extension base unit in the level immediately below the section where the error occurred cannot be accessed.

- Section where only one extension cable is connected
- Section where only one extension cable is recognized as connected properly due to an extension cable error

Therefore, although a stop error occurs in the CPU module of the control system, and system switching occurs, a stop error also occurs in the new control system. In this case, communication with the extension base unit is performed by the new control system.

■When system switching is disabled

When an error occurs in the extension cable on the active side between the extension base units in a redundant system where system switching is disabled due to a stop error in the standby system, the CPU module of the control system detects a stop error, but no system switching is performed, and communication with the extension base unit is also performed by the control system. In addition, the communication route of extension level 1 and later, including the communication route in the section where the extension cable error occurred, is not switched, and the extension base unit in the level immediately below the section where the extension cable error occurred cannot be accessed.

When an extension cable error (inactive side) occurs between extension base units

When the extension cable is redundant and an error occurs in the extension cable on the inactive side between the extension base units in a redundant system, a continuation error occurs in the CPU module of the control system. Unlike having an extension cable error on the active side, system switching or switching of the communication route does not occur.

How to check the extension cable faulty area

When an error occurs in the extension cable, the extension cable information can be checked in the detailed information for the error code.

Precautions

- If no communication can be performed with the extension base unit on the previous level or with the main base unit due to an extension cable error or the input power supply of the power module mounted on the extension base unit turning off, the output of that module on the extension base unit and the extension base unit in the level immediately below will be cleared regardless of the setting. Therefore, configure the system so that clearing the output of the module on the extension base unit does not cause a problem.
- If the standby system cannot be switched due to a stop error, even if a stop error occurs in the control system due to an extension cable error on the active side, system switching and switching of the communication route are not performed, and control cannot continue. Therefore, if a cause of the system switching failure occurs, eliminate the cause immediately.

Access to modules on the extension base unit

This section describes precautions for accessing modules on the extension base unit.

- When an instruction to access the buffer memory of a module on the extension base unit from the standby system is executed by SM1762 (operation setting for access from the standby system to the extension base unit), whether the operation of the instruction is handled as a stop error or as non-processing can be selected.
- Do not set the interrupt condition for modules on the extension base unit. If the interrupt condition is set, a stop error is detected at startup time.
- The monitoring/test function cannot be executed from the standby system by specifying a device to access the module on the extension base unit. If executed, the following actions will be taken.

Function	Device	Operation
Device/buffer memory batch monitor	Un\G□	The engineering tool displays an error message. Execute the monitor function again from the control system, not from the standby system.
	DX	The engineering tool monitor displays the X value for the own system.
Circuit monitor/watch	Un\G□	The fixed value FFFFH(-1) is displayed.
	DX	The engineering tool monitor displays the X value for the own system.
Device test	Un\G□	The engineering tool displays an error message.
	DY	When DY is tested from the engineering tool, the test is performed for Y of the own system.
Intelligent function module monitor	Un\G□	The engineering tool displays an error message. Execute the monitor function again from the control system, not from the standby system.
	X□/Y□	

- Module extension parameters cannot be written from the standby system to a module on the extension base unit by programmable controller writing.
- Set a program execution time of the standby system to be within 200ms. When the program execution time exceeds 200ms, the standby system detects a continuation error. If operation is performed while the program execution time exceeds 200ms, a major error of the CPU module of the control system causes system switching, and because a stop error occurs in the CPU module of the new control system, control may not be continued.
- Configuration of the tracking setting for the following data is recommended, because the standby system does not refresh modules on the extension base unit. If the tracking setting is not configured, the program is executed with the value before refresh in the first scan of the CPU module of the new control system after system switching.
  - Devices where refresh settings were made for the intelligent function module on the extension base unit
  - Devices where refresh settings were made for the CC-Link module on the extension base unit
- If the control system is powered off or reset while the systems cannot be switched due to a tracking communication failure, an error may occur in a module on an extension base unit. In this case, restart the CPU modules of both systems.

28 PROCEDURE FOR STARTING UP A REDUNDANT SYSTEM

This chapter describes the procedures for starting up a redundant system starting from the start-up procedure of CPU modules to execution of programs.

28.1 Overview

The following table lists two procedures for starting up a redundant system.

Start-up procedure	Description
Starting up both systems at a time	Prepare both systems and follow this procedure to start up both systems as a redundant system when the systems are powered on.
Starting up the systems one by one	In the configuration with the main base unit only, follow this procedure to start up a control system first to perform controls, and then start up a standby system to build a redundant system. • Debugging before operation can be performed with only one system. • If one system has failed and the system is powered off before module replacement, powering on this system again starts up only this system. In the redundant extension base unit configuration, the system that was started up first automatically starts up as the control system.

Starting up both systems at a time

1. Installing batteries

Install a battery to the CPU module of each system. (MELSEC iQ-R CPU Module User's Manual (Startup))

2. Mounting extended SRAM cassettes and inserting SD memory cards

Mount an extended SRAM cassette and insert an SD memory card to the CPU module of each system as needed. (MELSEC iQ-R CPU Module User's Manual (Startup))

Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card. (Page 461 Using an SD memory card)

3. Mounting modules and wiring each device

Mount modules on the base unit of each system and wire each device. Use modules of the same model, and mount them on the same slots on each base unit. (MELSEC iQ-R Module Configuration Manual, Page 485 Wiring)

4. Powering on the systems

Check the system for the following and power on the main and extension base units of both systems.

- Wiring to the power supply module is correct.
- The power supply voltage is within the range of specifications.
- The CPU module is in the STOP state.

Check that the following LEDs turn on after the system is powered on.

- Power supply modules on the main base unit and extension base unit: POWER LED
- CPU module: READY LED
- Redundant function module: RUN LED

In step 4, the ERROR LED of each CPU module flashes and the ERR LED of each redundant function module turns on. Proceed to the next step.

5. Connecting a personal computer and a CPU module

Start an engineering tool installed on a personal computer. (Page 487 Creating a Project)

Connect the personal computer on which the engineering tool has been installed and a CPU module. (Page 487 Connecting a Personal Computer and a CPU Module)

6. Initializing the CPU module

Initialize the CPU module with the engineering tool. (MELSEC iQ-R CPU Module User's Manual (Startup))
After initialing the CPU module, connect the other CPU module and the personal computer. (☞ Page 487 Connecting a Personal Computer and a CPU Module)
Initialize the CPU module in the same way.

7. Setting parameters

Set system parameters, CPU parameters, and module parameters. (MELSEC iQ-R CPU Module User's Manual (Startup))

- To use the functions that require an SD memory card, set memory card parameters.
- To mount an intelligent function module, set intelligent function module parameters.

Point

Users can set system parameters by reading the actual system configuration to the module configuration of the engineering tool.

8. Creating a program

Create a program with the engineering tool. After creating the program, convert the program and save the project. (MELSEC iQ-R CPU Module User's Manual (Startup))

9. Writing the system A/B setting

Set the system A or B with the engineering tool. (☞ Page 488 System A/B Settings)

10. Writing data to the programmable controller

Write the set parameters and created programs to both systems with the engineering tool. (☞ Page 490 Writing Data to the Programmable Controller)

11. Resetting the CPU modules

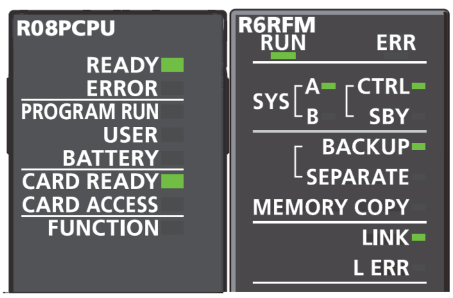
Restart both systems with either of the following methods.

- Powering on the systems
- Resetting the CPU modules

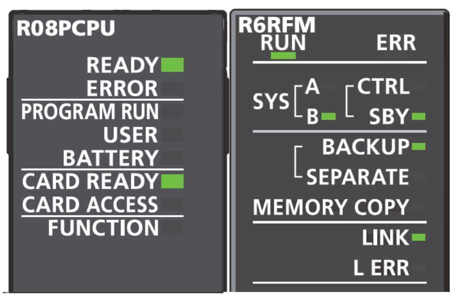
12. Checking the LEDs

Check that the LEDs of each CPU module and redundant function module are in the following states. The CARD READY LED status depends on whether an SD memory card has been installed to each CPU module or not.

- LEDs of System A



- LEDs of System B



When an error has occurred, the following LEDs are on. Check details of the error with the engineering tool and eliminate the error cause.

- CPU module: ERROR LED (MELSEC iQ-R CPU Module User's Manual (Startup))
- Redundant function module: ERR LED, L ERR LED (☞ Page 925 When an error has occurred in a redundant function module)

To start up the system when the data logging function is used, refer to the following.
☞ Page 484 Precautions on starting up the system when the data logging function is used

13. Checking the connection of the extension cable

When extension base units at extension level 2 and later are connected, check that the following LEDs are turned on.

- CONNECT LED for the extension cable connected to the extension base unit
- ACTIVE LED on either of the extension base units

14. Running a program

Power off both systems. Set the RUN/STOP/RESET switch of the CPU module of each system to the RUN position and power on both systems.

Check that the PROGRAM RUN LED of the CPU module of the control system turns on.

Point

Individually setting the RUN/STOP/RESET switch of each CPU module to the RUN position with the CPU module powered on causes a continuation error due to the operating status mismatch and the error is detected in the standby system. Therefore, to start up both systems simultaneously, performing step 13 is recommended.

15. Monitoring the Program

Check that the program is normally running on the engineering tool. (☞ Page 491 Monitoring the Program)

Starting up the systems one by one

To debug a program with only one system before operation, start up the control system.
To start up the control system for a purpose other than debugging a program, start up the standby system to build a redundant system.

Starting up the control system

Start up only the control system.

1. Installing a battery

Install a battery to the CPU module. (MELSEC iQ-R CPU Module User's Manual (Startup))

2. Mounting an extended SRAM cassette and inserting an SD memory card

Mount an extended SRAM cassette and insert an SD memory card to the CPU module as needed. (MELSEC iQ-R CPU Module User's Manual (Startup))

Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card. (Page 461 Using an SD memory card)

3. Mounting modules and wiring each device

Mount modules on the base unit and wire each device. (MELSEC iQ-R Module Configuration Manual, Page 486 Wiring to the power supply modules in a redundant system)

4. Powering on the system

Check the system for the following and power on the main and extension base units.

- Wiring to the power supply module is correct.
- The power supply voltage is within the range of specifications.
- The CPU module is in the STOP state.

Check that the following LEDs turn on after the system is powered on.

- Power supply modules on the main base unit and extension base unit: POWER LED
- CPU module: READY LED
- Redundant function module: RUN LED

In step 4, the ERROR LED of the CPU module flashes and the ERR LED of the redundant function module turns on. Proceed to the next step.

5. Connecting a personal computer and the CPU module

Start an engineering tool installed on a personal computer. (Page 487 Creating a Project)

Connect the personal computer on which the engineering tool has been installed and the CPU module. (Page 487 Connecting a Personal Computer and a CPU Module)

6. Initializing the CPU module

Initialize the CPU module with the engineering tool. (MELSEC iQ-R CPU Module User's Manual (Startup))

7. Setting parameters

Set system parameters, CPU parameters, and module parameters. (MELSEC iQ-R CPU Module User's Manual (Startup))

- To use the functions that require an SD memory card, set memory card parameters.
- To mount an intelligent function module, set intelligent function module parameters.

Point

Users can set system parameters by reading the actual system configuration to the module configuration of the engineering tool.

8. Creating a program

Create a program with the engineering tool. After creating the program, convert the program and save the project. (MELSEC iQ-R CPU Module User's Manual (Startup))

9. Writing the system A/B setting

Set the system A or B with the engineering tool. (☞ Page 488 System A/B Settings)

10. Writing data to the programmable controller

Write the set parameters and created programs to the CPU module with the engineering tool. (☞ Page 490 Writing Data to the Programmable Controller)

11. Resetting the CPU module

Restart the control system with either of the following methods.

- Powering on the system
- Resetting the CPU module

12. Starting up the system (in configuration with the main base unit only)

Perform the following operation on the engineering tool to start up the system as a control system within the time set in "Other system Start-up Timeout Setting".

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

Select "Forced Start of Control System while Waiting for Other System to Start" and click the [Execute] button. (When "Other system Start-up Timeout Setting" of "CPU Parameter" has not been changed, a stop error occurs in 60 seconds.)

Check that the CTRL LED of the redundant function module turns on.

Point

When the CPU parameters have been set, users can start up only one system with the following operations. (☞ Page 551 Redundant System Operation Setting)

- Switch operation: Set the RUN/STOP/RESET switch of the CPU module to the RUN, STOP, and then the RUN position again.
- Input (X) operation: Turn on the input (X) set with the parameters.

13. Starting up the system (in redundant extension base unit configuration)

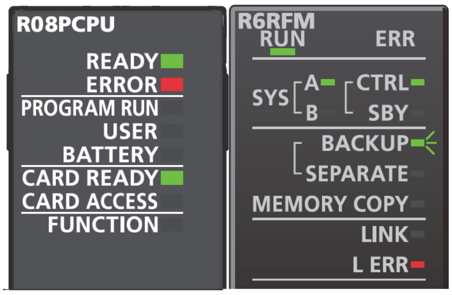
The system is started up as the control system. (The system does not wait for the start-up of the other system and is started up as the control system.)

Point

- When the RUN/STOP/RESET switch of the CPU module is set to RUN, the operating status is changed to RUN, and the standby system is not started up. Therefore, to start control when both systems are started up, set the RUN/STOP/RESET switch to STOP and start up the system.
- The CPU module does not start up if the extension cable between the extension base units is connected incorrectly, so be careful not to connect the extension cable incorrectly.

14. Checking the LEDs

Check that the LEDs of the CPU module and redundant function module are in the following states. The following figure shows the LED status when own system is set as the system A in the system settings. The CARD READY LED status depends on whether an SD memory card has been installed to the CPU module or not.



Point

- Since only the control system has been started up, a continuation error occurs. Thus, the ERROR LED of the CPU module and the L ERR LED of the redundant function module turn on. The BACKUP LED flashes because a system switching failure cause has been generated.
- When "Watching Standby System Setting" of "CPU Parameter" is set to "Disable", the ERROR LED of the CPU module will turn off.

15. Checking the connection of the extension cable

When extension base units at extension level 2 and later are connected, check that the following LEDs are turned on.

- CONNECT LED for the extension cable connected to the extension base unit
- ACTIVE LED on either of the extension base units

Point

When starting up the systems one by one, start up the other system only after the system that was started first has finished starting up (after procedures 1 to 15 have been performed). Because tracking communication cannot be performed by a system that is starting up, a stop error may occur in the CPU module of the system that was started up later. In this case, restart and restore the system with the CPU module with a stop error. (When automatic recovery is set, automatic restart can be executed without manual operation.) (☞ Page 557 Automatic recovery of the CPU module of the standby system)

To start up the system when the data logging function is used, refer to the following.

☞ Page 484 Precautions on starting up the system when the data logging function is used

16. Running a program

Set the CPU module to the RUN state and check that the PROGRAM RUN LED of the CPU module turns on.

17. Monitoring the program

Check that the program is normally running on the engineering tool. (☞ Page 491 Monitoring the Program)

Starting up the standby system

Start up the standby system while the control system keeps operating.

Follow step 1 to 6 in the start-up procedure for the control system to start up the standby system. (☞ Page 480 Starting up the control system)

Point

Check if the model of the module which is mounted on the base unit and insertion of the extended SRAM cassette or SD memory card are the same between the control system and standby system before the starting up.

1. Powering off the system

Power off the standby system.

2. Wiring the redundant function modules

Connect the redundant function modules of both systems with tracking cables. (☞ Page 485 Wiring the redundant function modules)

3. Starting up the system

Power on the standby system.

Connect the personal computer to the control system and copy the memory of the control system to the standby system with the engineering tool by following the procedure described below. (☞ Page 530 Memory Copy from Control System to Standby System)

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

Select "Memory Copy" and click the [Execute] button.

When copying the memory is completed, the MEMORY COPY LED of the redundant function module of the standby system turns on.

After that, reset the CPU module of the standby system and set it to the RUN state.

Point

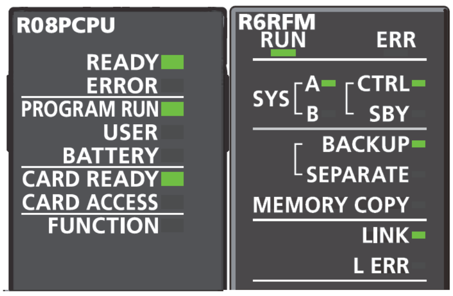
If "Auto Memory Copy Setting" has been set to "Enable" in the CPU parameter settings in advance, the system can be started up only with the following operation. (☞ Page 551 Redundant System Operation Setting)

- Set the RUN/STOP/RESET switch of the CPU module of the standby system to the RUN position and power on the system.

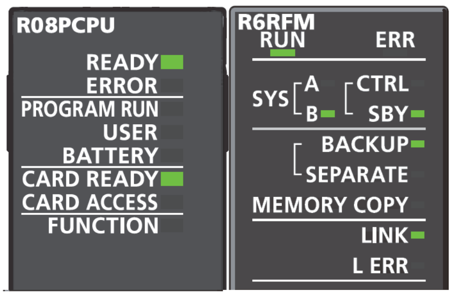
4. Checking the LEDs

Check that the LEDs of the CPU module and redundant function module are in the following states. The following figure shows the LED status when the control system has been started up as the system A. The CARD READY LED status depends on whether an SD memory card has been installed to the CPU module or not.

- LEDs of System A



- LEDs of System B



5. Clearing errors

When the ERROR LED of the CPU module of the control system is on, clear the error with the engineering tool. (☞ Page 145 Error Clear)

Point

When "Watching Standby System Setting" of the CPU parameter has been set to "Disable" in the procedure for starting up the control system, the ERROR LED may be off. (☞ Page 551 Redundant System Operation Setting)

6. Monitoring the program

Check that the program is normally running on the engineering tool. (☞ Page 491 Monitoring the Program)

Precautions on starting up the system when the data logging function is used

If the systems are switched after data logging has failed (processing overflow) due to frequent and continuous data logging collection, data logging continues frequently and continuously in the new control system, and thus the scan time of the new control system increases considerably. Therefore, a WDT error may occur in the new control system and both systems may stop. To avoid the above problem, check the following at system start-up.

- After the redundant system is started, the control system does not fail in data logging collection (processing overflow).*1
- The new control system does not fail in data logging collection (processing overflow) and the CPU module operates without errors after system switching.*1

*1 Whether a processing overflow has occurred or not can be checked with the special register (Number of processing overflow occurrences) corresponding to the data logging No. (☞ Page 966 List of Special Register Areas)

To prevent both systems from stopping, review the following settings.

- Collection interval and data to be collected in the data logging setting
- "Scan Time Monitoring Time (WDT) Setting" of the CPU parameter (☞ Page 136 Scan time monitoring time setting)

Point

When using the data logging function, note that the number of writings to the SD memory card is limited.

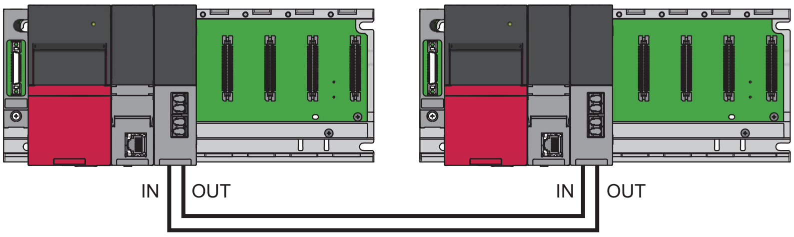
28.2 Wiring

Wiring the redundant function modules

This section describes how to wire the redundant function modules.

Wiring method

Connect the tracking cables from the OUT connector of a redundant function module to the IN connector of the other redundant function module.



Point

For the specifications of the tracking cables connected to redundant function modules, refer to the following.
MELSEC iQ-R CPU Module User's Manual (Startup)

Connecting/disconnecting tracking cables

■Connection procedure

1. Pay attention to the direction of a tracking cable connector and insert the cable to a redundant function module until it clicks. A redundant function module has one IN connector and one OUT connector. Connect the IN connector of the system A and the OUT connector of the system B with a tracking cable and connect the OUT connector of the system A and the IN connector of the system B.
2. Pull each cable lightly and check that it has been connected securely.

■Disconnection procedure

1. Disconnect the tracking cables while pressing the connector hook.

Precautions

- There are restrictions on the cable bending radius. For details, refer to the specifications of the tracking cables used.
- Place the cables in a duct or clamp them. If not, dangling cables may swing or inadvertently be pulled, resulting in damage to the module or cables or malfunction due to poor contact.
- When connecting tracking cables, pay attention not to touch optical fiber cores of the connectors on the cables and module sides and prevent dirt and dust from adhering to them. If oil on hands, dirt, and dust adhere to the optical fiber cores, the transmission loss increases and tracking may not work properly.
- Hold the connector of the tracking cables to connect or disconnect the cable. Pulling the cable connected to the module may result in malfunction or damage to the module or cable or malfunction due to poor contact.

Wiring the extension base unit for the redundant system

For the wiring of the extension base unit for the redundant system, refer to the following.
MELSEC iQ-R Module Configuration Manual

Wiring to the power supply modules in a redundant system

This section describes the wiring to the power supply modules.

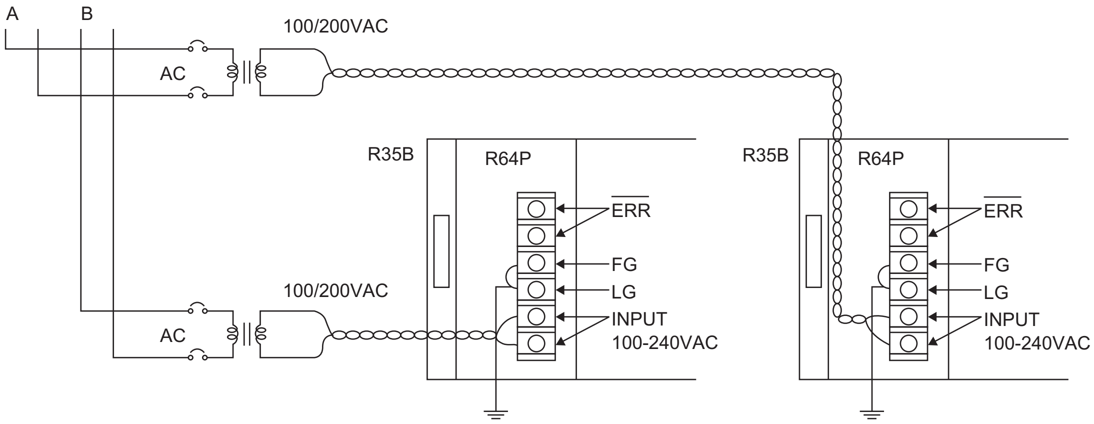
The terminal block of each power supply module has a screw size of M4. Wire cables to the terminal block with the applicable solderless terminal RAV1.25-4 or RAV2-4.

Separately supply power to the system A and B.

Wiring example

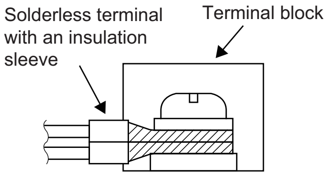
The following figure shows a wiring example of power cables to each main base unit and ground cables.

For wiring examples for each power supply module, refer to the manual included with the power supply module. (📖 Before Using the Product)



Point

- 100VAC, 200VAC and 24VDC wires must be twisted starting from the terminal connected, and connect modules at the shortest distance. Also, use the thickest wire (maximum 2mm²) to reduce the voltage drop.
- For the wiring to a terminal block, use a solderless terminal.
- To prevent the short-circuit because of loosening screws, use the solderless terminal with an insulation sleeve of 0.8mm or less. Note that up to two solderless terminals can be connected per terminal block.

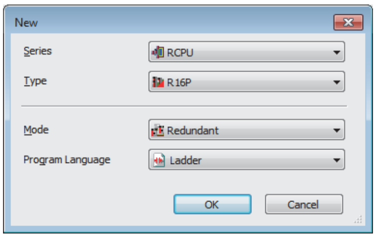


- Ground the LG and FG terminals after short-circuiting them. Failure to do so may cause the terminals to be susceptible to noise. The LG terminal has a half potential of the input voltage.
- When two redundant power supply modules operate in parallel as a redundant power supply system, it is recommended to connect the one redundant power supply module to an AC power supply and the other one to an uninterruptible power supply (UPS).

28.3 Creating a Project

Start the engineering tool and create a project.

[Project] ⇨ [New]



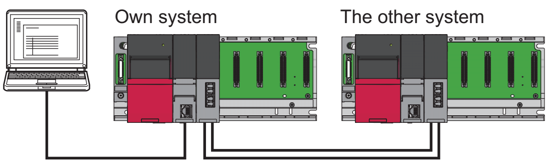
1. Select the Process CPU to be used for "Type". Select "Redundant" for "Mode". Select a programming language to be used for "Programming Language" and click the [OK] button.

28.4 Connecting a Personal Computer and a CPU Module

Connect the personal computer on which the engineering tool has been installed and a CPU module.

Connection procedure

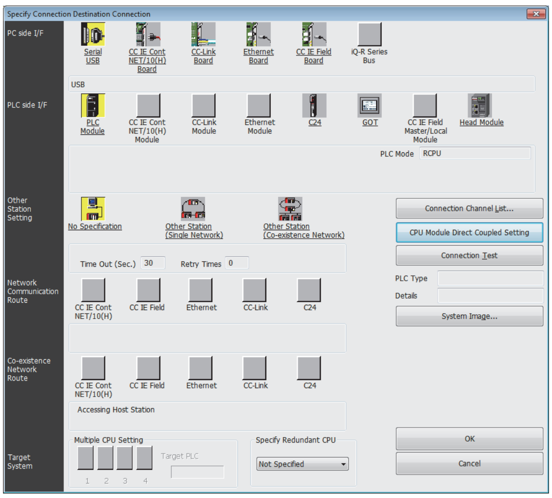
The following describes the procedure for directly connecting a CPU module and the personal computer.



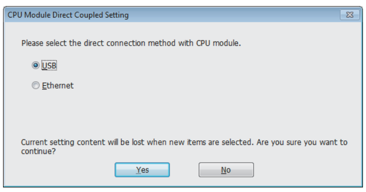
1. Connect a CPU module to the personal computer with a USB cable or an Ethernet cable. The CPU module connected with a USB cable or an Ethernet cable becomes own system.

2. Select [Online] ⇨ [Current Connection Destination] from the menu of the engineering tool.

3. Click the [CPU Module Direct Coupled Setting] button on the "Specify Connection Destination Connection" window.



4. Select a method of connection with the CPU module and click the [Yes] button.



5. Set "Specify Redundant CPU" to "Not Specified" on the "Specify Connection Destination Connection" window and click the [Connection Test] button to check that the personal computer has been connected with the CPU module.

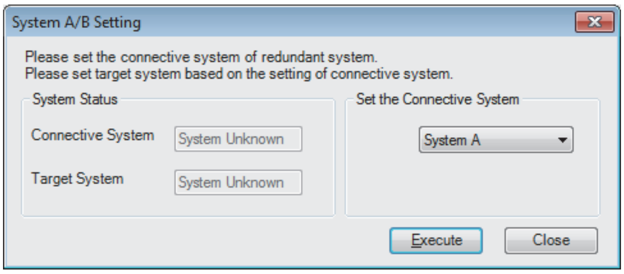
When connecting the personal computer and the CPU module with a USB cable for the first time, install a USB driver. (GX Works3 Operating Manual)

28.5 System A/B Settings

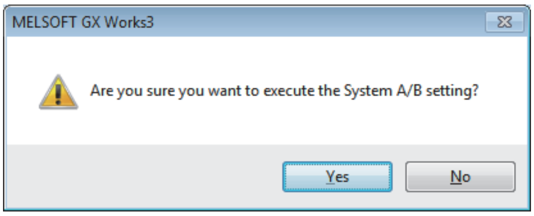
Set the system A or B with the engineering tool and write the system settings to the CPU module.

[Online] ⇨ [Redundant PLC Operation] ⇨ [System A/B Setting]

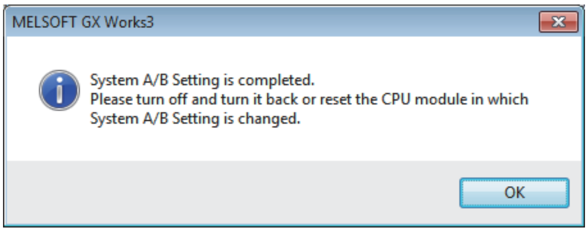
Configuration procedure



1. Set the CPU module to the STOP state.
2. Select the setting to be written to own on the "System A/B Setting" window.
3. Click the [Execute] button.



4. Click the [Yes] button on the window on the left. Check that the SYS A LED or SYS B LED of the redundant function module flashes in accordance with the setting.



5. To change the system A/B settings, power off and on the system or reset the CPU module, and then click the [OK] button. Check that the SYS A LED or SYS B LED of the redundant function module turns on in accordance with the new setting.*1

*1 If the system A/B settings are set to the Process CPU that is not running in the redundant mode, the SYS A LED or SYS B LED of the redundant function module will turn off. Follow the procedure below (writing data to the programmable controller).

Point

Besides the method that uses the engineering tool for setting system A and B, system A and B can be automatically set by using the systems.
When one system is set to system A and gets ready for tracking communications, the other system is automatically set to system B, and vice versa.
For the system to which system A or system B has been assigned automatically, the system setting will be overwritten automatically. In that case, the system setting change can be checked in the event history. (Event code: 00700)

- If a system that has no system A/B setting yet is connected to the system specified as system A, the system that has no system A/B setting will be automatically set to system B. (This event is not recorded in the event history.)
- If a system specified as system A is connected to another system specified as system A, the system that is turned off and on or reset will be automatically set to system B. (This event is recorded in the event history.)

Checking method

Check the LED of each redundant function module to check the system status.

Setting on the engineering tool	LED of the redundant function module
System A	
System B	
When the system setting is switched from system A to B	 When the system setting is switched to the system B with the engineering tool, The SYS B LED will flash. Powering on or resetting the CPU module turns off the SYS A LED and turns on the SYS B LED.

Point

Users can check the system A/B setting with the engineering tool. (📖 GX Works3 Operating Manual)
- System monitor
- "System A/B Setting" window
- Monitor status bar

Precautions

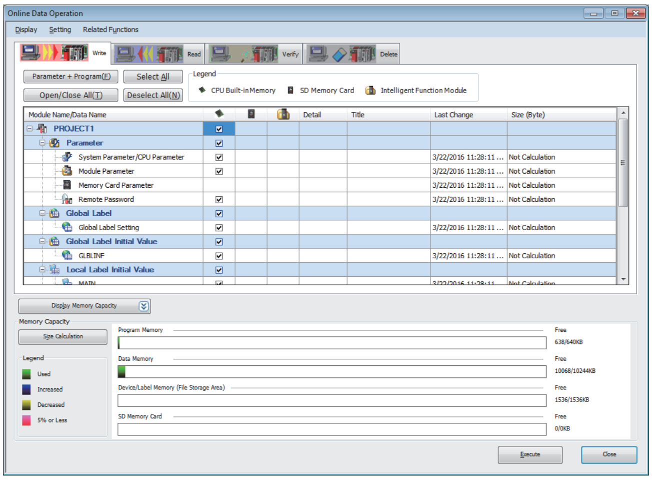
- Do not power off or reset the CPU module during system A/B setting. Doing so may not properly reflect the system A/B setting. If the system A/B setting has not been properly reflected, set the system again.
- For the system A/B setting, use the CPU module that has been connected with a USB cable or an Ethernet cable. Users cannot set a system if they have changed the connection destination on the engineering tool.
- The set system A/B information cannot be deleted. Only changing the set system A/B information is supported.
- When starting up both systems, if both systems have no system A/B setting yet or the same system A/B setting, a stop error will occur when tracking communications are established. Set a different system A/B setting for each system.
- If the system that has no system A/B setting yet is started first, a stop error will occur when tracking communications are established. Set a different system A/B setting for each system. If the system A/B settings are not configured for both systems in the redundant extension base unit configuration, modules on the extension base unit do not start up.
- Be careful not to connect between running system As or between system Bs with tracking cables. Otherwise, a continuation error will occur.

28.6 Writing Data to the Programmable Controller

Write the set parameters and created programs to the CPU module.

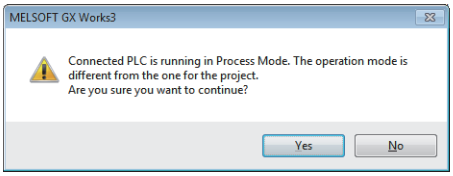
[Online] ⇨ [Write to PLC]

Operating procedure

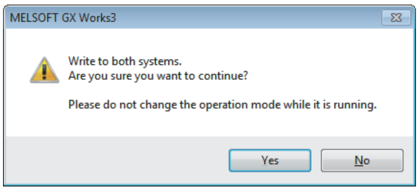


1. Select system parameters, CPU parameters, module parameters, and program files on the "Online Data Operation" window. When FBs are used, select the corresponding FB/FUN files.

2. Click the [Execute] button.



3. When the Process CPU is not in the redundant mode, the window shown on the left appears. Click the [Yes] button.



4. Click the [Yes] button.*1

5. When writing the data to the programmable controller is completed, click the [Close] button.

*1 When one of both systems is started up first, the window asking for writing the data to only the connected system appears.

Point

- The same data can be written to both systems when the operation mode is the backup mode. Always write the same data to both systems to prevent the occurrence of a file mismatch in the system consistency check. When writing programs, FB files, or global label setting after performing [Convert] or [Rebuild All], always write them to both systems. Otherwise, a file mismatch will be detected.
- To operate CPU modules, write system parameters, CPU parameters, and program files. To operate I/O modules and intelligent function modules, write module parameters or module extension parameters.
- When new parameters have been set or the set parameters have been changed, reset the CPU module. (MELSEC iQ-R CPU Module User's Manual (Startup))
- Use the [Select Favorites] button to register frequently used items such as system parameters, CPU parameters and programs. Select [Setting] ⇨ [Register Favorites Selection] from the menu on the "Online Data Operation" window and set the items for the [Select Favorites] button.

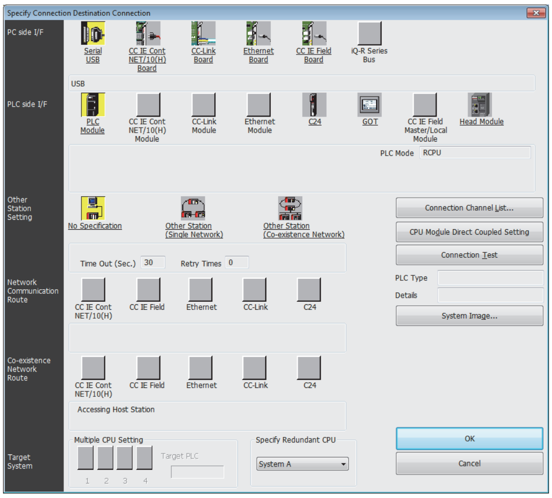
Precautions

Do not write data to both systems during the initial processing (or while the READY LED is on) after the power of the other system is turned off and on or reset. Writing to the programmable controller may fail due to the operating status mismatch.

28.7 Monitoring the Program

Check the operation of a program on the engineering tool.
Change the connection destination with the engineering tool and check the operating status of the system A or B.

[Online] ⇨ [Current Connection Destination]



1. Select a system in "Specify Redundant CPU" on the "Specify Connection Destination Connection" window.
2. Click the [Connection Test] button to check whether the CPU module of the selected system has been connected.

For how to check the operation, refer to the following.
GX Works3 Operating Manual

28

29 FUNCTIONS

This chapter describes the redundant functions and the functions that are different (modified or restricted) from those of the process mode. Functions not described in this chapter are the same as those described in PART 2.
For the availability of the functions in the redundant system, refer to the following.
MELSEC iQ-R CPU Module User's Manual (Startup)

Function			Description	Reference
Redundant function	Operation mode change		Switches the operation mode of the redundant system between the backup mode for normal operation and the separate mode for system maintenance while it is running.	Page 496 Operation Mode Change
	System switching		Switches the systems between the control system and the standby system to continue operation of the redundant system when a failure or an error occurs in the control system. For debugging and maintenance purpose, users can switch the systems at any desired timing.	Page 499 System Switching
	Tracking transfer		Transfers the control data from the control system to the standby system and maintains the consistency of the data in the two systems to continue operation of the redundant system when a failure or an error occurs in the control system.	Page 513 Tracking Transfer
	Memory copy from control system to standby system		Transfers data such as parameters and programs in the CPU module of the control system to the CPU module of the standby system to maintain the consistency of the memory in the two CPU modules.	Page 530 Memory Copy from Control System to Standby System
	System consistency check		Checks whether the system configurations and files in the CPU modules are the same between the control system and the standby system when the redundant system is in backup mode.	Page 539 System Consistency Check
	Program execution in both systems		Detects an error in the external device or network of the systems (control system and standby system) by executing a program that diagnoses external devices or networks of both systems.	Page 544 Program Execution in Both Systems
	Redundant system operation setting		Set the redundant system operation in the redundant system settings of the CPU parameter.	Page 551 Redundant System Operation Setting
	Redundant function module communication test		Checks a redundant function module for error when the communications of the redundant function module are unstable.	Page 555 Redundant Function Module Communication Test
	Redundant extension base unit configuration	Redundant extension base unit configuration setting	Set the redundant extension base unit configuration setting operation in the redundant settings of the CPU parameter.	Page 556 Redundant Extension Base Unit Configuration Setting
		Automatic recovery of the CPU module of the standby system	Allows the CPU module of the standby system to automatically recover from a specific error that occurs in the CPU module of the standby system. (Manual operation (turning off and on or resetting the system) is not required to recover the system.)	Page 557 Automatic recovery of the CPU module of the standby system
		Replacement/ addition of an extension cable (online)	The extension cable on the inactive side (ACTIVE LED is turned off) between the redundant extension base units can be replaced or added while the system is running.	Page 605 Replacement/ Addition of an Extension Cable (Online)

Function			Description	Reference
Functions different (modified or restricted) from those of the process mode	Constant scan		• In the standby system, when the system is powered off, a hardware failure has occurred, a tracking cable has a failure, or the system is switched to the separate mode, a continuation error may occur due to the excess of constant scan time. • The constant scan function is invalid for the standby system in backup mode.	Page 559 Constant Scan
	Device/label access service processing constant wait function		Operation in redundant mode and the setting method	Page 578 Device/label Access Service Processing Constant Wait Function
	Interrupt function		When the systems are switched during time measurement of I28 to I31, I48, or I49, the time measurement interrupts and the new control system starts the time measurement from 0.	—
	Clock function	Time synchronization	The time of the standby system is synchronized with the time of the control system in backup mode. The time setting using the time setting function (SNTP client) is available only for the control system.	—
		System clock	After system switching, SM420 (User timing clock No.0) to SM424 (User timing clock No.4) remain off on the new control system CPU module. To use SM420 to SM424 on the new control system CPU module, execute the DUTY instruction again.	—
	Writing data to the CPU module	Online change (ladder block)	• When the online change (ladder block) is performed on the CPU module in one system in the backup mode, the change is also reflected on the CPU module in the other system. • Data cannot be written to the programmable controller while its CPU module is running or while connecting to a module on the extension base unit.	Page 560 Online Change
		File batch online change of FB files and the global label setting file	Data can be written only when the firmware version of the CPU modules of both systems supports writing of FB files and the global label setting file online.	Page 1139 Added and Enhanced Functions
	RAS function	Scan monitoring function	• During system switching, scan time monitoring with the watchdog timer is interrupted. Thus, no error is detected even if the scan time monitoring time has elapsed. Therefore, a time taken for system switching does not need to be considered in the scan time monitoring time setting. • Scan time monitoring with the watchdog timer is interrupted while the standby system is waiting for tracking data reception. Thus, no error is detected even if the scan time monitoring time has elapsed while the standby system is waiting for tracking data reception. Therefore, a time taken for tracking data reception does not need to be considered in the scan time monitoring time setting.	—
		Self-diagnostic function	The following items related to the redundant system are added to the contents that can be checked with the system monitor. • Tracking cable status • Display of the main base units of systems A and B • Module configuration on the base unit of the selected system • Event history of own system • System to which a CPU module is belonged (control system/standby system, system A/B) and operation mode of the CPU module (backup mode/separate mode)	GX Works3 Operating Manual
		Error clear	A continuation error occurred in own system can be cleared by using the special relay (SM50) or the connected engineering tool.	Page 145 Error Clear
		Clearing errors on the standby system CPU module from the control system CPU module	Errors on the standby system CPU module can be cleared by using the special relay (SM1679) of the control system CPU module.	Page 561 Clearing errors on the standby system CPU module from the control system CPU module

Function			Description	Reference
Function different (modified or restricted) from that of the process mode	RAS function	Event history function	Events that occur in a module on the extension base unit are saved in the event history of the CPU module of the control system.	Page 561 Event history function
		Event history logging restriction	The CPU module of the control system monitors and restricts the logging of events that occur in a module on the extension base unit.	Page 561 Event history logging restriction
	Remote operation	Using a contact	When the backup mode setting is enabled in the CPU parameter, both systems may be recognized as being mismatched in the system consistency check even if the remote operation is simultaneously performed on the systems because their operating statuses are mismatched depending on the timing.	—
		Using an engineering tool	In a redundant system, the operation target of remote operations depends on the operation mode and type.	Page 562 Remote Operation
	Boot operation		Use the boot operation only to simultaneously start up both systems. When one system is restarted and the boot operation is used, an error occurs.	Page 564 Boot Operation
	External input/output forced on/off function		Forced on/off is reflected to the input/output devices of both systems and external outputs by registering or canceling forced on/off for the control system. (Forced on/off is reflected to both systems without setting tracking transfer setting in the CPU parameter.)	Page 565 External Input/ Output Forced On/Off Function
	Device test with execution condition		Using the engineering tool, device/label values can be set for each execution of specified steps of programs for the control system and standby system.	Page 568 Device Test with Execution Condition
	Monitoring/test function		If system switching occurs while the monitoring/test function is being used for a module on the extension base unit in the redundant extension base unit configuration, the following process is performed. • If the monitor function is executed during system switching, an error occurs. • If the new standby system is connected after system switching, monitoring stops or a fixed value (FFFFH(-1)) is displayed.	—
	Test function		Changes in values of devices and labels are transferred (tracking) from the control system to the standby system before the END processing. If the systems are switched between the change and the tracking transfer, the change is not reflected on the new control system.	—
	Data logging function		In a redundant system, the data logging function collects data only in the control system regardless of the operation mode.	Page 570 Data Logging Function
	CPU module data backup/restoration function		The internal data of the CPU module in each own system is backed up and is restored only to the CPU module in the own system.	Page 573 CPU Module Data Backup/ Restoration Function
	Multiple CPU system function		This function cannot be used in redundant mode.	—
	Security function		The security key setting is written or deleted on both systems individually, regardless of the operation mode.	—
	Latch function		When the systems are switched, the new control system and the new standby system start the value in "Interval Setting at Time Setting" in "Device Latch Interval Setting" from 0.	—
	Device/label initial value setting		The device/label initial value specified for the control system is reflected in the buffer memory of the module on the extension base unit by turning off and on the control system or switching from STOP to RUN.	—
	Label access setting from external device		In the transfer setup of external devices that communicate with the CPU module through GOT, SLMP, or other methods, select "Control System/Standby System", "System A/System B", or "Not Specified" to start a communication with a global label name specified.	—
	Program restoration information write selection		Only when the firmware versions of the CPU modules of both systems support the program restoration information write selection, "Not to Write" can be selected.	Page 1139 Added and Enhanced Functions

Function			Description	Reference
Function different (modified or restricted) from that of the process mode	SFC function		• The SFC program cannot be set to be executed in both systems. • POFF(P) and PSCAN(P) instructions cannot be used for the SFC program.	—
	SLMP communication		• There are notes for system switching when the system IP address matching function is not used. • If a remote operation command is executed, the CPU modules enter different operating statuses and thus systems cannot be switched. • For SLMP communications via the built-in Ethernet port of the CPU module, when communications are performed to the other system that cannot respond (power-off, reset, or tracking cable disconnection), a timeout error may occur. • The SLMP frame send instruction (SP.SLMPSND) performs communications using IP addresses of the system A and B.	Page 579 SLMP Communication
	Ethernet function	File transfer function (FTP client)	• This function performs communications using IP addresses of the system A and B.	MELSEC iQ-R Ethernet User's Manual (Application)
	Inter-module synchronization function		This function cannot be used in redundant mode.	—

29.1 Operation Mode Change

This function switches the operation mode of the redundant system between the backup mode for normal operation and the separate mode for system maintenance while it is running.

Switching procedure

Switch the operation mode of the control system CPU module in the "Redundant Operation" window of the engineering tool.

Mode switching to the separate mode

The following describes the switching procedure for the separate mode.
To prevent tracking transfer in separate mode, turn off the tracking trigger in advance. (Page 519 Tracking trigger)

1. Connect the engineering tool to the control system CPU module.
2. Open the "Redundant Operation" window of the engineering tool.

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

3. Select "Separate Mode" in "Change the Operation", and click the [Execute] button. The SEPARATE LED of the redundant function module turns on.

4. The standby system CPU module is set to the RUN-transition instruction waiting state (same as the STOP state), and the PROGRAM RUN LED flashes.
5. Set the standby system as the connection destination with the engineering tool, and execute the remote STOP and then the remote RUN with the remote operation function.
6. The CPU module is set to the RUN state and executes a program, and the PROGRAM RUN LED turns on.

Point

For the step 5, the following methods are also available for setting the CPU module to the RUN state.
- Setting the RUN/STOP/RESET switch in the STOP position to the RUN position
- Setting the module in the STOP state to the RUN state by using the RUN-PAUSE contact

■Powering off and on or resetting the CPU module in separate mode

In separate mode, the systems are not switched even when the CPU module of the control system is powered off and on or reset.[*1]
- When the control system CPU module is turned off and on or reset, the system starts up as the control system in separate mode at the next startup.
- When the standby system CPU module is turned off and on or reset, the system starts up as the standby system in separate mode at the next startup.
- When the CPU modules of both systems are simultaneously turned off and on or reset, the systems start up in the backup mode at the next startup.

*1 In the redundant extension base unit configuration, do not turn off or reset the CPU module of the control system during operation in separate mode, because control cannot be continued.

Before turning off or resetting one system in the separate mode, check if the other system is in the following status.
- The other system has been started and the READY LED is on.
- When the other system is set to the RUN state, the PROGRAM RUN LED is on.

If one system is turned off or reset while the other system is in the initial processing or initial processing (when switched to RUN), the operation modes of both systems become different and a stop error occurs.

Mode switching to the backup mode

The operation mode can be switched to the backup mode only in the communication path of when the operation mode was switched to the separate mode.

The following describes the switching procedure for the backup mode.

1. Set the CPU modules of the control system and standby system to have the same file configuration and operating status. (To prevent an error detected by a restarted system consistency check after switching the separate mode to the backup mode)

2. Connect the engineering tool to the control system CPU module.

3. Open the "Redundant Operation" window of the engineering tool.

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

4. Select "Backup Mode" in "Change the Operation", and click the [Execute] button. The BACKUP LED of the redundant function module turns on.

5. When the tracking trigger has been turned off at the mode switching to the separate mode, turn on the tracking trigger. (☞ Page 519 Tracking trigger)

Point

- Even when the CPU modules of both systems are simultaneously turned off and on or reset, the systems start up in the backup mode at the next startup.
- When the control system CPU module has been turned off or reset in separate mode, the operation mode can be switched from the separate mode to the backup mode in a different communication path by using the engineering tool.
- When the operation mode is switched to the separate mode, the standby system CPU module is set to the RUN-transition instruction waiting state (the PROGRAM RUN LED flashes). If the operation mode is switched to the backup mode in this state, the CPU module is set to the RUN state (the PROGRAM RUN LED stops flashing).

Precautions

The following describes the precautions for switching the operation mode.

Switching of the operation mode in the RUN-transition instruction waiting state

When the operation mode is switched to the separate mode, the standby system CPU module is set to the RUN-transition instruction waiting state (same as the STOP state). If the operation mode is switched to the backup mode in this state, the standby system CPU module goes to the RUN state.

The following describes the precautions for this state.

- If the SP.CONTSW instruction is executed before the standby system CPU module goes to the RUN state (during initial processing (when switched to RUN)), the operating status of each of both systems becomes different and a continuation error will occur. (☞ Page 512 System switching during initial processing/initial processing (when switched to RUN))
- If the standby system CPU module takes time to switch to the RUN state due to causes such as access to a file in the standby system CPU module, the operating status of each of both systems becomes different and a continuation error occurs.

Switching the mode during initial/initial processing (when switched to RUN)

When the operation mode is switched during initial processing or initial processing (when switched to RUN), the operation mode is switched after the initial processing or initial processing (when switched to RUN) is finished. Even if the initial processing or initial processing (when switched to RUN) takes time and a communication error occurs, the operation mode is switched after the initial processing or initial processing (when switched to RUN) is finished.

Consecutive operation mode switching

To consecutively switch the operation mode, switch the operation mode in certain intervals. If the operation mode is changed again before the operating status is switched by the previous operation mode switching, the operating status of each of both systems becomes different and a continuation error will occur.

29.2 System Switching

This function switches the systems between the control system and the standby system to continue operation of the redundant system when a failure or an error occurs in the control system. The systems can also be switched manually by a user for debug or maintenance.

System switching method

The following two methods are available for system switching: system switching (automatically performed by a redundant system) and manual system switching (performed by a user).
The following table lists the system switching types, causes, execution availability by operation mode, and priority of when multiple causes are simultaneously generated.
○: Operation possible, ×: Operation not possible

System switching type	System switching cause	Operation mode		Priority	
		Backup mode	Separate mode		
Automatic system switching	Power-off, reset, hardware failure of the CPU module	○	×	High ↑ ↓ Low	1
	Stop error of the CPU module				2
	System switching request from a network module				3
Manual system switching	System switching request by the SP.CONTSW instruction	○	○		4
	System switching request from the engineering tool				5

- When multiple system switching causes are simultaneously generated, the systems are switched according to the cause with higher priority. The switching cause determined according to the priority is stored in the event history and SD1643 (System switching cause).
- The manual system switching is requested for the control system CPU module.

Point

The systems are not switched even when the CPU module is set to the STOP state. They are switched when any of the above switching causes is generated.

Automatic system switching

A redundant system judges whether system switching is required or not in backup mode, and automatically switches the systems between the control system and the standby system as required.

■System switching due to power-off, reset, or hardware failure of the CPU module

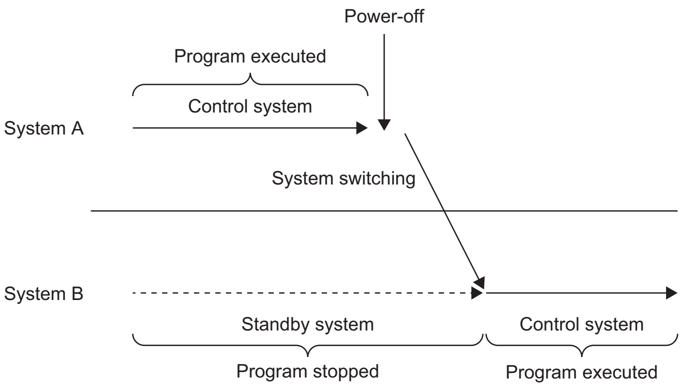
In a redundant system, the standby system CPU module monitors the control system status. If the control system is unable to control the redundant system in the following cases, the standby system CPU module is switched to the control system CPU module and the new control system CPU continues the control over the redundant system.

- The control system CPU module has been turned off.
- The control system CPU module has been reset.
- A hardware failure has occurred on the control system CPU module.*1

*1 The control system CPU module is switched to the standby system CPU module even when the existing standby system CPU module is not ready to switch. (☞ Page 506 Execution availability of system switching)

Ex.

Operation of when the control system (system A) CPU module is turned off



Point

In the redundant extension base unit configuration or when a network is established using the CC-Link IE Field Network module, system switching is also performed in the following cases.

- When the redundant function module has been removed from the base unit
- When the base unit has failed

■Precautions when system switching is performed by a reset from the RUN/STOP/RESET switch

When system switching is performed by a reset from the RUN/STOP/RESET switch, the CPU module may be reset after the operating status of the CPU module is switched to STOP. In this case, system switching is performed after the output (Y) is turned off when the operating status is STOP. Therefore, when reset is performed by the RUN/STOP/RESET switch of the control system when both systems are running, check that system switching after the output (Y) is turned off does not cause a problem. If there is any problem, manually switch the control system to the standby system, and then perform reset of the RUN/STOP/RESET switch.

■Stop error of the CPU module

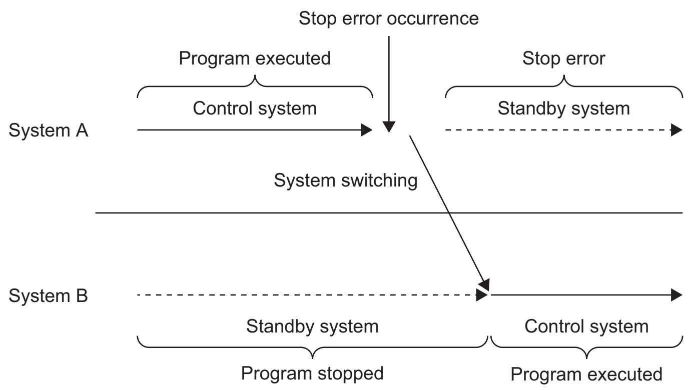
When a stop error occurs on the control system CPU module, the standby system is notified of a system switching due to the stop error, and the standby system CPU module is switched to the control system CPU module. The control system CPU module where the stop error has occurred is switched to the standby system.*1

*1 If a WDT error has occurred, the control system CPU module is switched to the standby system CPU module even when the existing standby system CPU module is not ready to switch. (☞ Page 506 Execution availability of system switching)

29

Ex.

Operation of when a stop error occurs on the control system (system A) CPU module



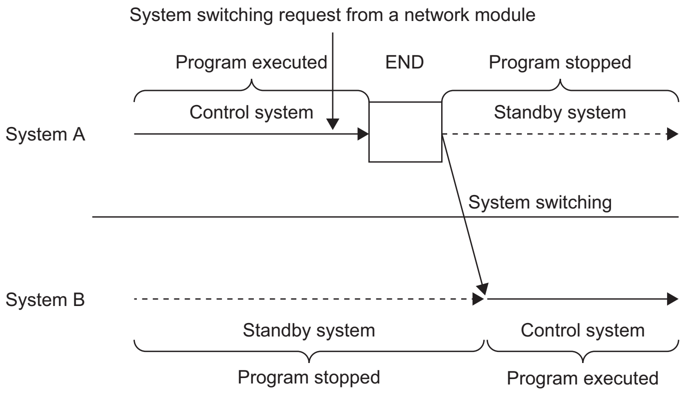
■System switching request from a network module

The network module on the main base unit of the control system requests the CPU module to switch systems when a communication error or disconnection is detected. (The network module on the extension base unit does not request the CPU module to switch systems when an error is detected.)

When the control system CPU module receives the system switching request from the network module, the systems are switched after the END processing.

Ex.

Operation of when the CPU module receives a system switching request from a network module



Point

When disconnecting the network cable of the control system due to maintenance or for other reasons, disconnect the network cable of the standby system in advance. Otherwise, system switching will not occur even when the network cable of the control system is disconnected.

The following network modules send a system switching request.

- CC-Link IE Controller Network module
- CC-Link IE Field Network module
- Ethernet interface module with built-in CC-Link IE
- MELSECNET/H network module
- PROFIBUS-DP module

If a network module cable is disconnected, the systems may not be switched depending on the timing of error detection on the control system and the timing on the standby system. (☞ Page 512 When the cable for the network module is disconnected)

Manual system switching

The user can manually switch the systems between the control system and the standby system.

Point

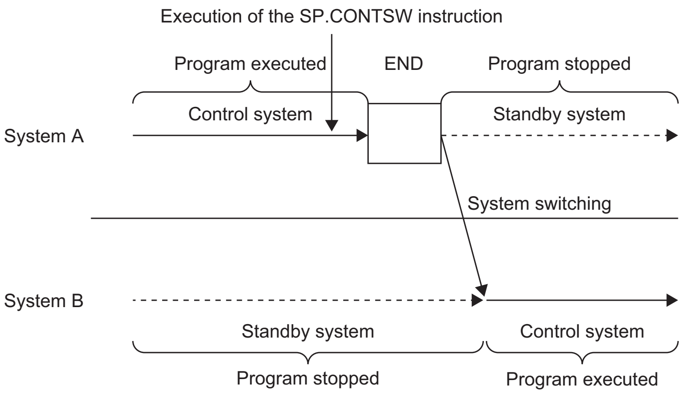
- After turning on SM1646 (System switching by a user), perform the manual system switching in the control system.
- When the manual system switching is disabled by the DCONTSW instruction, execute the ECONTSW instruction. The system switching is enabled in the initial status.

■System switching by the SP.CONTSW instruction

When the SP.CONTSW instruction is executed on the control system CPU module, the systems are switched at the END processing after the instruction execution.

Ex.

Operation of system switching by the SP.CONTSW instruction



Restriction

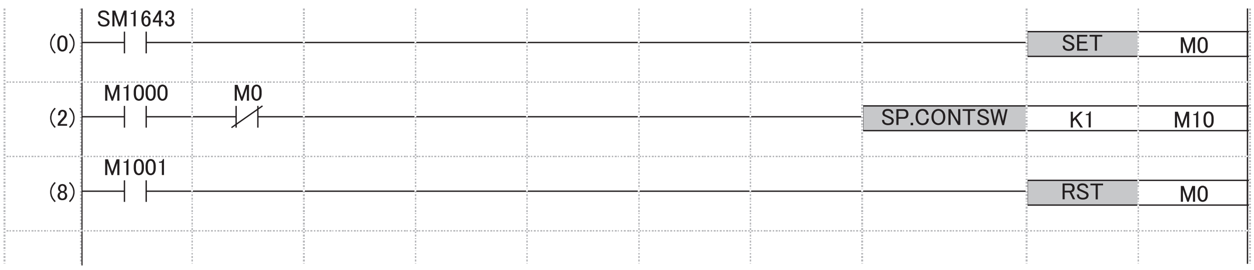
When the SP.CONTSW instruction is executed while the tracking transfer is in asynchronous transfer mode, such as immediately after the transition to RUN in the backup mode or in separate mode, several scans may be required for the system switching.

Point

Once the devices and labels specified in "Tracking Setting" are transferred, the CPU modules in both of the control system and standby system have the same specified devices and labels. Thus, if the systems are switched by using the SP.CONTSW instruction on the control system CPU module, switching may also be performed on the new control system CPU module.

When the SP.CONTSW instruction is used, create a program that does not execute the SP.CONTSW instruction again on the new control system CPU module by using SM1643 (ON for only one scan after system switching (standby system to control system)), as shown below.

M1000: System switching command, M1001: Clear signal



For details on the SP.CONTSW instruction, refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

■System switching request from the engineering tool

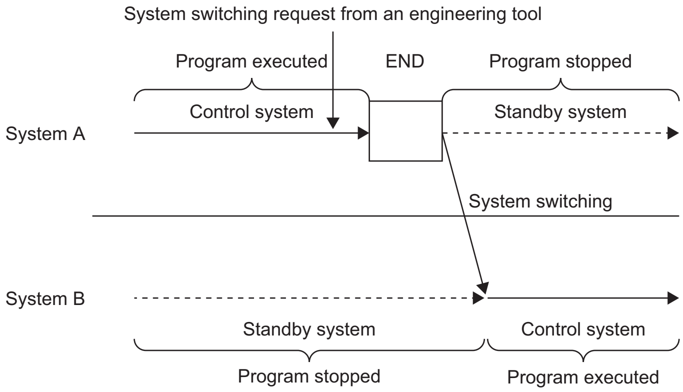
When the engineering tool sends a system switching request to the control system CPU module, the systems are switched after the END processing.

Switch the systems from the "Redundant Operation" window of the engineering tool.

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

Ex.

Operation of system switching from the engineering tool



Operation at system switching

The following table shows the operations of the CPU modules of when the control system and the standby system are switched.

These operations are for when both systems are operating and the operating statuses of the CPU modules are the same.

Item		New control system CPU module	New standby system CPU module
Program execution (Except the SFC program)	Initial execution type program	This type of program is not executed. However, when an initial execution type program has not been completed on the old control system at system switching, it is executed again from its head.	The program execution stops.[*1]
	Scan execution type program	This type of program is executed from the step 0.	
	Fixed scan execution type program	The fixed scan execution interval is measured from 0.	The program execution stops.
	Standby type program	This type of program is not executed.	
	Event execution type program	• Interrupt program: The program is executed when an interrupt factor is generated. • ON of bit data (TRUE): The program is executed when a specified factor is generated. • Passing time: A specified time is measured from 0.	
Execution of SFC program	Block active status	The status before system switching is held.	
	Step active status	The status before system switching is held.	
	Execution of action	The program execution is started in the active action using transferred device data.	The program execution stops.
Tracking transfer		Tracking data is transferred to the new standby system.	Tracking data is received. However, when a stop error occurs on the new standby system, the new standby system does not receive the tracking data.
Online change		Write operation at system switching continues.	
Device/label memory		The state before system switching is held.	
Signal flow memory (Page 521 Tracking transfer setting for the signal flow memory)		• In backup mode, the signal flow of the old control system is reflected.[*2] • In separate mode, the signal flow of the old standby system is reflected.	The state before system switching is held.
Special relay (SM), special register (SD)		The state before system switching is held. However, when the CPU module is in the RUN state, SD520 to SD531 are cleared.	
Output (Y)		The state before system switching is held and output refresh is performed.	After the output (Y) is turned off, output refresh stops.[*3]
Direct access input (DX)		In the program execution after system switching, data is fetched when an instruction using the direct access input (DX) is executed.	No operation is performed because the program does not operate.[*1]
Direct access output (DY)		In the program execution after system switching, data is output when an instruction using the direct access output (DY) is executed.	
FROM/TO instructions		In the program execution after system switching, this instruction is executed when the instruction execution condition is satisfied.	
Instruction that requires several scans		In the program execution after system switching, this instruction is executed when the instruction execution condition is satisfied.	• No operation is performed because the program does not operate.[*1] • When the systems are switched while the instruction is being executed, the instruction execution continues. However, the completion device does not turn on in the new standby system, and it turns on in the control system after the next system switching.[*1]
Constant scan		Constant scan is enabled.	• The constant scan function is disabled in the backup mode. • The constant scan function is enabled in the separate mode.

*1 The operation varies in a program executed in both systems. (Page 547 Operation at system switching)

*2 The operation varies when the signal flow memory is not transferred. (Page 584 Instructions whose operations vary depending on tracking of the signal flow memory)

*3 The operation varies when the standby system output setting is enabled. (Page 551 Redundant System Operation Setting)

Execution availability of system switching

The following tables show the execution availability of system switching in each operation mode.

In backup mode

○: Switching possible, ×: Switching not possible

Redundant system status		Execution availability of system switching						
		Automatic system switching					Manual system switching	
		Power-off or reset of the CPU module	Hardware failure of the CPU module	Stop error of the CPU module		System switching request from a network module*3	System switching request by the SP.CONTSW instruction*3	System switching request from the engineering tool*4
				WDT error	Error other than WDT error			
Normal operation or continuation error in the standby system		○	○	○	○	○	○	○
Tracking communications disabled (cable disconnection)		×	○*6	○*6	×	×	×	×
Power-off, reset, hardware failure of the standby system CPU module		×	○*6	○*6	×	×	×	×
Stop error in the standby system	WDT error	○	○*6	○*6	×	×	×	×
	Error other than WDT error	○	○	○*6	×	×	×	×
Network error detected in the standby system*1		○	○	○	○	×	×	×
During memory copy from control system to standby system		○	○	○*6	×	×	×	×
During online change		○	○	○	○	○*2*5	×	×
Mismatch between the CPU module operating status of both systems		○	○	○	○	×	×	×
During system switching		○	○	○	○	○	×	×
During online module change	Redundant module	×	○*6	○*6	×	×	×	×
	Module on the main base unit in the redundant extension base unit configuration	○	○	○	○	○	×	×
	Other modules	○	○	○	○	○	○	○
System switching disabled by the DCONTSW instruction		○	○	○	○	○	×	×

*1 When the group specification has been set in the standby system, a network error is not detected if communication is available with the line of an Ethernet-equipped module after a communication error has occurred on the other Ethernet-equipped module. (MELSEC iQ-R Ethernet User's Manual (Application))

*2 After the online change has been completed, the switching cause is detected and the systems are switched. However, when an online change is being executed only in the standby system, the systems cannot be switched.

*3 If system switching is disabled when a system switching request is sent, a continuation error occurs and a cause of a system switching failure is stored in SD1644 (Cause of system switching failure).

*4 If system switching is disabled when a system switching request is sent, the error code corresponding to a cause of the system switching failure is returned.

*5 When the systems are switched during an online change, a mismatch between the files is detected in the system consistency check and a stop error may occur on the new standby system. (Page 541 File)

*6 Although the control system CPU module is switched to the standby system CPU module, the existing standby system is not switched.

In separate mode

○: Switching possible, ×: Switching not possible

Redundant system status		Execution availability of system switching				
		Automatic system switching			Manual system switching*4	
		Power-off, reset, hardware failure of the CPU module	Stop error of the CPU module	System switching request from a network module	System switching request by the SP.CONTSW instruction*2	System switching request from the engineering tool*3
Normal operation or continuation error in the standby system		×	×	×	○	○
Tracking communications disabled (cable disconnection)		×	×	×	×	×
Power-off, reset, hardware failure of the standby system CPU module		×	×	×	×	×
Stop error in the standby system		×	×	×	×	×
Network error detected in the standby system*1		×	×	×	○	○
During memory copy from control system to standby system		×	×	×	×	×
During online change		×	×	×	○	○
Mismatch between the CPU module operating status of both systems		×	×	×	○	○
During system switching		×	×	×	×	×
During online module change	Redundant module	×	×	×	×	×
	Module on the main base unit in the redundant extension base unit configuration	×	×	×	×	×
	Other modules	×	×	×	○	○
System switching disabled by the DCONTSW instruction		×	×	×	×	×

*1 When the group specification has been set in the standby system, a network error is not detected if communication is available with the line of an Ethernet-equipped module after a communication error has occurred on the other Ethernet-equipped module. (MELSEC iQ-R Ethernet User's Manual (Application))

*2 If system switching is failed when a system switching request is sent, a continuation error occurs and a cause of a switching failure is stored in SD1644 (Cause of system switching failure).

*3 If system switching is disabled when a system switching request is sent, the error code corresponding to a cause of the system switching failure is returned.

*4 If the system switching timing and the interrupt timing of the internal timer overlap in the redundant extension base unit configuration, the interrupt interval for the internal timer may be extended in the CPU module of the control system after system switching.

Occurrence of a cause of the system switching failure

When a cause of the system switching failure occurs, the BACKUP LED flashes in backup mode and the SEPARATE LED flashes in separate mode.

Point

The cause to flash the BACKUP LED or SEPARATE LED can be checked in SD1642 (BACKUP/SEPARATE LED flashing cause). Check SD1642 and eliminate the cause to flash the LED.

Check method of system switching information

The following table lists the check methods of system switching information at system switching (automatic system switching and manual system switching).

Check method	Information	Reference
Event history	System switching result, system switching cause, and control system/standby system transition	Page 508 Event history
Special relay (SM)/Special register (SD)	System switching result and detailed information	Page 508 Special relay/Special register
CTRL LED and SBY LED of the redundant function module	System switching result	Page 509 CTRL LED and SBY LED of the redundant function module

When the systems have been switched, check the switching cause or detailed information and take corrective action to restore the system to a normal state as required.

Event history

The information related to system switching results, system switching cause, and control system/standby system transitions can be checked with the event history of the engineering tool.

When the systems are switched, the following items are stored in the event history of both systems.

- Automatic system switching: "System switching (system)" (event code: 00F00)
- Manual system switching: "System switching (user)" (event code: 2B000)

When the systems are switched by the SP.CONTSW instruction, the system switching instruction ID number specified with the SP.CONTSW instruction is also stored.

Special relay/Special register

System switching results and detailed information can be checked with the special relay and special register.

- Whether the systems have been switched or not can be checked by checking SD1649 (System switching Cause (when the systems are successfully switched)). When the systems have been switched, the switching cause is stored in SD1649 of the control system and standby system.
- If the systems have not been switched even after a switching cause is generated, the cause why the systems have not been switched can be checked by checking SD1644 (Cause of system switching failure). The switching cause is stored in SD1643 (System switching cause) of the control system.

For the values stored in the special relay and special register, refer to the following.

- Special relay (☞ Page 961 Redundant function)
- Special register (☞ Page 1012 Redundant function)

■Special relay

The following table shows the special relay for system switching and the storage status of the CPU modules in the control system and standby system.

○: Stored, ×: Not stored

SM number	Name	Storage status at system switching	
		New control system CPU module	New standby system CPU module
SM1637	System switching detection (standby system to control system)	○	×
SM1643	System switching check flag (standby system to control system)	○	×
SM1644	System switching check flag (control system to standby system)	×	○
SM1645	System switching request from a network module	○	○
SM1646	System switching by a user	×	×

■Special register

The following table shows the special register for system switching and the storage status of the CPU modules in the control system and standby system.

○: Stored, ×: Not stored

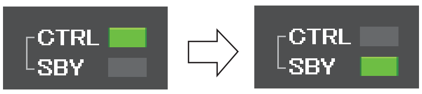
SD number	Name	Storage status at system switching	
		New control system CPU module	New standby system CPU module
SD1642	BACKUP/SEPARATE LED flashing cause	×	×
SD1643	System switching cause	×	○
SD1644	Cause of system switching failure*1	×	×
SD1645	System switching request status from a network module of own system	○	○
SD1646	System switching request status from a network module of the other system	○	○
SD1648	Cause of the other system monitoring error	○	○
SD1649	System switching Cause (when the systems are successfully switched)	○	○
SD1650	System switching instruction ID number	○	○

*1 When the system switching has not been normally completed, a value is stored in this special register area of the control system.

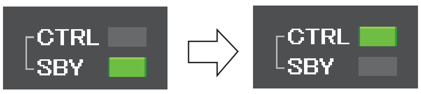
CTRL LED and SBY LED of the redundant function module

System switching results can be checked with the CTRL LED and SBY LED on the front of the redundant function module.

Switching from the standby system to the control system



Switching from the control system to the standby system



Precautions

The following describes the precautions on system switching.

Item	Description	Reference
Error in the redundant function module	• When an error has been detected on the redundant function module, the control system and standby system continue operating without being switched. • When a communication error has been detected in the communication between the redundant function module and a CPU module, the systems are switched.	Page 511 Error in the redundant function module
When both systems operate as standby systems	• If a communication error is generated due to a tracking cable error during system switching, both systems may operate as standby systems. • If a WDT error or a hardware failure of the CPU module has occurred in the control system while a stop error exists in the standby system, both systems operate as standby systems.	Page 511 When both systems operate as standby systems
When both systems operate as control systems	If the tracking cable and the network cable that is connected to the network module (control system or standby system) are simultaneously pulled out or disconnected, both systems may operate as control systems. In addition, if the tracking cable and the extension cable connected to the main base unit of the control system are simultaneously pulled out or disconnected (if double failures occur in one system), both systems may operate as control systems due to system switching. In this case, connect the tracking cable again. After a stop error has occurred in the system B, power off and on or reset the system B.	—
Scan time monitoring function	During system switching, scan time monitoring with the watchdog timer is interrupted. Thus, in a scan in which the systems are switched, no error is detected even if the scan time monitoring time has elapsed. A time taken for system switching does not need to be considered in the scan time monitoring time (WDT) setting.	—
System switching disabled by online change	In a redundant system, the manual system switching is disabled during an online change.	Page 511 System switching disabled by online change
Operation of an event execution type program	When "ON of bit data (TRUE)" has been specified in the trigger type and the systems are switched before tracking transfer of device data, an event execution type program does not operate in the new control system.	—
When the cable for the network module is disconnected	If a network module cable is disconnected, the systems may not be switched depending on the timing of error detection on the control system and the timing on the standby system.	Page 512 When the cable for the network module is disconnected
System switching during initial processing/ initial processing (when switched to RUN)	• When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the control system, the systems are switched after the initial processing or initial processing (when switched to RUN).*1 • When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the standby system, whether the systems are switched or not depends on the system switching cause.	Page 512 System switching during initial processing/initial processing (when switched to RUN)
CC-Link control at system switching	When the master operating station is switched with a program by using the CC-Link standby master function, the CC-Link control can be continued even after system switching.	MELSEC iQ-R CC-Link System Master/Local Module User's Manual (Application)

*1 When the both systems are simultaneously turned on in the redundant extension base unit configuration, if a stop error occurs during initial processing of the control system, the standby system also detects the stop error, and the system switching fails.

Error in the redundant function module

- When an error has been detected on a redundant function module, a continuation error occurs on the CPU module, and the control system and standby system continue operating without being switched. Check the error code, and perform online module change if the redundant function module has failed. (MELSEC iQ-R Online Module Change Manual)
- When the redundant function module was removed from the base unit or a failure has occurred in the base unit and a communication error has been detected in the communication between the redundant function module and a CPU module, a stop error occurs on the CPU module and the systems are switched. In this case, check the error code and eliminate the error cause.

Restriction

When the CC-Link IE Field Network module or the extension base unit for the redundant system has not been mounted and a communication error has been detected in the communications between the redundant function module and the CPU module, the systems may not be switched.
To switch the systems even though the redundant function module has been removed from the base unit or a failure has occurred in the main base unit, mount the CC-Link IE Field Network module or the extension base unit for the redundant system.

29

When both systems operate as standby systems

- If a communication error is generated due to a tracking cable error during system switching, both systems may operate as standby systems. When an error occurs with the tracking cable, each L ERR LED of the redundant function modules turns on. In this case, replace the tracking cable with a new one as soon as possible. If both systems operate standby systems, connect a new tracking cable properly and turn off and on or reset the CPU module of one system so that the other system will operate as the control system.
- If a WDT error or a hardware failure of the CPU module has occurred in the control system while a stop error exists in the standby system, the control system is switched to the standby system and the both systems operate as standby systems. In this case, eliminate error causes, and then power off and on or reset the CPU modules in the both systems.

The following table lists the operations of when both systems operate as standby systems.

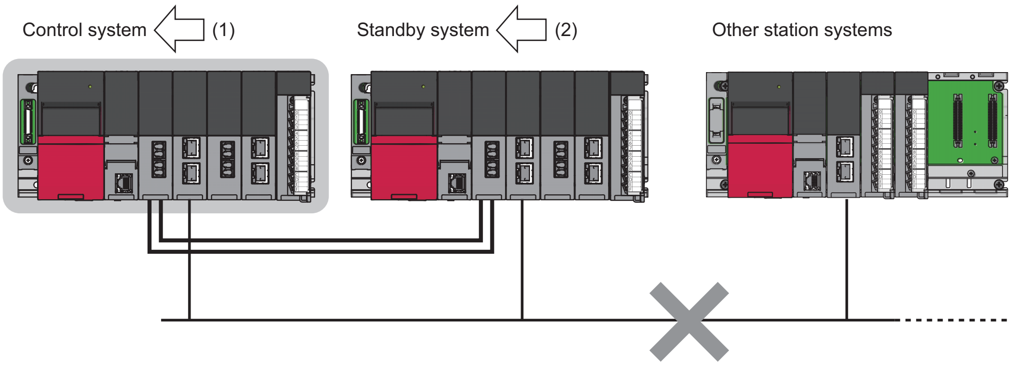
Item		Operation of both systems
LED of the redundant function module	BACKUP LED	Flashing (in backup mode)
	SEPARATE LED	Flashing (in separate mode)
	CTRL LED	Off
	SBY LED	On
Special relay	SM1635 (Standby system judgment flag)	On
System switching request	System switching request from a network module	System switching disabled
	System switching request by the SP.CONTSW instruction	
	System switching request from the engineering tool	

System switching disabled by online change

In a redundant system, the manual system switching is disabled during an online change. To disable the manual system switching during an online change, the CPU module is set to the manual system switching disabled state before an online change starts. After the online change is completed, it is set to the manual system switching enabled state.
If a communication with the engineering tool is interrupted due to communication cable disconnection or other causes during an online change, the online change cannot be completed, and the CPU module remains in the manual system switching disabled state. When the CPU module is in the manual system switching disabled state, the systems cannot be switched by manual system switching or a system switching request from a network module.
If an online change has failed, refer to the following and take actions.
☞ Page 560 Action for when an online change has failed

When the cable for the network module is disconnected

If a network module cable is disconnected, the systems may not be switched depending on the timing of error detection on the control system and the timing on the standby system.



(1) When the control system detects cable disconnection first, the systems are switched.
(2) When the standby system detects cable disconnection first, the systems are not switched. At this time, a continuation error occurs on the control system CPU module and the BACKUP LED flashes since a cause of the system switching failure has been generated.

In both case, replace the network module cable with a new one and clear the network error.

System switching during initial processing/initial processing (when switched to RUN)

When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the control system, the systems are switched after the initial processing or initial processing (when switched to RUN). Even if the initial processing or initial processing (when switched to RUN) takes time and a communication error occurs, the systems are switched after the initial processing or initial processing (when switched to RUN).*1

When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the standby system, whether the systems are switched or not depends on the system switching cause.

System switching cause	Whether to switch the systems
Power-off, reset, hardware failure of the CPU module	The systems are switched after the initial processing or initial processing (when switched to RUN).
Stop error of the CPU module	
System switching request from a network module	The systems are not switched.
System switching request by the SP.CONTSW instruction	• In backup mode, the systems are not switched. • In separate mode, the systems will be switched after the initial processing or initial processing (when switched to RUN).
System switching request from the engineering tool	

*1 When the both systems are simultaneously turned on in the redundant extension base unit configuration, if a stop error occurs during initial processing of the control system, the standby system also detects the stop error, and the system switching fails.

Extension cable errors in the redundant extension base unit configuration

■When the extension cable is redundant

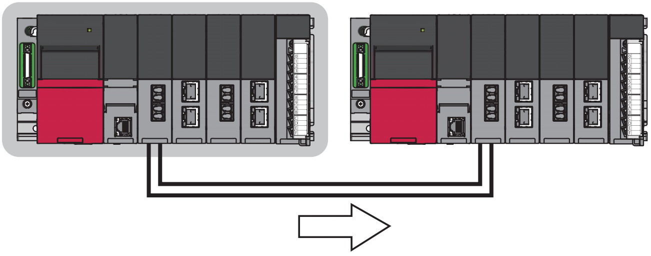
When the extension cable between the main base module of the control system and the extension base unit or the extension cable (on the active side) between extension base units fails or is disconnected, a bus access error occurs. After the systems are switched due to a stop error, control is continued by the new control system.

■When the extension cable is not redundant

- When the extension cable between the main base module of the control system and the extension base unit fails or is disconnected, a bus access error occurs. After the systems are switched due to a stop error, operation is continued by the new control system.
- When the extension cable between the extension base units fails or is disconnected, a bus access error occurs, and the systems are switched due to a stop error. A stop error also occurs in the new control system, and so control stops for both systems.

29.3 Tracking Transfer

This function transfers the control data from the control system to the standby system and maintains the consistency of the data in the two systems to continue operation of the redundant system when a failure or an error occurs in the control system.



- The control system CPU module sends tracking data to the standby system CPU module before the END processing. The tracking data includes device/label data, special relay (SM), special register (SD), PID control instruction information, and system data.
- The standby system CPU module receives the tracking data, and then reflects the received data on the device/label memory.
- Tracking transfer is performed in both of backup mode and separate mode.
- Device/label data can be divided and transferred as required. To transfer device/label data within a desired range, set a range of global devices to be transferred and whether to transfer local devices/global labels/local labels/module labels (extension base unit) for each tracking block and turn on the tracking triggers (SD1667 to SD1670) assigned for each tracking block.
- Up to 1M-word device/label data can be transferred in one scan.
- When a stop error has occurred, only system data is transferred. (☞ Page 517 System data)

Point

- In the CPU parameter, "Tracking Device/Label Setting" is set to "Transfer collectively" by default. Thus, tracking transfer is performed without setting the parameter. (☞ Page 522 Batch transfer)
- When a program file is converted with the process control extension setting enabled, "Tracking Device/Label Setting" is automatically set to "Detailed setting". In the tracking block No.64, the range of the file register specified in the system resource of the process control extension setting is registered. When the system is operating in backup mode, the tracking transfer trigger (bit 15 of SD1670) corresponding to the tracking block No.64 automatically turns on at every scan. (☞ Page 520 When the "Process Control Extension Setting" is enabled)

Restriction

When a tracking transfer is not performed, the possible causes are the following:

- The standby system CPU module has been powered off or reset.
- Hardware failure of the CPU module has occurred.*1
- An error has occurred in the redundant function module.*1
- A WDT error has occurred.*1
- Tracking cables has been pulled out or disconnected.

*1 When the tracking communication line properly works, "Link-up" (event code: 00100) may be stored in the event history.

Tracking data

The following table lists the tracking data that can be transferred from the control system to the standby system.

Item		Operation mode		Reference
		Backup mode[*3]	Separate mode[*4]	
Device data	User device	○	○ (Except step relay (S))	Page 515 Devices that can be specified
	Special relay	○ (Auto transfer)	×	Page 516 Special relay
	Special register	○ (Auto transfer)	×	Page 517 Special register
Global label[*2] (Except module labels)		○	○	Page 523 Detailed setting
Module label (main base unit)		×	×	
Module label (extension base unit)		○[*5*6]	○[*5*6]	
Local device		○[*1]	×	
Local label		○[*1]	×	
Signal flow memory		○[*1]	×	Page 521 Tracking transfer setting for the signal flow memory
SFC information		○ (Auto transfer)	×	—
PID control instruction information		○ (Auto transfer)	×	Page 517 PID control instruction information
CPU buffer memory		×	×	—
System data		○ (Auto transfer)	○ (Auto transfer)	Page 517 System data

*1 The data that is used in a program executed in both systems is not transferred. (☞ Page 544 Program Execution in Both Systems)
*2 Global labels with devices assigned are not transferred as global labels. They are transferred according to the tracking transfer settings of the assigned devices.
*3 During online change on both systems or on the control system only, only system data is transferred. During online change on the standby system only, data is transferred according to the table.
*4 During online change on the control system, system data and device data (global devices except step relay (S)) are transferred. During online change on the standby system, data is transferred according to the table.
*5 This applies to the module labels of modules on the extension base unit in the module label assignment area. (The module labels of modules on the main base unit are excluded.)
*6 When the module using the module label is set to "Empty" in the system parameters, tracking data is not transferred for the module label.

Devices that can be specified

The following table lists the data that can be specified for tracking transfer.

○: Specifiable, ×: Not specifiable, —: Not settable as a local device

Classification	Device name	Transfer	
		Global device	Local device
User device	Input (X)	○	—
	Output (Y)	○	—
	Internal relay (M)	○	○
	Link relay (B)	○	—
	Annunciator (F)	○	—
	Link special relay (SB)	○	—
	Edge relay (V)	○	○
	Step relay (S)	○	—
	Timer (T)	○	○
	Retentive timer (ST)	○	○
	Long timer (LT)	○	○
	Long retentive timer (LST)	○	○
	Counter (C)	○	○
	Long counter (LC)	○	○
	Data register (D)	○	○
	Link register (W)	○	—
	Link special register (SW)	○	—
	Latch relay (L)	○	—
System device	Function input (FX)	×	—
	Function output (FY)	×	—
	Function register (FD)	×	—
	Special relay (SM)	○*1	—
	Special register (SD)	○*1	—
Index register	Index register (Z)	○	○
	Long index register (LZ)	○	○
File register	File register (R)	×	—
	File register (ZR)	○	—
Refresh data register	Refresh data register (RD)	○	—

*1 The data is automatically transferred regardless of parameter settings. (☞ Page 516 Auto tracking data)

Point

When using the SFC program, set all points of step relay (S) within the device transfer range.

29

Auto tracking data

The following tables list the data that is automatically transferred by the system regardless of parameter settings of tracking transfer.

■Special relay

The following table lists the special relay areas that are automatically transferred by the system.

SM number	Name
SM315	Service processing constant wait setting flag
SM321	Start/stop SFC program
SM322	SFC program start status
SM323	Presence/absence of continuous transition for entire block
SM325	Output mode at block stop
SM326	SFC device/label clear mode
SM327	Output mode at execution of the end step
SM328	Clear processing mode when the sequence reaches the end step
SM752	Dedicated instruction End bit control flag
SM754	BIN/DBIN instruction error control flag
SM755	Scaling data check settings
SM756	Module access completion wait control flag
SM775	Selection of link refresh processing during the COM instruction execution
SM776	Local device setting at CALL
SM777	Local device setting in interrupt programs
SM792	PID bumpless processing (for the complete differentiation PIDCONT instruction)
SM794	PID bumpless processing (for the inexact differential S.PIDCONT instruction)
SM816	Hold mode (S.IN instruction)
SM817	Hold mode (S.OUT instruction)
SM960	Upper limit setting flag for the number of CPU module backup data
SM1646	System switching by a user
SM1762	Operation setting for access from the standby system to the extension base unit

For details on the special relay, refer to the following.

☞ Page 930 List of Special Relay Areas

■Special register

The following table lists the special register areas that are automatically transferred by the system.

SD number	Name
SD49	Error detection invalidation setting
SD250	Loaded maximum I/O
SD315	Service processing constant wait status setting
SD414	2n second clock setting
SD415	2n ms clock setting
SD771	Specification of the number of write instruction executions to data memory
SD775	Selection of refresh processing during the COM instruction execution
SD792, SD793	PID limit setting (for complete derivative)
SD794, SD795	PID limit setting (for incomplete derivative)
SD816, SD817	Basic period
SD818	Bumpless function availability setting for the S.PIDP instruction
SD819	Process value output type setting for the S.PHPL2 instruction
SD944	Backup function setting
SD947	Day and time setting for automatic backup (day)
SD948	Day and time setting for automatic backup (hour)
SD949	Day and time setting for automatic backup (minute)
SD950	Time and day of the week setting for automatic backup (hour)
SD951	Time and day of the week setting for automatic backup (minute)
SD952	Time and day of the week setting for automatic backup (day of the week)
SD954	Restoration target data setting
SD955	Restoration function setting
SD956, SD957	Restoration target date folder setting
SD958	Restoration target number folder setting
SD1353	Upper limit value setting for the number of CPU module backup data
SD1662	Tracking transfer data receive completion wait time
SD1667 to SD1670	Tracking transfer trigger

For details on the special register, refer to the following.

☞ Page 966 List of Special Register Areas

■PID control instruction information

The PID control data that is specified with the PIDINIT or S.PIDINIT instruction is transferred. (📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))

■SFC information

To keep executing the SFC program in the new control system, tracking transfer of the required data is executed.

■System data

The data related to the redundant system, such as system switching and operation mode change, is transferred.

Tracking block and tracking trigger

The devices or labels of a specified range is transferred by setting a range of devices or labels to be transferred for each tracking block and turning on the tracking trigger which is assigned for each tracking block.

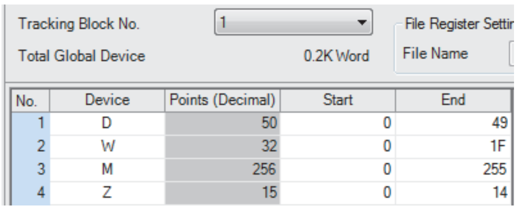
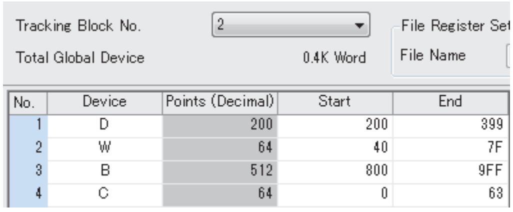
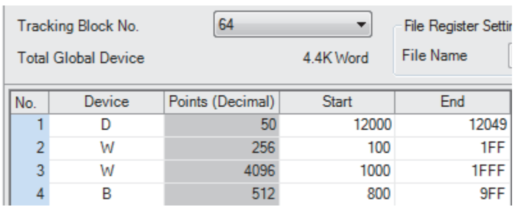
Tracking block

A tracking block is used for setting a tracking transfer range of global devices and whether or not to transfer local devices/global labels/local labels/module labels (extension base unit).

- Up to 64 blocks (No.1 to 64) are available.
- Up to 2048 devices can be set in a block. Up to 2048 devices can be transferred in a single tracking.
- The available capacity of tracking devices or labels is 1M words per block.

Ex.

When only the tracking block No.3 has no setting

<table>
<tr><th>Tracking block No.</th><th>Setting example</th></tr>
<tr><td>1</td><td>
</td></tr>
<tr><td>2</td><td>
</td></tr>
<tr><td>3</td><td>Not set</td></tr>
<tr><td>⋮</td><td>⋮</td></tr>
<tr><td>64</td><td>
</td></tr>
</table>

Point

Local devices, global labels, local labels, and module labels (extension base unit) are transferred only once in one scan. Even when the same local devices, global labels, local labels, and module labels (extension base unit) are specified in multiple tracking blocks, the overlap does not affect the data capacity to be transferred.

Tracking trigger

By turning on a tracking trigger, the devices or labels of a range specified in the corresponding tracking block are transferred. Bits used as tracking triggers change depending on the CPU parameter setting, as shown below.

■When "Tracking Device/Label Setting" is set to "Transfer collectively"

The bit 0 of SD1667 is used as a tracking trigger. The bit is automatically turned on by the system at an initial processing or operating status change (STOP to RUN), and the tracking transfer is started. To stop the tracking transfer, turn off the bit. To restart, turn on the bit.

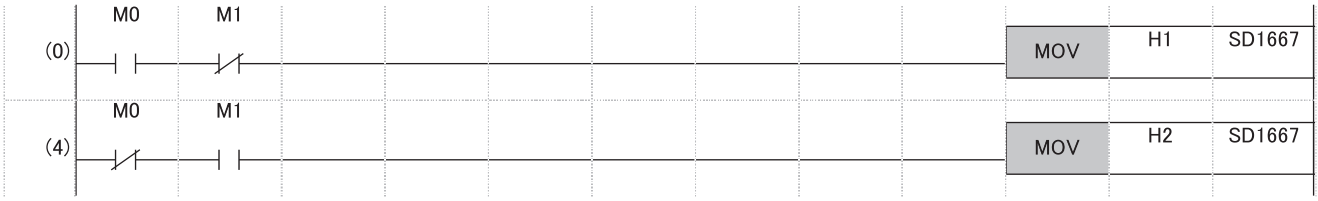
■When "Tracking Device/Label Setting" is set to "Detailed setting"

SD1667 to SD1670 (64 bits) are used as tracking triggers. The bit 0 of SD1667 to the bit 15 of SD1670 correspond to the tracking blocks No.1 to 64. To start the tracking transfer of a tracking block, turn on the corresponding bit. To stop the tracking, turn off the bit. To restart the tracking transfer, turn on the bit again.

When "Tracking Block No.1 Auto Transfer Setting" is set to "Transfer Automatically" in the CPU parameter, the bit 0 of SD1667 is automatically turned on by the system at initial processing or operating status change from STOP to RUN, and the tracking transfer is started.

Ex.

Program example for changing the status of tracking transfer triggers according to conditions



- When only M0 (Trigger switching condition 1) is on, the tracking transfer trigger of the tracking block No.1 is turned on. (Tracking blocks No.2 to No.64 are not transferred.)
- When only M1 (Trigger switching condition 2) is on, the tracking transfer trigger of the tracking block No.2 is turned on. (Tracking blocks No.1 and No.3 to No.64 are not transferred.)

Setting procedure for tracking transfer

The following describes the setting procedure for tracking transfer.

1. Set "Tracking Setting" in the CPU parameter. (☞ Page 520 Tracking transfer setting)
2. Add a program that controls tracking triggers. (☞ Page 519 Tracking trigger)
3. Write the set parameters and program to the CPU module with the engineering tool.
4. Start up the system again.
5. Turn on the tracking trigger with the program to start the tracking transfer.

When "Tracking Device/Label Setting" is set to "Transfer collectively", the above steps 2 and 5 are not required.

Point

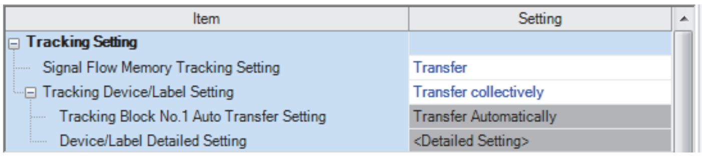
To continue the same control as before even after system switching, configure the settings in "Tracking Setting" so that all data necessary for the program is transferred.

Tracking transfer setting

The following describes the CPU parameters related to tracking transfer.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Tracking Setting]

Window



Displayed items

Item	Description	Setting range	Default
Signal Flow Memory Tracking Setting	Sets whether to transfer the signal flow memory or not. (☞ Page 521 Tracking transfer setting for the signal flow memory)	• Do Not Transfer • Transfer	Transfer
Tracking Device/Label Setting	Sets "Transfer collectively" or "Detailed setting" for devices or labels to be transferred. • When "Transfer collectively" is set, global devices, local devices, global labels, module labels (extension base unit), and local labels are assigned to the tracking block No.1 and automatically transferred. (☞ Page 522 Batch transfer) • When "Detailed setting" is set, detailed settings can be configured in "Tracking Block No.1 Auto Transfer Setting" and "Device/Label Detailed Setting".	• Transfer collectively • Detailed setting	Transfer collectively
Tracking Block No.1 Auto Transfer Setting	Sets whether to automatically transfer the tracking block No.1 or not. (☞ Page 518 Tracking block and tracking trigger)	• Do Not Transfer Automatically • Transfer Automatically	Transfer Automatically
Device/Label Detailed Setting	Sets devices and labels to be transferred. (☞ Page 523 Detailed setting)	—	—

Point

Set "Detailed setting" of "Tracking Device/Label Setting" for the following purposes.
- To shorten a tracking transfer time
- To add tracking data, such as the file register (ZR)
- To transfer each data set individually

When the "Process Control Extension Setting" is enabled

Settings and operations are as follows:
- When a program file is converted, "Tracking Device/Label Setting" is automatically set to "Detailed setting".
- In the tracking block No.64, the range of the file register specified in the system resource of the process control extension setting is registered. Because the other devices are not automatically registered, specify the other tracking block numbers for those devices. The devices and their ranges set in "Device/Label Memory Area Setting" can be imported at a time by clicking the [Device Setting Reflection] button in the "Global Device Setting" window. (☞ Page 524 Global device setting)
- To transfer the tracking block No.64, the bit 15 of SD1670 automatically turns on at every scan when the system is operating in backup mode. Do not turn off the bit 15 of SD1670.

Tracking transfer setting for the signal flow memory

By transferring the signal flow memory, operations of rising/falling instructions in the old control system are taken over to the new control system even after system switching.

Point

"Signal Flow Memory Tracking Setting" is set to "Transfer" by default. Transferring the signal flow memory is recommended. For the operation not to transfer it, refer to the following.

☞ Page 584 Instructions whose operations vary depending on tracking of the signal flow memory

■Tracking of individual POUs

The following table shows whether the signal flow memory is transferred or not for each POU.

○: Transferred, ×: Not transferred, —: No signal flow memory

POU			Both systems program executions setting	
			Control system execution	Both systems executions
Program block			○	×
Function block	Macro type		○	×
	Subroutine type	Global FB	○	○*1
		Local FB	○	×
Function			—	

*1 To prevent the signal flow memory in the standby system from being overwritten in a both systems execution program, use a local FB. When a global FB is used, the signal flow memory in the standby system is overwritten with the signal flow memory in the control system.

Tracking device/label setting

The following two methods are available for transferring devices and labels: automatically transferring all the devices and labels in a batch or transferring specified devices and labels of a specified tracking block.

■Batch transfer

When "Tracking Device/Label Setting" is set to "Transfer collectively", the following devices and labels are assigned to the tracking block No. 1 and automatically transferred.

Type	Description
Global device	■Bit device • Input (X) • Output (Y) • Internal relay (M) • Link relay (B) • Step relay (S) • Edge relay (V) • Latch relay (L) ■Word device • Timer (T) • Long timer (LT) • Retentive timer (ST) • Long retentive timer (LST) • Counter (C) • Long counter (LC) • Data register (D) • Link register (W) • Index register (Z) • Long index register (LZ)
Local device*1	All local devices
Global label*2	All the global labels assigned to the device/label memory
Module label (extension base unit)	The range that was set for the refresh target on the module label in the refresh settings parameter for each module is transferred.
Local label*1	All local labels

*1 The data used in both systems execution programs is not transferred.

*2 Global labels with devices assigned are not transferred as global labels. They are transferred according to the tracking settings of the assigned devices. To transfer global labels with devices assigned, specify the assigned global devices in the global device setting. (☞ Page 524 Global device setting)

Point

To transfer the annunciator (F), link special relay (SB), link special register (SW), file register (ZR), or refresh data register (RD), specify the corresponding data in "Device/Label Detailed Setting" of "Tracking Device/Label Setting". (☞ Page 523 Detailed setting)

Restriction

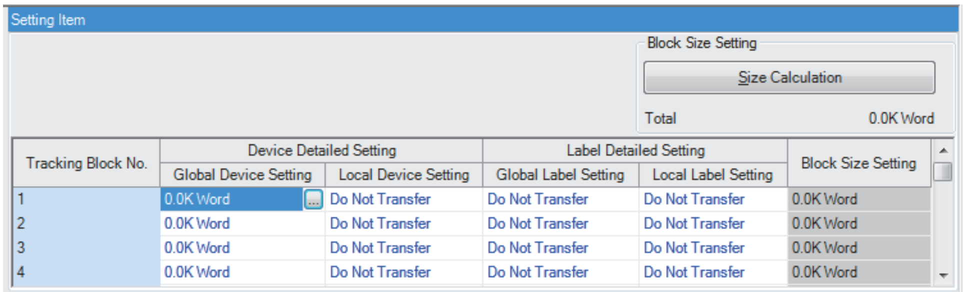
After setting "Transfer collectively", perform a test operation in the system design phase and check if the size of tracking data is 1M words or smaller. If the size of the tracking data is larger than 1M words, a stop error occurs when the CPU module is powered off and on or reset.

■Detailed setting

Set devices and labels to be transferred for each tracking block (No.1 to 64) to be used.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Tracking Setting] ⇨ [Device/Label Detailed Setting]

Window



Displayed items

Item	Item	Description	Setting range	Default
Device Detailed Setting	Global Device Setting	Sets global devices to be transferred. (☞ Page 524 Global device setting)	—	0.0K Word
	Local Device Setting	Sets whether to transfer local devices or not. When "Transfer" is set, all the local devices are transferred. For the devices that can be set as local devices, refer to the following. ☞ Page 515 Devices that can be specified	• Do Not Transfer • Transfer	Do Not Transfer
Label Detailed Setting	Global Label Setting*1	Sets whether to transfer global labels or not. When "Transfer" is set, all the global labels assigned to the device/label memory are transferred.	• Do Not Transfer • Transfer	Do Not Transfer
	Local Label Setting	Sets whether to transfer local labels or not. When "Transfer" is set, all the local labels are transferred.	• Do Not Transfer • Transfer	Do Not Transfer
	Module Label (Extension) Setting	Sets whether to transfer module labels of the modules on the extension base unit for each tracking block or not. When "Transfer" is set, the range that was set for the refresh target on the module label in the refresh settings parameter for each module is transferred.	• Do Not Transfer • Transfer	Do Not Transfer
Block Size Setting		Calculates "Block Size Setting" of each tracking block and "Total" of the setting capacity.	—	—

*1 Global labels with devices assigned are not transferred as global labels. They are transferred according to the tracking settings of the assigned devices.

Point

When setting devices and labels for tracking transfer, click the [Size Calculation] button to check if the tracking data capacity in one scan is equal to or less than 1M words. (☞ Page 529 Data capacity for tracking transfer).

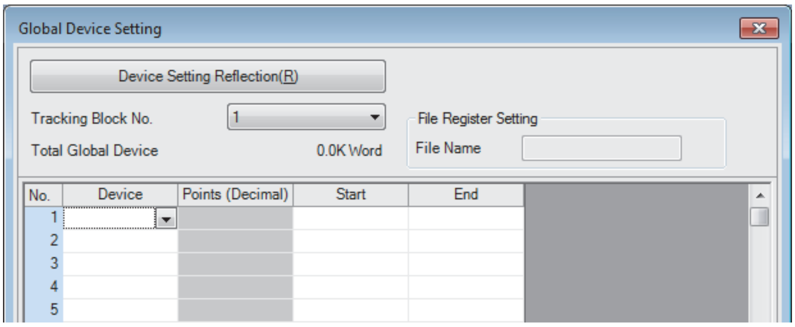
29

■Global device setting

Set devices and their ranges for each tracking block No.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Tracking Setting] ⇨ [Device/Label Detailed Setting] ⇨ [Global Device Setting]

Window



Displayed items

Item	Description
Device Setting Reflection	Reflects the device setting of "Device/Label Memory Area Setting" in the CPU parameter. (Except for the annunciator (F), link special relay (SB), and link special register (SW))
Tracking Block No.	Selects the number of a tracking block to be set.
File Register Setting	Enters a file name of the file register file. This item is valid only when the file register (ZR) is selected in the device name field.
Device	Selects a device to be transferred. (☞ Page 515 Devices that can be specified)
Start/End	Specifies a range of devices to be transferred.

Point

- To input devices and their ranges set in "Device/Label Memory Area Setting" in a batch, click the [Device Setting Reflection] button.
- When using the SFC program, set all points of step relay (S) within the device transfer range.

Tracking mode

The following two modes are available for tracking.

Item	Description
Synchronous tracking mode	Tracking data is always transferred to the standby system once every scan of the control system. During a tracking transfer from the control system to the standby system, the next scan does not start in the control system.
Asynchronous tracking mode	When a tracking transfer from the control system is to be performed and the previous tracking is still in progress, the tracking transfer from the control system is canceled and the previous tracking continues. The control system starts the next scan without waiting for notifications of data reception/reflection completion from the standby system.

The tracking mode depends on the operation mode and the CPU module operating status.

Operating status of standby system and control system		Operation mode	
Control system	Standby system	Backup mode	Separate mode
RUN	RUN	Synchronous tracking mode*1	Asynchronous tracking mode
	STOP, PAUSE	Asynchronous tracking mode	
STOP, PAUSE	RUN		
	STOP, PAUSE		

*1 When the RUN/STOP/RESET switch of the CPU module of each system is set to the RUN position and both systems are powered on, data is transferred in the asynchronous tracking mode at first. After the tracking data is reflected to the standby system, the mode is switched to the synchronous tracking mode.

In the following conditions, the tracking mode is the asynchronous tracking mode.

- During an online change
- When the operation mode is switched
- When the systems are switched
- When tracking communications disabled is detected

29

Effect on the scan time

The following describes the effect on the scan time depending on the tracking mode.

Point

For the calculation method for an increase in the scan time due to tracking transfer, refer to the following.

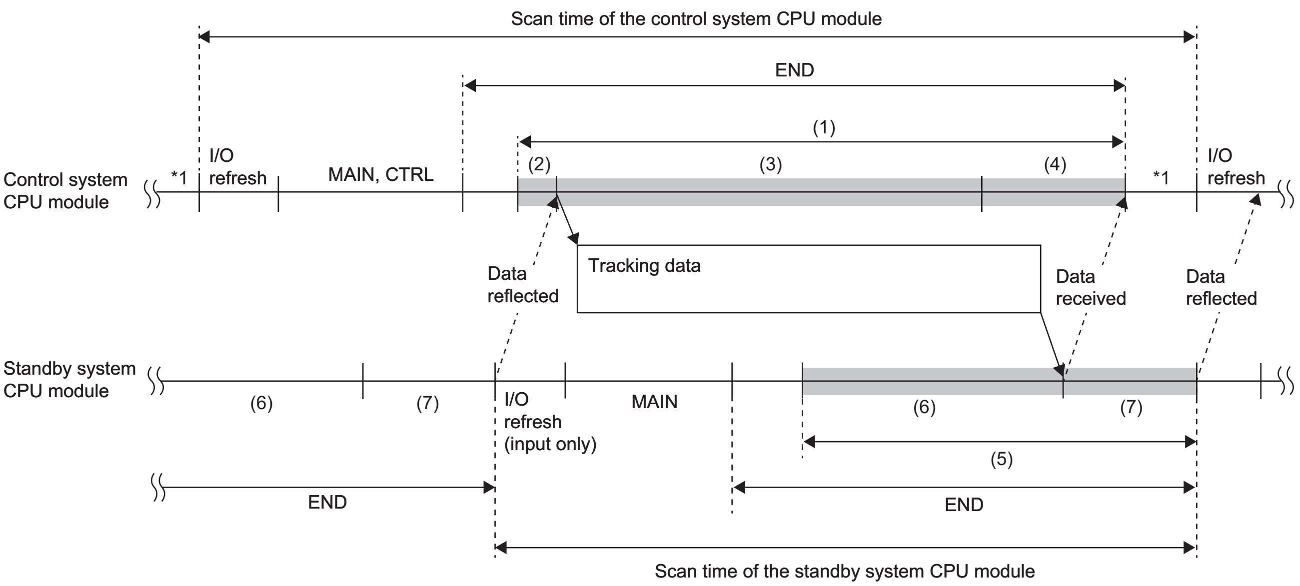
☞ Page 1078 Increase in the scan time due to tracking transfer

■Synchronous tracking mode

In the synchronous tracking mode, tracking transfer is always performed once every scan during the END processing. Until the standby system receives the tracking data sent from the control system, the next scan is not started. Thus, the scan time of the control system and standby system increases by the time for tracking data send/receive processing.

Ex.

CTRL: Program executed in the control system, MAIN: Program executed in both systems (in configuration with the main base unit only*2)



*1 If the constant scan is used, waiting time for the constant scan is generated.

*2 In the redundant extension base unit configuration, the processing order in the scan is different (☞ Page 459 Scan Configuration), but the operation is the same as in the configuration with the main base unit only.

In the control system, the scan time is extended by the following tracking send processing time.

Item		Description
(1) Tracking send processing	(2) Waiting for completion of tracking data reflection	After receiving a notification of the reflection completion from the standby system, the control system sends the tracking data.
	(3) Sending tracking data	The control system sends the tracking data.
	(4) Waiting for completion of tracking data reception	The control system waits for a notification of receive completion from the standby system. After receiving the notification of the receive completion from the standby system, the control system starts another END processing.

In the standby system, the scan time increases by the following tracking receive processing time.

Item		Description
(5) Tracking receive processing	(6) Waiting for tracking data reception	The standby system receives the tracking data from the control system. After receiving the tracking data, the standby system notifies the control system of the receive completion and reflects the tracking data.
	(7) Reflecting tracking data	The standby system reflects the tracking data. After the reflection completion, the standby system notifies the control system of the reflection completion and starts another END processing.

■Asynchronous tracking mode

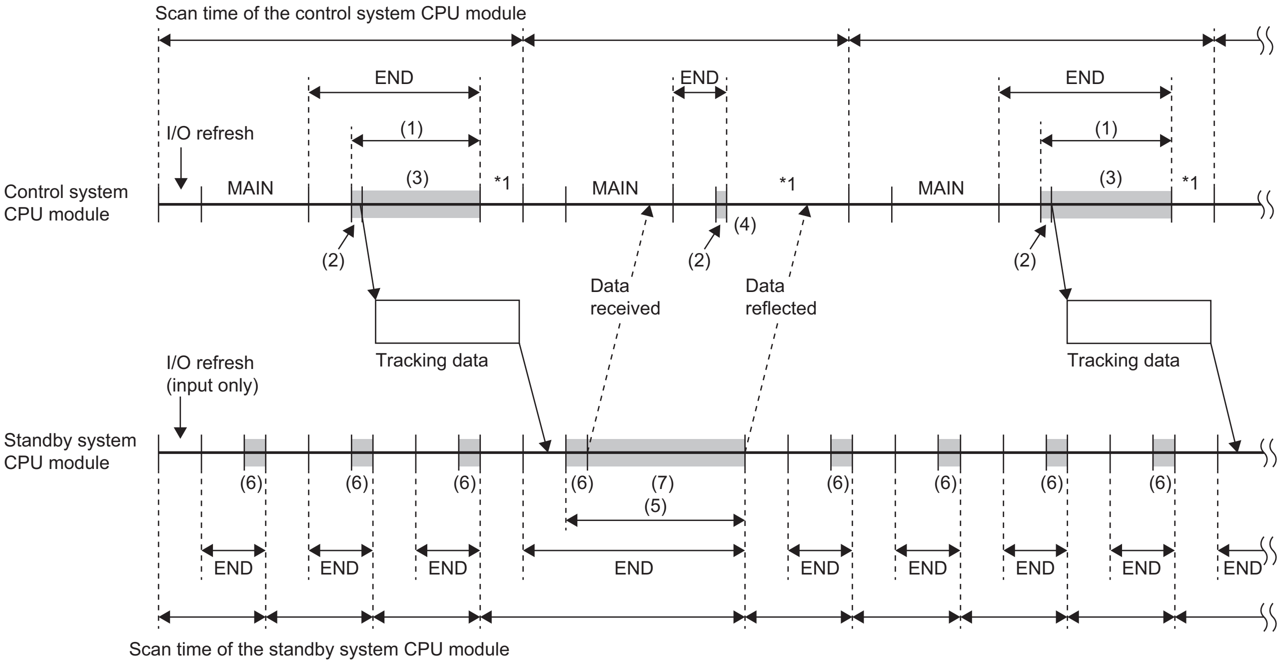
In the asynchronous tracking mode, the control system starts the next scan without waiting for notifications of data reception/reflection completion from the standby system.

Unlike the synchronous tracking mode, the scan time is not affected by waiting time for data reception/reflection completion.

When the standby system does not receive the tracking data from the control system, the standby system starts the next scan.

Ex.

When the control system CPU module is in the RUN state and the standby system CPU module is in the STOP state (in configuration with the main base unit only*2)



*1 If the constant scan is used, waiting time for the constant scan is generated.

*2 In the redundant extension base unit configuration, the processing order in the scan is different (☞ Page 459 Scan Configuration), but the operation is the same as in the configuration with the main base unit only.

In the control system, the scan time is extended by the following tracking send processing time.

Item		Description
(1) Tracking send processing	(2) Waiting for completion of tracking data reflection	After receiving a notification of the reflection completion from the standby system, the control system sends the tracking data. When a notification of reflection completion is not arrived (4), the control system does not send tracking data in the scan.
	(3) Sending tracking data	The control system sends the tracking data. After completing the send, the control system starts another END processing without waiting for a notification of receive completion from the standby system.

In the standby system, the scan time increases by the following tracking receive processing time.

Item		Description
(5) Tracking receive processing	(6) Waiting for tracking data reception	The standby system receives the tracking data from the control system. When the standby system does not receive the tracking data, the standby system starts the next scan. After receiving the tracking data, the standby system notifies the control system of the receive completion and reflects the tracking data.
	(7) Reflecting tracking data	The standby system reflects the tracking data. After the reflection completion, the standby system notifies the control system of the reflection completion and starts the next scan.

■When the mode is switched from the asynchronous tracking mode to the synchronous tracking mode

When the mode is switched from the asynchronous tracking mode to the synchronous tracking mode, the standby system receives the tracking data twice in one scan. Therefore, the scan time of the standby system is extended by the following time.

Scan time of the standby system: Standby system scan time × 2 + Control system scan time

Precautions

Operation at power-on

In the configuration with the main base unit only, when the RUN/STOP/RESET switch of the CPU module of each system is set to the RUN position and both systems are powered on, the control system CPU module starts in the STOP state and is switched to the RUN state after reflecting the tracking data is completed in the standby system CPU module. (The same operation is performed when one system is powered on while the other system is waiting for the start-up of the other system.) Since the operating status of the control system and that of the standby system CPU modules do not march until the control system is switched from the STOP state to RUN state, the BACKUP LED flashes.
When the time of the initial processing and that of the initial processing (when switched to RUN) of the standby system CPU module are longer than those of the control system CPU module, the control system CPU module may not be immediately switched to the RUN state after power-on.
In the redundant extension base unit configuration, the CPU module is switched to RUN after about three seconds from when one system is powered on, without waiting for the standby system tracking data to be reflected.

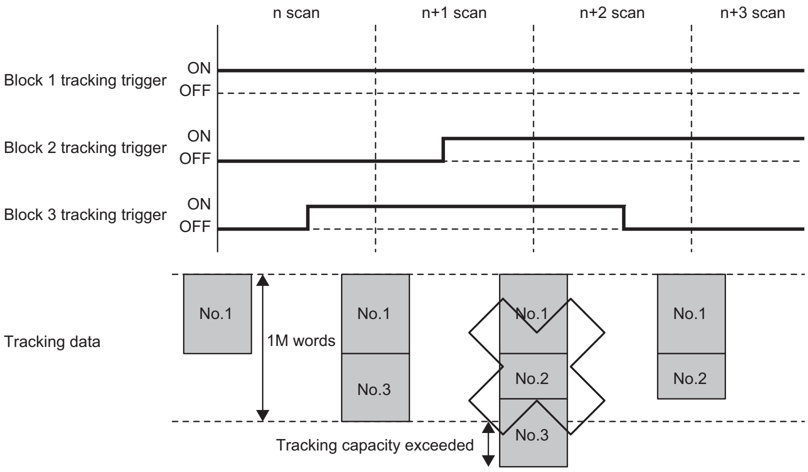
Device data used by the new control system CPU module

After system switching, the new control system CPU module starts operations by using the device/label data received in tracking transfer. The following table shows device/label data used by the new control system CPU module by tracking data receive status at system switching.

<table>
<tr><th>Device/label initial value</th><th>No tracking data received</th><th>Tracking data received once at least</th></tr>
<tr><td>Not set</td><td>Operation starts based on the latched device/label data.</td><td rowspan="2">Operation starts based on the device/label data that is transferred from the old control system.
When the synchronous tracking mode is used as a tracking mode, the data in the old control system at the final scan start is used by the new control system.</td></tr>
<tr><td>Set</td><td>Operation starts based on the device/label data that is set with device/label initial values.</td></tr>
</table>

Data capacity for tracking transfer

Up to 1M words of device/label data can be transferred in one scan. Set the tracking data capacity within 1M words.
If the data capacity exceeds 1M words, global devices, local devices, global labels, local labels, and module labels (extension base unit) are not transferred in the scan. In this case, check which tracking trigger turned on in the event history of the engineering tool and set the tracking data capacity within 1M words.



Even though the size of the tracking data to be transferred is 1M words or less, the size of the data may become larger than 1M words depending on the label type or data type to be used after the data or all the data is converted with the engineering tool. When configuring "Tracking Setting" with the CPU parameter, click the [Size Calculation] button in "Detailed setting" of "Device/Label Detailed Setting" to check if the size of the tracking data 1M words or less and transferred in one scan. (☞ Page 523 Detailed setting)
To reduce the size of the tracking data, consider the following.

- Exclude devices/labels that are not required to continue the system operation.
- Divide the tracking data into multiple blocks to transfer the data in multiple scans.

When data is different between the control system and the standby system

Store the same program, FB file, CPU parameter, and global label setting file in the control system CPU module and the standby system CPU module for tracking transfer. (☞ Page 530 Memory Copy from Control System to Standby System)
If there is any difference, only global devices (except step relay (S)), system data, and PID control instruction information are transferred.

When the communication load from an external device via the other system is high

When the communication load from an external device via the other system (such as MELSOFT connection and SLMP communications) is high, the load is imposed on the tracking transfer and an error may occur. In such case, review and correct the communication path so that communication from an external device does not go through the other system or reduce the communication load from the external device.

When the redundant function module restarts

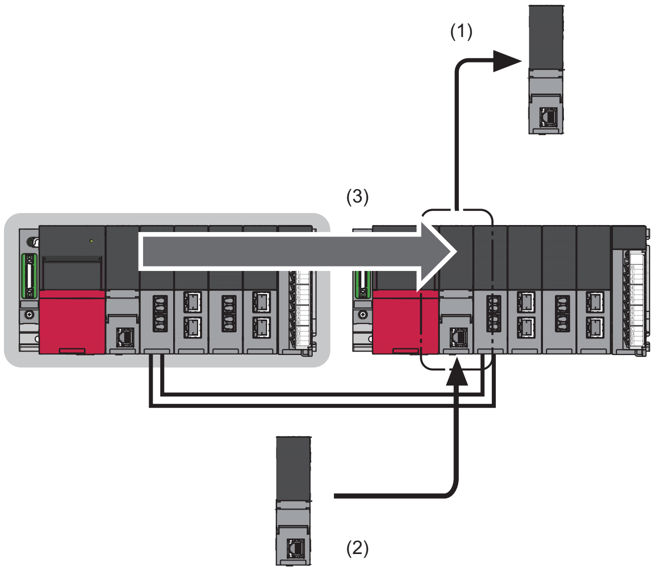
The redundant function module can be temporarily affected by noise, and it can cause an unintended restart of the redundant function module and an error occurs. In this case, clear the error. (☞ Page 145 Error Clear)
To check whether the redundant function has restarted, check the event history. (☞ Page 908 Event list)

29.4 Memory Copy from Control System to Standby System

This function transfers data such as parameters and programs in the CPU module of the control system to the CPU module of the standby system to maintain the consistency of the memory in the two CPU modules.

Ex.

Replacement of the standby system CPU module using the memory copy



(1) Remove the standby system CPU module.
(2) Mount a new CPU module.
(3) Execute the memory copy to match the memory contents of the control system CPU module and the standby system CPU module.

The following memories are available for the memory copy.

- Program memory
- Device/label memory (copying only the file area of file register)
- Data memory (copying only the files in the system folder ($MELPRJ$))
- SD memory card (copying only the files in the system folder ($MELPRJ$))
- System memory (copying only the system operation setting*1)*2

The memory copy is executed on memories that have differences between the control system and the standby system. It is not executed on memories without difference between the systems.

*1 Set in SM384 (System operation setting request) and SD384 (System operation setting).
*2 Check the version of the CPU module used. (☞ Page 1139 Added and Enhanced Functions)

Point

For the module replacement method and maintenance procedure using the memory copy, refer to the following.

☞ Page 596 Module Replacement in the Redundant System

Files copied by the memory copy function

The following table lists the files to be copied by the memory copy function.

○: Memory copy possible, ×: Memory copy not possible, —: Storage not possible

File type		Copy			
		CPU built-in memory			SD memory card
		Program memory	Device/label memory	Data memory	
Program		○	—	○	○
FB file		○	—	○	○
CPU parameter		—	—	○	○
System parameter		—	—	○	○
Module parameter		—	—	○	○
Module extension parameter		—	—	○	○
Module-specific backup parameter		—	—	×	×
Memory card parameter		—	—	—	○
Device comment		—	—	○	○
Initial device value		—	—	○	○
Global label setting file		—	—	○	○
Initial label value file	Initial global label value file	—	—	○	○
	Initial local label value file	—	—	○	○
File register		—	○	—	×
Event history		—	—	×	×
Device data storage file		—	—	○	×
General-purpose data		—	—	×	×
Data logging setting file	Common setting file	—	—	—	○
	Individual setting file	—	—	○	○
Remote password		—	—	○	○

Point

When the memory copy is executed, only the files in the memory with differences between control system and standby system are copied. The event history is stored only when the memory copy is completed successfully. (☞ Page 537 Event history when memory copy is executed)

■When the security key authentication function is used

The security key of the CPU module is not copied to the standby system. Write the security key from the personal computer where the security key is registered again after completion of the memory copy.

If the memory copy is executed when the security key is written in the CPU module, program files and their security keys are copied but the security key of the CPU module is not copied. Starting up the CPU module without re-writing of the security key causes an error of the CPU module because the security keys do not match between those program files and the CPU module.

Point

If the security key of the CPU module is written in an extended SRAM cassette, the replaced CPU module can take over the security key by simply replacing the extended SRAM cassette at replacement of the CPU module. In this case, re-writing of the security key after completion of the memory copy is not required.

Execution method of memory copy

The following methods are available for memory copy.

Item	Description	Application
Automatic memory copy	Automatically executes the memory copy by the system. CPU parameter settings are required in advance.	Executing the memory copy without using an engineering tool or external devices (such as a GOT)
Memory copy with the engineering tool	Executes the memory copy with an online operation of the engineering tool that is connected to the standby system CPU module.	Using the engineering tool
Memory copy with the special relay and special register	Executes the memory copy with operations of the special relay and special register.	Executing the memory copy from external devices (such as a GOT)

Execution of memory copy

The memory copy can be executed regardless of the operating statuses (RUN/STOP/PAUSE, stop error) of the control system and standby system CPU modules.

The automatic memory copy can be executed only in backup mode.

Automatic memory copy

The following describes the execution procedure of automatic memory copy.
To execute the automatic memory copy, turn off and on or reset the standby system CPU module while the control system is operating.

Point

- Set the automatic memory copy in the CPU parameter in advance. Therefore, consider the use of automatic memory copy when determining the maintenance policy in the system design phase.
- The automatic memory copy is executed when the parameter setting has been configured for the control system CPU module. Even if the parameter setting has not been configured for the standby system CPU module, the automatic memory copy is executed when the parameter setting has been configured for the control system CPU module.

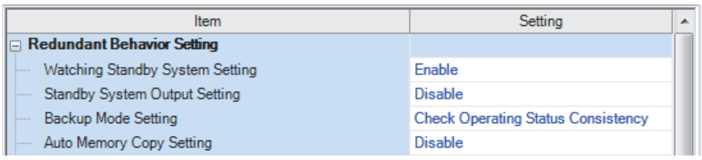
Parameter

29

To execute the automatic memory copy, set "Auto Memory Copy Setting" to "Enable" in the CPU parameter.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Redundant Behavior Setting]

Window



Displayed items

Item	Description	Setting range	Default
Auto Memory Copy Setting	Sets whether to enable the automatic memory copy function or not.	• Disable • Enable	Disable

Restriction

- When the control system power supply and the standby system power supply are simultaneously turned on, the automatic memory copy is not executed.
- The standby system CPU module does not operate in boot operation using an SD memory card, but operates based on the files copied by the automatic memory copy function. However, when the automatic memory copy cannot be executed due to a communication error or other causes at power-on or reset of the standby system CPU module, the standby system CPU module operates with the files transferred by the boot operation.

Execution procedure

1. When the operation mode is the separate mode, switch it to the backup mode. (☞ Page 496 Operation Mode Change)

2. Turn off and on or reset the standby system CPU module. The system executes the memory copy. During the memory copy, the MEMORY COPY LEDs of the redundant function modules in both systems flash (at 200ms intervals).

3. The standby system CPU module is automatically reset and restarted, and the memory copy is completed. The MEMORY COPY LEDs of both systems turn off.

Memory copy with the engineering tool

The following describes the execution procedure of memory copy using the engineering tool.

Execution procedure

1. Connect the engineering tool to the control system CPU module.

2. Open the "Redundant Operation" window of the engineering tool.

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

3. Select "Memory Copy" in the "Redundant Operation" window and click [Execute]. During the memory copy, the MEMORY COPY LEDs of the redundant function modules in both systems flash (at 200ms intervals).

4. When the memory copy is completed, the MEMORY COPY LED of the control system turns off and that of the standby system turns on.

5. Turn off and on or reset the standby system CPU module. The MEMORY COPY LED of the standby system turns off.

Memory copy with the special relay and special register

The following describes the execution procedure of memory copy using the special relay and special register.

Special relay to be used

The following table lists the special relay areas used in memory copy.

SM number	Name
SM1653	Memory copy start
SM1654	Memory copy being executed
SM1655	Memory copy completion
SM1656	Auto memory copy

For details on the special relay (SM), refer to the following.

☞ Page 930 List of Special Relay Areas

Special register to be used

The following table lists the special register used in memory copy.

SD number	Name
SD988*1	Memory copy completion status (latch)
SD1653	Memory copy destination I/O number
SD1654	Memory copy completion status

*1 Even after the CPU module is turned off and on or reset, the result of the memory copy executed just before this reset operation can be checked since this register area is a latch area.

For details on the special register (SD), refer to the following.

☞ Page 966 List of Special Register Areas

Execution procedure

To execute the memory copy with the special relay and special register, use the special relay and special register of the standby system CPU module.

1. Check that SM1654 (Memory copy being executed) and SM1655 (Memory copy completion) are off.

- When SM1654 is on, the memory copy is being executed. Another memory copy cannot be started until the memory copy is completed.
- When SM1655 is on, the memory copy with the special relay and special register cannot be started. Execute the memory copy again after turning off SM1655.

2. Store 03D1H (Standby system CPU module) in SD1653 (Memory copy destination I/O number).

3. Turn on SM1653 (Memory copy start) to execute the memory copy. During the memory copy, the MEMORY COPY LEDs of the redundant function modules in both systems flash (at 200ms intervals).

Control system

Standby system

4. When the memory copy is completed, the MEMORY COPY LED of the control system turns off and that of the standby system turns on.

Control system

Standby system

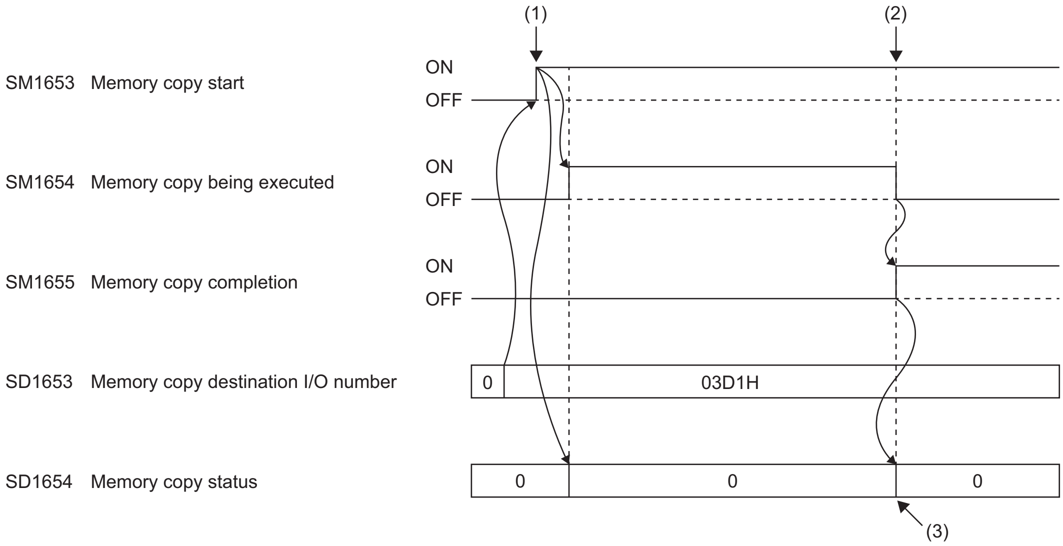
5. Turn off and on or reset the standby system CPU module. The MEMORY COPY LED of the standby system turns off.

Control system

Standby system

Operation of the special relay and special register

The following chart shows the operation of the special relay and special register in memory copy.



(1) The memory copy is started.
(2) The memory copy is completed.
(3) When the memory copy is successfully completed, 0 is stored. When it is completed with an error, an error code is stored.

Precautions

The following describes precautions on the memory copy function.

Restrictions

■Restrictions on memory copy execution

Do not execute the memory copy in the following cases.

- The standby system CPU module is off or being reset.
- During initial processing (when switched to RUN)
- The tracking cable has an error or is pulled out.
- An error occurs on a redundant function module.
- A system configuration in which communications via a redundant function module are unavailable is used.
- Different models of CPU module are used in the control system and standby system.
- Online operation from the engineering tool
- The systems are being switched.
- An online module change is being performed (for replacing a redundant function module or modules of the standby system).
- During backup of CPU module (CPU module data backup/restoration function)
- In separate mode (for the automatic memory copy)
- The memory copy has been executed on the standby system.
- The memory copy is being executed.
- A value other than 03D1H is stored in SD1653 (for the memory copy with the special relay and special register).
- The memory copy has been executed on the standby system CPU module that does not support the program restoration information write selection when the program restoration information is not written to the control system CPU module.

■Restrictions during memory copy execution

Do not execute the following functions during memory copy execution.

- Online operation from the engineering tool
- Operation of the RUN/STOP/RESET switch
- Removal of the SD memory card
- Online module change
- Backup of CPU module (CPU module data backup/restoration function)
- Powering off or resetting the CPU modules
- Removing tracking cables

Whether to delete files before memory copy

The standby system CPU module may detect a stop error because the files are deleted from the standby system CPU module memory to which the data is copied when copying memory. When a stop error has already occurred in the standby system, no error is detected. The files in the memory to execute memory copy for are deleted, and the files in the memory not to execute memory copy for are not deleted.
When the firmware version of the CPU module is "20" or later, even if the files before copying are deleted, the files in or lower than the system folder ($MELPRJ$) which is not applicable to memory copy is not cleared.

Event history when memory copy is executed

Back up the event history before memory copy. The files before memory copy are deleted when executing memory copy, so the files in the memory including the event history may be deleted and the event history before memory copy may be cleared.[*1]

*1 If the firmware version of the CPU module of the standby system is "20" or later, the event history before memory copy is not cleared.

Errors during memory copy

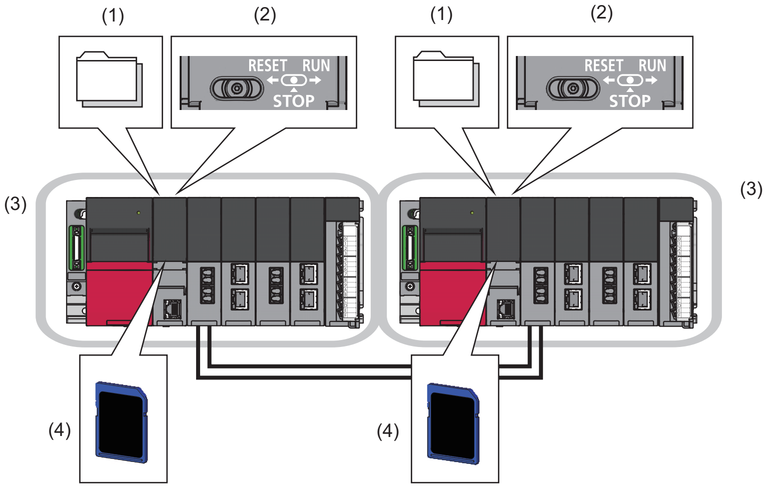
When the memory copy is completed with an error, the MEMORY COPY LED of the control system turns off and that of the standby system flashes (at 1s intervals). In this case, the memory copy has not been normally executed on the standby system CPU module.
After checking the error code of memory copy stored in SD1654 (Memory copy completion status) and eliminating the error cause, execute the memory copy again. (☞ Page 966 List of Special Register Areas)

When labels accessible from external devices are used

When executing the memory copy, pay attention to the firmware version of the CPU modules used.
Depending on the firmware version, an error may occur in the CPU module of the standby system.
In this case, the following operations enable external devices to access the labels: Read parameters and programs including the global label setting file from the CPU module of the standby system, write them to the CPU module, and restart the CPU module of the standby system only.

29.5 System Consistency Check

This function checks whether the system configurations and files in the CPU modules are the same between the control system and the standby system when the redundant system is in backup mode.



The following table lists the items to be checked in the system consistency check.

No.	Item	Description	Reference
(1)	File	Parameter file, program file, and other files	Page 541 File
(2)	Operating status	Operating status of CPU module (RUN/STOP/PAUSE)	Page 542 Operating status
(3)	Mounting status of main base unit	CPU module and other modules mounted on the main base unit	Page 542 Mounting status of main base unit
(4)	SD memory card	Installing status of the SD memory card and status of the write protect switch	Page 543 SD memory card

Execution timing

The following table shows the execution timing of the system consistency check.

Item	Execution timing
File	• When both systems are simultaneously turned on or reset • When one system is turned on or reset while waiting for the start-up of the other system • When the standby system is turned on or reset while the control system is operating • When the operation mode is switched to the backup mode • When the operating status of the CPU module is switched from STOP to RUN • At END processing • At online change completion • At system switching • When a tracking cable is reconnected
Operating status*1	• When the standby system is turned on or reset while the control system is operating • When the operation mode is switched to the backup mode • When the operating status of the CPU module is switched from STOP to RUN • At END processing • When a tracking cable is reconnected
Mounting status of main base unit	• When both systems are simultaneously turned on or reset • When one system is turned on or reset while waiting for the start-up of the other system • When the standby system is turned on or reset while the control system is operating • At mode switching to the backup mode • At reconnection of tracking cable
SD memory card	• When both systems are simultaneously turned on or reset • When one system is turned on or reset while waiting for the start-up of the other system • When the standby system is turned on or reset while the control system is operating

*1 Checking this item can be disabled by setting "Backup Mode Setting" of "Redundant Behavior Setting". (☞ Page 551 Redundant System Operation Setting)

In the following conditions, the system consistency check is not performed.

- One of the systems has not started up. (The CPU module is powered off or is reset or during an initial processing.)
- A stop error has occurred in the CPU module.
- The operation mode is in separate mode.
- The memory copy is being executed.

File

Whether both systems have the same files is checked.

The following table shows whether or not to perform the check on each file type.

○: Checked, ×: Not checked, —: Storage not possible

File type		Check target memory	
		Built-in memory of CPU module	**SD memory card*3**
Program*1		○	×
FB file*1		○	×
CPU parameter		○	×
System parameter		○	×
Module parameter		○	×
Module extension parameter*4		○	○
Module-specific backup parameter		×	×
Memory card parameter		—	○
Device comment		×	×
Initial device value		○	○
Global label setting file		○	×
Initial label value file	Initial global label value file	○	○
	Initial local label value file	○	○
File register file*2		○	×
Event history		×	×
Device data storage file		×	×
General-purpose data		×	×
Data logging setting file	Common setting file	—	×
	Individual setting file	×	×
Remote password		○	×

*1 The reserved area for online change and program restoration information write status are also checked. (☞ Page 116 Configuration of a program file, Page 1101 Program Restoration Information Write Selection)
The SFC information device setting and the operation mode when an active block is activated ("Act at Block Multi-Activated") are also checked. (📖 MELSEC iQ-R Programming Manual (Program Design))

*2 Whether a file is stored or not is checked. The file contents are not checked.

*3 The check is not performed at the point when SD memory cards are inserted during operation.

*4 The check is performed only for modules on the main base unit. (The system consistency check is not performed for the module extension parameter for modules on the extension base unit, because the parameter cannot be stored in the CPU module.)

Operation of when a mismatch is detected

If a mismatch between the files is detected, a stop error occurs on the standby system CPU module.

Set the CPU modules of both systems to have the same files with either of the following methods.

- Writing files to both systems
- Executing the memory copy from the control system to the standby system for storing the same files in both systems (☞ Page 530 Memory Copy from Control System to Standby System)

Operating status

Whether the CPU modules of the control system and standby system are in the same operating status (RUN/STOP/PAUSE) is checked.

Operation of when a mismatch is detected

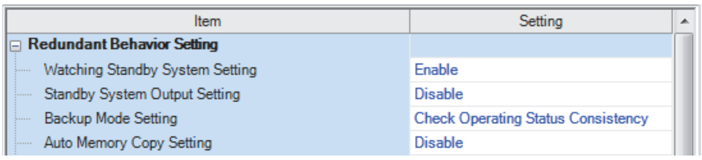
If a mismatch between the operating statuses is detected, a continuation error occurs on the standby system CPU module. The BACKUP LEDs of the redundant function modules of the control system and standby system flash because a cause of the system switching failure has occurred.

Backup mode setting

To prevent a continuation error from being detected when the operating status is changed during operation, set the system consistency check so that the consistency check on the operating status is not performed.
With this setting, a continuation error is not detected even if the CPU modules of both systems are in different statuses.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Redundant Behavior Setting]

Window



Displayed items

Item	Description	Setting range	Default
Backup Mode Setting	Sets whether to check the operating status consistency between the two systems or not when the redundant system is in backup mode.	• Do Not Check Operating Status Consistency • Check Operating Status Consistency	Check Operating Status Consistency

Mounting status of main base unit

The system consistency check is performed for the following items.

- Models of CPU modules
- Models and types of modules mounted on each slot on the main base unit*1
- Firmware version supporting functions added and changed by the upgrade and its settings*2

*1 For a slot set to be empty in the module status setting under the I/O assignment setting for system parameters, no error occurs even if the module model names are not the same.

*2 For functions added and changed by the upgrade and supported firmware versions, refer to the following.
Page 1139 Added and Enhanced Functions

Operation of when a mismatch is detected

If a mismatch between the mounting statuses of the main base units is detected, a stop error occurs on the standby system CPU module. If a mismatch between the mounting statuses of the main base units is detected when the CPU modules of both systems are simultaneously turned on or reset, a stop error occurs on the control system CPU module as well. ("Simultaneously" here means that one CPU module is started up within three seconds after the other CPU module is started.)

Checking the redundant system configuration in the redundant extension base unit configuration

The redundant system configuration between both systems is checked in the following cases.

- At power-on or reset
- At tracking cable connection

Checking the module configuration between both systems

If at least one of the following conditions is different between both systems at power-on or reset, a stop error occurs.

- Main base units with the same number of slots are used in both systems.
- The mounting status of each slot of the main base units and the model names of the mounted modules are the same in both systems. However, for a slot set to be empty in the module status setting under the I/O assignment setting for system parameters, no error occurs even if the module model names are not the same.
- The I/O Assignment Settings for the system parameters are the same in both systems.

29

Point

- When both systems are simultaneously turned on or reset, a stop error occurs in both systems.
- When the standby system is turned on or reset while the control system is operating, a stop error occurs in the standby system. While the online module change process is performed in the control system, the mounting status of each slot is different, but a stop error does not occur.

Checking the extension cable connection between both systems

If the main base unit is not mounted on the same extension base unit for the redundant system, a stop error occurs at the following timing.

- At power-on or reset
- At tracking cable connection

Point

- When both systems are simultaneously turned on or reset, a stop error occurs in both systems.
- When the standby system is turned on or reset while the control system is operating, a stop error occurs in the standby system.
- When the tracking cable is connected in the condition where no tracking cable is connected, a stop error occurs in the standby system.

SD memory card

The system consistency check is performed to check the installation of the SD memory cards and the status of the write protect switch. The SD memory card type or capacity is not checked.

Point

The system checks if the SD memory card is inserted while the control system is running even in the case only the standby system is turned off and on or reset. When an SD memory card is used, removing it more than necessary is not recommended.

Operation of when a mismatch is detected

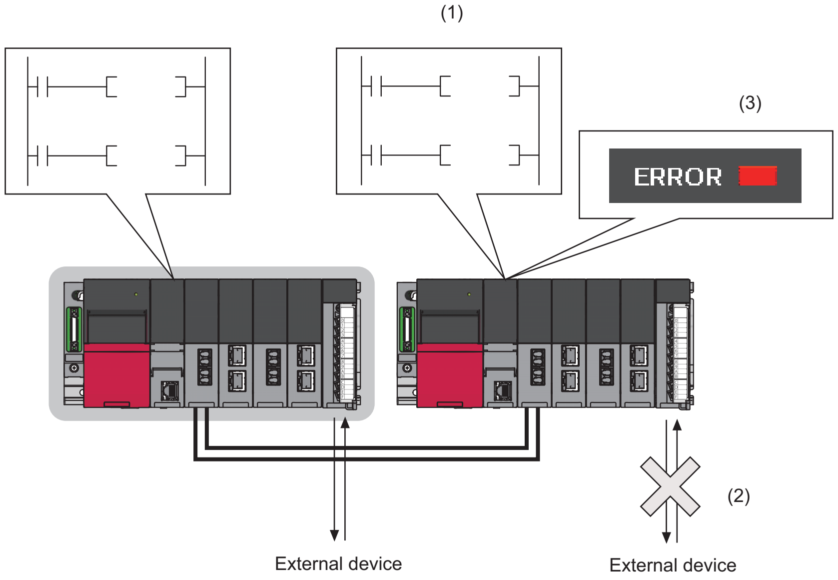
If a mismatch of the installation of the SD memory cards or the status of the write protect switch is detected, a stop error occurs on the standby system CPU module. If a mismatch between the mounting statuses of the main base units is detected when the CPU modules of both systems are simultaneously turned on or reset, a stop error occurs on the control system CPU module as well.

29.6 Program Execution in Both Systems

This function detects an error in the external device or network of the systems (control system and standby system) by executing a program that diagnoses external devices or networks of both systems.
The program set in "Both Systems Program Executions Setting" is executed on the CPU modules of both systems.

Ex.

When an error in the external device of the standby system is notified through a continuation error



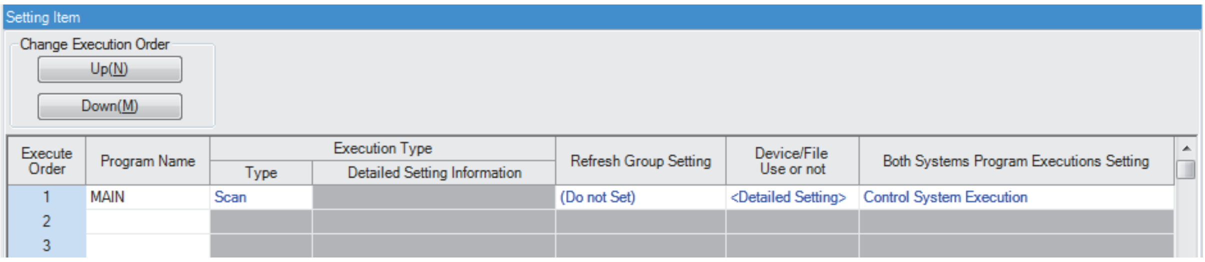
(1) The diagnostic program set as a both systems execution program is executed.
(2) An error is detected on the standby system external device by the diagnostic program.
(3) The PALERT instruction is executed, and the detailed information on a continuation error is displayed.

Setting of program execution in both systems

Configure the setting for each program to be executed on both systems.

[CPU Parameter] ⇨ [Program Setting]

Window



Displayed items

Item	Description	Setting range	Default
Both Systems Program Executions Setting	Sets whether to execute a program only in the CPU module of the control system or in the CPU modules of both systems. • "Both Systems Executions" can be set for programs whose execution type is the initial execution type, scan execution type, or standby type program. • Only "Control System Execution" can be set for programs whose execution type is the fixed scan execution type or event execution type.	• Control system execution • Both systems executions	Control System Execution

Point

To enable the output (Y) from a standby system external device using a program executed in both systems, configure settings in "Standby System Output Setting" of the CPU parameter. (☞ Page 551 Redundant System Operation Setting)

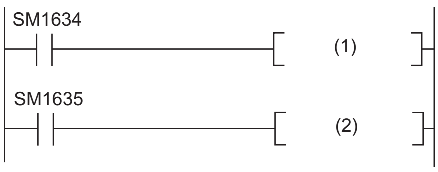
Operation of a program executed in both systems

The following describes the operation of a program executed in both systems.

Control system/ Standby system	Backup mode	Separate mode
Control system	A program is executed according to its execution type.	A program is executed according to its execution type regardless of the setting in "Both Systems Program Executions Setting".
Standby system	A program executed in both systems is executed according to its execution type. When an initial execution type program is set as a program executed in both systems, the program is executed in the first scan while the CPU module are in the RUN state. When an initial execution type program is not set as a program executed in both systems, a program executed in both systems of scan execution type is executed in the first scan while the CPU modules are in the RUN state.	

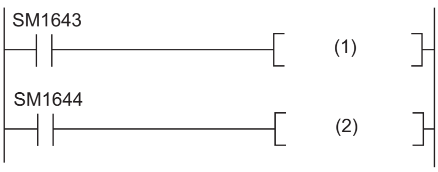
Point

- To perform different diagnostic processing in the control system and standby system by using a program executed in both systems, use the special relay. SM1634 (Control system judgment flag) turns on only in the control system and SM1635 (Standby system judgment flag) only in the standby system. The following shows a program example of when different processing is performed. For (1) and (2) in the following example, use the SET instruction or others to prevent duplication of a coil in the processing.



(1) Diagnostic processing in the control system, (2) Diagnostic processing in the standby system

- When a program executed in both systems is used, device/label values may differ between the control system and the standby system. If the systems are switched in this state, the program is started based on the different data, causing an unintended operation. The device/label values to be used can be initialized by using SM1643 (ON for only one scan after system switching (standby system to control system)) and SM1644 (ON for only one scan after system switching (control system to standby system)). For (1) and (2) in the following example, use the SET instruction or others to prevent duplication of a coil in the processing.



(1) Initialization processing (standby system → control system), (2) Initialization processing (control system → standby system)

Operation at system switching

The operation at system switching is different for a program executed in both systems. The following table shows the operation at system switching.

Item		New control system CPU module	New standby system CPU module
Program execution	Initial execution type program	When an initial execution type program has not been completed on the old control system at system switching, it is executed again from its head.	When an initial execution type program has not been completed on the old control system at system switching, it is not executed on the new standby system because system switching is performed after the program completion on the old control system.
	Scan execution type program	This type of program is executed from the step 0.	
Direct access input (DX)		In the program execution after system switching, data is fetched when an instruction using the direct access input (DX) is executed. However, no data is fetched for the direct input for the module mounted on the extension base unit.	
Direct access output (DY)		In the program execution after system switching, data is output when an instruction using the direct access output (DY) is executed. However, no data is output even when an instruction using the direct output of the module mounted on the extension base unit is executed.	
FROM/TO instructions		In the program execution after system switching, this instruction is executed when the instruction execution condition is satisfied. However, when the FROM/TO instructions are executed for the module mounted on the extension base unit and when the status of SM1762 (Operation setting for access from the standby system to the extension base unit) is set to OFF, a stop error occurs. When the SM1762 status is set to ON, the operation is handled as non-processing. (When "Continue" is selected in the operation error in the RAS setting for the CPU parameter, the error can be set as a continuation error.)	
Instruction that requires several scans		• In the program execution after system switching, this instruction is executed when the instruction execution condition is satisfied. • When the systems are switched while the instruction is being executed, the instruction execution continues and the completion device turns on at a completion of the instruction execution.	

For the operation of programs which are not set as both systems execution programs at system switching, refer to the following.

☞ Page 505 Operation at system switching

Precautions

The following lists the precautions for using a program executed in both systems.

<table>
<tr><th colspan="2">Item</th><th>Description</th><th>Reference</th></tr>
<tr><td colspan="2">Program execution time</td><td>Set a program execution time of the standby system to be shorter than that of the control system. In the redundant extension base unit configuration, set a program execution time of the standby system to be within 200ms.</td><td>Page 549 Program execution time</td></tr>
<tr><td colspan="2">Constant scan</td><td>The constant scan function is invalid for the standby system in backup mode.</td><td>—</td></tr>
<tr><td colspan="2">Time required for system switching</td><td>If system switching is attempted to be performed while a program executed in both systems is being executed, the system is switched after the END instruction is completed. Thus, a time required for system switching may be extended.</td><td>—</td></tr>
<tr><td rowspan="3">Program execution type</td><td>Execution type change</td><td>The execution type of a program executed in both systems is not taken over at system switching.
Even if the systems are switched after the execution type is changed by using a program control instruction (PSCAN(P), PSTOP(P), or POFF(P) instruction) on the old control system, the program is executed on the new control system with the execution type of the old standby system.</td><td>—</td></tr>
<tr><td>Initial execution type</td><td>When the systems are switched while an initial execution type program set as a both systems execution program is being executed, the initial execution type program is executed two times on the new control system CPU module.
The old standby system is switched to the new control system after the initial execution type program has been completed, and then the initial execution type program is executed again.</td><td>—</td></tr>
<tr><td>Event execution type</td><td>When the trigger type is set to "ON of bit data (TRUE)" and the module access device (Un\G) of the module on the extension base unit is set for the device in the redundant extension base unit configuration, the event execution program is not performed from the standby system in the separate mode. Therefore, be careful when performing maintenance on the system using separate mode.</td><td>—</td></tr>
<tr><td colspan="2">SFC program</td><td>The SFC program cannot be set to be executed in both systems.</td><td>—</td></tr>
<tr><td colspan="2">Interrupt disabled or enabled state</td><td>The control system and standby system have individual interrupt disabled or enabled state because the states are not transferred.</td><td>—</td></tr>
<tr><td colspan="2">Tracking transfer</td><td>Do not set global devices used in a program executed in both systems as a tracking transfer target. When using labels in a program executed in both systems, use local labels. When using FBs in a program executed in both systems, use local FBs.</td><td>Page 549 Tracking transfer</td></tr>
<tr><td rowspan="3">Device</td><td>Timer (T)</td><td>When the control system is switched to the standby system, the current value of the timer is not updated in the first scan of the new standby system and a timeout does not occur. As a result, an error of the time required for system switching + one scan is produced at system switching.</td><td>—</td></tr>
<tr><td>Long timer (LT) and long retentive timer (LST)</td><td>When the long timer (LT) or long retentive timer (LST) is used in the standby system, time is not measured and a timeout does not occur.
After the standby system has been switched to the control system, the long timer (LT) or long retentive timer (LST) is started up. To measure time in the standby system, use the timer (T).</td><td>—</td></tr>
<tr><td>Interrupt pointer (I)</td><td>The interrupt pointer (I) cannot be used for the standby system in backup mode.</td><td>—</td></tr>
<tr><td colspan="2">Buffer memory address</td><td>In the redundant extension base unit configuration, do not access the buffer memory of the module on the extension base unit from a program executed in the standby system in the separate mode or in a program executed in both systems using instructions or the module access device. If the buffer memory is accessed from the standby system, a stop error occurs. (When "Continue" is selected in the operation error in the RAS setting for the CPU parameter, the error can be set as a continuation error.)
The error can be also set as non-processing with SM1762 (Operation setting for access from the standby system to the extension base unit).</td><td>—</td></tr>
<tr><td colspan="2">Restricted instructions</td><td>Some instructions have restrictions when they are used in a program executed in both systems.</td><td>Page 550 Restricted instructions</td></tr>
<tr><td rowspan="2">Standard function block</td><td>TIMER_□_M</td><td>When the control system is switched to the standby system, the current value of the timer is not updated in the first scan of the new standby system and a timeout does not occur. As a result, an error of the time required for system switching + one scan is produced at system switching.</td><td>—</td></tr>
<tr><td>TP(_E), TON(_E), TOF(_E)</td><td>When these function blocks are used in the standby system, time is not measured and a timeout does not occur.
After the standby system has been switched to the control system, TP(_E), TON(_E), or TOF(_E) are started up.</td><td>—</td></tr>
<tr><td rowspan="2">Ethernet</td><td>Socket communications</td><td rowspan="2">When data is sent to the Ethernet-equipped module of the standby system, the data receive processing is not performed because the Ethernet-equipped module discards the received data.</td><td rowspan="2">MELSEC iQ-R Ethernet User's Manual (Application)</td></tr>
<tr><td>Communications using the fixed buffer</td></tr>
</table>

Program execution time

Set a program execution time of the standby system to be shorter than that of the control system. When a program execution time of the standby system is longer than the total program execution time of the control system, the control system starts the next scan before the standby system has received tracking data, causing a continuation error. If the systems are switched in this state, the latest data may not be reflected on the new control system.
When a program execution time of the standby system cannot be shortened, set SD1662 (Tracking transfer data receive completion wait time) to extend a waiting time for tracking data receive completion of the control system. With this setting, the control system CPU module starts the next scan after the standby system has received tracking data. Even if the systems are switched, the control can be continued based on the latest data.
In the redundant extension base unit configuration, set a program execution time of the standby system to be within 200ms. When the program execution time exceeds 200ms, the standby system detects a continuation error. If operation is performed while the program execution time exceeds 200ms, a hardware failure of the CPU module of the control system causes system switching, and because a stop error occurs in the CPU module of the new control system, control may not be continued.

Tracking transfer

- Do not set global devices used in a program executed in both systems as a tracking transfer target. Due to tracking transfer, the standby system data is overwritten with that of the control system, and the standby system program may operate in an unintended way.
- When using a label in a program executed in both systems, use a local label.
- When using an FB in a program executed in both systems, use a local FB. If a global FB is used, a program execution result of the control system is transferred and written over a program execution result of the standby system, causing an unintended operation.

Restricted instructions

The following table lists the instructions that are restricted in a program executed in both systems.

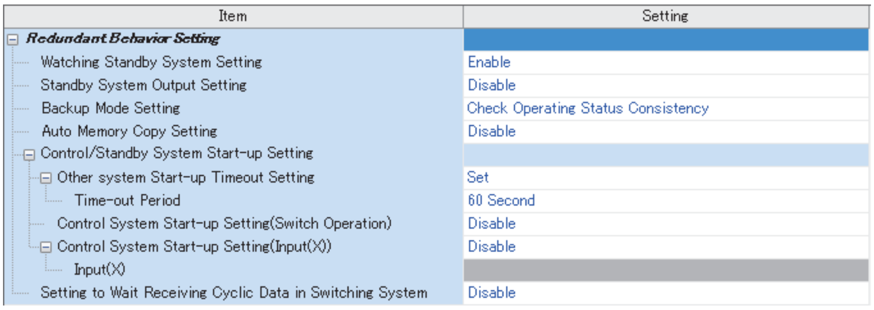
Classification	Notation	Description
Calling a subroutine program	CALL(P)	These instructions may not normally operate if a subroutine with any of a rising instruction, falling instruction, or SCJ instruction in a program which is not set as a both systems execution program is called in the standby system in the backup mode.
Calling a subroutine program and turning the output off	FCALL(P)	
Calling a subroutine program in the specified program file	ECALL(P)	
Calling a subroutine program in the specified program file and turning the output off	EFCALL(P)	
Calling a subroutine program with output off	XCALL	
Program control instruction	PSTOP(P)	No operation is performed if these instructions are executed on a program which is not set as a both systems execution program in the standby system in the backup mode.
	POFF(P)	
	PSCAN(P)	
Redundant system instruction	SP.CONTSW	No operation is performed if this instruction is executed in the standby system.
PID control instructions (Inexact differential)	S(P).PIDINIT	The PID control instruction information of the control system is transferred and written over the PID control instruction information of the standby system. When these instructions are executed in the standby system in the backup mode, a stop error may occur.
	S(P).PIDCONT	
	S(P).PIDSTOP	
	S(P).PIDRUN	
	S(P).PIDPRMW	
PID control instructions (Exact differential)	PIDINIT(P)	
	PIDCONT(P)	
	PIDSTOP(P)	
	PIDRUN(P)	
	PIDPRMW(P)	
SFC control instruction	SET [BL□]	No operation is performed if this instruction is executed in the standby system in the backup mode.
	RST [BL□]	
	PAUSE [BL□]	
	RSTART [BL□]	
	SET [S□/BL□\S□]	
	RST [S□/BL□\S□]	
Module access instructions	RFS(P)	No operation is performed if inputs (X) or outputs (Y) of a module on an extension base unit are specified from the standby system.
	COM(P)	Refresh of modules on an extension base unit is not performed if the instruction is executed from the standby system.
	S(P).ZCOM	
	FROM(P)	An error occurs if access to the buffer memory of a module mounted on an extension base unit is attempted from the standby system. (The error can be set as non-processing with SM1762 (Operation setting for access from the standby system to the extension base unit).)
	FROMD(P)	
	DFROM(P)	
	DFROMD(P)	
	TO(P)	
	TOD(P)	
	DTO(P)	
	DTOD(P)	
	TYPERD(P)	The module model name cannot be read if the module on an extension base unit is specified from the standby system.
	UNIINFRD(P)	The module information cannot be read if the module on an extension base unit is specified from the standby system.

29.7 Redundant System Operation Setting

Set the redundant system operation in the redundant system settings of the CPU parameter.

[CPU Parameter] ⇨ [Redundant System Settings]

Window



Displayed items

Item		Description	Setting range	Default
Watching Standby System Setting		Sets this item not to detect continuation errors when: • A Communication error with the other system has occurred or communications with the other system has been disabled. • The standby system CPU module is off, has been reset, or a stop error has occurred. This item is enabled only for the control system CPU module.	• Disable • Enable	Enable
Standby System Output Setting		Sets this item to enable the output (Y) from the standby system for system inspection or adjustment in the standby system in backup mode. Even though this parameter setting has been set to "Disable" in separate mode, the output (Y) is enabled. • To diagnose devices that are connected to the standby system by using a both systems execution program, set "Enable". However, set the standby system output setting to "Disable" when common devices have been connected to both the control system and standby system. • When "Enable" is set, do not specify outputs (Y) to be used in the standby system as the data to be tracked. If such an operation is performed, the outputs (Y) are overwritten with the control system data and output from the standby system. (Even though the standby system CPU module is in the STOP state, the output (Y) received in tracking transfer is output.)	• Disable • Enable	Disable
Backup Mode Setting		☞ Page 542 Operating status		
Auto Memory Copy Setting		☞ Page 533 Automatic memory copy		
Control/Standby System Start-up Setting*2	Other system Start-up Timeout Setting	Sets a timeout time taken until communications with the other system are enabled after own system is started up and the initial processing is completed. • When "Set" is selected, a stop error will occur in the own system unless communications are performed with the other system even though the timeout time has come. • When "Not Set" is set, the own system will wait until communications with the other system are established.	• Not Set • Set	Set
	Time-out Period		3 to 1800s (in increments of 1s)	60s
	Control System Start-up Setting (Switch Operation)	Sets this item to enable the operation to be started in the control system with the switch operation (RUN→STOP→RUN) while own system that has started is waiting for the start-up of the other system.	• Disable • Enable	Disable
	Control System Start-up Setting (Input (X))	Sets this item to enable the operation to be started in the control system with the contact input (X) while own system that has started is waiting for the start-up of the other system.	• Disable • Enable	Disable
	Input (X)		X0 to X2FFF	—
Setting to Wait Receiving Cyclic Data after Switching System*1		Sets this item to wait for the execution of the sequence program until all CC-Link IE Field Network modules on the main base unit receive the latest cyclic data after system switching. ☞ Page 554 Setting to wait cyclic data receive after system switching	• Disable • Enable	Disable

*1 Before enabling the setting to wait cyclic data receive after system switching, check the versions of the CPU module, CC-Link IE Field Network module, and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

*2 This setting is invalid in the redundant extension base unit configuration. (The system that has started up first starts up as a control system regardless of this setting.)

Standby system output setting

Output timing

When "Standby System Output Setting" has been set to "Enable", the output timing from the standby system in backup mode is at the completion of the END processing or depends on the refresh group setting or refresh settings of each module. (☞ Page 72 Group setting for refresh)
Therefore, when a control system execution program is set in the refresh group setting, the I/O refresh is not performed while the module is running because the program is not executed in the standby system. (When the CPU module is the STOP/PAUSE state, the I/O refresh is performed at the timing of the END processing.)
To execute a program set in the refresh group setting in the standby system in backup mode, set "Both Systems Program Executions Setting". (☞ Page 544 Program Execution in Both Systems)

Operation at system switching

The operation performed at standby system differs when "Standby System Output Setting" is set to "Enable". The following table shows the operation performed when "Standby System Output Setting" has been set to "Enable".

Item	New control system CPU module	New standby system CPU module
Output (Y)	The status of the old standby system is held and output refresh is performed.	The status of the old control system is held and output refresh is performed. However, no data is output, although the output Y of the module mounted on the extension base unit is held.
Direct access output (DY)	In the program execution after system switching, data is output when an instruction using the direct access output (DY) is executed.	■For program executed in both systems In the program execution after system switching, data is output when an instruction using the direct access output (DY) is executed. However, no data is output even when an instruction using the direct output of the module mounted on the extension base unit is executed. ■For program executed in the control system No operation is performed because the program does not operate.

For the program execution in both systems, refer to the following.
☞ Page 544 Program Execution in Both Systems

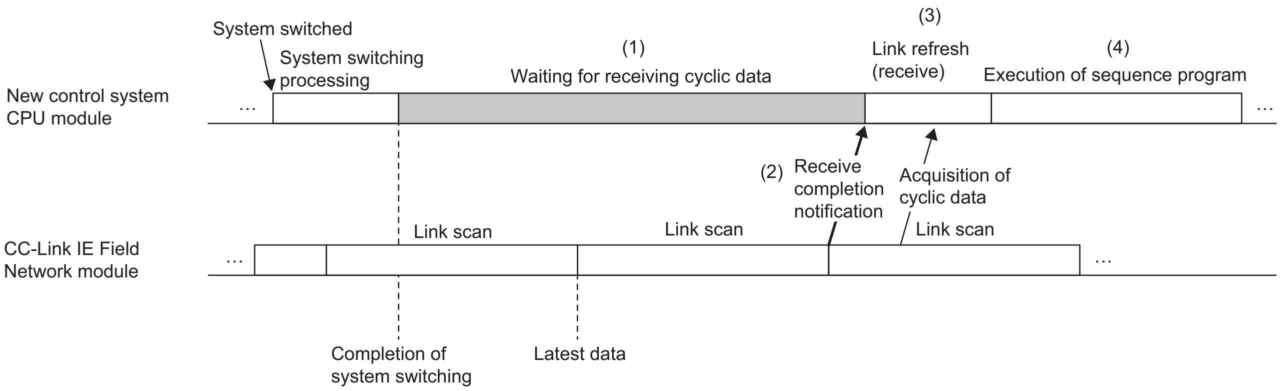
Setting to wait cyclic data receive after system switching

When system switching occurs in the redundant line structure of CC-Link IE Field Network, set this item to start the execution of the program in the new control system with new cyclic data obtained after system switching.

Restriction

Before enabling the setting to wait cyclic data receive after system switching, check the versions of the CPU module, CC-Link IE Field Network module, and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

When this setting is enabled, the execution of the program is suspended until new cyclic data is received in the new control system after system switching.



(1) After system switching has been completed, the CPU module stands by until it is notified of the completion of the cyclic data receipt after system switching by all CC-Link IE Field Network modules on the main base unit.
(2) When the CC-Link IE Field Network modules receive cyclic data from the remote I/O station after system switching, the CC-Link IE Field Network modules notify the CPU module of the completion of the cyclic data receipt.
(3) When notified of the completion of the cyclic data receipt by the CC-Link IE Field Network modules, the CPU module executes link refresh (receive).
(4) The CPU module executes the sequence program using the new cyclic data.

- When this setting is enabled, the extended time from the completion of system switching to the first output is added to the cyclic data receipt waiting time. (☞ Page 1086 Delay time until initial output after system switching (Tjo))
- This setting is executed in the new control system after system switching regardless of the operation mode (backup mode/ separate mode). This setting operates when the operating status of the CPU module in the new control system is RUN, STOP, or PAUSE. (This setting does not operate when a stop error has occurred in the CPU module in the new control system.)
- This setting operates when the link scan mode of the CC-Link IE Field Network is set to "Sequence Scan Asynchronous" or "Constant Link Scan". This setting does not operate when the link scan mode of the CC-Link IE Field Network is set to "Sequence Scan Synchronous Setting". (📖 MELSEC iQ-R CC-Link IE Field Network User's Manual (Application))
- When cyclic data cannot be received within the timeout time due to disconnection of the network cable or other causes while cyclic data receipt is waited for, waiting for cyclic data receipt is stopped, and the sequence program is executed. Occurrence of a timeout can be checked in SM1756 (wait timeout for receiving cyclic data after system switching) and SD1756 (module information on wait timeout for receiving cyclic data after system switching).

Precautions

- Scan time monitoring with the watchdog timer is interrupted while cyclic data receipt is waited for. Thus, no error is detected even if the scan time monitoring time has elapsed while cyclic data receipt is waited for.
- Constant scan is disabled while cyclic data receipt is waited for. Thus, no error is detected even if the constant scan setting time has elapsed. Constant scan is enabled after waiting for cyclic data receipt has been completed, and the sequence program is executed.
- Because device/label access service processing is not accepted while the cyclic data receipt is waited for, set the communication timeout time with external devices in consideration of the cyclic data receipt waiting time. (☞ Page 1087 Waiting time for cyclic data receive after system switching (Twcyc))

29.8 Redundant Function Module Communication Test

The hardware of the redundant function module is checked for an error when its communication is unstable.
The following table shows the test items included in the module communication test.

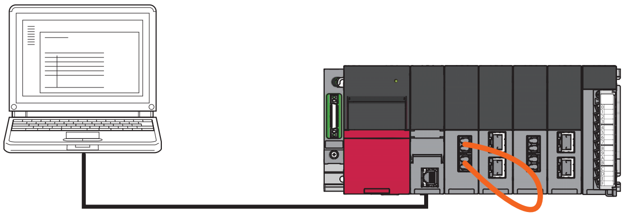
Test item	Description
Internal selfloopback test	Checks whether the communication function of the redundant function module normally operates.
External selfloopback test	Checks whether communications can be normally performed with a tracking cable that connects the IN connector and OUT connector of the redundant function modules.

Execution procedure of the module communication test

1. Connect the engineering tool directly to the standby system CPU module.

2. Set the CPU module operating status to the STOP state.

3. Connect the IN and OUT connectors of the standby system redundant function module with the tracking cable. (A cable disconnection error is detected in the control system.)



4. Open the "Redundant Operation" window of the engineering tool.

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

5. Click the [Execute Test] button of "Module Communication Test".

- LED status of the redundant function module during the module communication test

Status	RUN LED	ERR LED
Module communication test in execution	Flashing	Off
Completed successfully	On	Off
Completed with an error	On	On

6. When the test is completed with an error, take actions according to "Corrective Action" in the test result window.

7. When the test is completed successfully, connect the control system and standby system with the tracking cable. (☞ Page 485 Wiring the redundant function modules)

8. Click the [Close] button in the "Redundant Operation" window to exit the module communication test.

9. Set the CPU module operating status to the RUN state.

Point

The module communication test can be executed regardless of whether the backup mode or separate mode is selected as the operation mode. In the backup mode, a tracking communication error is detected when a tracking cable is disconnected.

Precautions

- Always connect the IN and OUT connectors of the redundant function module with the tracking cable before performing the module communication test.
- Perform the module communication test on the standby system CPU module. Performing the test on the control system CPU module may cause an unintended operation.

29.9 Redundant Extension Base Unit Configuration Setting

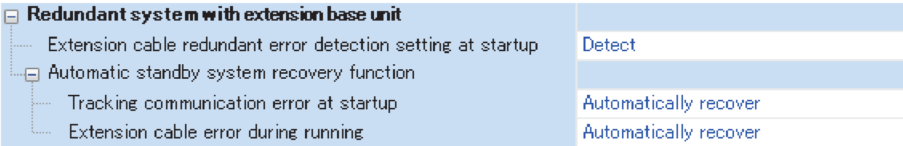
Set the redundant extension base unit configuration setting operation in the redundant settings of the CPU parameter.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Redundant system with extension base unit]

Restriction

Before using the redundant extension base unit configuration setting, check the versions of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

Window



Displayed items

Item		Description	Setting range	Default
Extension cable redundant error detection setting at startup		Set whether to detect an error when the extension cable between the extension base units is not redundant at startup.	• Detect • Not Detected	Detect
Automatic standby system recovery function	Tracking communication error at startup	Set whether to recover the standby system automatically when the standby system fails to start up due to any of the following causes. • One of the systems is turned off and on before both systems start up. • The other system is turned on before the first system starts up. • It took a long time to start up the standby system.	• Automatically recover • Not to automatically recover	Automatically recover
	Extension cable error during running	Set whether to perform automatic recovery for the new standby system when the system is switched due to an error in the extension cable (between extension base units) on the active side (the side with the ACTIVE LED lit). (☞ Page 557 Automatic recovery of the CPU module of the standby system).	• Automatically recover • Not to automatically recover	Automatically recover

Automatic recovery of the CPU module of the standby system

This function allows the CPU module of the standby system to automatically recover from a specific error (☞ Page 557 Target errors and operations when the automatic recovery is enabled) that occurs in the CPU module of the standby system. (Manual operation (turning off and on or resetting the system) is not required to recover the system.)

Point

This function is valid in the backup mode. This function is invalid in the configuration with the main base unit only.

Target errors and operations when the automatic recovery is enabled

The following describes the target of automatic recovery and the operation at automatic recovery setting.

Target error	Operation	Reference
Tracking Communication Error at Startup	If the standby system fails to start up due to any of the following causes, the standby system automatically recovers and both systems can be operated. • One of the systems is turned off and on before both systems start up. • The other system is turned on before the first system starts up. • It took a long time to start up the standby system. (Such as when automatic restoration or boot operation takes a long time or SD memory card diagnosis is executed)	Page 557 When a tracking communication error occurs at system start-up
Extension cable error on the active side (between the extension base units)	If an error occurs in the extension cable (between extension base units) on the active side, the new standby system automatically recovers and both systems can be operated.	Page 558 When an extension cable error occurs during system operation

Point

- If automatic recovery fails and the cause is eliminated, automatic recovery is not performed again by this function. Therefore, recover the standby system by manual operation (turning off and on or resetting the system). (☞ Page 558 Cause of failure of automatic recovery).
- The event is registered when automatic recovery is performed or fails. Therefore, whether automatic recovery has been executed or failed can be checked in the event history.

When a tracking communication error occurs at system start-up

Tracking communication cannot be performed during the initial processing of the CPU module that was turned on first when the redundant system starts up. Therefore, a tracking communication error may occur to the CPU module that was turned on later. If "Tracking communication error at startup" is set to "Automatically recover" for the CPU parameter, when a stop error of a tracking communication error occurs in the CPU module of the standby system, the standby system is automatically recovered.

When an extension cable error occurs during system operation

When an error occurs in the extension cable on the active side (the side with the ACTIVE LED lit) between the extension base units during system operation, an extension cable error occurs in the CPU module of the control system and the systems are switched.

If "Tracking communication error at startup" is set to "Automatically recover" for the automatic recovery setting, when a stop error occurs in the CPU module of the new standby system due to an extension cable error after system switching, the new standby system is automatically recovered.

Point

When redundant extension cables are used (when extension base units for the redundant system are used at extension level 2 and later), automatic recovery is performed only when a cable error occurs between extension base units. Therefore, automatic recovery is not performed when any of the following errors occur. Take action according to the procedures for a stop error that has occurred, and then restore the standby system manually.

- Extension cable error between the main base unit and the extension base unit
- Extension cable error when the extension cable is not redundant

If the extension cable is redundant, automatic recovery is performed even if there are both a level with one extension cable and a level with two extension cables between the extension base units and an extension cable error occurs in the level with two cables.

Cause of failure of automatic recovery

Automatic recovery may fail due to the following causes. If automatic recovery fails, recover the standby system by manual operation (turning off and on or resetting the system).

- Tracking communications disabled (cable disconnection or control system power-off)
- A file on the CPU module of the standby system being accessed*1
- Stop error in the CPU module of the control system
- Memory copy being executed
- Online change being executed*2
- System switching being executed

*1 Only external file access is disabled. (Access to files by a system such as event history or data logging is reset after the completion of the access.)

*2 If any cause of automatic recovery occurs during backup, it is not handled as automatic recovery failure, and automatic recovery is executed after the completion of backup.

Precautions

The following describes the precautions for automatic recovery.

- If automatic recovery is performed by this function during writing to the programmable controller on the CPU module of the standby system, the writing may fail and a file invalid error may occur at the time of recovery. In this case, write the data to the programmable controller again.

29.10 Constant Scan

The following describes the precautions for setting the constant scan in the redundant system.

Increase in scan time

In the standby system, when the CPU module is powered off, a hardware failure has occurred, or a tracking cable has a failure, the scan time will increase in the control system.

When setting the constant scan, take one of the following measures.

- Set the setting time of the constant scan by adding the increase in the scan time when an error occurs. (☞ Page 1078 Increase in the scan time due to tracking transfer)
- When a failure has occurred and a continuation error occurs due to the excess of constant scan time, clear the error. (☞ Page 145 Error Clear)

System switching in separate mode

When the systems are switched in separate mode, the scam time of the control system will increase by the increase of the standby system scam time.

For program executed in both systems

The constant scan function is invalid for the standby system in backup mode.

29

29.11 Online Change

When the online change (ladder block) is performed on the CPU module in one system in backup mode, the change is also reflected on the CPU module in the other system.
In separate mode, only the ladder block of the CPU module in the system specified in the transfer setup of the engineering tool is changed.

Precautions

During an online change, avoid the following conditions.
- Changing the operating status of the CPU module from STOP (PAUSE) to RUN
- System switching
- Operation mode switching (switching to the backup mode)
- The tracking cable has been removed.
- The control system or standby system CPU module has been off or reset.

When either of the above conditions is satisfied during an online change, a file mismatch is detected in the system consistency check or the online change fails.

■Action for when an online change has failed

If an online change has failed, take the following actions.

1. Get ready for an online change and execute the online change with the same data as before the failure.
- If the online change succeeds, the action is completed.
- If the online change fails again, take the following actions.

2. Verify the data in the control system CPU module with the programmable controller from the engineering tool, and check whether the online change has been completed successfully.
- When the verification result is mismatched, the online change has not been completed successfully. Connect the engineering tool to the control system CPU module, execute an online change only to the control system, and proceed to step 3.
- When the verification result is matched but the program memory transfer window was not displayed at the online change, the possible cause is that data transfer to the program memory has not been completed successfully. Batch-write files only to the control system, and proceed to step 3.
- When the verification result is matched and the program memory transfer window is displayed (the online change is completed successfully) during online change, proceed to step 3.

3. Execute the memory copy from the control system to the standby system. When the memory copy has failed, power off and on or reset the standby system CPU module and execute the memory copy again. (☞ Page 530 Memory Copy from Control System to Standby System)

4. Power off and on or reset the standby system CPU module.

■When the program restoration information is written in the background

If the program restoration information is written by the online change using the engineering tool with version "1.045X" or earlier and the CPU module with firmware version "15" or later, an unsupported error by the engineering tool may occur. In this case, take either of the following actions.
- Update the version of the engineering tool.
- If the version of the engineering tool cannot be updated, set "Write with Execution Program" to the following.

[Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Online Program Change" ⇨ "Operational Setting" ⇨ "Write Program Restore Information"

■Writing data to the CPU module in the redundant extension base unit configuration

Data cannot be written to the programmable controller while its CPU module is running or while connecting to a module on the extension base unit. Otherwise, an error occurs.

29.12 RAS Function

Clearing errors on the standby system CPU module from the control system CPU module

Errors on the standby system can be cleared from the control system CPU module by using SM1679 (Error reset (the other system)) in programs or external devices.

Error clearing procedure

Use SM1679 to clear errors.

1. Eliminate the causes of all the continuation errors that have occurred on the standby system.
2. Turn off and on SM1679 of the control system CPU module to clear the errors. When multiple continuation errors have occurred, all the errors are cleared at once.

29

Precautions

- Error clear operation with SM1679 can be performed by the control system CPU only. A continuation error on the standby system CPU module cannot be cleared by turning off and on SM1679 of the standby system CPU module.
- Error clear operation with SM1679 can be performed in backup mode only.
- The cause of an error which has occurred in a module other than the target CPU module for the error clear cannot be eliminated even though the error is cleared using SM1679.
- If the error cause is not eliminated after the error clear operation, the same error is detected again.
- The error clear processing is performed in the END processing. To clear an error, execute the END instruction while SM1679 is on.

Event history function

Events that occur in a module on the extension base unit are saved in the event history of the CPU module of the control system.
However, events that occur in the module on the extension base unit may be saved in the event history of the CPU modules of both systems when the system is switched by power-off or reset of the control system, depending on the timing of saving.

Checking the event history of both systems chronologically

The engineering tool displays the event history saved in the CPU module of the connected system. Therefore, when checking the event history of both systems chronologically, output the event history of each system from the Event History window of the engineering tool to a CSV file and check the event history on the CSV file.

Event history logging restriction

When the number of minor events from the CPU module or intelligent function module such as link-up and link-down exceeds the upper limit value, event history logging is restricted (stopped). (☞ Page 153 Event history logging restriction)
The CPU module of the control system monitors and restricts the logging of events that occur in a module on the extension base unit.
When systems are switched, the event history logging restriction state is cleared. Therefore, even if the old control system is placed in the logging restriction state before system switching, the new control system is placed in the state of no event history logging restriction.

29.13 Remote Operation

In a redundant system, the operation target of remote operations (with the engineering tool) depends on the operation mode and method.

Operation mode	Remote operation	
	Remote RUN, remote STOP, remote PAUSE	Remote RESET
Backup mode	The CPU module operating status of a system specified in the transfer setup of the engineering tool or both systems can be changed.	The CPU modules of both systems can be reset by performing the remote RESET operation on the control system CPU module. Only the standby system CPU module can be reset by performing the remote RESET operation on the standby system CPU module.
Separate mode	The CPU module operating status of a system specified in the transfer setup of the engineering tool can be changed.	

Remote RUN, remote STOP, remote PAUSE

The following describes the remote RUN, remote STOP, and remote PAUSE operations in a redundant system.

In backup mode

The CPU module operating status of a system specified in the transfer setup of the engineering tool or both systems can be changed.

■Remote operation performed on the CPU module of a system specified in the transfer setup of the engineering tool

When any of the following items is selected in "Execution Target", the remote operation is performed only on the CPU module of a system specified in the transfer setup of the engineering tool.

- Currently Specified Station
- All Stations Specified
- Specify Group No.

[Online] ⇨ [Remote Operation]

■Remote operation performed on both systems

When the following item is selected in "Specify Execution Target", the remote operation is performed on the CPU modules of both systems.

- Specify Both Systems

[Online] ⇨ [Remote Operation]

Point

When "Specify Redundant CPU" is set to "Not Specified" in the transfer setup, select "All Stations Specified" or "Specified Group" to perform a remote operation on the CPU modules of both systems.

In separate mode

The CPU module operating status of a system specified in the transfer setup of the engineering tool can be changed. (☞ Page 562 Remote operation performed on the CPU module of a system specified in the transfer setup of the engineering tool)

Remote RESET

The following describes the remote RESET operation in a redundant system.

In backup mode

The CPU modules of both systems can be reset by performing the remote RESET operation on the control system CPU module. Only the standby system CPU module can be reset by performing the remote RESET operation on the standby system CPU module.
Select "Control System" or "Standby System" in the transfer setup, and select "Currently Specified Station/Specify Both Systems" in "Specify Execution Target" of "Remote Operation".

■Precautions

The following describes the precautions on the remote RESET operation performed on both systems in backup mode.

- When the control system CPU module is in the STOP state and the standby system CPU module is in the RUN state, performing the remote RESET operation on the control system CPU module causes system switching. To prevent system switching in the remote RESET operation, perform the remote RESET operation after setting both of the CPU modules to the STOP state.
- When a remote operation is performed on the CPU module of the control system or standby system through another path, the standby system CPU module cannot be reset by performing a remote RESET operation on the control system CPU module. To perform a remote RESET operation on the CPU module of the control system or standby system, use the path that was used to perform the remote operation on the standby system CPU module. Cancel the remote operation first and perform the remote RESET operation.
- When performing a remote RESET operation on both systems, select "Control System" in "Specify Redundant CPU". When "Not Specified" is selected, an error may occur depending on the timing of the operating status change of each CPU module.

[Online] ⇨ [Current Connection Destination] ⇨ [Specify Redundant CPU]

In separate mode

Only the CPU module of a system specified in the transfer setup of the engineering tool can be reset. The operating status of the CPU module in a system not specified is not changed.

Precautions

The following describes the precautions on the remote operation in the redundant system.

- When the control system or standby system is in the initial processing (the READY LED is flashing), do not perform the remote operation of "Specify Both Systems". If such an operation is performed, both systems may be recognized as being mismatched in the system consistency check because the operating statuses may be mismatched.
- If a remote RESET is performed for both systems when the load of each CPU module is high in the redundant extension base unit configuration, the module on the extension base unit may not be reset and each CPU module may not start up successfully. If each CPU module does not start up successfully, execute a remote RESET for both systems again.

29.14 Boot Operation

This section describes precautions on using the boot operation in the redundant system.

Boot operation at start-up of the redundant system

Use the boot operation only to simultaneously start up both systems.

1. Attach SD memory cards that have boot setting data to both systems.
2. Simultaneously turn on or reset both systems and perform the boot operation for them.

Do not use the boot operation to restart one system. If only one system is restarted with the boot operation, a system consistency check error may occur. The following operations are recommended.

- Remove the SD memory cards for boot operation after simultaneously starting up both systems.
- When using an SD memory card for other applications, such as the data logging function, prepare another one.

Online change in boot operation

It is recommended to remove SD memory cards for boot operation while the system is operating. Do not perform the online change (online change (ladder block)). To copy project data to an SD memory card for boot operation, set the CPU module in the STOP state and write the project data to the SD memory card.

Precautions

When starting up the CPU modules of both systems simultaneously by boot operation in the redundant extension base unit configuration, it is recommended to use the automatic recovery function of the CPU module of the standby system.
Even if the CPU modules of both systems are started up simultaneously, if the start-up timing of one system is delayed due to the time taken for booting, the CPU module of the system that was started up first cannot perform tracking communications during initialization. Therefore, a stop error may occur in the CPU module of the system that was started up later.
In this case, if the automatic recovery function of the CPU module of the standby system is used, the system can be recovered by automatically restarting the CPU module with a stop error. If the automatic recovery function of the CPU module of the standby system is not used, the CPU module with a stop error must be restarted by manual operation.

29.15 External Input/Output Forced On/Off Function

Forced on/off is reflected to the input/output devices of both systems and external outputs by registering or canceling forced on/off for the control system. (Forced on/off is reflected to both systems without setting tracking transfer setting in the CPU parameters.)

Restriction

Before executing the external input/output forced on/off function, check the versions of the CPU module and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

Availability of forced on/off operations

Forced on/off operations are performed for the control system. However, registration status update can be performed for both systems. (There is no difference between operation modes.)

○: Operation possible, ×: Operation not possible

Operation item	Control system	Standby system
Forced on registration	○	×
Forced off registration	○	×
Registration cancellation	○	×
Registration batch cancellation	○	×
Registration status update	○	○

Reflection to external outputs of the standby system

The forced on/off is reflected to output devices of the standby system regardless of the operation mode and settings. However, for external outputs of the standby system, the reflection is as follows depending on the operation mode and the output setting of the standby system.

Operation mode	Standby system output setting	Reflection to output devices of the standby system	Reflection to external outputs of the standby system
Backup mode	Disable	Reflected	Not reflected
	Enable		Reflected*1
Separate mode	Disable		
	Enable		

*1 Since cyclic data of the network module is not sent on the standby system, forced on/off is reflected only to the modules mounted on the main base unit. (Forced on/off is not reflected to external outputs of the remote I/O module regardless of the standby system output setting.)

Input/output behavior in forced on/off registration

The reflection of forced on/off does not differ depending on the system or operation mode.

■Forced on/off of input devices

Forced on/off is reflected to input devices when the END processing is executed. The following table lists the behavior of the input devices within the refresh range or the input devices specified in the tracking transfer setting.

Item	Behavior
Input devices within the refresh range	Forced on/off is reflected to the input devices after the data is refreshed from the module.
Input devices specified in the tracking transfer setting	Forced on/off is reflected to output devices after the transferred device data is reflected to the standby system.

■Forced on/off of output devices

Forced on/off is reflected to output devices when the END processing is executed. The following table lists the behavior of the output devices within the refresh range or the output devices specified in the tracking transfer setting.

Item	Behavior
Output devices within the refresh range	After forced on/off is reflected to the output devices, data is refreshed to the module.
Output devices specified in the tracking transfer setting	Forced on/off is reflected to the output devices on both systems after the device data transferred from the standby system is reflected.

Forced on/off timing

The following table lists the timing to reflect forced on/off registration information to the input/output devices or external outputs.

Input/output for which forced on/off can be set	Reflection timing for the input devices	Reflection timing for the output devices or external outputs*4
Input/output of the modules mounted on the base unit	• END processing (at input refresh) • At COM instruction execution (at input refresh) • At instruction execution using the direct access input (DX) (LD, LDI, AND, ANI, OR, ORI, LDP, LDF, ANDP, ANDF, ORP, ORF, LDPI, LDFI, ANDPI, ANDFI, ORPI, ORFI) • At RFS instruction execution • At execution of instructions used in the interrupt by the system (UDCNT2) • At program execution*1 • At reflection of the tracking receive data to devices	• END processing (at output refresh) • At COM instruction execution (at output refresh) • At instruction execution using the direct access output (DY) (OUT, SET, DELTA(P), RST, PLS, PLF, FF, MC, SFT(P))*1 • At RFS instruction execution • At program execution*1 • At reflection of the tracking receive data to devices*2
Input/output of the CPU module assigned to LX and LY of the CC-Link IE Controller Network module or MELSECNET/H network module*3	• END processing (at link refresh of the CC-Link IE Controller Network module or MELSECNET/H network module) • At COM instruction execution (at link refresh of the CC-Link IE Controller Network module or MELSECNET/H network module) • At ZCOM instruction execution (at link refresh of the CC-Link IE Controller Network module or MELSECNET/H network module)	
Input/output of the CPU module assigned to RX and RY of the CC-Link module*5	• END processing (at link refresh) • At COM instruction execution (at link refresh) • At ZCOM instruction execution (at link refresh)	
Input/output of the CPU module assigned to RX and RY of the CC-Link IE Field Network module*3	• END processing (at link refresh) • At COM instruction execution (at link refresh) • At ZCOM instruction execution (at link refresh)	

*1 At input/output refresh execution where input/output refresh is registered for each program, and for interrupt programs.
*2 Forced on/off is reflected only to output devices. (Since refresh to external outputs is not executed.)
*3 Forced on/off is reflected only to input devices on the standby system. (Since refresh from the CPU module to link devices is not executed.)
*4 Whether external outputs to the standby system are reflected or not differs depending on the standby system output setting. (☞ Page 565 Reflection to external outputs of the standby system)
*5 Refresh can be executed only for the module mounted on the extension base unit.

Behavior of forced on/off

This section describes the behavior of forced on/off in the following cases.

■At operation mode change

Even if the operation mode has been changed (backup mode → separate mode, or separate mode → backup mode), the forced on/off registration information before the change remains.

■At system switching

Even if a system switching is performed, the forced on/off registration information before the switching remains since forced on/off registration information is transferred from the control system to the standby system.

■When the CPU module of only one system is powered off and on or is reset

Condition		Behavior
Backup mode	Control system	Although a system switching is performed, the forced on/off registration information of the new control system remains while the CPU module of the old control system is off or in the reset state. After the CPU module of the old control system is powered on or is reset, the forced on/off registration information of the new control system is reflected to the new standby system.
	Standby system	The forced on/off registration information of the control system remains while the CPU module of the standby system is off or in the reset state. After the CPU module of the standby system is powered on or is reset, the forced on/off registration information of the control system is reflected to the standby system.
Separate mode	Control system	The forced on/off registration information of the standby system remains while the CPU module of the control system is off or in the reset state. After the CPU module of the control system is powered on or is reset, the forced on/off registration of the standby system is canceled in a batch and no forced on/off is registered in both systems.
	Standby system	The forced on/off registration information of the control system remains while the CPU module of the standby system is off or in the reset state. After the CPU module of the standby system is powered on or is reset, the forced on/off registration information of the control system is reflected to the standby system.

■When tracking communications are disabled

When tracking communications with the standby system are disabled or the system is operating only with the control system, forced on/off can be registered or registration of forced on/off can be canceled to the control system. Forced on/off registration information of the control system is reflected to the standby system when tracking communications are enabled.

Reflection timing of forced on/off registration information to the standby system

This section describes the reflection timing of forced on/off registration information to the standby system.

■When the synchronous tracking mode is used as a tracking mode

The reflection of forced on/off registration information to the standby system delays for one scan (maximum).

■When the asynchronous tracking mode is used as a tracking mode

The reflection of forced on/off registration information to the standby system may delay for several scans.

Precautions

The following describes precautions on this function.

- When reflecting forced on/off registration information in the first scan is required, start the standby system in the STOP state, and switch it to the RUN state after checking that the forced on/off registration information of the control system is reflected to the standby system. When the standby system is powered on while the control system is operating, forced on/off registration information may not be reflected in the first scan if the standby system is started with its RUN/STOP/RESET switch set to the RUN state.
- When reflecting the external input/output forced on/off function to the standby device in the redundant extension base unit configuration, perform tracking transfer for the device.

29.16 Device Test with Execution Condition

Using the engineering tool, device/label values can be set for each execution of specified steps of program.

Configurable data

The following devices cannot be used in redundant mode because they do not support the multiple CPU system.

- U3En\G to which other than the host CPU module is specified
- U3En\HG

Operation when device test with execution conditions is registered

This section describes the operation of the device test with execution conditions in redundant mode.

■Operation at operation mode change

Even if the operation mode is changed, the registration information of the device test with execution conditions before changing the operation mode is retained.

■Operation at system switching

Even if the system is switched, the registration information of the device test with execution conditions before system switching is retained.

■Operation when the CPU module of only one system is powered off and on or is reset

Even if the CPU module of the other system is powered off and on or is reset, the registration information of the device test with execution conditions of the CPU module of the own system is retained. The registration information of the device test with execution conditions is released in the CPU module which is powered off and on or reset.

■When the device test with execution conditions is registered to the standby system CPU module of backup mode

For the backup mode CPU module, if a program for which "Both Systems Program Executions Setting" is set to "Control System Execution" is specified and the device test with execution conditions is registered, the execution conditions are not satisfied and the device test is not executed.

Precautions

The following describes precautions on this function in redundant mode.

■When the program which is not executed in the standby system CPU module of backup mode is registered

If system switching from backup mode standby system or changing operation mode, the execution conditions of the program that previously had not been executed are satisfied and the device test may be executed unintentionally. To prevent the above, be careful with executing system switching and operation mode change. For example, they shall be executed after all the registrations of the device test with execution conditions are released.

■Registration and release of registration of the device test with execution conditions during online change

The registration or release of registration of the device test with execution conditions for the target system of online change cannot be executed during online change to both systems or only one system.

■When the online change was executed in the state that the device test with execution conditions is registered

The registration information of the device test with execution conditions in the system where the online change is executed may be released according to the ON/OFF state of SM940 (Operation setting of the device test with execution conditions) or the addition/deletion/change contents of the labels or programs.

■CPU module to connect to when the device test with execution conditions is executed

When the device test with execution conditions is executed, specify "Not Specified", "System A", or "System B" to the CPU module to connected to. If the device test with execution conditions is registered in the state that the control system/standby system CPU module to connect to is specified, the CPU module is changed by system switching, and the registration of the device test with execution conditions cannot be released. In this case, change the CPU module to connect to and release the registration again.

■When the tracking transfer is set

When the device test with execution conditions is registered to the standby system CPU module in the state that the tracking transfer of devices/labels is set, if the execution conditions are satisfied, the device/label values are changed. The devices/labels are overwritten using tracking transfer at END processing. When registering the device test with execution conditions to the standby system CPU module, take the tracking transfer into consideration.

■When the device test with execution conditions is registered to the standby system

When the device test with execution conditions specifying buffer memory, link direct device, or direct access output of the module on the extension base unit is registered in the standby system in the redundant extension base unit configuration, the value of the device is not changed even if the execution conditions are satisfied.

29.17 Data Logging Function

In a redundant system, the data logging function collects data only in the control system regardless of the operation mode.

Point

For details on CPU Module Logging Configuration Tool used for the data logging function in a redundant system, refer to the following.

☞ Page 1104 Specifications of the Data Logging Function

Data logging states

The following table describes each data logging status in the redundant system.

Data logging state	Description
Stop	No data logging settings are registered and data collection is inactive.
Stop (Execute other functions after completing collection)	Transition from "Collection completed" to "Stop" has occurred due to the execution of another function*1.
Stop (Execute other functions after error)	Transition from "Error" to "Stop" has occurred due to the execution of another function*1.
Waiting RUN Not collected	• Data collection has not yet been started because the CPU module is not in the RUN state. • After the operation mode was switched to the separate mode, the standby system CPU module is waiting for the transition to the RUN state.
Waiting Start Not Collected	Data collection has not yet been started and the CPU module is waiting for the start operation.
Pause	Data logging is suspended and data collection has not yet been started.
Waiting to establish collection conditions Not collected	Waiting for the first collection timing after the start operation
Collecting	Continuous logging is active and data collection is being performed.
Waiting trigger Collecting before trigger	Trigger logging is active, data collection is being performed, and waiting until the trigger condition is met
After Trigger Collecting	Trigger logging is active and data collection is being performed after the trigger condition is met
Collection Completed	• Continuous logging: Data collection has finished upon reaching "Number of files to be saved" specified as part of the "Stop" setting configured in "Operation when the number of files exceeds the set value". • Trigger logging: Trigger logging has finished collecting data as much as the specified number of records.
Error	Data logging has failed due to the occurrence of an error.
Standby system start waiting	Data collection has not been performed, waiting for the start operation in the standby system.
Standby system pause	Data logging is suspended in the standby system.
Standby system no collection	Data logging is started in the standby system, but data collection is inactive.

*1 The execution of another function includes:

· Execution of data logging with the same trigger conditions (Trigger conditions = Specified conditions)

· Auto logging

· Online change

Data logging states at system switching

The following table lists the data logging states that change at system switching.

■When the control system is switched to the standby system

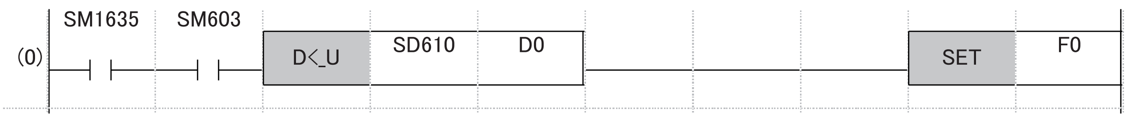
Before system switching (old control system)	After system switching (new standby system)
Waiting Start Not Collected	Standby system start waiting
Pause	Standby system pause
Waiting to establish collection conditions Not collected	Standby system no collection
Collecting	
Waiting trigger Collecting before trigger	
After Trigger Collecting	

■When the standby system is switched to the control system

Before system switching (old standby system)	After system switching (new control system)
Standby system start waiting	Waiting Start Not Collected
Standby system pause	Pause
Standby system no collection	Waiting to establish collection conditions Not collected

Point

Before using the data logging function, check if the SD memory card inserted in the standby system CPU module has sufficient free space with a program. When the SD memory card inserted in the standby system CPU module does not have sufficient free space and the systems are switched, an error occurs at the data collection in the new control system and the data collection is finished.
To prevent occurrence of an error, use a program that turns on the annunciator when the free space of the SD memory card of the standby system is smaller than the value of the criteria, such as the one shown below.



- For the above program, set "Both Systems Executions" for "Both Systems Program Executions Setting" of the CPU parameter. (☞ Page 544 Program Execution in Both Systems)
- Set the value of the criteria in D0 in increments of K of the double word [unsigned].

LED status

The following table shows the LED status in the data logging states of the standby system.

State of the data logging function		LED status
Execution in progress	All of the registered data logging sessions are in the "Standby System Start Waiting", "Standby System Pause", or "Standby System No Collection" state.	• CARD READY LED: On • CARD ACCESS LED: Off • FUNCTION LED: On

Data collection conditions

When time specification has been set as a data collection condition and the systems are switched during data logging, time measurement starts from 0 in the new control system, and data is not collected at the specified interval.

Trigger logging

If the systems are switched while trigger logging is being executed, logging data collected in the old control system is discarded.

Switching to a storage file

After system switching, the file name in the old control system is not taken over to the new control system. (A number added to the file name is not a serial number.)

An identifier added to the end of the file name indicates whether the systems are switched or not.*3

Identifier	Description
_NC*1*2	The systems have not been switched.
_CS	The control system has been switched to the standby system.
_SC	The standby system has been switched to the control system.
_SS	The standby system has been switched to the control system and back to the standby system. (The system is switched back to the control system while data is being written to the save file.)

*1 With the trigger logging, the identifier "_NC" is added because the logging data is discarded at system switching.

*2 With the auto logging, the identifier "_NC" is added because the auto logging is completed when the systems are switched.

*3 Since three characters of an identifier are added to a file name, the save file name should be specified within 61 one-byte characters (including an underscore (_), a serial number (eight digits), a period, and an extension).

Point

When the simple setting (default) is applied for the save file name, only a serial number is added to the file name. If the systems are switched with this setting, which file is sequentially output cannot be determined. Thus, the following settings are recommended.

- Simple setting: Select "Date" and "Time", and select "Date to establish file switching condition" in the add date type.
- Optional setting: Enter a string indicating a date (YYYYMMDD) and time (hhmmss) in "Format", and select "Date to establish file switching condition" in the add date type.

Storage file switching condition

Switch the storage file at system switching at the following timing.

Storage file switching condition	Storage file switch timing
System switching from control system to standby system	When the data collected before system switching to the standby system is transferred to the SD memory card

Writing the data logging setting

When the data logging setting is written, please note the following.

- While the data logging setting is being written, do not perform the following operations on both of the control system and standby system.
  - Turning off or resetting the CPU modules
  - Connecting/disconnecting the tracking cable
  - Online module change of redundant function modules
- In the following cases, the setting is written only to own system.
  - The other system CPU module is off or reset.
  - A hardware failure (a failure of the CPU module or redundant function module) has occurred on the other system.
  - A tracking cable is disconnected or incorrectly connected, or failed.
  - Tracking communications stops due to an error in the CPU module.

Auto logging function

- The auto logging function cannot be used in the standby system. Even though an SD memory card that stores the data logging setting file for which the auto logging function has been enabled is inserted, the data logging is not started automatically.
- When the systems are switched during auto logging, the auto logging is completed.

29.18 CPU Module Data Backup/Restoration Function

The internal data of the CPU module in each own system is backed up and is restored only to the CPU module in the own system.

Point

In the CPU module of both systems, when the data which is backed up in the different system is restored, a system mismatch is detected. Restore the data that is backed up in the same system.

Differences in usage from the memory copy function

When combining the data of the standby system CPU module with the data of the control system CPU module, use the memory copy function. When changing the data of both systems at the same time, use the CPU module data backup/restoration function.

Function	Application
CPU module data backup/restoration function	When adding the same system as the redundant system that has already operated
	When restoring programs and parameters of both systems to the data before change due to trouble occurrence
Memory copy function (☞ Page 530 Memory Copy from Control System to Standby System)	When replacing only the standby system CPU module due to a failure and others
	When recovering the data due to error occurrence in the system consistency check

Backup Function

This section describes the backup function in redundant system.

Backup specifications

The following describes the backup system in a redundant system.

■System A/B setting information

The system A/B setting information is backed up. Restore the data included in the CPU module backup of each system.

Point

Manage SD cards so as to not accidentally restore the data to a different system by labeling the SD memory cards for backup with the label "For A system" or "For B system" respectively.

■System status

The system status is not backed up. Therefore, even if system B is backed up in the control system or system A is backed up in the standby system, each system starts up according to system A/B setting information.

■Operation mode

The operation mode is not backed up. Therefore, even if backed up in separate mode, the CPU module starts up in backup mode after restoration. Before starting up the CPU module in separate mode, change the operation mode to the separate mode after restoration.

■Setting to operate the CPU module in the status at data backup

Even if bit 15 of SD955 (Continuous operation with the status at backup) is on, the device/label data cannot be maintained in the status at data backup. (A control system CPU module starts to operate in the STOP status and changes to the RUN status after restoration of both systems. Therefore, the module changes the operating status in a similar manner.*1)

The SFC program and event history operate in the status at data backup.*2

Therefore, set this setting when using the SFC program and event history in the status at data backup on the system that has no problem even if the device/label data is operated from the initial status.

*1 For details on the operation of the CPU module at the operation status change, refer to the following.
☞ Page 96 Operation Processing When Operating Status Is Changed

*2 When the firmware version of the CPU module is different between backup and restoration, the SFC program will perform initial start regardless of selection for "Resume Start".

Precautions for backup function

The following describes the precautions for the backup function.

■Function cannot be executed simultaneously with backup function

If the backup function is executed in the state that the following function is executed, the backup function is completed with an error.

- System A and system B setting
- Memory copy from control system to standby system

When Retry execution for the automatic backup (bit 10 of SD944 (Backup function setting)) is set to on, retry the automatic backup. When the automatic backup cannot be executed due to the above functions operating even after the specified number of retries are attempted, the automatic backup is completed with an error.

When the above functions are executed during backup, errors in them are detected.

■Backup execution request

The tacking transfer does not cover SM1351 (CPU module data backup execution request). To perform the backup by turning on SM1351, turn on SM1351 both in the control system and standby system before executing a backup.

Restoration Function

This section describes the backup function in redundant system.

Automatic restoration using SD955

This section describes the procedure of the automatic restoration using SD955 in a redundant system

How to restore	Restoration operation	Description
Use the restoration function in both systems.	Start up the CPU module in both systems.	• Use this procedure when both systems can be powered on at the same time. • Use the two SD memory cards used at data backup. • After restoration, the backup data of both systems can be recovered.
	Start up the systems one by one.	• Use this procedure when both systems cannot be powered on at the same time because the systems are far apart. • Use the two SD memory cards used at data backup. • After restoration, the backup data of both systems can be recovered.
After restoring the data of one system, copy the data to the other system by memory copy.	—	• Use one of the SD memory cards used at data backup. • The target data for memory copy is copied from the control system CPU module to the standby system CPU module. (Data that is not target for memory copy will not be the data at data backup.) • In "Redundant Behavior Setting" of the CPU parameter, set "Enable" to "Auto Memory Copy Setting" and enable either or both of the control system start-up settings in "Control/Standby System Start-up Setting" beforehand. (The memory copy can be executed after the module starts as a control system.)

■Use the restoration function in both systems. (Start up the CPU modules of both systems at the same time.)

1. Set the restoration function to the CPU module of each system. (☞ Page 319 Automatic restoration using SD955)
2. Turn on the bit 0 of SD955 (Restoration function setting) to the CPU module of each system.
3. Power off and on or reset the CPU module of each system at the same time.

■Use the restoration function in both systems. (Start up the systems one by one.)

1. Set the restoration function to the CPU module of system A. (☞ Page 319 Automatic restoration using SD955)
2. Turn on the bit 0 of SD955 (Restoration function setting) to the CPU module of system A.
3. Power off and on or reset the CPU module of system A.*1
4. Set the restoration function to the CPU module of system B.
5. Turn on the bit 0 of SD955 (Restoration function setting) to the CPU module of system B.
6. Power off and on or reset the CPU module of system B.

*1 The system A that starts up earlier will wait for the start-up of the other system.

■After restoring the data of one system, copy the data to the other system by memory copy.

1. Enable the following settings in "Redundant Behavior Setting" of the CPU parameter beforehand: "Auto Memory Copy Setting" and either or both of the control system start-up settings under "Control/Standby System Start-up Setting".

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Redundant Behavior Setting]

2. Set the restoration function to the CPU module of the own system. (☞ Page 319 Automatic restoration using SD955)
3. Turn on the bit 0 of SD955 (Restoration function setting) to the CPU module of the own system.
4. In the configuration with the main base unit only, start up the standby system CPU module of the own system. In the redundant extension base unit configuration, the system does not wait for the start-up of the other system and is started up as the control system.
5. Check that the SD memory card is inserted into the CPU module of the other system, and then power off and on or reset the module.

Automatic restoration with the SD CARD OFF button

This section describes the procedure of the automatic restoration with the SD CARD OFF button in a redundant system.

How to restore	Restoration operation	Description
Use the restoration function in both systems.	Start up the systems one by one.	• Use the two SD memory cards used at data backup. • After restoration, the backup data of both systems can be recovered.
After restoring the data of the one system, copy the data to the other system by memory copy.	—	• Use one of the SD memory cards used at data backup. • The target data for memory copy is copied from the CPU module of control system to that of standby system. (Data that is not target for memory copy will not be the data at data backup.) • In "Redundant Behavior Setting" of the CPU parameter, set "Enable" to "Auto Memory Copy Setting" and enable either or both of the control system start-up settings in "Control/Standby System Start-up Setting" beforehand. (The memory copy can be executed after the module starts as a control system.)

■Use the restoration function in both systems. (Start up the systems one by one.)

1. Power on or reset the CPU module while pressing the SD CARD OFF button of the CPU module of system A.*1*2

2. Power on or reset the CPU module while pressing the SD CARD OFF button of the CPU module of system B.*1

*1 Release the SD CARD OFF button within 10 seconds after the READY LED begins to flash. If the switch is pressed more than 10 seconds, restoration may fail.

*2 The system A that starts up earlier will wait for the start-up of the other system.

■After restoring the data of one system, copy the data to the other system by memory copy.

1. Enable the following settings in "Redundant Behavior Setting" of the CPU parameter beforehand: "Auto Memory Copy Setting" and either or both of the control system start-up settings under "Control/Standby System Start-up Setting".

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Redundant Behavior Setting]

2. Set the restoration function to the CPU module of the own system. (☞ Page 320 Automatic restoration with the SD CARD OFF button)

3. Power on or reset the CPU module while pressing the SD CARD OFF button of CPU module of the own system.*1

4. In the configuration with the main base unit only, start up the standby system CPU module of the own system. In the redundant extension base unit configuration, the system does not wait for the start-up of the other system and is started up as the control system.

5. Check that the SD memory card is inserted into the CPU module of the other system, and then power off and on or reset the module.

*1 Release the SD CARD OFF button within 10 seconds after the READY LED begins to flash. If the switch is pressed more than 10 seconds, restoration may fail.

Precautions for restoration

This section describes the precautions for restoration.

■Target backup data

Restore the backup data in the CPU module of each system in the same redundant system as at backup processing. If the backup data is restored in a system different from at backup processing, an error may be detected using system consistency check.

■When restoring automatically by the procedure to start up each system separately

Set "Other system Start-up Timeout Setting" of "CPU Parameter" to "Not Set". When setting to "Set", an error may be detected during restoration of the CPU module of the other system.

■When starting up the CPU module by automatic restoration

When starting up the CPU module by automatic restoration in the redundant extension base unit configuration, it is recommended to use the automatic recovery function of the CPU module of the standby system.

In the following operations performed by automatic restoration, the CPU module of the system that was started up first cannot perform tracking communications during initialization. Therefore, a stop error may occur in the CPU module of the system that was started up later.

- The CPU modules of both systems are started up simultaneously, but the start-up timing of one system is delayed due to the time taken for restoration.
- In the procedure for starting up systems one by one, one system is started up while the other system has been started up.

In this case, if the automatic recovery function of the CPU module of the standby system is used, the system can be recovered by automatically restarting the CPU module with a stop error. If the automatic recovery function of the CPU module of the standby system is not used, the CPU module with a stop error must be restarted by manual operation.

29.19 Device/label Access Service Processing Constant Wait Function

This section describes operation in redundant mode and the setting method.

Operation in redundant mode

The following describes operation in redundant mode.

■Backup mode

This item is enabled only for the control system CPU module.

■Separate mode

This item is enabled for the CPU modules of both systems.

Setting method

The following describes the setting method in redundant mode.

■Backup mode

Set SM315 (Service processing constant wait setting flag) and SD315 (Service processing constant wait status setting) of the CPU module of the control system. The setting details of the CPU module of the control system are reflected in the CPU module of the standby system by tracking transfer.

■Separate mode

Set SM315 (Service processing constant wait setting flag) and SD315 (Service processing constant wait status setting) of each CPU module.

29.20 SLMP Communication

The following describes the precautions on SLMP communications.

System switching

There are the following notes when the system IP address matching function is not used.

■Re-setting of the connection destination

When the relay CPU module is in the communication-disabled state (power-off, reset, or tracking cable disconnection) at system switching, the connection destination needs to be set again for SLMP communications.

■Re-execution of the write command

When "Control System" or "Standby System" has been selected in the transfer setup and the systems are switched, a target system mismatch is detected by a command issued during the system switching, and a communication error occurs. If a communication error occurs while a data write command is being issued, the data write command needs to be issued for the new control system.

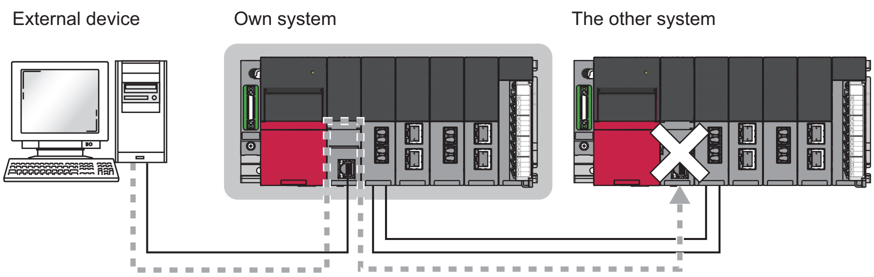
29

Remote operation

If a remote operation command is executed, the CPU modules enter different operating statuses and thus systems cannot be switched.

Communicating with other systems

For SLMP communications via the built-in Ethernet port of the CPU module, when communications are performed to the other system that cannot respond (power-off, reset, or tracking cable disconnection), a timeout error may occur.



30 PRECAUTIONS ON PROGRAMMING

This chapter describes the precautions on programming for a redundant system.

30.1 Instructions Not Available in Redundant System

This section describes the instructions not available in a redundant system.

Instructions that cause stop errors

Do not use the following instructions in backup mode. Doing so causes an error when the operating status of the CPU module is changed from STOP to RUN.

Classification		Instruction symbol
Special counter instructions	Counting up or down the current value (1-phase input)	UDCNT1
	Counting up or down the current value (2-phase input)	UDCNT2
Special timer instructions	Teaching timer	TTMR
	Special function timer	STMR
Shortcut control instruction	Rotary table shortest direction control	ROTC
Ramp signal instruction	Ramp signal	RAMPQ
Pulse related instructions	Measuring the density of pulses	SPD
	Outputting pulses at regular intervals	PLSY
	Performing the pulse width modulation	PWM
Matrix input instruction	Matrix input	MTR
Multiple CPU dedicated instructions	Reading device data from another CPU module	D.DDRD
		DP.DDRD
		M.DDRD
		MP.DDRD
	Writing device data to another CPU module	D.DDWR
		DP.DDWR
		M.DDWR
		MP.DDWR

Instructions that need to be executed again in a new control system

For an instruction that requires several scans for completing the processing, the instruction will be continuously executed when the system switching is performed during execution of the instruction. When a completion device has been used in an execution program of the control system, the completion device will not turn on even though the instruction is completed after the system switching from the control system to the standby system. However, the completion device will turn on after the system switching from the control system to the standby system and then to the control system again. Note that the completion state of the completion device is not reflected to the tracking device. When the system switching is performed during execution of an instruction, execute the instruction again as required.

Classification	Instruction symbol
Data processing instructions	SORTD(_U), DSORTD(_U)
Reading/writing data instructions	SP.DEVST, SP.FREAD, SP.FWRITE
Open/close processing instructions	SP.SOCOPEN, SP.SOCCLOSE
Socket communications instructions	SP.SOCRCV, S.SOCRCVS, SP.SOCSND, SP.SOCCINF, SP.SOCCSET, SP.SOCRMODE, S(P).SOCRDATA
SLMP frame send instruction	SP.SLMPSND
File transfer function instruction	SP.FTPPUT, SP.FTPGET
Module dedicated instructions	Instructions that require several scans for completing processing

For some instructions, an error will occur if an instruction is executed during execution of the same instruction. For the operation for each instruction that is executed again during execution of the same instruction, refer to the following.

📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

📖 MELSEC iQ-R Programming Manual (Module Dedicated Instructions)

Re-execution of instruction when systems are switched during instruction execution

When the system switching is performed while an instruction that requires several scans is being executed, the instruction can be executed again in the new control system after the system switching by using such as the following programs.

■REMFR instruction

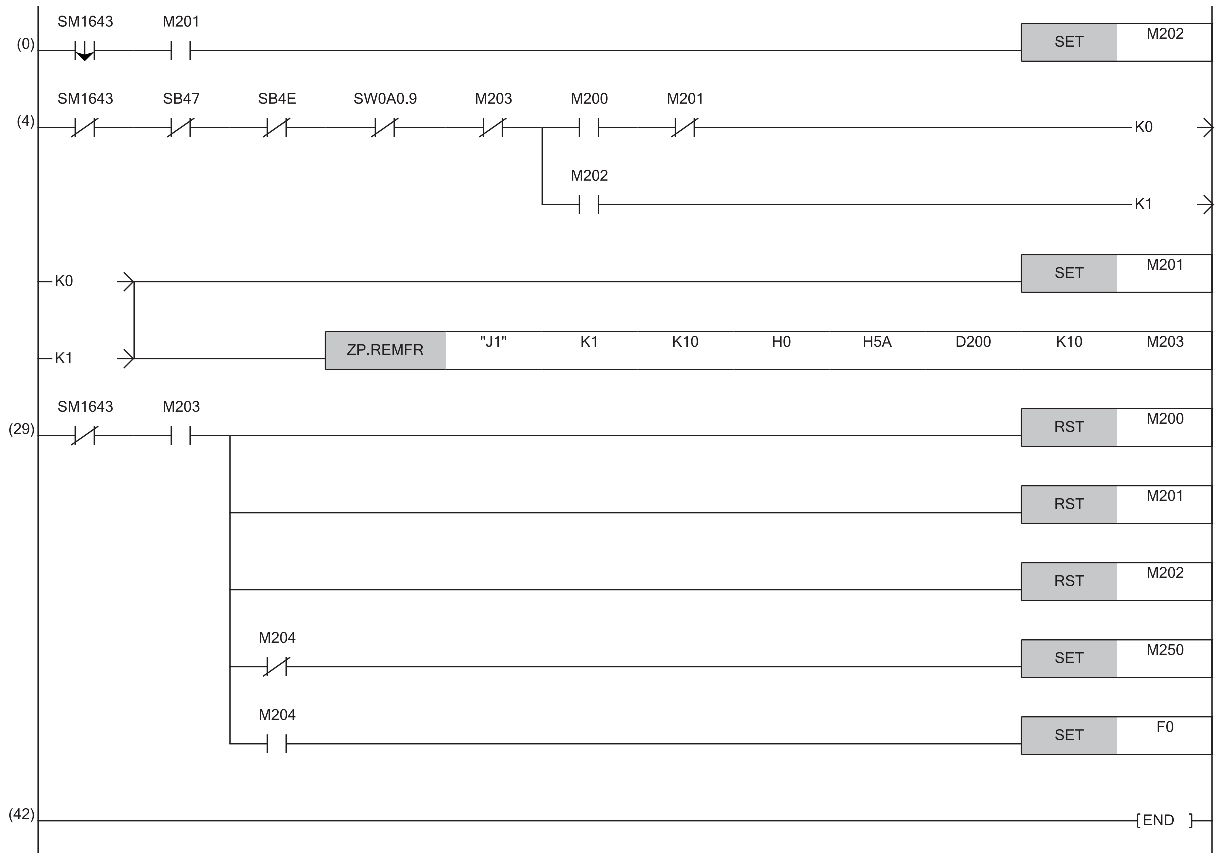
When the system switching is performed while the instruction is being executed (M201 = ON), SM1643 (ON only in one scan after system switching (standby to control)) will turn on for one scan in the new control system and the REMFR instruction will be executed again on the station number 10 of the network number 1.

- DEVICES

DEVICES	Data stored
SM1643	System switching check flag (standby system to control system)
SB47*1	Baton pass status of own station
SW0A0.9*1	Baton pass status of each station
M200*2	On: Reading request
M201*2	At instruction execution
M202*2	ON at instruction re-execution request due to system switching
M203*2	Completed without an error
M204*2	Completed with an error

*1 For details on the link special relays (SB) and link special registers (SW), refer to the manuals for the network used.
*2 Change the device number according to the system.

- Program example



■REMTO instruction

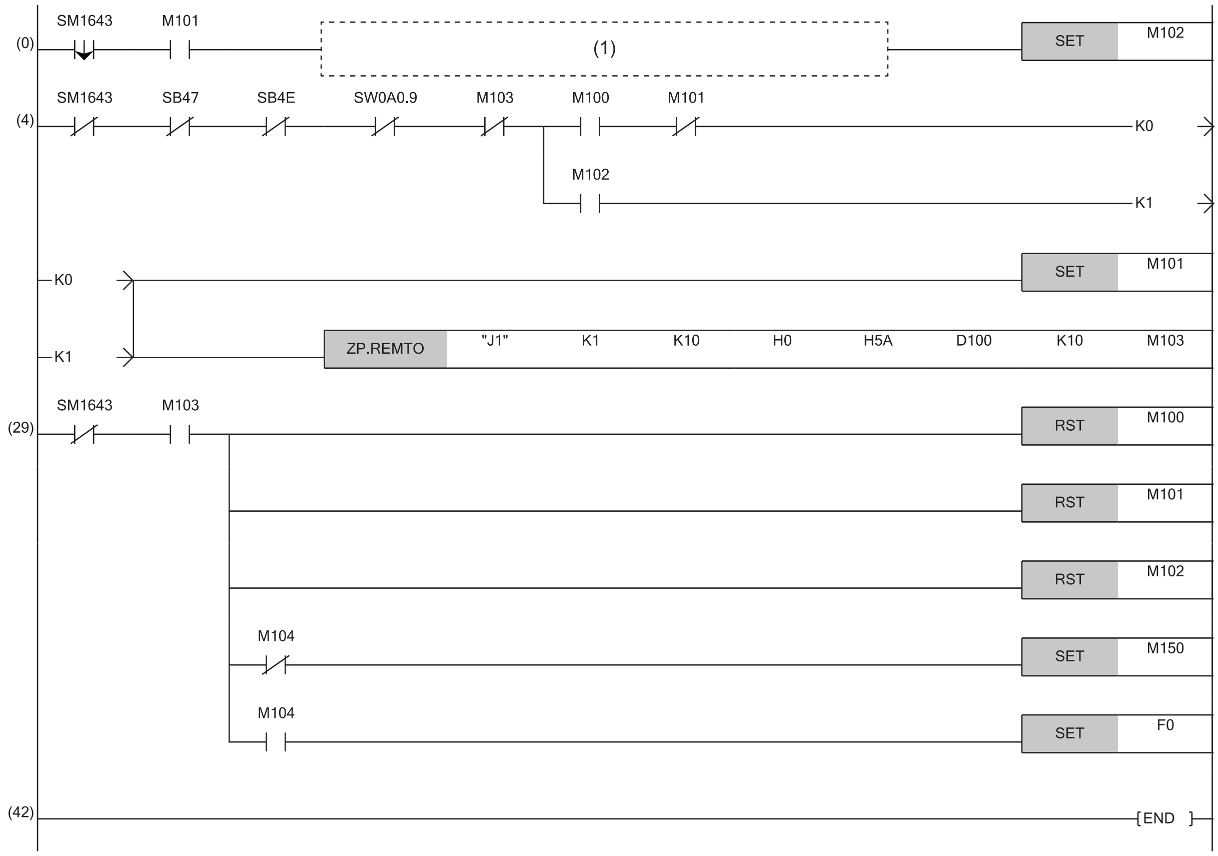
When the system switching is performed while the instruction is being executed (M101 = ON), SM1643 (ON only in one scan after system switching (standby to control)) will turn on for one scan in the new control system and the REMTO instruction will be executed again on the station number 10 of the network number 1. When the system switching is performed while a write instruction such as the REMTO instruction is being executed, execution of the instruction may have been suspended before completion of writing data to the target module. Thus, insert an interlock in the new control system to read the X signals and buffer memory status of the target module and to determine whether or not to execute the instruction again.

- DEVICES

DEVICES	Data stored
SM1643	System switching check flag (standby system to control system)
SB47*1	Baton pass status of own station
SW0A0.9*1	Baton pass status of each station
M100*2	Write request
M101*2	At instruction execution
M102*2	ON at instruction re-execution request due to system switching
M103*2	Completed without an error
M104*2	Completed with an error

*1 For details on the link special relays (SB) and link special registers (SW), refer to the manuals for the network used.
*2 Change the device number according to the system.

- Program example



(1) Add an interlock circuit for determining whether or not to execute the instruction again (according to the X signals and buffer memory status of the target module) as required.

Instructions whose operations vary depending on tracking of the signal flow memory

The following describes the instructions whose operations after the system switching vary depending on whether the signal flow memory is tracked or not in backup mode. The operations vary when one of the following instructions is executed among program organization units that have the signal flow memory, memory to which tracking can be performed.

Classification/type of special relay	Instruction symbol
Rising instruction	LDP, ANDP, ORP, LDPI, ANDPI, ORPI, PLS, MEP, EGP, SET F, RST F, FF, LEDR, DUTY, LOGTRG, LOGTRGR, □P (such as MOVP and INCP), SP.□, JP.□, GP.□, ZP.□
Falling instruction	LDF, ANDF, ORF, LDFI, ANDFI, ORFI, PLF, MEF, EGF
SCJ instruction	SCJ
Data processing instructions	SORTD(_U), DSORTD(_U)
TIMCHK instruction	TIMCHK
XCALL instruction	XCALL
Rising instruction using SM1643 as an execution condition	—

Rising instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, a rising instruction whose execution condition turned on during system switching will not be executed.

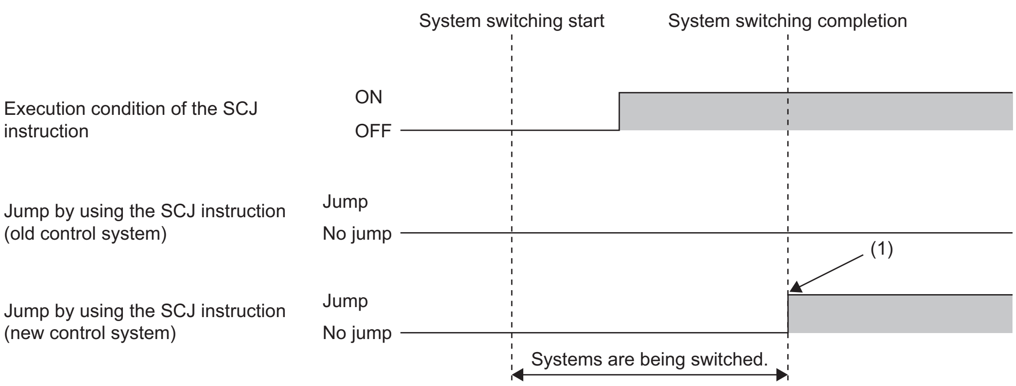
Falling instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, a falling instruction whose execution condition turned off before the system switching will be executed.

SCJ instruction

■When the signal flow memory is not tracked

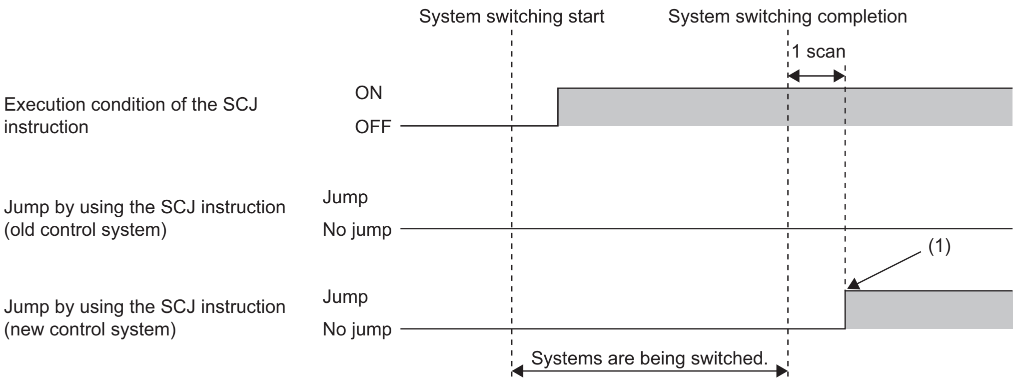
Once the system switching is performed, the execution condition of the SCJ instruction turns on after the signal flow memory of the new control system turns on. In the new control system, processing jumps to the pointer specified by the SCJ instruction in the first scan.



(1) After system switching, the processing jumps in the first and later scans.

■When the signal flow memory is tracked

When the system switching is performed, the execution condition of the SCJ instruction turns on while the signal flow memory remains off. In the new control system, the processing jumps to the pointer specified by the SCJ instruction in the second scan.



(1) After system switching, the processing jumps in the second and later scans.

Data processing instructions

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, the SORTD(_U)/DSORTD(_U) instruction in the first scan after the system switching will be executed not as the first execution but as continuous execution (continuous processing). When the instruction is executed for the first time, the instruction is executed without data to be stored in the devices used by the system, causing an unintended operation.

TIMCHK instruction

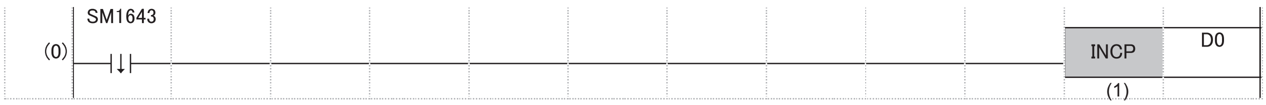
If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, the TIMCHK instruction in the first scan after the system switching will be executed not as the first execution but as continuous execution (continuous processing). At the first execution, the current value is not cleared and the device that turns on at timeout is not turned off. The instruction is executed with the status at the previous measurement.

XCALL instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, the subroutine program will not be executed when the execution condition of the XCALL instruction remains off in the first scan after system switching.

Rising instruction using SM1643 as an execution condition

SM1643 is the special relay that turns on for one scan in the new control system after system switching. If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on and a rising instruction cannot be executed. To execute a rising instruction where SM1643 has been set as an execution condition, use a falling edge pulse operation contact (LDF/AND/ORF instruction) and create a program in which the rising instruction is to be executed at the falling edge of SM1643, as shown below. However, when the falling edge of SM1643 has been set as the execution condition, the target instruction will be executed in the second scan after the system switching.



(1) The instruction will be executed in the second scan after the system switching.

Instructions that affect the status of another instruction when executed

When one of the following instructions is executed and the status of another instruction changes, the new status will not be tracked to the other system. When the system switching is performed during execution of an instruction, execute the instruction again as required.

Classification		Instruction symbol
Program execution control instructions	Disabling interrupt programs	DI
	Enabling interrupt programs	EI
	Disabling interrupt programs with specified priority or lower	DI
	Interrupt program mask	IMASK
	Disabling/enabling the specified interrupt pointer	SIMASK
File register operation instructions	Switching the file register block number	RSET(P)
	Changing the file register file name	QDRSET(P)
Timing check instruction	Generating timing pulses	DUTY
SFC control instruction	Target block switching	BRSET

Instruction that causes different operation results between the control system and standby system

The following describes the instruction that causes different operation results in both systems after system switching.

PID control instruction

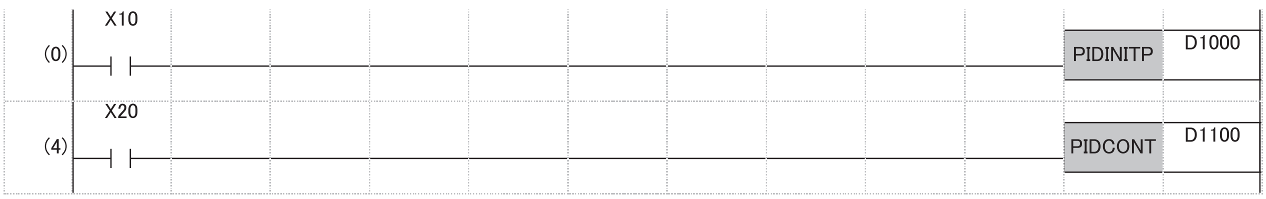
When using the following PID control instructions, include the number of device points used by the systems in tracking target data. Otherwise, the instructions cause different operation results between the control system and standby system after system switching.

n: Total number of loops

Classification		Instruction symbol	Number of device points used
PID control instructions (Inexact differential)	Registering the PID control data to the CPU module	S(P).PIDINIT	2 + n × 4
	PID operation	S(P).PIDCONT	10 + n × 23
PID control instructions (Exact differential)	Registering the PID control data to the CPU module	PIDINIT(P)	2 + n × 10
	PID operation	PIDCONT(P)	10 + n × 18

Ex.

When the number of loops is eight, the PIDINITP instruction needs 2 + 8 × 10 = 82 words and the PIDCONT instruction needs 10 + 8 × 18 = 154 words. Thus, include D1000 to D1081 and D1100 to D1253 in tracking target data.



Precautions for using the COM or ZCOM instruction

When refresh is performed by using the COM or ZCOM instruction, output from the remote I/O station or a module on the extension base unit may change after system switching. To prevent this, do not perform refresh with the COM or ZCOM instruction. For the COM instruction, whether or not to perform refresh can be set using SM775 (Selection of refresh processing during the COM instruction execution) and SD775 (Selection of refresh processing during the COM instruction execution). Set SM775 and SD775 and perform only the device/label access service processing with the COM instruction.

○: Selectable, ×: Not selectable

Instruction	Refresh processing	In a redundant system
COM instruction	I/O refresh	○*1
	Network module link refresh	○*1
	Intelligent function module refresh	○*1
	Refresh using the CPU buffer memory of the multiple CPU system (END)	×*2
	Device/label access service processing (communications with the engineering tool, GOT, or other external devices)	○
ZCOM instruction	Network module link refresh	○*1
	Intelligent function module refresh	○*1

*1 At the timing when the COM or ZCOM instruction is executed, tracking transfer is not performed. Thus, when the system switching is attempted after execution of the instruction and before completion of tracking transfer, the system switching is performed without tracking transfer. Thus, even though output to the module is changed by the COM or ZCOM instruction with the CPU module in the control system, the change will not be reflected to the CPU module in the standby system, but after the systems are switched, the CPU module in the new control system outputs the signals before the system switching, the output may change.

*2 This processing is not selectable because a multiple CPU system cannot be built in a redundant system.

Precautions for using the ADRSET instruction

Even though a file is written to both the control system and standby system, different addresses are assigned to the systems. To continue the processing in the new control system, use the ADRSET instruction to obtain indirect addresses again.

30.2 Interrupt from Modules

The following describes the precautions for interrupts from modules.

System switching in backup mode

When the old control system is switched to the new standby system

The old control system retains the interrupt factors that have occurred even after the system is switched to the new standby system by system switching before execution of an interrupt program. After the systems are switched again, the interrupt program of an interrupt factor that the old control system has retained will be executed.
Since the interrupt factor that has occurred on the old control system is not inherited to the new control system, the interrupt program of an interrupt factor that has occurred on the old control system will not be executed on the new control system.

When the old standby system is switched to the new control system

The old standby system retains the interrupt factors that have occurred. After the old standby system is switched to the new control system by system switching, the interrupt program of an interrupt factor that the old standby system has retained will be executed on the new control system.
When the old standby system has retained multiple interrupt factors, the scan time may greatly increase.

System switching in separate mode

When the old control system is switched to the new standby system

The old control system executes the interrupt factors that the old control system has retained even after the system is switched to the new standby system by system switching before execution of an interrupt program.

When the old standby system is switched to the new control system

The old standby system retains the interrupt factors that have occurred. The new control system executes the interrupt program of an interrupt factor that the old standby system has retained regardless of the system switching operation.

When the operation mode is switched to the separate mode

For the control system

The control system retains the interrupt factors before switching to the separate mode. After the operation mode is switched to the separate mode, the control system executes the interrupt programs of the interrupt factors that have occurred in the backup mode.

For the standby system

The interrupt factors that the standby system has retained before switching to the separate mode will be discarded. Therefore, after the operation mode is switched to the separate mode, the interrupt programs of the interrupt factors that have occurred before switching to the separate mode will not be executed.

When the operation mode is switched to the backup mode

For the control system

When the interrupt programs of the interrupt factors before switching to the backup mode have not been executed yet, the control system will execute the programs that the system has retained in the separate mode.

For the standby system

When the interrupt programs of the interrupt factors before switching to the backup mode have not been executed yet, the standby system will retain the interrupt factors that the system has retained in the separate mode. Interrupt programs will not be executed.

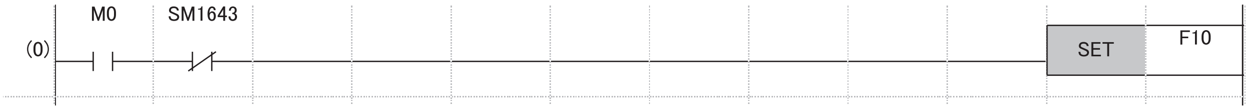
30.3 Precautions for Using the Annunciator (F)

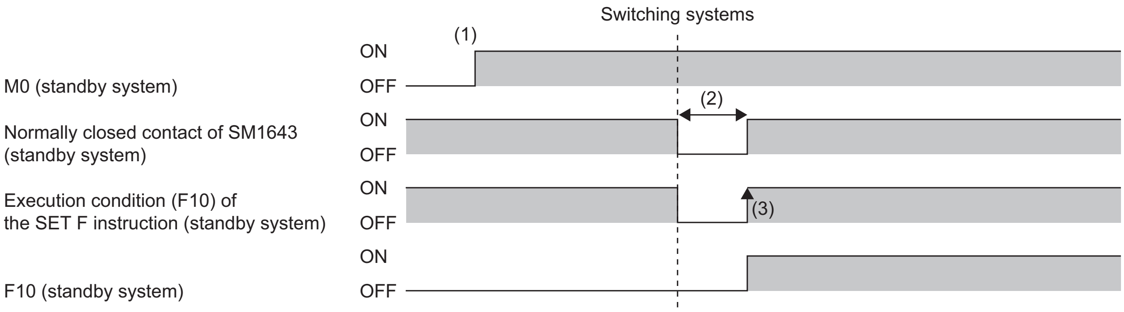
This section describes the precautions for using the annunciator (F).

When the SET F instruction is used to register the annunciator

When the execution condition of the SET F□ instruction has been satisfied at system switching, the annunciator that is turned on by the SET F□ instruction can be registered in the new control system. The SET F□ instruction registers the annunciator at the rising edge of the execution condition. To register the annunciator in the new control system at system switching, add a normally closed contact of SM1643 (ON for only one scan after system switching (standby system to control system)) as the AND condition in the execution conditions of the SET F□ instruction.

Ex.





(1) When M0 turns on in the control system before system switching, M0 of the standby system also turns on as a result of tracking transfer.
(2) The contact turns off for one scan after system switching.
(3) When the execution condition turns on, annunciator information is registered by the SET F□ instruction.

When the OUT F instruction is used to register the annunciator

When the execution condition of the OUT F□ instruction has been satisfied at system switching, annunciator information is registered in the new control system at system switching.

30.4 Precautions on Timers and Timer Function Blocks

The following describes the precautions on timers and timer function blocks at system switching.

Current values at system switching

For the timer (T), retentive timer (ST), and a timer function block TIMER_□_M, the current values of the timers in the first scan of the CPU module of the new control system will not be updated after system switching.

Timeout before system switching

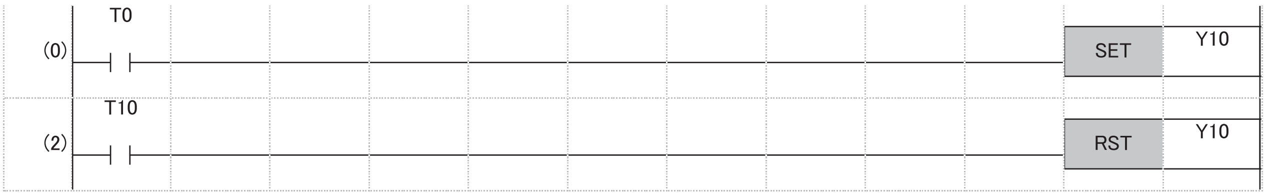
Depending on the timing to perform system switching such as power-off, tracking transfer processing is suspended and tracking data may not be reflected to the CPU module in the new control system. For the timer (T), retentive timer (ST), long timer (LT), long retentive timer (LST), and timer function blocks TIMER_□_M, TP(_E), TON(_E), and TOF(_E), the timer whose time has been up before system switching may go into the state in which no timeout has occurred in the first scan after system switching.

When values are output (writing values to the buffer memory and the output (Y)) with a timer contact or an output variable, the timer statues may go into the state in which no timeout has occurred as described above, causing chattering of the output. To transfer data with modules or external devices using the output (Y) or buffer memory, a program may not properly function due to chattering of output after the system switching. To output values (writing data to the buffer memory and the output (Y)) to modules or external devices with a timer contact or an output variable, consider the time taken for data to be transferred from the CPU module in the control system to the CPU module in the standby system after the time is up.

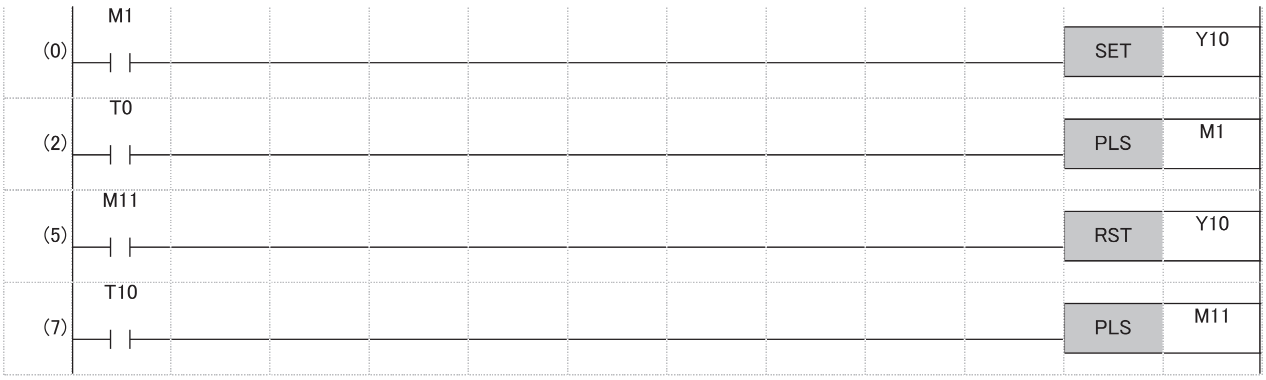
Ex.

Program that delays outputting values by one scan after the time of the timer (T) is up

[Without measures]



[With measures]



Point

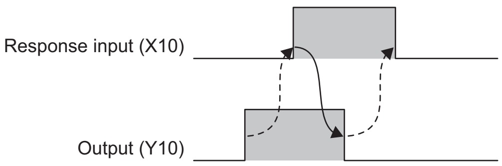
In the CPU parameter, select "Transfer" (default setting) in "Signal Flow Memory Tracking Setting" of "Redundant System Settings". (☞ Page 521 Tracking transfer setting for the signal flow memory)

30.5 Precautions on Access to Intelligent Function Module or External Devices

Depending on the timing of system switching cause to be caused, such as power-off, tracking processing is suspended and device/label data may not be applied to the CPU module in the new control system after the system switching.
Consequently, output data may differ from device/label data of the CPU module of the new control system. In communication with intelligent function modules or external devices using the output (Y) or buffer memory, programs may not properly function due to a mismatch in the device data after system switching.
For command output (such as output (Y), startup by writing data to the buffer memory, and clear) to the intelligent function module and external devices, consider the time taken for tracking the execution condition of command output from the CPU module in the control system to the CPU module in the standby system.
The following example shows a program that outputs data one scan later after the command output condition is satisfied.

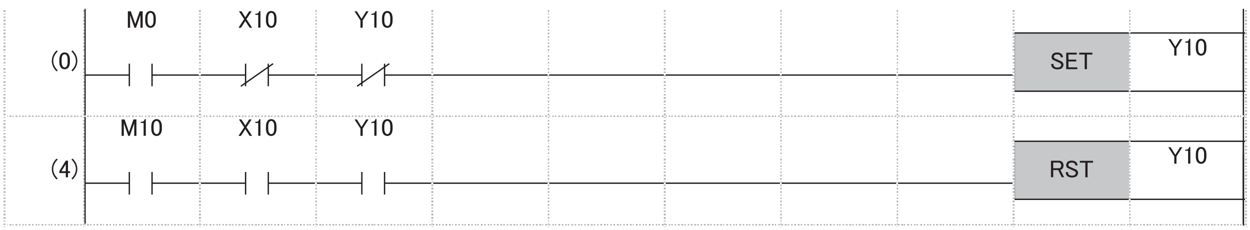
Ex.

When response input is returned to output

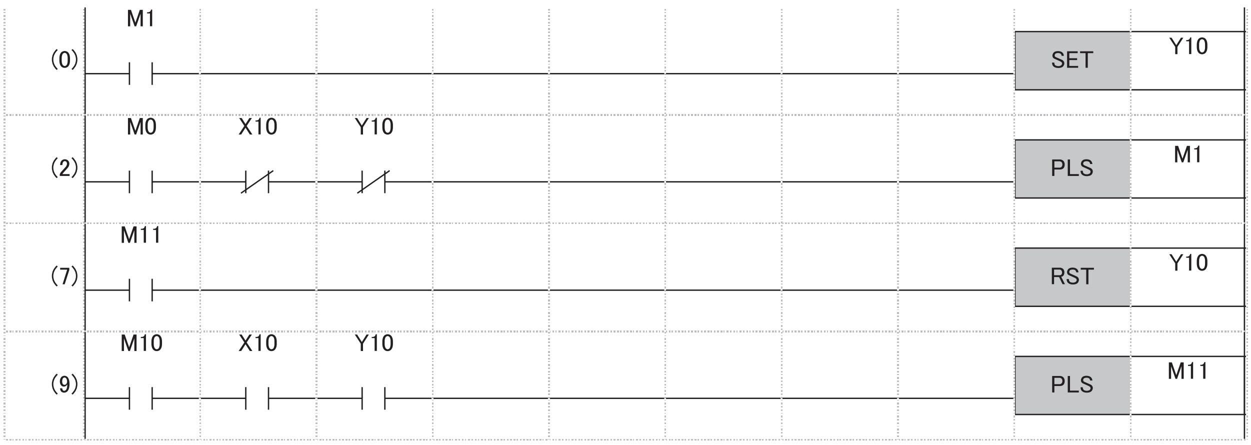


In the following program, turning on M0 turns on the output (Y10) and turning on M10 turns off the output (Y10).

[Without measures]



[With measures]



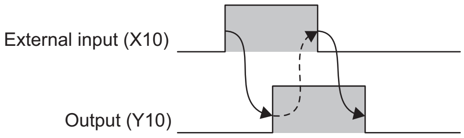
(2) PLS M1 delays SET Y0 by one scan.
(9) PLS M11 delays RST Y10 by one scan.

Point

In the CPU parameter, select "Transfer" (default setting) in "Signal Flow Memory Tracking Setting" of "Redundant System Settings". (☞ Page 521 Tracking transfer setting for the signal flow memory)

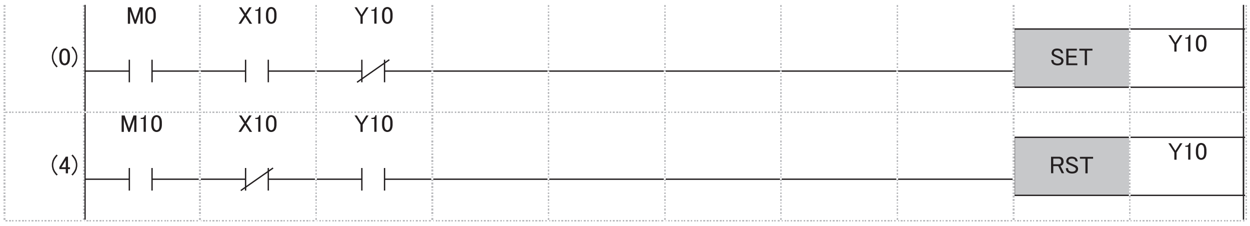
Ex.

When output is returned to external input

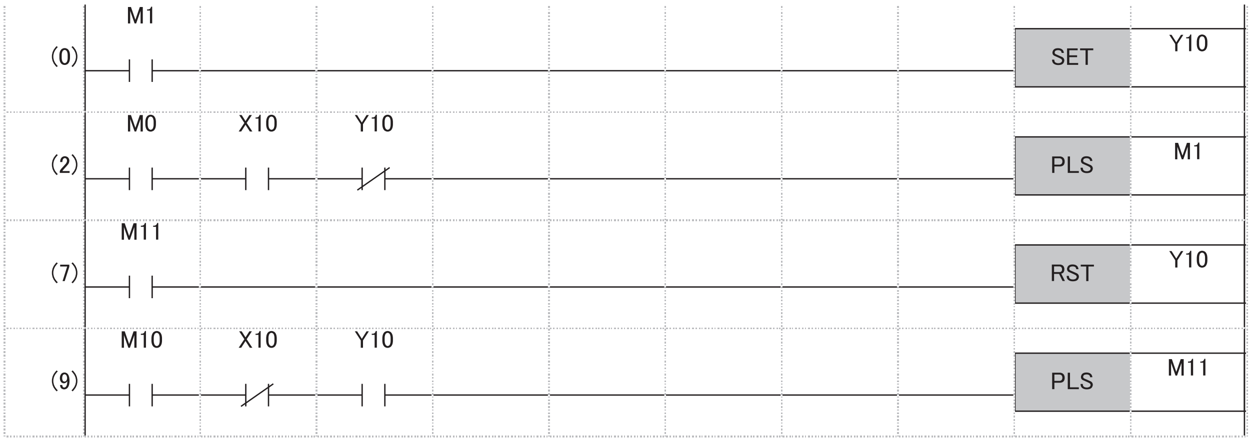


In the following program, turning on M0 turns on the output (Y10) and turning on M10 turns off the output (Y10).

[Without measures]



[With measures]



(2) PLS M1 delays SET Y10 by one scan.

(9) PLS M11 delays RST Y10 by one scan.

Point

In the CPU parameter, select "Transfer" (default setting) in "Signal Flow Memory Tracking Setting" of "Redundant System Settings". (☞ Page 521 Tracking transfer setting for the signal flow memory)

30.6 Precautions on Writing Data from GOT or External Devices

When data is written from the GOT or external devices, the tracking data may not be applied to the CPU module of the new control system depending on the timing of system switching cause to be caused, such as power-off.
Consequently, data that is written from the GOT or external devices before system switching may be lost. Write the data again after the system switching.

30.7 Precautions on Outputting in the Middle of the Scan

When values are output during execution of a program with the following devices or setting, the program will be executed again from the step 0 in the new control system after system switching. Therefore, values may be output twice, before and after system switching. Output results may differ between the two outputs.

- Link direct device (Jn\Y)
- Direct access output (DY)
- Refresh at execution of a specified program ("Refresh Group Setting" of "Program Setting")

Before outputting values with one of the above devices or setting, check that the above operation does not cause any problem in the system design phase. If any problem occurs, do not perform the external output until the status of the received output signal becomes stable, or take other measures with external circuits.

Point

In the program of the redundant system, performing the output refresh by the END processing is recommended instead of outputting with the link direct device (Jn\Y), direct access output (DY), or refresh at execution of a specified program ("Refresh Group Setting") in the middle of the scan.

30.8 Precautions for the redundant extension base unit configuration

This chapter describes the precautions on programming in the redundant extension base unit configuration.

Dedicated instruction

Dedicated instruction cannot be used for modules mounted on the extension base unit. An error occurs if the dedicated instruction is executed. Note that the error can be set as a continuation error in the CPU module operation setting for error detection under the RAS setting of the CPU parameter. (☞ Page 141 CPU module operation upon error detection setting)

Module label

When using a module label for a module mounted on the extension base unit, transfer the tracking data of the module label (extension base unit). (☞ Page 513 Tracking Transfer)

Module function block

In the module function block for a module mounted on the extension base unit, do not use a module function block that uses dedicated instructions. When using a module function block that does not use dedicated instructions, transfer the tracking data of the module label (extension base unit). (☞ Page 513 Tracking Transfer)

When using input values in the program

When the following devices/labels are used in the program, the values of the old control system are inherited even after system switching. Therefore, transfer the tracking data.

- Input devices (X)/labels assigned to a module on the extension base unit
- Devices/labels where automatic refresh settings are made for the intelligent function module on the extension base unit
- Devices/labels where automatic refresh settings are made for the CC-Link module on the extension base unit

31 MAINTENANCE AND INSPECTION FOR A REDUNDANT SYSTEM

This chapter describes the maintenance and inspection for a redundant system.

31.1 Module Replacement in the Redundant System

This section describes the module replacement in the redundant system.

Overview

■Replacing a module on the main base unit of the control system

- A module can be replaced by turning on SM1646 (System switching by a user) to execute system switching and powering off the main base unit in the standby system after system switching.
- A module supporting the online module change function can be replaced by online module change.

■Replacing a module on the main base unit of the standby system

- A module can be replaced by powering off the main base unit of the standby system.
- In a redundant system without extension base units, a module supporting the online module function change can be replaced by online module change. (In a redundant system with redundant extension base unit, a module on the main base unit of the standby system cannot be replaced by online module change.)

■Replacing a module on the extension base unit

- A module supporting the online module change function can be replaced by online module change.
- When replacing a module not supporting the online module change function, power off both systems before replacing.

Replacement of each module

The following is the list indicating whether each module can be replaced.

○: Replaceable, ×: Not replaceable, —: Inapplicable

Replacement target	Replacement							Reference
	Redundant system without extension base units			Redundant system with redundant extension base unit				
	Module on the main base unit of the control system	Module on the main base unit of the standby system		Module on the main base unit of the control system	Module on the main base unit of the standby system		Module on the extension base unit	
	Replacement while power is on	Replacement after power is turned off	Replacement while power is on	Replacement while power is on	Replacement after power is turned off	Replacement while power is on	Replacement while power is on	
CPU module	×*2	○	×	×*2	○	×	—	Page 599 Replacing a CPU module
Power supply module	×*2	○	×	×*2	○	×	×	Page 601 Replacing a power supply module
Redundant power supply module	○*1	○	○*1	○*1	○	○*1	○*1	Page 601 Replacing a redundant power supply module

Replacement target	Replacement							Reference
	Redundant system without extension base units			Redundant system with redundant extension base unit				
	Module on the main base unit of the control system	Module on the main base unit of the standby system		Module on the main base unit of the control system	Module on the main base unit of the standby system		Module on the extension base unit	
	Replacement while power is on	Replacement after power is turned off	Replacement while power is on	Replacement while power is on	Replacement after power is turned off	Replacement while power is on	Replacement while power is on	
Redundant function module	○	○	○	○	○	×	—	Page 602 Replacing a redundant function module
I/O module	○	○	○	○	○	×	○	Page 602 Replacing I/O modules
Intelligent function module	○*3*4	○	○*3	○*3*4	○	×	○*3	Page 603 Replacing an intelligent function module

*1 When the power supply module is redundant, only one of the two power supply modules can be replaced.
*2 A module can be replaced by turning on SM1646 (System switching by a user) to execute system switching and powering off the main base unit in the standby system after system switching.
*3 A module supporting the online module change function can be replaced by online module change. (MELSEC iQ-R Online Module Change Manual)
*4 When replacing a module not supporting the online module change function, the module can be replaced by turning on SM1646 (System switching by a user) to execute system switching and powering off the main base unit in the standby system after system switching.

Point

- When a module is replaced after powering off the standby system, a continuation error occurs on the CPU module of the control system. After the replacement, clear the continuation error that has occurred on the CPU module of the control system.
- When replacing the CPU module, if functions added or changed by the upgrade are used, replace it with a CPU module with a firmware version that supports the functions used. (Page 1139 Added and Enhanced Functions)

■Replacement/addition of a base unit and extension cable

The following is the list indicating whether a base unit and extension cable can be replaced/added.

Replacement target		Replacement	Reference
Main base unit	Main base unit of the standby system	Can be replaced by powering off the main base unit of the standby system.	Page 603 Replacing a main base unit
	Main base unit of the control system	Can be replaced by turning on SM1646 (System switching by a user) to execute system switching and powering off the main base unit in the standby system after system switching.	
Extension base unit		The extension base unit cannot be turned off and replaced while the system is operating.	—
Extension cable	Between the main base unit of the standby system and the extension base unit	Can be replaced by powering off the main base unit of the standby system.	Page 604 Replacement procedure for extension cable between main base unit and extension base unit
	Between the main base unit of the control system and the extension base unit	Can be replaced by turning on SM1646 (System switching by a user) to execute system switching and powering off the main base unit in the standby system after system switching.	
	Between extension base units	Only the inactive extension cable can be replaced. If the extension cable on the active side is removed, system switching occurs, and the cable cannot be replaced.	Page 604 Replacement procedure for extension cable between extension base units

Replacing a CPU module

The following describes the procedure for replacing the CPU module of the standby system after turning the power off while the redundant system is operating. When the replacement target is in the control system, turn on SM1646 (System switching by a user) in advance to execute system switching before switching to the standby system.
The replacement procedure differs depending on whether the automatic memory copy function is enabled or disabled. (☞ Page 533 Automatic memory copy)

When the automatic memory copy function is enabled

1. Check the operation mode.

Check the BACKUP LED of the redundant function module in the control system to check if the CPU module in the control system is in the backup mode.

- The BACKUP LED of the redundant function module in the control system is on or flashes.

2. Check the system of the replacement target CPU module.

Check the CTRL LED and SBY LED of the redundant function module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

3. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes.

4. Replace the CPU module in the standby system.

Replace the CPU module in the standby system with a new CPU module whose model is the same as that of the CPU module in the control system. If an SD memory card or an extended SRAM cassette has been inserted to the replacement target CPU module, insert it to the new CPU module.
When using functions added or changed by the upgrade, replace with a CPU module with a firmware version that supports the functions used. (☞ Page 1139 Added and Enhanced Functions)

5. Change the switch status of the CPU module in the standby system.

Set the RUN/STOP/RESET switch of the CPU module in the standby system to the RUN position.

6. Power on the standby system.

Power on the standby system.

7. Memory copy from control system to standby system

The memory is automatically copied. When the memory copy is completed, the MEMORY COPY LEDs of the redundant function modules of both systems turn off. Then, the CPU module in the standby system is automatically reset and restarted. (☞ Page 533 Automatic memory copy)
The BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

When the automatic memory copy function is disabled

1. Check the system of the replacement target CPU module.

Check the CTRL LED and SBY LED of the redundant function module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace the CPU module in the standby system.

Replace the CPU module in the standby system with a new CPU module whose model is the same as that of the CPU module in the control system. If an SD memory card or an extended SRAM cassette has been inserted to the replacement target CPU module, insert it to the new CPU module.

When using functions added or changed by the upgrade, replace with a CPU module with a firmware version that supports the functions used. (☞ Page 1139 Added and Enhanced Functions)

4. Change the switch status of the CPU module in the standby system.

Set the RUN/STOP/RESET switch of the CPU module in the standby system to the RUN position.

5. Power on the standby system.

Power on the standby system.

6. Memory copy from control system to standby system

The memory is copied with the engineering tool or the special relay and special register. (☞ Page 534 Memory copy with the engineering tool, ☞ Page 535 Memory copy with the special relay and special register)

When the memory copy is completed, the MEMORY COPY LED of the redundant function module in the control system turns off and that in the standby system turns on.

7. Restart the standby system.

Restart the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

Replacing a power supply module

The following describes the procedure for replacing the power supply module on the main base unit of the standby system by turning the power off while the redundant system is operating. When the replacement target is in the control system, turn on SM1646 (System switching by a user) in advance to execute system switching before switching to the standby system.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace the power supply module in the standby system.

Replace the power supply module in the standby system.

4. Power on the standby system.

Check if wiring to the power supply module and the power supply voltage are correct, and power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

Replacing a redundant power supply module

To replace a redundant power supply module while the redundant system is operating, if the power supply system is made redundant, replace one power supply module at a time by powering it off while the system is operating. Since the redundant power supply module that is not the replacement target supplies power to the modules on the base unit, controls can be continuously performed during replacement of the other redundant power supply module.

For the replacement procedure, refer to the following.

📖 MELSEC iQ-R Module Configuration Manual

Replacing a redundant function module

The following describes the procedure for replacing the redundant function module of the standby system by turning the power off while the redundant system is operating. When the replacement target is in the control system, turn on SM1646 (System switching by a user) in advance to execute system switching before switching to the standby system.
To replace this module without turning the power off while the system is operating, use the online module change function.
(📖 MELSEC iQ-R Online Module Change Manual)

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace the redundant function module in the standby system.

Replace the redundant function module in the standby system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

5. Check the modules for errors.

Check that no error has occurred on the CPU module or redundant function module in the standby system. If the ERROR LED of either of the modules is on or flashes, check the cause of an error and eliminate the error cause.

Replacing I/O modules

The following describes the procedure for replacing the I/O module on the main base unit of the standby system by turning the power off while the redundant system is operating. When the replacement target is in the control system, turn on SM1646 (System switching by a user) in advance to execute system switching before switching to the standby system. To replace this module without turning the power off while the system is operating, use the online module change function. (📖 MELSEC iQ-R Online Module Change Manual)

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace the I/O modules in the standby system.

Replace the I/O modules in the standby system with new I/O modules whose models are the same as the ones of the I/O modules in the control system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

Replacing an intelligent function module

The following describes the procedure for replacing the intelligent function module on the main base unit of the standby system by turning the power off while the redundant system is operating. When the replacement target is in the control system, turn on SM1646 (System switching by a user) in advance to execute system switching before switching to the standby system.

To replace this module without turning the power off while the system is operating, use the online module change function.
(📖 MELSEC iQ-R Online Module Change Manual)

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace an intelligent function module in the standby system.

Disconnect the cables such as the communication cable connected to the intelligent function module in the standby system. Replace the intelligent function module with an intelligent function module whose model is the same as that of the intelligent function module in the control system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

5. Perform module diagnostics on the standby system.

Check with error information in the module diagnostics that no error has occurred. In addition, when the network module is replaced, check the network status with the network diagnostics.

Replacing a main base unit

The following describes the procedure for replacing the main base unit of the standby system by turning the power off while the redundant system is operating. When the replacement target is in the control system, turn on SM1646 (System switching by a user) in advance to execute system switching before switching to the standby system.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace the main base unit in the standby system.

Replace the main base unit in the standby system with a new main base unit whose model is the same as that of the main base unit in the control system. When mounting modules to the new base unit, mount each module to the slots of the unit in the same order as that of the base unit of the control system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

5. Check the modules for errors.

Check that no error has occurred on the CPU module or redundant function module in the standby system. If the ERROR LED of either of the modules is on or flashes, check the cause of an error and eliminate the error cause.

Replacement procedure for extension cable between main base unit and extension base unit

The following describes the procedure for replacing the extension cable between the main base unit and the extension base unit of the standby system by turning the power off while the redundant system is operating. When replacing the extension cable between the main base unit and the extension base unit of the control system, turn on SM1646 (System switching by a user) in advance to execute system switching before switching to the standby system.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes.

3. Replace the extension cable between the main base unit and the extension base unit.

Replace the extension cable that connects the main base unit of the standby system and the extension base unit.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on.

5. Check the modules for errors.

Check that no error has occurred on the CPU module or redundant function module in the standby system. If the ERROR LED of either of the modules is on or flashes, check the cause of an error and eliminate the error cause.

Replacement procedure for extension cable between extension base units

Only the extension cable on the inactive side can be replaced. (☞ Page 605 Replacement/Addition of an Extension Cable (Online))

If the extension cable on the active side is removed, system switching occurs, and the cable cannot be replaced.

31.2 Replacement/Addition of an Extension Cable (Online)

The extension cable on the inactive side (ACTIVE LED is turned off) between the redundant extension base units can be replaced or added while the system is running.

Point

- Extension cables can be replaced or added regardless of the operating status of the CPU module.
- Extension cables can be replaced or added regardless of the operating mode.
- Extension cables can be replaced or added when the main base unit of the CPU module of the control system and the extension base unit for the redundant system are connected properly.

Procedures for replacement or addition

This section describes the procedures for replacing or adding extension cables.

■Replacement procedure for extension cables

1. Check that the target extension cable is inactive based on SD1761 (extension cable route information) or the ACTIVE LED of the extension base unit for the redundant system.
2. Remove the target extension cable. (CONNECT LED of the extension base unit for the redundant system (on the side of the removed extension cable) turns off.)

Point

Due to the removal of the extension cable, a continuation error, which indicates that the CPU module of the control system has an extension cable error, is detected.
Even when the replacement is completed successfully, this extension cable error is not cleared, and the ERROR LED stays on. After replacement, clear the error using an engineering tool.
The date and time when the extension cable was disconnected or failed can be checked in the event history.

3. Attach the replacement extension cable. (CONNECT LED of the extension base unit for the redundant system (on the side of the attached extension cable) turns on.)
4. Check that the extension cable is attached properly based on SD1760 (extension cable connection status) or the CONNECT LED of the extension base unit for the redundant system.

■Addition procedure for extension cables

1. Check the place to add an extension cable.
2. Attach the extension cable to be added. (CONNECT LED of the extension base unit for the redundant system (on the side of the attached extension cable) turns on.)
3. Check that the extension cable is attached properly based on SD1760 (extension cable connection status) or the CONNECT LED of the extension base unit for the redundant system.

Point

The date and time when the extension cable was attached properly can be checked in the event history.

Precautions

This section describes the precautions for replacement/addition of the extension cable (online).

■Precautions for handling extension cables

- Keep the overall cable distance within 20m in total length of extension cables. For the two extension cables connecting the main base unit and the extension base unit for the redundant system for the system A/B and for the two extension cables connecting the extension base units for the redundant system, extension cables with different lengths can be used. However, when connecting cables with different lengths, calculate using the longer length.

For the precautions for handling extension cables other than those above, refer to the following.

📖 MELSEC iQ-R Module Configuration Manual

■Precautions for replacing or adding extension cables

- Replace one extension cable at a time, and connect the new extension cable to the same connectors as the removed extension cable. When adding an extension cable, connect it to the IN and OUT connectors of the extension base units to which another extension cable is connected. Note that the units may not operate normally if connected incorrectly.
- Do not disconnect the extension cable on the active side while replacing the extension cable on the inactive side. Otherwise a stop error is detected in the CPU modules of both systems, and control cannot be continued.
- After the first use of the product, do not insert/remove the extension cable to/from the base unit more than 50 times. Exceeding the limit may cause malfunction.

PART 5 WHEN USING THE SAFETY CPU

This part consists of the following chapters. Please read these chapters when using the Safety CPU. Since information same as that of the standard CPU is not described in these chapters, refer to Part 1 to Part 3.

32 RUNNING A SAFETY PROGRAM

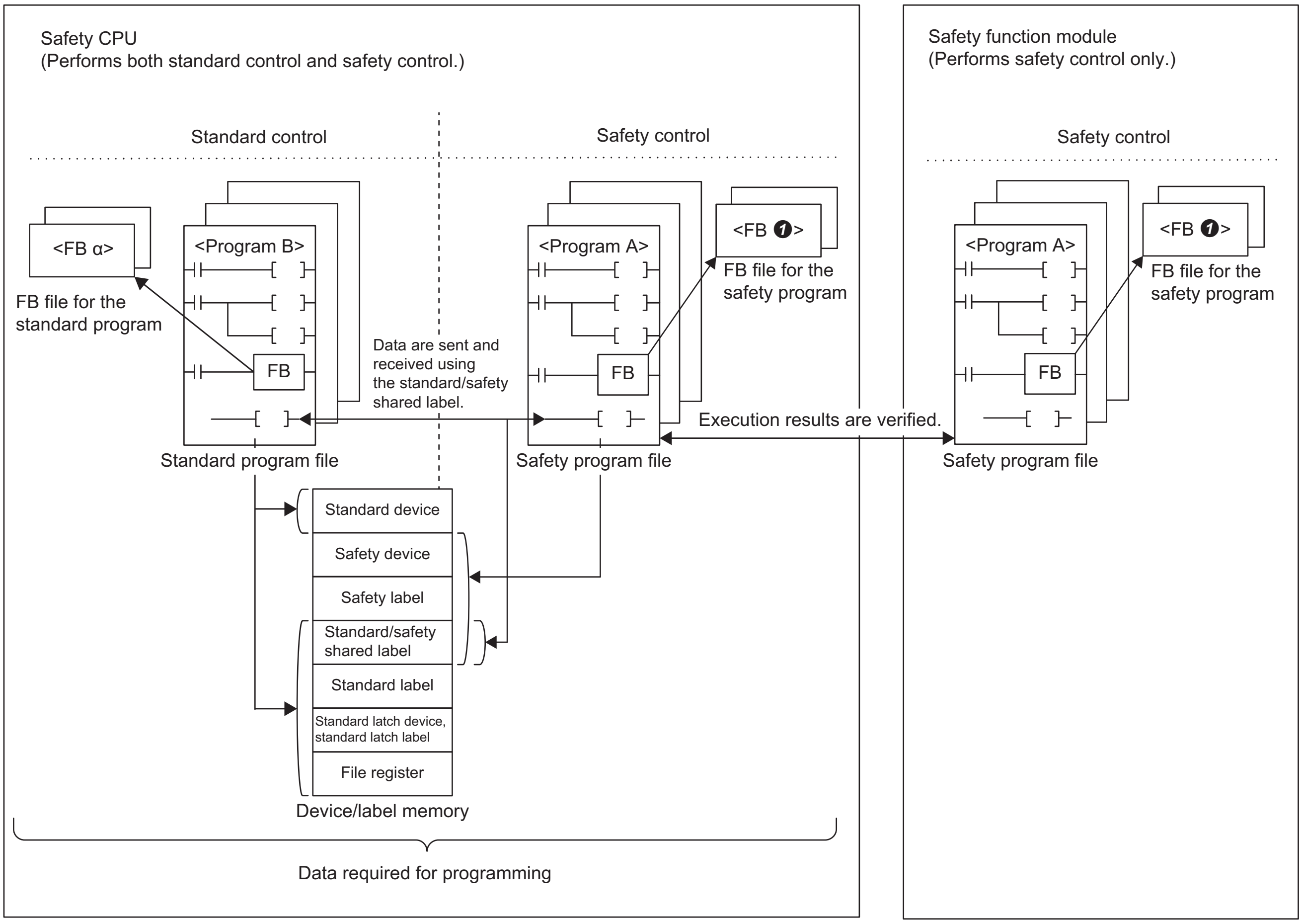
This section describes how to execute a safety program. Information not described in this chapter is same as that of the standard CPU. (☞ Page 40 RUNNING A PROGRAM to ☞ Page 100 MEMORY CONFIGURATION OF THE CPU MODULE)

Point

For details on the setting method (registration procedure) of safety programs, refer to the following.
📖 GX Works3 Operating Manual

32.1 Overview

The Safety CPU executes standard programs and safety programs, and the safety function module executes safety programs only. During operations, the Safety CPU verifies the execution results of safety programs with those of the safety function module.



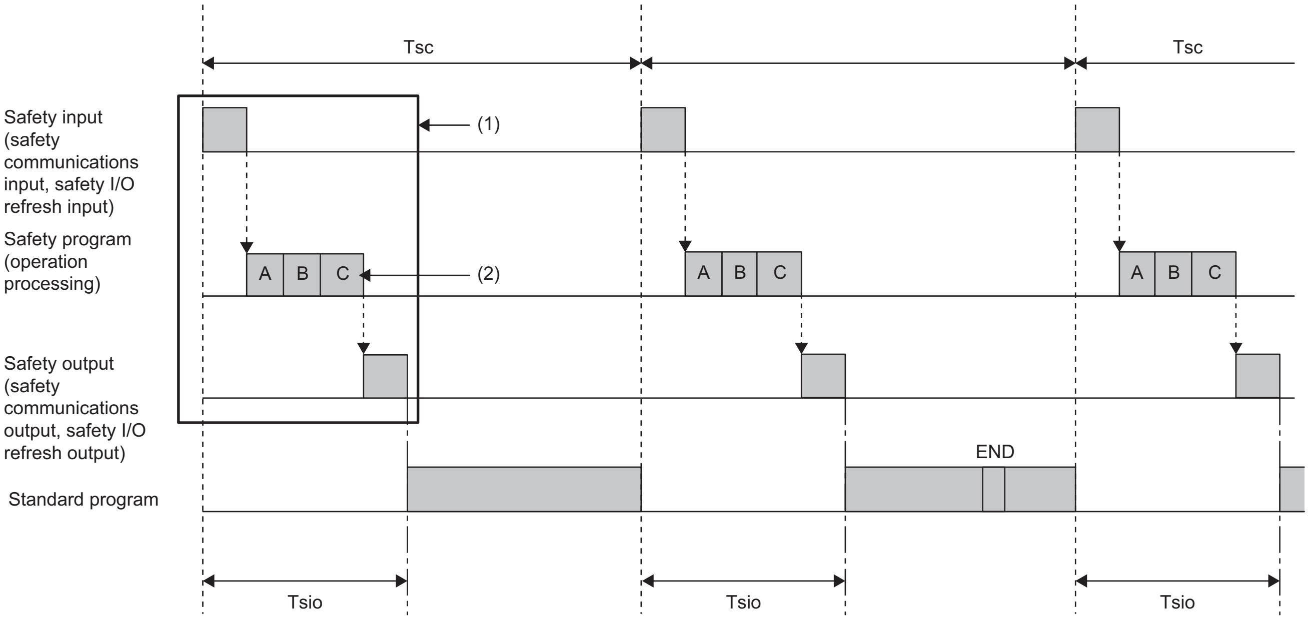
Devices/labels that can be used in safety programs

Only safety devices and safety labels can be used in safety programs.

- Safety device: ☞ Page 654 Safety Devices
- Safety label: ☞ Page 659 Safety Label

32.2 Safety Program

Safety programs are executed at every safety cycle. Safety cycle processing is performed in the following order: safety input (safety communications input, safety I/O refresh input), safety program, and safety output (safety communications output, safety I/O refresh output). (☞ Page 611 Safety Cycle Time)
Standard programs (+ END processing) are executed within the remaining time of the safety cycle time after safety programs are executed. (Standard programs are executed until next safety cycle time starts.)



Tsc: Safety cycle time
Tsio: Safety program + Safety input/output processing time
(1): Store the processing times below.
- Safety CPU: SD1890, SD1891
- Safety function module: Un\G62, Un\G63

(2): Execute all safety programs. (Execute safety programs A, B, and C.)

Point

- After executing a rising/falling instruction, this will be valid until the safety program for the next safety cycle processing is executed. (MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))
- In safety programs, the block number of the file register (R) is not saved or restored. Data in the index register (Z, LZ) are not saved or restored, either.
- Even if the operating status of Safety CPUs for which no safety program has been registered is STOP or PAUSE, safety program operations other than operation processing are performed. Consequently, the standard control processing time and device/label access service processing time will be longer.

Safety program execution type

The standard program and safety program execution types are as follows.

Program	Execution type
Standard program	• Initial • Scan • Fixed scan • Event • Standby
Safety program	Fixed scan

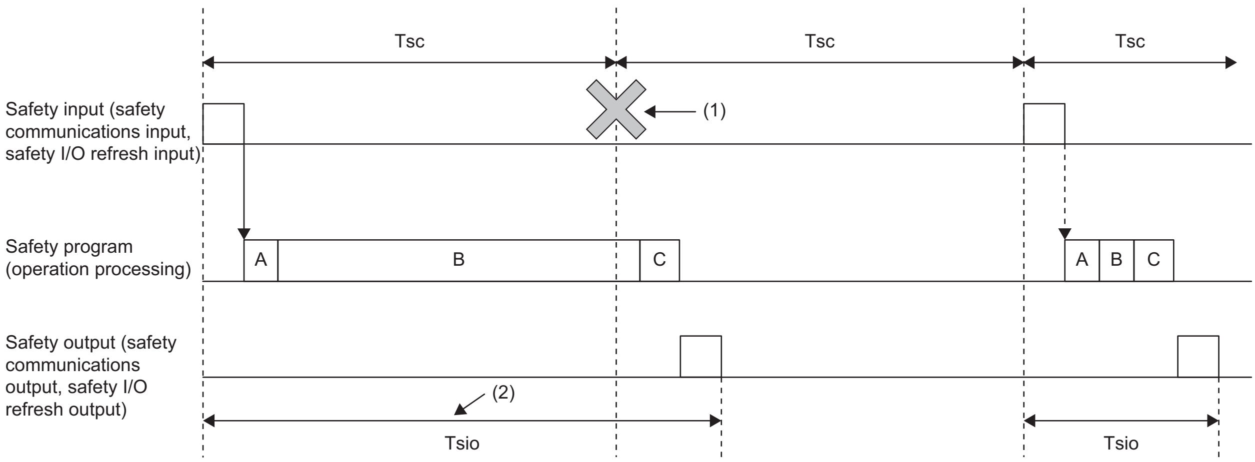
Safety programs are executed as a fixed scan execution type program. However, safety programs perform safety control, and therefore operation differs from standard fixed scan execution type programs in the following ways.

- They are unaffected by disabling interrupt instructions (EI instruction, DI instruction, and IMASK instruction), and are executed based on the safety cycle time.
- Of all interrupt programs that can be created by customers, the execution of safety programs is given highest priority.
- Safety cycle processing including safety programs prioritizes fixed periodicity. If the safety cycle time is exceeded, and safety cycle processing is executed, processing is not performed in succession, but based on the interrupt timing of the next cycle time.

When the program does not complete within the safety cycle time

Monitors whether execution of the safety program is completed within the safety cycle time, and if the program execution time exceeds the next safety cycle time, an error occurs at the Safety CPU and safety function module. Note that if the next safety cycle time is exceeded, the safety input → safety program is not executed following program completion, a safety cycle processing error (error code: 1A01H) is detected at the Safety CPU, and the safety program is executed at the next safety cycle time. Also, if execution of the program is completed after the safety cycle time has been exceeded, the safety cycle processing time execution cycle error flag turns on, and the safety cycle processing time execution cycle error count is incremented by one as follows.

Item	Safety cycle processing time execution cycle error flag	Safety cycle processing time execution cycle error count	Cause of safety cycle processing time execution cycle error count being incremented by one
Safety CPU	SM1888	SD1888	If error codes 1A00H, 1A01H occur
Safety function module	—	Un\G54	If error code 1A00H occurs



Tsc: Safety cycle time
Tsio: Safety program + Safety input/output processing time
(1): If safety input/output (safety communications input/output, safety I/O refresh input/output) and safety program execution are not completed within the safety cycle time, interrupts during this period are ignored. Execute safety input/output (safety communications input/output, safety I/O refresh input/output) and safety programs from the next interrupt.
(2): If the time from safety input (safety communications input, safety I/O refresh input) to safety output (safety communications output, safety I/O refresh output) exceeds the safety cycle time, a continuation error will occur after safety processing is complete.
- Safety CPU: SM1888 turns ON, and the value in SD1888 is incremented by one.
- Safety function module: The value in Un\G54 is incremented by one.

32.3 Safety Cycle Time

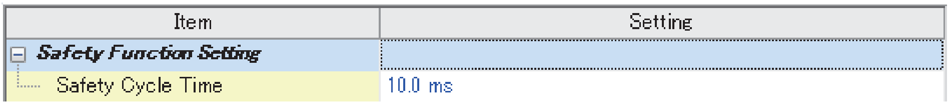
A safety cycle time is timing for performing safety input/output (safety communications input/output, safety I/O refresh input/output) and executing safety programs.

Setting method

Set a safety cycle time in the CPU parameter.

[CPU parameter] ⇨ [Safety Function Setting]

Window



Displayed items

Item	Description	Setting range	Default
Safety Cycle Time	Sets the timing (safety cycle time) for performing safety input/output (safety communications input/output, safety I/O refresh input/output) and executing safety programs.	1.0 to 1000.0ms (unit: 0.1ms)	10.0ms

Point

- If an instruction with long processing time is executed in the standard program, the start of safety cycle processing is delayed. To prioritize safety cycle processing, set "Enable" for the enabling interrupt setting during command execution. (☞ Page 85 Interrupt enabled during instruction execution)
- A safety communication timeout error and safety I/O refresh timeout error occur if the start of safety cycle processing is delayed and the monitoring time that is specified in the safety communication settings and safety I/O refresh settings has elapsed due to delay of the safety cycle processing. Therefore, the safety response time is not affected. (☞ Page 630 Safety Communication Function, ☞ Page 631 Safety I/O Refresh Function)
- When "Safety Cycle Time" is set to "Sampling Method" in "Sampling Interval Setting" of the recording settings, set a safety cycle time that includes the time required for processing of the recording function. If the time is delayed due to the processing of the recording function, a safety cycle processing error may occur. When using the recording function in the Safety CPU, check the version of the CPU module and the engineering tool. (☞ Page 1139 Added and Enhanced Functions)

32.4 Safety CPU Operating Status

The following is a list of Safety CPU operating statuses.

- RUN state
- STOP state
- PAUSE state

Operation processing based on safety CPU operating status

Operation processing based on the Safety CPU operating status is the same as that of the standard CPU. (☞ Page 95 Operation Processing by Operating Status)

Safety communications processing and safety I/O refresh processing

Safety communications processing and safety I/O refresh processing based on the Safety CPU operating status are as follows.

Safety CPU operating status	Program operation processing	Safety communications processing	Safety I/O refresh processing
RUN state	Executes the operation of the standard program/safety program stored in the CPU module.	Executes the set safety communications.	Executes the set safety I/O refresh.
STOP state	Stops the operation of the standard program/safety program stored in the CPU module.	Executes the set safety communications, however, send data is turned off (0).*1*2	Executes the set safety I/O refresh, however, send data and safety I/O refresh output are turned off (0).*1*2
PAUSE state	Suspends the operation of the standard program stored in the CPU module. Stops the operation of the safety program stored in the CPU module.	Executes the set safety communications, however, send data is turned off (0).*1	Executes the set safety I/O refresh, however, send data and safety I/O refresh output are turned off (0).*1

*1 When the safety operation mode is set to TEST MODE, the processing is as follows.
·Output is based on the output hold and clear settings of when the status of network modules (such as the RJ71GF11-T2) for which safety communication settings are specified is CPU STOP. (📖 Manual for network module used)
·The device value that is specified in the safety I/O refresh device settings is output. However, with the safety output device (SA\Y) specified, the value that is turned off (0) will be output because the safety output device (SA\Y) in the CPU module is turned off (0) when the operating status switches from RUN to STOP/PAUSE.
(📖 MELSEC iQ-R I/O Module (With Safety Functions) User's Manual)

*2 Safety communications and safety I/O refresh are not performed if a stop error occurs at the Safety CPU.

Operation processing when operating status of the Safety CPU changes

This displays operation processing with the safety program when the operating status of the Safety CPU is changed.

Safety CPU Operating Status	Safety CPU processing			
	Safety program	External output	Safety device data[*1]	
			Other than SA\Y	SA\Y
STOP → RUN	Executes safety programs.	Outputs the value after safety program execution.	Retains the safety device data status immediately before the RUN status. Clears safety local devices.	Retains the SA\Y value immediately before the RUN status.
RUN → STOP	Executes up to the END instruction and then stops.	Turns off all the outputs of the safety communications and safety I/O refresh.[*3]	Retains the device data status immediately before the STOP status.	Turns off all SA\Y values.[*2]
RUN → PAUSE	Stops operation after the execution of one scan.	Turns off all the outputs of the safety communications and safety I/O refresh.[*3]	Retains the device memory status immediately before the PAUSE state.	Turns off all SA\Y values.[*2]
PAUSE → RUN	Executes the program from the start.	Outputs the value after safety program execution.	Retains the device data status immediately before the RUN status. Clears safety local devices.[*2]	Retains the device data status immediately before the RUN status.
PAUSE → STOP	Operation remains stopped.	Turns off all the outputs of the safety communications and safety I/O refresh.[*3]	Retains the device data status immediately before the STOP status.	Retains the SA\Y value immediately before the STOP state.
STOP → PAUSE	Operation remains stopped.	Turns off all the outputs of the safety communications and safety I/O refresh.[*3]	Retains the device data status immediately before the PAUSE state.	Retains the SA\Y value immediately before the PAUSE state.

*1 Assignment device data for standard/safety shared labels is processed in the same way as operation processing with standard programs.

*2 Values for safety devices when the Safety CPU operating status changes are cleared or turn off, regardless of the safety operation mode.

*3 If the safety operation mode is TEST MODE, output is based on the output hold and clear settings of the following modules for which safety communication settings are specified is CPU STOP.
· Network modules (such as the RJ71GF11-T2) for which safety communication settings are specified (📖 Manual for network module used)
· I/O modules with safety functions for which safety I/O refresh device settings are specified (📖 MELSEC iQ-R I/O Module (With Safety Functions) User's Manual)

Operation processing with standard programs is the same as the following.

☞ Page 96 Operation Processing When Operating Status Is Changed

32

33 MEMORY SPECIFICATIONS

This chapter describes the memory specifications of the Safety CPU. Information not described in this chapter is same as that of the standard CPU. (☞ Page 100 MEMORY CONFIGURATION OF THE CPU MODULE)

33.1 Memory Configuration

Specifications of the device/label memory differ from the standard CPU in the memory configuration of the Safety CPU.

Device/label memory

In addition to standard devices and standard labels, the data such as safety devices and safety labels are allocated to each data area of the device/label memory.
The safety device area, safety label area, safety local device area, and standard/safety shared label area are allocated between the standard device area and standard label area. The capacities for each area can be changed in "Device/Label Memory Area Setting" of the engineering tool. (☞ Page 632 Device/Label Memory Area Setting)

<table>
<tr><td colspan="2">Standard device area</td></tr>
<tr><td rowspan="3">Safety device/label area</td><td>Safety device area</td></tr>
<tr><td>Safety label area</td></tr>
<tr><td>Safety local device area</td></tr>
<tr><td colspan="2">Standard/safety shared label area</td></tr>
<tr><td rowspan="2">Standard label area</td><td>Standard label area</td></tr>
<tr><td>Standard latch label area</td></tr>
<tr><td colspan="2">Standard local device area</td></tr>
<tr><td colspan="2">File storage area</td></tr>
</table>

Data to be allocated

The following table lists the data allocated to each area.

Area		Application
Standard device area		Standard user devices
Safety device/label area	Safety device area	Safety user devices
	Safety label area	Safety global labels and safety local labels
	Safety local device area	Safety local devices
Standard/safety shared label area		Standard/safety shared labels
Standard label area	Standard label area	Standard global labels and standard local labels
	Standard latch label area	Standard global labels and standard local labels with latch specified
Standard local device area		Standard local devices (excluding index register)
File storage area		File register files and other data

Point

Safety devices and safety labels cannot be latched, and therefore there is no latch area for safety devices and safety labels.

33.2 File Size Unit in Memory

The following table lists the unit of the file size (cluster size) of the Safety CPU memory.

Safety CPU	File size unit		
	Program memory	Device/label memory	Data memory
R08SFCPU	128 bytes	512 bytes	2048 bytes
R16SFCPU			4096 bytes
R32SFCPU			8192 bytes
R120SFCPU			16384 bytes

Point

Data is written in the unit of the file size (cluster size). For example, when 464 bytes of CPU parameter is written to the data memory on R08SFCPU, it is written as 2048 bytes of data because the file size unit of the data memory is 2048 bytes.

33.3 Memory Operation

Using an engineering tool, each memory can be initialized and cleared to zero. For details on the operation method, refer to the following.

GX Works3 Operating Manual

Items to be specified in the engineering tool				Target
Initialization	Data memory			Deletes all the folders and files in the program memory and data memory.
	Device/label memory			Deletes all the files in the file storage areas in the device/label memory.
	SD memory card			Deletes all the folders and files in the SD memory card.
Value clear	Device, label	Zero clear		Excluding devices and labels with latch specified, clears the following to zero: X, Y, M, B, F, SB, V, T, ST, LT, LST, C, LC, D, W, SW, FX, FY, FD, Z, LZ, RD, SA\X, SA\Y, SA\M, SA\B, SA\T, SA\ST, SA\C, SA\D, SA\W, and all labels (including module labels).
		Zero clear (including Latches (1) and (2))		Including devices and labels with latch specified, clears the following to zero: X, Y, M, B, F, SB, V, T, ST, LT, LST, C, LC, D, W, SW, FX, FY, FD, Z, LZ, RD, SA\X, SA\Y, SA\M, SA\B, SA\T, SA\ST, SA\C, SA\D, SA\W, and all labels (including module labels).
	File register	Zero clear	All files	Clears the contents of all the file registers to zero.
			File specification	Clears only the contents of the specified file register(s) to zero.
		Zero clear excluding Latch (2)		Clears the file registers other than Latch (2) to zero.
	Device/label/file register latch clear			Clears devices, labels, and file registers other than Latch (2) to zero.

Point

- If the power goes off during initialization or zero clear, the memory is left in the state of that point, and it is necessary to re-execute the memory operation.
- Following initialization, write the file required to run the Safety CPU. (GX Works3 Operating Manual)
- Following data memory initialization, if the power is turned off and on or reset, the device/label memory will revert to the default status (default capacity for each area).

33.4 Files

This section lists the files used by the Safety CPU.

File types and storage memory

The following table summarizes the types of files stored in the Safety CPU and storage memory.

◎: Required, ○: Can be stored, ×: Cannot be stored

File type		CPU built-in memory			SD memory card	File name and extension
		Program memory	Device/label memory	Data memory		
		Drive 0	Drive 3	Drive 4	Drive 2	
Standard program		◎*4*6	×	◎*4*6	×	ANY_STRING.PRG
Standard FB file		○*4	×	○*4	×	ANY_STRING.PFB
Standard CPU parameter		×	×	◎	×	CPU.PRM
System parameter		×	×	◎	×	SYSTEM.PRM
Standard module parameter		×	×	○	×	UNIT.PRM
Module extension parameter		×	×	○	○	• UEXmmmnn.PRM*1 • UEXmmm00.PPR*5
Memory card parameter		×	×	×	○	MEMCARD.PRM
Device comment		×	×	○	○	ANY_STRING.DCM
Initial device value		×	×	○	×	ANY_STRING.DID
Standard global label setting file		×	×	○	×	GLBLINF.IFG
Initial label value file	Initial global label value file	×	×	○	×	GLBLINF.LID
	Initial local label value file	×	×	○	×	PROGRAM_NAME.LID
File register		×	○	×	○*3	ANY_STRING.QDR
Event history		×	×	○	○	• EVENT.LOG • EVEN2.LOG
Device data storage file		×	×	○	○*3	DEVSTORE.QST
General-purpose data		×	×	○	○	ANY_STRING.CSV/BIN
Data logging setting file	Common setting file	×	×	×	○	LOGCOM.LCS
	Individual setting file	×	×	○	○	LOGnn.LIS*2
Memory dump setting file		×	×	○	○*3	MEMDUMP.DPS
Remote password		×	×	○	○	00000001.SYP
System file for backing up CPU module data		×	×	×	○	$BKUP_CPU_INF.BSC
Backup data file for backing up CPU module data		×	×	×	○	BKUP_CPU.BKD
Device/label data file for backing up CPU module data		×	×	×	○	BKUP_CPU_DEVLAB.BKD
Device station parameter file		×	×	○*10	○*10	SLAVEmmmnnnxxxx.NSP*9
Recording setting file		×	×	○	○	RECCFGn.RSI
Safety program*7		◎*4*6	×	◎*4*6	×	ANY_STRING.SPG
Safety FB file*7		○*4	×	○*4	×	ANY_STRING.SPB
Safety CPU parameter*7*8		×	×	◎	×	CPU.SPR
Safety module parameter*7		×	×	○	×	UNIT.SPR
Safety global label setting file*7		×	×	○	×	GLBLINF.SIF
Standard/safety shared label setting file*7		×	×	○	×	S_GLBLINF.SIF
Safety device station parameter file		×	×	○*10	×	SLAVEmmmnnnxxxx.SSP*9

*1 mmm represents the start I/O number (first three digits in four-digit hexadecimal representation) of each module. For the Safety CPU, it will be 3FFH. Also, nn represents the consecutive number (two-digit hexadecimal representation) of module extension parameter files of each module.

*2 nn corresponds to the setting number and is 01 through 10.

*3 Can be stored but cannot operate as a function.

*4 When this file is stored in the built-in memory of the Safety CPU, it is divided into program memory and data memory and stored. (☞ Page 116 Configuration of a program file)

*5 Module extension parameter for the protocol setting, storing protocol setting information in the predefined protocol support function
*6 One or more of either standard program or safety program is required.
*7 The file is also stored in the safety function module.
*8 This parameter relates to safety control inside the CPU parameters. (☞ Page 1091 CPU Parameters)
This item is yellow in the engineering tool CPU parameters.
*9 mmm represents the start I/O number (first three digits in four-digit hexadecimal representation) of the master station of CC-Link IE TSN, nnn represents the number of modules in the network configuration settings (three-digit hexadecimal representation), and xxxx represents the serial number of the parameter (four-digit hexadecimal representation).
*10 Up to 1024 files can be stored in total of device station parameter files and safety device station parameter files.

File operation available

The following lists file operations which are available for each file type.

In TEST MODE

The following lists file operations which are available for each file type in TEST MODE.

○: Available, ×: Not available, —: N/A

File type		Operation using engineering tool			Operation with SLMP and FTP server function			Operation with instructions in standard programs	
		Write	Read	Delete	Write	Read	Delete	Write	Read
Standard program		○[*1]	○	○[*2]	×	×	×	—	—
Standard FB file		○	○	○[*2]	×	×	×	—	—
Standard CPU parameter		○[*2]	○	○[*2]	×	×	×	—	—
System parameter		○[*2]	○	○[*2]	×	×	×	—	—
Standard module parameter		○[*2]	○	○[*2]	×	×	×	—	—
Module extension parameter		○[*2]	○	○[*2]	×	×	×	—	—
Memory card parameter		○[*2]	○	○[*2]	×	×	×	—	—
Device comment		○	○	○[*2]	×	×	×	—	—
Initial device value		○	○	○[*2]	×	×	×	—	—
Standard global label setting file		○	○	○[*2]	×	×	×	—	—
Initial label value file	Initial global label value file	○	○	○[*2]	×	×	×	—	—
	Initial local label value file	○	○	○[*2]	×	×	×	—	—
File register		○	○	○[*2]	×	×	×	○	○
Event history		—	—	—	×	×	×	—	—
Device data storage file		—	—	—	×	×	×	○	○
General-purpose data		○	○	○	○	○	○	○	○
Data logging setting file	Common setting file	○[*3]	○[*3]	○[*3]	○	○	○	—	—
	Individual setting file	○[*3]	○[*3]	○[*3]	○	○	○	—	—
Memory dump setting file		○	○	○	○	○	○	—	—
Remote password		○[*2]	○	○[*2]	×	×	×	—	—
System file for backing up CPU module data		—	—	—	○	○	○	—	—
Backup data file for backing up CPU module data		○	○	○	○	○	○	—	—
Device/label data file for backing up CPU module data		○	○	○	○	○	○	—	—
Device station parameter file		○	○	○	○	○	○	—	—
Safety program		○[*1]	○	○[*2]	×	×	×	—	—
Safety FB file		○	○	○[*2]	×	×	×	—	—
Safety CPU parameter[*4]		○[*2]	○	○[*2]	×	×	×	—	—
Safety module parameter		○[*2]	○	○[*2]	×	×	×	—	—
Safety global label setting file		○	○	○[*2]	×	×	×	—	—
Standard/safety shared label setting file		○	○	○[*2]	×	×	×	—	—
Safety device station parameter file		○	○	○	○	○	○	—	—

*1 To write files to the programmable controller, use the write function when the operating status of the Safety CPU is STOP. When the operating status is RUN, use the online change function. When the Safety CPU operating status is RUN, only the standard programs registered in the CPU parameter can be written to the programmable controller.

*2 Operation is possible only when the operating status of the Safety CPU is STOP/PAUSE. If performed in RUN state, the operating status of the Safety CPU is changed by the remote STOP function, and then the operation continues.

*3 This indicates operations performed using the CPU Module Logging Configuration Tool.

*4 This parameter relates to safety control inside the CPU parameters. (☞ Page 1091 CPU Parameters)
This item is yellow in the engineering tool CPU parameters.

In SAFETY MODE

The following lists file operations which are available for each file type in SAFETY MODE.

○: Available, ×: Not available, —: N/A

File type		Operation using engineering tool			Operation with SLMP and FTP server function			Operation with instructions in standard programs	
		Write	Read	Delete	Write	Read	Delete	Write	Read
Standard program		○*1*5	○	○*2	×	×	×	—	—
Standard FB file		○*2*5	○	○*2	×	×	×	—	—
Standard CPU parameter		×	○	×	×	×	×	—	—
System parameter		×	○	×	×	×	×	—	—
Standard module parameter		×	○	×	×	×	×	—	—
Module extension parameter		○*2	○	○*2	×	×	×	—	—
Memory card parameter		○*2	○	○*2	×	×	×	—	—
Device comment		○	○	○*2	×	×	×	—	—
Initial device value		○	○	○*2	×	×	×	—	—
Standard global label setting file		○*2*5	○	○*2	×	×	×	—	—
Initial label value file	Initial global label value file	○	○	○*2	×	×	×	—	—
	Initial local label value file	○	○	○*2	×	×	×	—	—
File register		○	○	○*2	×	×	×	○	○
Event history		—	—	—	×	×	×	—	—
Device data storage file		—	—	—	×	×	×	○	○
General-purpose data		○	○	○	○	○	○	○	○
Data logging setting file	Common setting file	○*3	○*3	○*3	○	○	○	—	—
	Individual setting file	○*3	○*3	○*3	○	○	○	—	—
Memory dump setting file		○	○	○	○	○	○	—	—
Remote password		○*2	○	○*2	×	×	×	—	—
System file for backing up CPU module data		—	—	—	○	○	○	—	—
Backup data file for backing up CPU module data		○	○	○	○	○	○	—	—
Device/label data file for backing up CPU module data		○	○	○	○	○	○	—	—
Device station parameter file		○	○	○	○	○	○	—	—
Safety program		×	○	×	×	×	×	—	—
Safety FB file		×	○	×	×	×	×	—	—
Safety CPU parameter*4		×	○	×	×	×	×	—	—
Safety module parameter		×	○	×	×	×	×	—	—
Safety global label setting file		×	○	×	×	×	×	—	—
Standard/safety shared label setting file		×	○	×	×	×	×	—	—
Safety device station parameter file		×	○	×	○	○	○	—	—

*1 To write files to the programmable controller, use the write function when the operating status of the Safety CPU is STOP. When the operating status is RUN, use the online change function. When the Safety CPU operating status is RUN, only the standard programs registered in the CPU parameter can be written to the programmable controller.

*2 Operation is possible only when the operating status of the Safety CPU is STOP/PAUSE. If performed in RUN state, the operating status of the Safety CPU is changed by the remote STOP function, and then the operation continues.

*3 This indicates operations performed using the CPU Module Logging Configuration Tool.

*4 This parameter relates to safety control inside the CPU parameters. (☞ Page 1091 CPU Parameters)
This item is yellow in the engineering tool CPU parameters.

*5 Writing is possible in file units, however, it is not possible to select files together with parameter files or files relating to safety and then write to the programmable controller CPU. If writing in file units, write after clearing the selection of other parameter files or files relating to safety, or write after switching to TEST MODE.

34 FUNCTIONS

This chapter describes the functions added to the Safety CPU and, among the functions of the standard CPU, the availability of functions for safety devices and safety programs and which have been modified or restricted.
For the availability of Safety CPU functions, refer to the following.
MELSEC iQ-R CPU Module User's Manual (Startup)

Function		Description	Reference
Function added to the Safety CPU	Safety operation mode	There are two safety operation modes: TEST MODE and SAFETY MODE. One mode is for changing the safety program or safety parameters, and the other is for operating the system as a safety system.	Page 624 Safety Operation Mode
	Continuous RUN prevention in TEST MODE	Prevents the Safety CPU from running continuously for a long time in TEST MODE.	Page 628 Continuous RUN Prevention in TEST MODE
	Safety diagnostic function	Self-diagnostic function of the Safety CPU	Page 629 Safety Diagnostic Function
	Identification check for safety data	Checks if the project data created using the engineering tool and the data in the Safety CPU are the same, and confirms that the program executed in SAFETY MODE is the one written by the user.	Page 630 Identification Check for Safety Data
	Safety communication function	Communicates data between the Safety CPU and modules supporting safety functions using safety protocols.	Page 630 Safety Communication Function
	Safety I/O refresh function	Periodically communicates I/O data between the I/O module with safety functions and safety CPU using safety protocols.	Page 631 Safety I/O Refresh Function
	User authentication function of CPU modules	Prevents an unauthorized access to a program or parameters written to a CPU module. Operations are restricted by registering a user name/password for a CPU module.	Page 653 User Authentication Function of CPU Modules

Function			Description	Reference
Function different (modified or restricted) from the standard CPU	Interrupt function	Interrupt period setting	All interrupt pointers (I) cannot be used in safety programs. The high-speed internal timer interrupt (I48, I49) cannot be used in standard programs. The inter-module synchronous interrupt (I44) and multiple CPU synchronous interrupt (I45) cannot be used in some firmware version of the safety CPU. (☞ Page 1139 Added and Enhanced Functions)	MELSEC iQ-R CPU Module User's Manual (Startup)
		Multiple interrupt function	Safety cycle processing is performed as a higher priority interrupt than other processing, and therefore even when set to "Enable" in the multiple interrupt settings, interrupts will never occur during safety cycle processing.	—
	Output Mode Setting of STOP to RUN		The safety output (SA\Y) for the safety output device (SA\Y) is off, regardless of this setting. If the value is changed with a device test or others while the CPU module is in the STOP state, the value will be applied when the status of the CPU module changes from STOP to RUN.	Page 655 Safety output (SA\Y)
	Clock function		When setting clock data for the Safety CPU at the "Set Clock" window of the engineering tool, "All Stations Specified" cannot be selected as an execution target.	—
	Writing data to the CPU module	Online change	• When online changes are executed in safety program, the firmware version of the safety CPU is restricted. (☞ Page 1139 Added and Enhanced Functions) It is available only when the safety operation mode is to the TEST MODE. (☞ Page 627 Operations restricted in SAFETY MODE) • When the safety program files, safety global label setting files, and standard/safety shared global label setting files cannot be changed online, the firmware version of the safety CPU is restricted. (☞ Page 1139 Added and Enhanced Functions) If standard/safety shared labels inside standard programs are edited, standard programs can be changed online. • If a disconnection, power-off or reset of the CPU module is detected while the program restoration information is being written, the processing will be suspended and a message will be displayed. If the processing is suspended by a disconnection, click the [Retry] button in the message after reconnecting cables. If the processing is suspended by power-off or reset, login to the user authentication function is disabled. Log in to the function and write data again to the programmable controller.	—
	RAS function	Self-diagnostic function	• The behavior for each setting differs. • Applicable error codes to each setting in the Safety CPU are listed.	Page 649 Self-diagnostic function
		Error clear	Errors in the Safety CPU are added. Errors of the safety function module can also be cleared.	Page 650 Error clear
		Event history function	Events in the Safety CPU are added. Note that events that occur at the safety function module are not retained in the event history.	Page 908 Event List
	Remote operation		When specifying the execution destination at the "Remote Operation" window of the engineering tool, the specification of all stations and group No. is not supported, and CPU module cannot issue the command if being issued by the own station. All stations and group No. specification is not executed by CPU module if those in which all stations and group No. have been specified are received. Also, if the Safety CPU is in SAFETY MODE when clearing device/label memory during remote RUN, safety global devices, safety global labels, safety local labels, and standard/safety shared labels are not cleared.	—
	Real-time monitor function		A safety device or safety label cannot be specified for the monitor target setting or monitor condition setting of the real-time monitor. For details, refer to the manual shown in the column on the right.	📖 GX LogViewer Version 1 Operating Manual
	Scan time measurement		This function cannot be used in safety programs.	—
	Specified program monitor		This function cannot be used in safety programs. However, the specified program monitor can be used in the standard/safety shared label of the standard program.	—

34

Function			Description	Reference
Function different (modified or restricted) from the standard CPU	Device test with execution conditions		Only standard programs (ladder programs) can be set. In addition, whether each operation of this function can be performed depends on the safety operation mode.	Page 637 Device Test with Execution Conditions
	Data logging function		A safety label or safety device cannot be specified as data that can be collected, data collecting condition, or trigger condition. For details, refer to the page shown in the column on the right.	Page 639 Data Logging Function
	Memory dump function		• For data that can be collected, a safety device cannot be specified. • A safety device cannot be specified when specifying a device as a trigger condition.	—
	CPU module data backup/restoration function		For restoration, only the restoration by enabling the automatic restoration setting is supported. However, turning on the bit 0 of SD955 does not start the automatic restoration. Set "Automatic Restore Setting" to "Enable" in the engineering tool and the bit 0 of SD955 automatically turns on.	Page 643 CPU Module Data Backup/ Restoration Function
	PID control function		The PID control instructions cannot be used in safety programs.	MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)
	Multiple CPU system function	Data communications between CPU modules	Depending on the firmware version of the Safety CPU, only data communications using the CPU buffer memory are available. Data communications using the fixed scan communication area are not supported. (☞ Page 1139 Added and Enhanced Functions) The data communications using the fixed scan communication area are available between the CPU modules other than the Safety CPU. (There are no restrictions on the firmware version of the Safety CPU.)	—
		Multiple CPU synchronous interrupt	When the multiple CPU synchronous interrupt (I45) is used in general program, the firmware version of the safety CPU is restricted. (☞ Page 1139 Added and Enhanced Functions) The multiple CPU synchronous interrupt (I45) cannot be used in safety program because the execution of safety programs is prioritized.	—
	Security function	Write-protect function for device data (from outside the CPU module)	• A safety device cannot be specified in the write-protect range. • Some of the operations and functions differ from those of the standard CPU.	Page 647 Write-Protect Function for Device Data (from Outside the CPU Module)
	Latch function		Safety devices, safety labels, and standard/safety shared labels cannot be latched.	—
	Device/label initial value setting		An initial value cannot be set for safety global devices, safety local devices, safety global labels, safety local labels, and standard/safety shared labels. To set an initial value to those devices and labels, create a program so that the safety state can be secured and the value is set only at the start-up.	—
	Device/label access service processing setting		Even though this function is set, the scan time of some functions (operations) becomes longer than the specified time. The following are added to those functions. • Safety local devices (label) batch/registration monitor • Current value change of the safety labels/safety devices • Switching safety operation mode • Identification check for safety data	—
	SLMP communication function		Safety devices cannot be accessed. Also, the command execution status differs.	Page 619 File operation available
	Ethernet function	File transfer function (FTP server)		

34.1 Safety Operation Mode

There are two safety operation modes: TEST MODE and SAFETY MODE. Running normally as a safety system or changing a safety program and safety parameter can be selected.

Safety operation mode	Description
SAFETY MODE	This is a mode for operating the safety system controlled by the Safety CPU. In this mode, Safety CPU safety programs and safety parameters cannot be changed. Only device data values in safety programs can be changed.
TEST MODE	This is a mode for performing maintenance (such as setting changes and tests) of the safety system controlled by the Safety CPU. In this mode, Safety CPU safety programs and safety parameters can be changed. Device data can be changed by performing a device test.
SAFETY MODE (wait-for-restart)*1	The system switches to this mode during the period from immediately after safety operation mode is changed to SAFETY MODE when in TEST MODE until the CPU module is turned off or is reset. Even if safety operation mode is switched to SAFETY MODE, safety operation mode becomes this mode until the CPU module is powered on or is reset. Programs are not executed in this mode. An error occurs if the CPU module is the RUN state.

*1 The same restrictions apply to this mode as those for SAFETY MODE.

Checking the safety operation mode

The safety operation mode can be checked in three ways.

- TEST LED of the safety function module (MELSEC iQ-R CPU Module User's Manual (Startup))
- "Safety Operation Mode Switch" window of the engineering tool (GX Works3 Operating Manual)
- Module information list of the engineering tool (GX Works3 Operating Manual)
- Safety special register of the Safety CPU (SA\SD205) (Page 1026 List of Safety Special Register Areas)

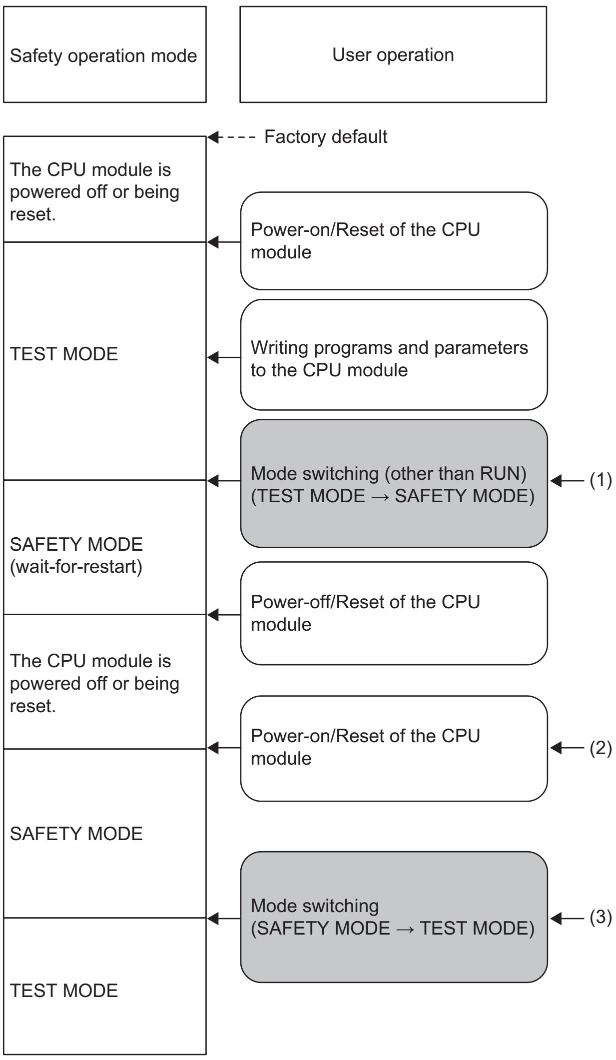
Switching the safety operation mode

This section describes how to switch the safety operation mode.

Safety operation mode transition timing

The timing at which safety operation mode switches is shown below.

Safety operation mode switching direction	Safety operation mode switch timing
TEST MODE → SAFETY MODE	When the CPU module is powered off and on or is reset after the safety operation mode switching operation
SAFETY MODE → TEST MODE	At the safety operation mode switching operation



(1) After switching from TEST MODE to SAFETY MODE, the safety operation mode will be SAFETY MODE (wait-for-restart). (The status will be wait-for-restart until the CPU module is powered off or is reset.) An error occurs if the Safety CPU is switched to the RUN state when in SAFETY MODE (wait-for-restart).

(2) After switching the safety operation mode (TEST MODE → SAFETY MODE), the mode will switch to SAFETY MODE when the CPU module is powered off and on or is reset.

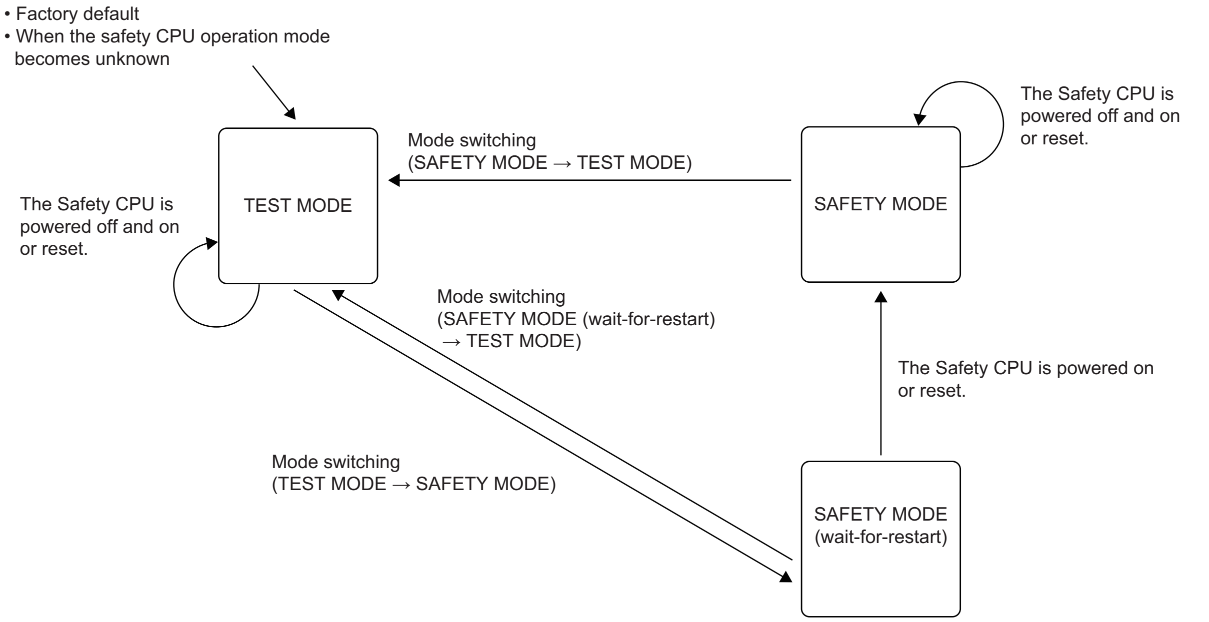
(3) After switching the safety operation mode (SAFETY MODE → TEST MODE), the mode will switch to TEST MODE.

Point

The Safety CPU retains the safety operation mode status even when the power is turned off, or a reset is performed.

Safety operation mode state transition

The Safety CPU retains the safety operation mode even in the event of a power failure, and therefore the current safety operation mode state will remain unchanged even if the CPU module is powered off and on or is reset. Safety function modules do not retain the safety operation mode in the event of a power failure, and the safety operation mode will be the same as that of the Safety CPU when the CPU module is powered off and on or is reset.



Safety operation mode state	Status transition by operation		
	Power off → on or reset	Safety operation mode switching operation	
		TEST MODE → SAFETY MODE	SAFETY MODE → TEST MODE
TEST MODE	→ TEST MODE	→ SAFETY MODE (wait-for-restart)	—
SAFETY MODE	→ SAFETY MODE	—	→ TEST MODE
SAFETY MODE (wait-for-restart)	→ SAFETY MODE	—	→ TEST MODE

Point

If [Online] ⇨ [User Authentication] ⇨ [Initialization of all PLC Data] is performed, the safety operation mode will revert to the default mode (TEST MODE). All programmable controller information can be initialized regardless of the safety operation mode.

Safety operation mode switching conditions

The conditions under which the safety operation mode can be changed are shown below.

Safety operation mode switching direction	Condition
TEST MODE → SAFETY MODE	The user who is making the switch has "Developers" access level or higher, and is currently logged on.
	The safety operation mode is currently set to the TEST MODE.
	The Safety CPU operating status is other than RUN.
	The following files held by the engineering tool, Safety CPU, and safety function module match. • Safety CPU parameter • Safety module parameter • Safety program • Safety FB file • Safety global label setting file • Standard/safety shared label setting file
	The files are not being written from another engineering tool.
SAFETY MODE → TEST MODE	The user who is making the switch has "Developers" access level or higher, and is currently logged on.
	The safety operation mode is currently set to SAFETY MODE or SAFETY MODE (wait-for-restart).

How to switch the safety operation mode

The safety operation mode can be switched using the engineering tool.

[Online] ⇨ [Safety PLC Operation] ⇨ [Safety Operation Mode Switch]

For details, refer to the following.

GX Works3 Operating Manual

Operations restricted in SAFETY MODE

When the safety operation mode is set to SAFETY MODE, the following operations cannot be performed.

Operation		Restrictions
Writing data to the programmable controller		Data cannot be written to files relating to safety control such as safety programs or safety CPU parameters. (☞ Page 619 File operation available)
Online Change		The online change cannot be executed in safety program.*1
Device/label test		Device/label tests (value changes) cannot be conducted from external devices for the following. (This includes value changes from the test function or monitor windows, and change requests from other external devices.) • Safety global devices • Safety local devices • Safety global labels • Safety local labels • Standard/safety shared label
Memory operation		Memory operations for the CPU built-in memory (including extended SRAM cassette) cannot be performed from the engineering tool. (☞ Page 616 Memory Operation) However, SD memory card initialization is possible.
Device test with execution condition		Only the registration status update can be performed. (☞ Page 637 Operations in each safety operation mode)
Security function	Security key authentication function	The following operations cannot be performed for the Safety CPU or extended SRAM cassette. • Registration • Delete
	File password function	The following operations cannot be performed for the Safety CPU. • Registration • Change • Delete

*1 Online changes (ladder block) in TEST MODE are not possible in some firmware version of the safety CPU. (Page 1139 Added and Enhanced Functions)

34.2 Continuous RUN Prevention in TEST MODE

This function prevents the Safety CPU from running continuously for a long time in TEST MODE. When the continuous RUN time has exceeded the allowed time, a continuation error occurs.

Measuring the continuous RUN time in TEST MODE

When the Safety CPU enters RUN state in TEST MODE, measurement of RUN time starts. When any of the following operations is performed, the Safety CPU stops the measurement and clears the measured value.

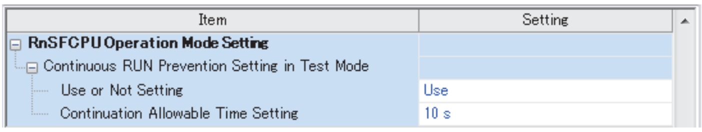
- The operating status of the Safety CPU is changed to STOP or PAUSE.
- The Safety CPU is powered off or is reset.
- The mode is changed to SAFETY MODE.
- An error is cleared.

Setting method

Set the continuous RUN allowed time in TEST MODE.

[CPU Parameter] ⇨ [RAS Setting] ⇨ [RnSFCPU Operation Mode Setting]

Window



Displayed items

Item		Description	Setting range	Default
Continuous RUN prevention setting in TEST MODE	Use or Not Setting	Selects whether to use the function or not.	• Use • Not Use	Use
	Continuation Allowable Time Setting	Sets the continuous RUN allowed time in TEST MODE.	1 to 86400 seconds	10 seconds

Checking the continuous RUN time in TEST MODE

The continuous RUN time in TEST MODE can be checked at SD1840 and SD1841 (Continuous RUN time in TEST MODE). (☞ Page 966 List of Special Register Areas)
Also, if the permissible TEST MODE continuous RUN time set in the parameters is exceeded, this can be checked by ensuring that SM1840 (Allowed time over flag for continuous RUN in TEST MODE) is on. (☞ Page 930 List of Special Relay Areas)

34.3 Safety Diagnostic Function

The following lists self-diagnostic functions specific to the Safety CPU.

Item*1		Description	Diagnostic timing	Error code
Memory diagnosis	RAM diagnosis	Detects errors occurring at program memory, device memory, and memory used by the system.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3C20H, 3C21H, 3C22H, 3C2FH, 3C33H
	F/W diagnosis	Diagnoses whether the firmware stored in the ROM is corrupt.	• When power is turned off and on • When the CPU module is reset	3C10H
Program diagnosis	Operation circuit diagnosis	Detects errors at operation circuits used to operate safety programs.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3C15H
	File verification	Detects errors in stored safety programs and safety parameters.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	2180H, 36E1H, 3C33H, 3C34H
		Diagnoses whether safety programs and safety parameters stored in both the Safety CPU and safety function module are the same.	• When power is turned off and on • When the CPU module is reset	2180H, 3640H
Operation result verification	Output data verification	Verifies output results calculated in the Safety CPU and safety function module.	When executing safety cycle processing	3E60H
System diagnosis	Time monitoring	Monitors the execution status of the respective Safety CPU and safety function module processing.	Always	3E61H
	Microcomputer diagnosis	Detects microcomputer internal register errors.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3C00H, 3C01H
Power supply voltage monitoring	Power supply voltage monitoring	Detects voltage errors to prevent operation with voltage outside the guaranteed operating range.	Always	— (Performs a shutdown.)
	Power supply voltage monitoring/shutdown circuit diagnosis	Monitors whether the circuit monitoring the power supply voltage and circuit used to perform shutdown are functioning properly.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3E10H, 3E11H
Reset circuit monitoring	Reset circuit monitoring diagnosis	Diagnoses whether the reset has been performed correctly.	• When power is turned off and on • When the CPU module is reset	3E12H
Clock stop detection	Clock diagnosis	Detects clock errors (frequency errors).	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3C16H
CRC calculation circuit diagnosis	CRC calculation circuit diagnosis	Diagnoses whether the CRC calculation circuit is capable of correctly calculating CRC.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3E01H
Module	Module OS diagnosis	Verifies whether the Safety CPU and safety function module are functioning normally without running out of control.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3C11H, 3C17H

*1 Self-diagnostics is performed in both Safety CPU and safety function module.

34.4 Identification Check for Safety Data

This function checks if the project data created using the engineering tool and the data in the Safety CPU are the same, and confirms that the program executed in SAFETY MODE is the one written by the user.

This function compares files in the engineering tool with files stored in the Safety CPU, and shows the comparison result.

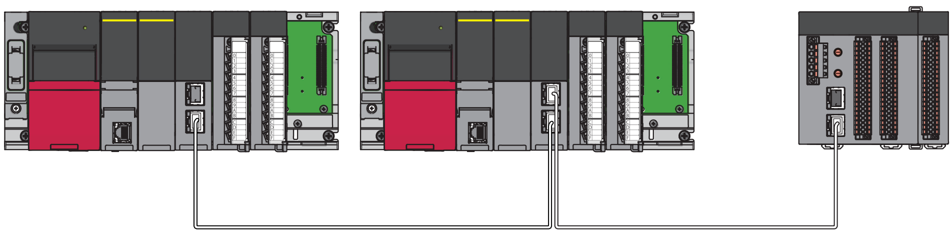
[Online] ⇨ [Safety PLC Operation] ⇨ [Check Safety Data Identity]

For details, refer to the following.

GX Works3 Operating Manual

34.5 Safety Communication Function

This function communicates data between the Safety CPU and modules supporting safety functions using safety protocols.



The Safety CPU performs safety communications between safety stations using the following network.

- CC-Link IE TSN (safety communication function)
- CC-Link IE Field Network (safety communication function)

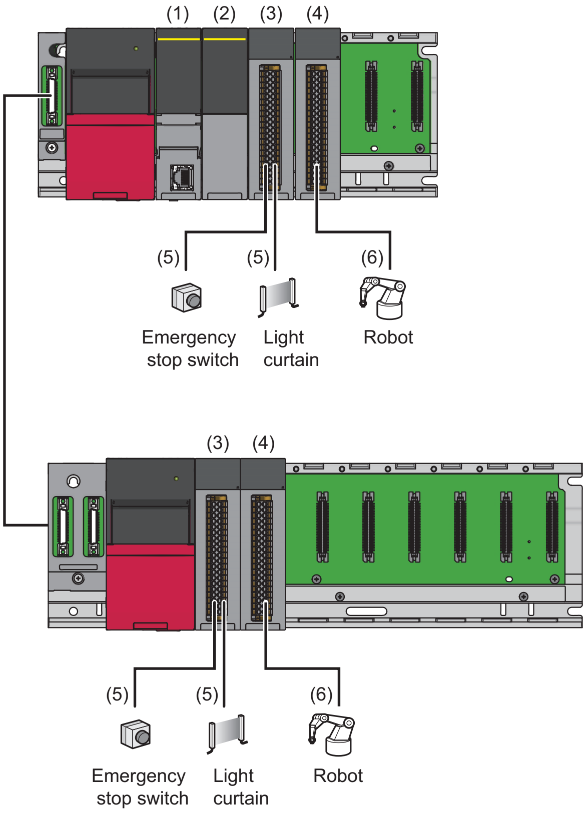
For details on the function and settings required for safety communications, refer to the following.

MELSEC iQ-R CC-Link IE TSN User's Manual (Application)

MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

34.6 Safety I/O Refresh Function

This function periodically communicates I/O data between the I/O module with safety functions and safety CPU using safety protocols.



(1) Safety CPU
(2) Safety function module
(3) Input module with safety functions
(4) Output module with safety functions
(5) Input signal
(6) Output signal

For details on the function and settings required for safety I/O refresh, refer to the following.

MELSEC iQ-R I/O Module (With Safety Functions) User's Manual

34.7 Device/Label Memory Area Setting

The capacity of each area on the device/label memory can be specified.

Default capacity

The default capacity of each area is as follows.

Item	R08SFCPU	R16SFCPU	R32SFCPU	R120SFCPU
Standard device area	40K words	40K words	40K words	40K words
Safety device area	20K words	20K words	20K words	20K words
Safety label area	20K words	20K words	20K words	20K words
Safety local device area	0K words	0K words	0K words	0K words
Standard/safety shared label area	10K words	10K words	10K words	10K words
Standard label area	40K words	50K words	90K words	110K words
Standard latch label area	2K words	2K words	4K words	4K words
Standard local device area	0K words	0K words	0K words	0K words
File storage area	457K words	713K words	969K words	1481K words

Point

The standard local device area capacity obtained by subtracting the total capacity of the standard device area, standard label area, standard latch label area, file storage area, and safety device/safety label area capacity, and the total standard/safety shared label area from the total device/label memory is set. Note, however, that even if the total capacity of the standard device area, standard label area, and safety device/safety label area capacity is smaller than the following, the remaining capacity cannot be assigned to the standard local device area. (The remaining area will be an unused area.)

- R08SFCPU: 50K words
- R16SFCPU: 60K words
- R32SFCPU: 70K words
- R120SFCPU: 90K words

Point

The safety local device area capacity obtained by subtracting the total capacity of the safety device area and safety label area from the safety device/label memory capacity is set.

Setting range of capacity of each area

The following tables list the setting range of the capacity of each area on the device/label memory.*1

*1 The remaining capacity of other areas is automatically set as the capacity of the standard local device area and safety local device area. (☞ Page 632 Default capacity)

R08SFCPU

Area	Setting range of capacity of each area				
	Without an extended SRAM cassette	With an extended SRAM cassette (1MB)	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (4MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 588K words	2 to 1100K words	2 to 1612K words	2 to 2636K words	2 to 4684K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 586K words	0 to 1098K words	0 to 1610K words	0 to 2634K words	0 to 4682K words
Standard latch label area	0 to 544K words	0 to 1056K words	0 to 1568K words	0 to 2592K words	0 to 4640K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 544K words	0 to 1056K words	0 to 1568K words	0 to 2592K words	0 to 4640K words

R16SFCPU

Area	Setting range of capacity of each area				
	Without an extended SRAM cassette	With an extended SRAM cassette (1MB)	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (4MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 854K words	2 to 1366K words	2 to 1878K words	2 to 2902K words	2 to 4950K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 852K words	0 to 1364K words	0 to 1876K words	0 to 2900K words	0 to 4948K words
Standard latch label area	0 to 800K words	0 to 1312K words	0 to 1824K words	0 to 2848K words	0 to 4896K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 800K words	0 to 1312K words	0 to 1824K words	0 to 2848K words	0 to 4896K words

R32SFCPU

Area	Setting range of capacity of each area				
	Without an extended SRAM cassette	With an extended SRAM cassette (1MB)	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (4MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 1152K words	2 to 1664K words	2 to 2176K words	2 to 3200K words	2 to 5248K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 1150K words	0 to 1662K words	0 to 2714K words	0 to 3198K words	0 to 5246K words
Standard latch label area	0 to 1088K words	0 to 1600K words	0 to 2112K words	0 to 3136K words	0 to 5184K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 1088K words	0 to 1600K words	0 to 2112K words	0 to 3136K words	0 to 5184K words

R120SFCPU

Area	Setting range of capacity of each area				
	Without an extended SRAM cassette	With an extended SRAM cassette (1MB)	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (4MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 1684K words	2 to 2196K words	2 to 2708K words	2 to 3732K words	2 to 5780K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 1682K words	0 to 2194K words	0 to 2706K words	0 to 3730K words	0 to 5778K words
Standard latch label area	0 to 1600K words	0 to 2112K words	0 to 2624K words	0 to 3648K words	0 to 5696K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 1600K words	0 to 2112K words	0 to 2624K words	0 to 3648K words	0 to 5696K words

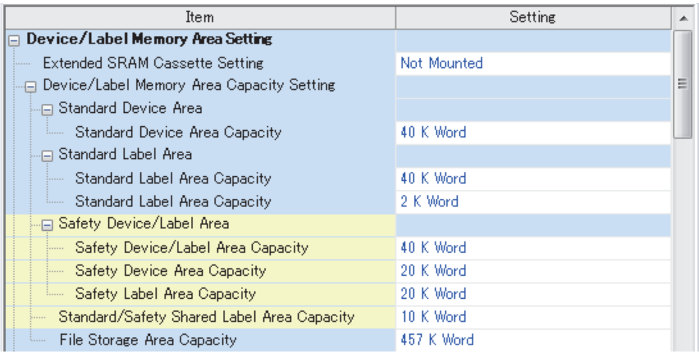
Setting method

The capacity in each data area of the device/label memory can be changed.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Setting]

Operating procedure

"Device/Label Memory Area Setting" window



1. Set whether to use an extended SRAM cassette or not in "Extended SRAM Cassette Setting".

2. Set the capacity of each area in "Device/Label Memory Area Capacity Setting".

Displayed items

Item			Description	Setting range	Default
Extended SRAM Cassette Setting			Sets the capacity of the extended SRAM cassette if used.	• Not Mounted • 1MB • 2MB • 4MB • 8MB	Not Mounted
Device/Label Memory Area Capacity Setting	Standard Device Area	Standard Device Area Capacity	Sets the capacity of the device area used for standard global devices.	☞ Page 633 Setting range of capacity of each area	☞ Page 632 Default capacity
	Standard Label Area	Standard Label Area Capacity	Sets the capacity of the label area used for standard global labels and standard local labels.		
		Standard Latch Label Area Capacity	Sets the capacity of the standard latch label area used for latch-type labels.		
	Safety Device/Label Area	Safety Device/Label Area Capacity	Sets the total capacity for the safety device area and safety label area.		
		Safety Device Area Capacity	Sets the capacity of the device area used for safety global devices.		
		Safety Label Area Capacity	Sets the capacity of the label area used for safety global labels and safety local labels.		
	Standard/Safety Shared Label Area Capacity		Sets the capacity of the label area used for standard/safety shared labels.		
	File Storage Area Capacity		Sets the capacity of the file storage area used to store files, such as file register files.		

Point

Note that the total of the capacity of each area (including the capacity of the local device area/safety local device area) and safety special relay/safety special register area capacity should not exceed the capacity of the device/label memory. (MELSEC iQ-R CPU Module User's Manual (Startup))

Standard device area setting range

The number of points of each device used in standard programs and capacity in which the total number of device points can be stored are set. Set the total number of device points within the device area range.

Type	Device name	Symbol	Range of use*1	Setting unit
Bit	Input	X	X0 to X2FFF	—
	Output	Y	Y0 to Y2FFF	—
	Internal relay	M	M0 to M94674943	64 points
	Link relay	B	B0 to B5A49FFF	64 points
	Annunciator	F	F0 to F32767	64 points
	Link special relay	SB	SB0 to SB5A49FFF	64 points
	Edge relay	V	V0 to V32767	64 points
	Latch relay	L	L0 to L32767	64 points
Word	Timer	T	T0 to T5259711	32 points
	Retentive timer	ST	ST0 to ST5259711	32 points
	Long timer	LT	LT0 to LT1479295	1 point
	Long retentive timer	LST	LST0 to LST1479295	1 point
	Counter	C	C0 to C5259711	32 points
	Long counter	LC	LC0 to LC2784543	32 points
	Data register	D	D0 to D5917183	4 points
	Link register	W	W0 to W5A49FF	4 points
	Link special register	SW	SW0 to SW5A49FF	4 points

*1 This is the maximum range when the R120SFCPU with an extended SRAM cassette (8MB) (NZ2MC-8MBS) is used. The number of points varies depending on the model of the Safety CPU, whether to use an extended SRAM cassette, and the type of the cassette.

Safety device area setting range

The number of points of each safety device used in safety programs and capacity in which the total number of safety device points can be stored are set. Set the total number of safety device points within the safety device area range.

Type	Device name	Symbol	Range of use	Setting unit
Bit	Safety input	SA\X	SA\X0 to SA\X2FFF*1	—
	Safety output	SA\Y	SA\Y0 to SA\Y2FFF*1	—
	Safety internal relay	SA\M	SA\M0 to SA\M638975	64 points
	Safety link relay	SA\B	SA\B0 to SA\B9BFFF	64 points
Word	Safety timer	SA\T	SA\T0 to SA\T35487	32 points
	Safety retentive timer	SA\ST	SA\ST0 to SA\ST35487	32 points
	Safety counter	SA\C	SA\C0 to SA\C35487	32 points
	Safety data register	SA\D	SA\D0 to SA\D39935	4 points
	Safety link register	SA\W	SA\W0 to SA\W9BFF	4 points

*1 If using the range from 2000 to 2FFF, check the versions of the CPU module and engineering tool. (☞ Page 1139 Added and Enhanced Functions)

34.8 Device Test with Execution Conditions

This section describes the device test with execution conditions for the Safety CPU. The main points of difference between this function for the Safety CPU and one for the standard CPU are the following:

Data that can be set

In addition to the data can be set in the standard CPU, digit-specified bit labels can be set in the Safety CPU. (In other CPU modules, digit-specified labels cannot be set.)

☞ Page 187 Data that can be set

Programs that can be set

Only standard programs (ladder programs) can be set.

Operations in each safety operation mode

The following table shows whether each operation can be performed in each safety operation mode.

○: Yes, ×: No

Operation		Safety operation mode	
		TEST MODE	SAFETY MODE
Registration		○	×
Registering/disabling settings from the list window	Reading registration status	○	○
	Disabling selected settings	○	×
	Registering settings in batch	○	×
	Disabling settings in batch	○	×
Disabling settings in batch*1		○	×

*1 This operation is performed from [Debug] ⇨ [Device Test with Execution Condition] ⇨ [Batch Disable] of the engineering tool.

Behavior when safety operation mode is switched

If the safety operation mode is switched from TEST MODE to SAFETY MODE while the CPU module has registered settings of the device test with execution conditions, all the settings are disabled.

Disabling device test with execution conditions

In addition to the operations from the engineering tool, the following operations can be used to disable the device test with execution conditions.

- Powering off and on
- Resetting the CPU module
- Switching the safety operation mode (TEST MODE to SAFETY MODE)
- Writing a program to the CPU built-in memory by writing data to the programmable controller while the CPU module is in the STOP state*1*4
- Deleting a program in the CPU built-in memory by deleting data in the programmable controller while the CPU module is in the STOP state*1
- Initializing the CPU built-in memory while the CPU module is in the STOP state*1
- Initialization of all programmable controller information while the CPU module is in the STOP state*1
- Changing a program online (the online change (ladder block)*6 or the file batch online change) (☞ Page 194 Operation during online change)
- Writing a standard global label to the CPU built-in memory by writing data to the programmable controller while the CPU module is in the STOP state*2*4
- Deleting a standard global label in the CPU built-in memory by deleting data in the programmable controller while the CPU module is in the STOP state*2
- Writing CPU parameters to the CPU built-in memory by writing data to the programmable controller while the CPU module is in the STOP state*3*4
- Deleting the CPU parameters and Safety CPU parameters in the CPU built-in memory by deleting data to the programmable controller while the CPU module is in the STOP state*5

*1 The registered settings of device test with execution conditions for the program to be changed are disabled when the operating status of the CPU module is changed from STOP to RUN after the operation. (They are not disabled at the completion of the operation.)

*2 All the registered settings that specify standard global labels are disabled when the operating status of the CPU module is changed from STOP to RUN after the operation. (They are not disabled at the completion of the operation.)

*3 When the CPU parameters are changed, all the registered settings that specify standard local devices or standard local labels are disabled when the operating status of the CPU module is changed from STOP to RUN after the operation. (They are not disabled at the completion of the operation.)

*4 When the same programs, standard global labels, or CPU parameters are written, the registered settings are not disabled.

*5 All the registered settings of device test with execution conditions are disabled when the operating status of the CPU module is changed from STOP to RUN after the operation. (They are not disabled at the completion of the operation.)

*6 When the online change (ladder block) is performed while the CPU module is in the STOP state, the device test with execution conditions registered for the program to be changed is disabled when the operating status of the CPU module is changed from STOP to RUN after the operation. (They are not disabled at the completion of the operation.)

In combination with safety cycle processing

The device test with execution conditions takes precedence over the safety cycle processing. For this reason, if a short interval such as 1.0ms is specified for the safety cycle time, the safety cycle processing may not be performed at the specified interval and an error may occur.

34.9 Data Logging Function

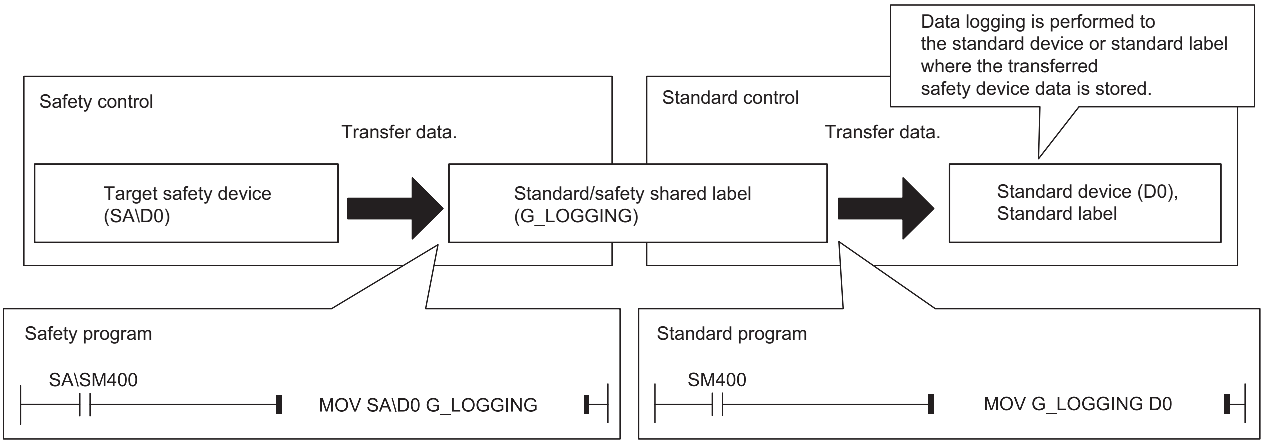
This section describes the data logging function of the Safety CPU. The main points are differences from the data logging function of the standard CPU.

Data to be collected

A safety device or safety label cannot be specified as the data that can be collected.
To specify a standard local device or standard label, check the versions of the CPU module, engineering tool, and CPU Module Logging Configuration Tool. (☞ Page 1139 Added and Enhanced Functions)

Point

Although data logging cannot be executed by specifying safety devices and safety labels, data in safety devices and safety labels can be collected by taking the following steps (❶ to ❸).
❶: Transfer the values of the target safety device or safety label to a standard/safety shared label.
❷: Transfer the values of the standard/safety shared label specified in Step ❶ to a standard device or standard label.
❸: Perform the logging of standard devices or standard labels specified in Step ❷.



Data collection conditions

■Time specification

For time specification, check the versions of the CPU module and CPU Module Logging Configuration Tool. (☞ Page 1139 Added and Enhanced Functions)

■Interrupt occurrence

- I48 and I49 cannot be specified.
- To specify interrupt occurrence, check the versions of the CPU module and CPU Module Logging Configuration Tool. (☞ Page 1139 Added and Enhanced Functions)

■Condition specification

Condition specification is not available.

Trigger conditions

A safety device and safety label cannot be specified as the monitoring data of when "Device/label change specification" is selected as the trigger condition.
In addition, step No. specification cannot be used for safety programs.

Device comment output

Device comments will not be output to the storage file even if "Device comment output" in the output setting is set to "Output device comment".

Storage format of data logging files

The CSV file format cannot be used as the storage format.

Storage location of data logging files

A function memory cannot be used.

File switching condition

Condition specification is not available.

Data logging file transfer (auto transfer to FTP server)

Data logging file transfer (auto transfer to FTP server) is not available.

Auto logging

If a sampling interval of the data logging function is set to the one other than "Each scanning cycle", data logging registration fails at the start of auto logging and an error occurs.

Precautions to take when using the data logging function

■Operation when online change is executed while data logging is in progress

In the operation when online change is executed while data logging is in progress, files specific to the Safety CPU are added to the change target files.

- When SD940 (Stop direction at file change on label specification) is off

Function to be executed later		Operation based on the setting details of data logging in progress*1				
Function	**Change target file**	**Standard global device**	**Standard local device**	**Standard global device**	**Standard local device**	**Step No.**
• Online change (ladder block) • File batch online change	Safety program file, safety FB file	Continue	Continue	Continue	Continue	Stop*2
	Safety global label setting file	Continue	Continue	Continue	Continue	Continue
	Standard/safety shared label setting file	Continue	Continue	Continue	Continue	Continue

*1 This is applicable for when the standard device and standard label are specified as "target data", "collection conditions", or "trigger conditions" or when the step No. is specified as "collection conditions" or "trigger conditions".

*2 During the execution of a "function to be executed later", if the change involves only a standard/safety local label name change or the addition of a standard/safety local label definition, an error may not occur, allowing the operation to continue.

- When SD940 (Stop direction at file change on label specification) is on

Function to be executed later		Operation based on the setting details of data logging in progress*3				
Function	**Change target file**	**Standard global device**	**Standard local device**	**Standard global device**	**Standard local device**	**Step No.**
• Online change (ladder block) • File batch online change	Safety program file	Continue	Continue	Continue	Continue	Stop*4
	Safety FB file	Continue	Continue	Continue	Continue	Stop*4
	Safety global label setting file	Continue	Continue	Continue	Continue	Continue
	Standard/safety shared label setting file	Continue	Continue	Continue	Continue	Continue

*3 This is applicable for when the standard device and standard label are specified as "target data", "collection conditions", or "trigger conditions" or when the step No. is specified as "collection conditions" or "trigger conditions".

*4 During the execution of a "function to be executed later", if the change involves only a standard/safety local label name change or the addition of a standard/safety local label definition, an error may not occur, allowing the operation to continue.

*5 An error occurs if a program file which includes the specified standard/safety local label is written.

■Operation when data logging is executed while online change is in progress

In the operation when data logging is executed while online change is in progress, files specific to the Safety CPU are added to the change target files.

Function that has been already executed		Operation based on the setting details of data logging in progress*1				
Function	**Change target file**	**Standard global device**	**Standard local device**	**Standard global device**	**Standard local device**	**Step No.**
• Online change (ladder block) • File batch online change	Safety program file, safety FB file	Can be executed	Can be executed	Cannot be executed	Cannot be executed	Cannot be executed
	Safety global label setting file	Can be executed	Can be executed	Cannot be executed	Cannot be executed	Cannot be executed
	Standard/safety shared label setting file	Can be executed	Can be executed	Cannot be executed	Cannot be executed	Cannot be executed

*1 This is applicable for when the standard device and standard label are specified as "target data", "collection conditions", or "trigger conditions" or when the step No. is specified as "collection conditions" or "trigger conditions".

■When a file operation related to the data logging is executed during the execution of online change

In the operation when a file operation related to the data logging is executed during the execution of online change, files specific to the Safety CPU are added to the change target files.

Function that has been already executed		Operation based on the setting details of data logging in progress*1				
Function	**Change target file**	**Standard global device**	**Standard local device**	**Standard global device**	**Standard local device**	**Step No.**
• Online change (ladder block) • File batch online change	Safety program file, safety FB file	Can be executed	Can be executed	Cannot be executed	Cannot be executed	Can be executed
	Safety global label setting file	Can be executed	Can be executed	Cannot be executed	Cannot be executed	Can be executed
	Standard/safety shared label setting file	Can be executed	Can be executed	Cannot be executed	Cannot be executed	Can be executed

*1 This is applicable for when the standard device and standard label are specified as "target data", "collection conditions", or "trigger conditions" or when the step No. is specified as "collection conditions" or "trigger conditions".

■Combined operation of the data logging function with another function, and the transition of the CPU module status from STOP to RUN

In the combined operation of the data logging function with another function, and the transition of the CPU module status from STOP to RUN, files specific to the Safety CPU are added to the change target files.

- When SD940 (Stop direction at file change on label specification) is off

Function to be executed later		Operation based on the setting details of data logging in progress*1				
Function	**Change target file**	**Standard global device**	**Standard local device**	**Standard global device**	**Standard local device**	**Step No.**
Writing data to the programmable controller	Safety CPU parameter file	Continue	Continue	Continue	Continue	Continue
	Safety program file	Continue	Continue	Continue	Continue	Continue
	Safety global label setting file	Continue	Continue	Continue	Continue	Continue*2
	Standard/safety shared label setting file	Continue	Continue	Continue	Continue	Continue

*1 This is applicable for when the standard device and standard label are specified as "target data", "collection conditions", or "trigger conditions" or when the step No. is specified as "collection conditions" or "trigger conditions".

*2 The operation stops if the step No. does not exist when the CPU module operating status is changed from STOP to RUN.

- When SD940 (Stop direction at file change on label specification) is on

Function to be executed later		Operation based on the setting details of data logging in progress*3				
Function	**Change target file**	**Standard global device**	**Standard local device**	**Standard global device**	**Standard local device**	**Step No.**
Writing data to the programmable controller	Safety CPU parameter file	Continue	Continue	Continue	Continue	Continue
	Safety program file	Continue	Continue	Continue	Continue	Continue
	Safety global label setting file	Continue	Continue	Continue	Continue	Continue
	Standard/safety shared label setting file	Continue	Continue	Continue	Continue	Continue

*3 This is applicable for when the standard device and standard label are specified as "target data", "collection conditions", or "trigger conditions" or when the step No. is specified as "collection conditions" or "trigger conditions".

34.10 CPU Module Data Backup/Restoration Function

This section describes the data backup/restoration function for the Safety CPU. The main points of difference between this function for the Safety CPU and one for the standard CPU are the following:

Backup/restoration target data

■Backup/restoration target files

In the Safety CPU, the following files are included in the backup/restoration target files.

- Safety program file
- Safety FB file
- Safety CPU parameter file
- Safety module parameter file
- Safety global label setting file*1
- Standard/safety shared label setting file*1

*1 These files are restored when 1 (Device/label data only) is specified in SD954 (Restoration target data setting). When all the target data except for device/label data are specified, these files are not restored.

■File size

Among the files created during backup, the files shown below have different file sizes.

File type	File size
Backup data file for backing up CPU module data	• When the data logging setting has been registered: 492 bytes • When the data logging setting has not been registered: 64 bytes
Device/label data file for backing up CPU module data	1096178 + (Number of programs × 142) + (Number of file registers × 134) + S1 + S2*1 + S3 bytes*2 • S1: Total size of the set devices (global devices) = (N1 + N2 × 2) × 18 + ((G1 ÷ 16) + G2 + (G3 × 2) + (G4 + (G4 ÷ 16) × 2) + (G5 × 2) + (G5 ÷ 16) × 2) + (G6 × 8)) × 2 • N1: Number of device types used from M, L, B, F, SB, V, S, D, W, SW, SA\X, SA\Y, SA\M, SA\B, SA\D, SA\W, U3E0\HG, U3E1\HG, U3E2\HG, U3E3\HG, Z, LZ, and RD • N2: Number of device types used from T, ST, C, LC, LT, LST, SA\T, SA\ST, and SA\C • G1: Total number of points of M, L, B, F, SB, V, S, SA\X, SA\Y, SA\M, and SA\B • G2: Total number of points of D, W, SW, SA\D, SA\W, U3E0\HG, U3E1\HG, U3E2\HG, U3E3\HG, Z, and RD • G3: Number of points of LZ • G4: Total number of points of T, ST, C, SA\T, SA\ST, and SA\C • G5: Number of points of LC • G6: Total number of points of LT and LST • S2: Total size of the set local devices = 16 + number of standard programs using the local devices*3 × (12 + ((N2 + N3 × 2) × 18) + ((L1 ÷ 16) + L2 + (L3 × 2) + (L4 + (L4 ÷ 16) × 2) + ((L5 × 2) + (L5 ÷ 16) × 2) + (L6 × 8)) × 2) + number of safety programs using the local devices × (12 + ((N4 + N5 × 2) × 18) + ((L7 ÷ 16) + L8 + (L9 + (L9 ÷ 16) × 2) • N2: Number of local device types used from M, V, D, Z, and LZ • N3: Number of local device types used from T, ST, C, LC, LT, and LST • N4: Number of local device types used from SA\M and SA\D • N5: Number of local device types used from SA\T, SA\ST, and SA\C • L1: Total number of points of the local devices M and V • L2: Total number of points of the local devices D and Z • L3: Number of points of the local device LZ • L4: Total number of points of the local devices T, ST, and C • L5: Number of points of the local device LC • L6: Total number of points of the local devices LT and LST • L7: Number of points of the local device SA\M • L8: Number of points of the local device SA\D • L9: Total number of points of the local devices SA\T, SA\ST, and SA\C • S3 = (Label area capacity (word) + latch label area capacity (word) + safety label area capacity (word) + standard/safety shared label area capacity (word) + module label assignment area capacity (word)) × 2

*1 This is added only when local devices are used.
*2 The total file size is adjusted by the system so that the size will be a multiple of 4 bytes. (Up to 3 bytes are added.)
*3 When the number of device points assigned to local devices is 0, this becomes 0.

■Backup/restoration target device data

The following table lists the safety devices of the Safety CPU and shows whether they can be backed up or restored.

○: Yes

Classification	Device name	Symbol	Backup*1	Restoration*1
Safety user device	Safety input	SA\X	○	○
	Safety output	SA\Y	○	○
	Safety internal relay	SA\M	○	○
	Safety link relay	SA\B	○	○
	Safety timer	SA\T	○	○
	Safety retentive timer	SA\ST	○	○
	Safety counter	SA\C	○	○
	Safety data register	SA\D	○	○
	Safety data register	SA\W	○	○
	Safety special relay	SA\SM	○	○*2*3
	Safety special register	SA\SD	○	○*2*3

*1 Device data may be overwritten by safety inputs (refresh).

*2 Areas used by the system may be overwritten.

*3 Whether to restore these areas can be set to the bit 14 of SD955 (Restoration function setting).

Execution of the backup/restoration function

The following table shows whether the backup/restoration function can be executed while the system is in each safety operation mode.

■Backup/restoration of safety operation mode status

The safety operation mode status (TEST MODE or SAFETY MODE) is not backed up/restored. Therefore, the mode after restoration of the data, which are backed up when the system is in SAFETY MODE, will not be SAFETY MODE.

○: Yes, ×: No

Function	Safety operation mode	
	TEST MODE	SAFETY MODE
Backup function	○	○
Restoration function	○	×

Firmware version of the CPU module at executing restoration

To execute restoration, check that the firmware version of the restoration destination CPU module is the same as or later than the firmware version at the CPU module backup. Otherwise, the functions may not operate normally after restoration.

Ex.

When a CPU module with firmware version "29" is backed up with its safety protocol version set to "2" in the safety communication setting, the firmware version of the restoration destination CPU module must be "29" or later. (If a CPU module with firmware version "28" or earlier is used, the safety communication function will not operate normally.)

Backup function

This section describes the data backup function for the Safety CPU.

Precautions

This section describes the precautions for the backup function for the Safety CPU.

■Operation and function cannot be performed or executed simultaneously with backup function

In the Safety CPU, the following operation and function also cannot be performed or executed simultaneously with the backup function, in addition to those for the programmable controller CPU. Other operations and functions than the following are common to the programmable controller CPU. (☞ Page 314 Operations and functions that cannot be performed)

Operation and function	
Operation from the engineering tool	Switching safety operation mode
	User authentication function (password change of the programmable controller, writing user information to the programmable controller, initialization of all programmable controller information)

Restoration function

This section describes the data restoration function for the Safety CPU.

Restoration by enabling the automatic restoration setting

Enabling the automatic restoration setting performs the automatic restoration.*1

*1 Enable the automatic restoration setting from menus of the engineering tool. (☞ Page 645 Enable/Disable the automatic restoration setting)

■Enable/Disable the automatic restoration setting

Set "Automatic Restore Setting" to "Enable" in the engineering tool and the bit 0 of SD955 (Restoration function setting) turns on automatically. Setting it to "Disable" turns off the bit 0 of SD955 automatically.

[Online] ⇨ [Safety PLC Operation] ⇨ [Automatic Restore Setting]

When the system is operating in SAFETY MODE, the "Automatic Restore Setting" cannot be set.

■Operating procedure

1. Set the data to restore in SD954.
2. Set a restoration target folder in the areas from SD956 (Restoration target date folder setting) to SD958 (Restoration target number folder setting). (These settings are not required when the bit 13 of SD955 is on.)
3. Set values to the bit 1 and bits from 13 to 15 of SD955.
4. Enable "Automatic Restore Setting" in the engineering tool. (☞ Page 645 Enable/Disable the automatic restoration setting)
5. Power off and on, or reset the CPU module.*1

*1 If the 301BH error is detected, reset the CPU module again. With other errors than 301BH, restoration is completed with an error.

Point

The status of the automatic restoration setting is kept enabled until the automatic restoration is completed successfully or the setting is disabled from the engineering tool.

34

Precautions

This section describes the precautions for the restoration function for the Safety CPU.

■Operation and function cannot be performed or executed simultaneously with restoration function

In the Safety CPU, the following operation and function also cannot be performed or executed simultaneously with the restoration function, in addition to those for the programmable controller CPU. Other operations and functions than the following are common to the programmable controller CPU. (☞ Page 323 Operations and functions that cannot be performed)

Operation and function	
Operation from the engineering tool	Switching safety operation mode
	User authentication function (password change of the programmable controller, writing user information to the programmable controller, initialization of all programmable controller information)

■When online operations requiring authentication are performed from engineering tool after restoration

When 0 (All the target data) or 2 (All the target data except for the device/label data) is specified in SD954 (Restoration target data setting), the user management information at the time of backup is restored to the CPU module. Because of this, the user has to log on with that user management information.

■Switching safety operation mode

The safety operation mode cannot be switched with the automatic restoration setting enabled.

34.11 Write-Protect Function for Device Data (from Outside the CPU Module)

This section describes the write-protect function for device data from outside the Safety CPU. The main points of difference with the standard CPU are described below.

Operations and functions

The following table lists the operations and functions that cannot be executed for devices in the write-protect range.

Operation and function			
Operation from the engineering tool	Current value change in the watch window		
	Remote operation	Remote RUN*1	
	CPU memory operation	Device/label memory value clear	Device/label zero clear
			File register zero clear
			Device/label/file register latch clear
	Writing data to the programmable controller	Device memory	
		Device initial value*2	
		File register	
Operation using a GOT	Device write		
Operation using SLMP	Device write using SLMP		
Operation using instructions	Writing device data from another CPU module		
	Device write from the programmable controller of another station		
	Remote RUN at device memory clear specification from the programmable controller of another station		
Other operations	Writing device data using the simple CPU communication function		
	Predefined protocol support function		

*1 This function cannot be executed when the device memory clear is specified.

*2 When this function is enabled, the files cannot be written.

■Operation using SLMP

Device write processing is disabled when the write-protect range is set in the device write using SLMP. In addition, device write processing is disabled when the following commands are executed in the operations such as the predefined protocol support function, SLMP frame send instruction, or access from an external device.

Type	Operation	Command	Subcommand
Device	Write	1401	00□0, 00□1
			00□2, 00□3
	Write Random	1402	00□0, 00□1
			00□2, 00□3
	Write Block	1406	00□0
			00□2
Label	Array Label Write	141A*1	0000
	Label Write Random	141B*1	0000
Remote Control	Remote Run	1001*2	0000
	Remote Latch Clear	1005	0000

*1 Device write processing is disabled only when a device is assigned to a label.

*2 Device write processing is disabled only when the device memory clear is specified.

■Operation using instructions

Write processing is disabled when the device write is performed by a multiple CPU dedicated instruction or a module dedicated instruction (including a device clear instruction (dedicated instruction)) to devices in the write-protect range.

Major classification	Classification		Instruction name
Writing device data from another station	Module dedicated instructions	Writing data to the programmable controller on another station	JP.WRITE, GP.WRITE
		Writing data to the programmable controller on another station (with notification)*1	JP.SWRITE, GP.SWRITE
		Reading data from the programmable controller on another station (with notification)*1	JP.SREAD, GP.SREAD
		Writing data to target station	J(P).RIWT, G(P).RIWT
Writing device data from another CPU module	Multiple CPU dedicated instruction	Writing device data to another CPU module	D(P).DDWR, M(P).DDWR
Clearing device data another station	Module dedicated instructions	Remote RUN*2	J(P).RRUN, G(P).RRUN, Z(P).RRUN, J(P).REQ, G(P).REQ

*1 Writing to notification devices is also disabled.
*2 Write processing is disabled only when the device clear is specified.

34.12 RAS Functions

This section describes the RAS functions of the safety CPU.

Self-diagnostic function

This section describes the self-diagnostic function of the Safety CPU module and safety function module. As for the Safety CPU, only the parts that differ from the standard CPU are described.

How to check errors

Errors in the Safety CPU can be checked in the same way as for the standard CPU. (☞ Page 138 Self-Diagnostics Function)

Errors in the safety function module can be checked in the following ways.

■Using the buffer memory

When the safety function module detects an error, the corresponding error code is stored in Un\G0 (Latest self-diagnostic error code). If multiple errors are detected, the latest error code is stored in Un\G0. Up to 16 error codes can be stored in Un\G10 (Self-diagnostic error code 1) to Un\G25 (Self-diagnostic error code 16). (The error codes of the 17th error and later are not stored.)

■Using the LED of the safety function module

Like intelligent function modules, the error status can be checked with the ERROR LED. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

■Using the engineering tool

Like intelligent function modules, the error status of the entire system and the history of errors currently occurred or events can be checked on the "Module Diagnostics" window. (📖 GX Works3 Operating Manual)

Error detection setting

■Applicable errors to the error detection setting

The following table lists errors in the Safety CPU for which whether or not to detect the errors can be set.

Error name	Error code
Battery error	1090H
Module verification error	2400H, 2401H
Fuse blown error	2420H

CPU parameter setting for when an error is detected

The "Instruction Execution Error" setting under the "CPU Module Operation Setting at Error Detected" is applied only for standard programs. A stop error always occurs in safety programs. (Even if "Continue" is selected, a stop error occurs.)

[CPU Parameter] ⇨ [RAS Setting] ⇨ [CPU Module Operation Setting at Error Detected]

■Applicable errors to the CPU module operation upon error detection setting

The following table lists the errors in the Safety CPU applicable to the setting that specifies the CPU module operation of when the specific errors are detected.

Error name	Error code
Memory card error	2120H, 2121H
Module verification error	2400H, 2401H
Fuse blown error	2420H
I/O number or network number specification error	2800H, 2801H, 2802H, 2803H, 2804H, 2805H, 2806H, 2807H, 2810H
Device, label, or buffer memory specification error	2820H, 2821H, 2822H, 2823H, 2824H
File name specification error	2840H, 2841H, 2842H
Operation error	3400H, 3401H, 3402H, 3403H, 3404H, 3405H, 3406H, 3420H, 3421H, 3422H, 3423H, 3440H, 3441H, 34A0H

CPU module operation setting

■Applicable errors to the CPU module operation setting

The following table lists the errors in the Safety CPU applicable to the setting that specifies the CPU module operation of when the specific errors have occurred in each intelligent function module.

Error name	Error code
Module moderate error	1200H
Module major error	2441H, 2442H, 2450H

Stop setting

■Applicable errors to the stop setting

The following table lists the errors in the Safety CPU applicable to the setting that specifies the operation of all the CPU modules of when a major or moderate error has occurred in any of the CPU modules.

Error name	Error code
Another CPU module moderate error	1220H
Another CPU module major error	2461H, 2462H, 2470H

Error clear

This function clears all the existing continuation errors occurring at the Safety CPU or safety function module at once.

Errors that can be cleared

This function can be used to clear only the continuation errors listed in the following table.

■Safety CPU

Error name	Error code
Power shutoff	1000H
ROM write count error	1080H
Battery error	1090H
Memory card access error	1100H
SNTP clock setting error	1120H
Default gateway/gateway IP address error	1124H
Own node port number error	1128H
Open specification port number error	1129H
Specified IP address error	112DH
Connection establishment failed	112EH
IP address duplication error	1130H
Socket communications response send error	1133H
TCP connection timeout	1134H
IP address error	1152H
Connection number acquisition error	1155H
Receive buffer securement error	1157H
UDP/IP transmission failed	1165H
TCP/IP transmission failed	1166H
Unsend data transmission error	1167H
PID operation error	11A0H to 11B8H
Module moderate error	1200H, 1210H
Another CPU module moderate error	1220H
Inter-module synchronization processing error	1240H, 1241H
Multiple CPU synchronization processing error	1260H, 1262H
Annunciator ON	1800H
Receive queue full	1830H
Receive processing error	1831H

Error name	Error code
Transient data error	1832H
Constant scan time exceeded	1900H
Safety cycle time exceeded	1A00H
Safety cycle processing error	1A01H
Continuous RUN time in TEST MODE exceeded	1A20H
Destination station error in safety communications	1A40H*1
Parameter mismatch in safety communication destination station	1A50H*1
Device mismatch in safety communication destination station	1A51H*1
Device version mismatch in safety communication destination station	1A52H*1
Timeout in safety communications	1A60H*1, 1A61H*1, 1A62H*1, 1A63H*1, 1A64H*1, 1A65H*1, 1A66H*1
Safety communications receipt data error	1A70H*1, 1A71H*1, 1A72H*1, 1A73H*1
Safety I/O refresh error	1AA0H*2, 1AB0H*2, 1AB1H*2, 1AB2H*2, 1AB3H*2
Safety I/O refresh timeout	1AC0H*2, 1AC1H*2, 1AC2H*2, 1AC3H*2, 1AC4H*2, 1AC5H*2, 1AC6H*2
Safety input refresh data error	1AD0H*2, 1AD1H*2, 1AD2H*2, 1AD3H*2
Memory card error	2120H, 2121H
Module verification error	2400H, 2401H
Fuse blown error	2420H
Module major error	2441H, 2442H
Major error in module	2450H
Another CPU module major error	2461H, 2462H
Another CPU module major error	2470H
Inter-module synchronization signal error	2610H
Multiple CPU synchronization signal error	2630H
I/O number or network number specification error	2800H, 2801H, 2802H, 2803H, 2804H, 2805H, 2806H, 2807H, 2810H
Device, label, or buffer memory specification error	2820H, 2821H, 2822H, 2823H, 2824H
File name specification error	2840H, 2841H
Operation error	3400H, 3401H, 3402H, 3403H, 3404H, 3405H, 3406H, 3420H, 3421H, 3422H, 3440H, 3441H, 34A0H

*1 If a safety communications error (minor error) occurs between safety stations, a safety station interlock will be triggered for safety communications between relevant stations, and safety communications will not be resumed until the safety station interlock is released. The safety station interlock is released with a safety special register (safety station interlock release request for each safety connection).
(☞ Page 1026 List of Safety Special Register Areas)
For details on the safety station interlock status, refer to the following.
MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

*2 If a minor error occurs in safety I/O refresh, the safety I/O refresh is interlocked.
For details on safety I/O refresh interlock, refer to the following.
MELSEC iQ-R I/O Module (With Safety Functions) User's Manual

■Safety function module

Error name	Error code
ROM write count error	1080H
Safety cycle time exceeded	1A00H
Destination station error in safety communications	1A40H*1
Parameter mismatch in safety communication destination station	1A50H*1
Device mismatch in safety communication destination station	1A51H*1
Device version mismatch in safety communication destination station	1A52H*1
Timeout in safety communications	1A60H*1, 1A61H*1, 1A62H*1, 1A63H*1, 1A64H*1, 1A65H*1, 1A66H*1
Safety communications receipt data error	1A70H*1, 1A71H*1, 1A72H*1, 1A73H*1
Safety I/O refresh error	1AA0H*2, 1AB0H*2, 1AB1H*2, 1AB2H*2, 1AB3H*2
Safety I/O refresh timeout	1AC0H*2, 1AC1H*2, 1AC2H*2, 1AC3H*2, 1AC4H*2, 1AC5H*2, 1AC6H*2
Safety input refresh data error	1AD0H*2, 1AD1H*2, 1AD2H*2, 1AD3H*2

*1 If a safety communications error (minor error) occurs between safety stations, a safety station interlock will be triggered for safety communications between relevant stations, and safety communications will not be resumed until the safety station interlock is released. The safety station interlock is released with a safety special register (safety station interlock release request for each safety connection). (☞ Page 1026 List of Safety Special Register Areas)
For details on the safety station interlock status, refer to the following.
MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

*2 If a minor error occurs in safety I/O refresh, the safety I/O refresh is interlocked.
For details on safety I/O refresh interlock, refer to the following.
MELSEC iQ-R I/O Module (With Safety Functions) User's Manual

How to clear errors

Errors in the Safety CPU can be cleared in the same way as for the standard CPU. (☞ Page 147 How to clear errors)
The following describes how to clear errors of the safety function module.

■Using the engineering tool

Clear errors with the module diagnostics of GX Works3. (GX Works3 Operating Manual)

■Using the buffer memory

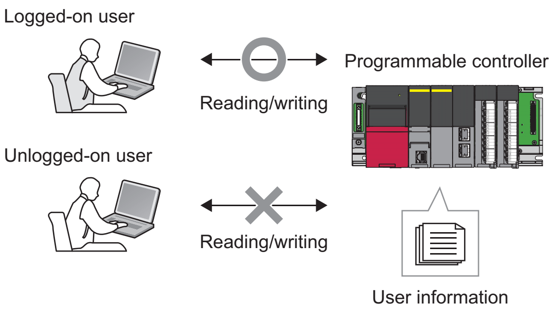
Clear errors using operations of buffer memory.

1. Check the standard program and identify the detected continuation error using Un\G0 (Latest self-diagnostic error code).
2. Eliminate the cause of the detected continuation error.
3. Set 1 in Un\G50 (Error clear) in the standard program to clear the error.*1

*1 After the error is cleared, 0 is automatically set in Un\G50 by the system.

34.13 User Authentication Function of CPU Modules

This function prevents an unauthorized access to a program or parameters written to a CPU module. Operations are restricted by registering a user name/password for a CPU module.
To access a CPU module, logging on with the registered user information (user name/password) is required.



For details on the user authentication function of CPU modules, refer to the following.
GX Works3 Operating Manual

Point

- The user information registered in a CPU module needs to match with the user information of a project.
- For RnSFCPUs (firmware version 27 or later), enhanced measures to reduce vulnerability are taken and partial changes are made to the following functions.
  - · Functions and operations that require user authentication
  - · Writing user information to a CPU module
  - · Logging on to a CPU module

For GX Works3 (version 1.087R or later), enhanced measures to reduce vulnerability are also taken.
Communications with the vulnerability-measures taken are performed by using an RnSFCPU (firmware version 27 or later) and GX Works3 (version 1.087R or later) and enabling the setting to communicate only with GX Works3 with the vulnerability-measures enhanced version.
However, when that setting is enabled, some functions of the intelligent function module that read files, such as the user authentication and recording functions of the GOT sequence program monitor (circuit display), may become unavailable.
To use the GOT sequence program monitor (circuit display) or the intelligent function module that reads out files, disable the above-mentioned setting.
For details on the user authentication function, refer to the following.
GX Works3 Operating Manual

Precautions

■Restoring backup data to a CPU module with the enhanced vulnerability-measures taken

In the following case, the CPU module with the firmware version 27 or later can be logged on from GX Works3 without the enhanced vulnerability-measures (version 1.086Q or earlier) even if the user authentication function is set to communicate only with GX Works3 with the vulnerability-measures enhanced version: when the backed-up data of the CPU module with the firmware version 26 or earlier are restored to the CPU module with the firmware version 27 or later by using the CPU module data backup/restoration function.

35 SAFETY DEVICES, SAFETY LABELS, AND CONSTANTS

This chapter describes the safety devices, safety labels, and constants.

35.1 Safety Devices

A safety device is a device used in safety programs. The safety devices can be used only in the safety programs.

Restriction

- The safety devices cannot be used in the standard programs.
- An index modification and indirect specification cannot be performed in safety programs.

List of safety devices

The following table lists the safety devices.

Classification	Type	Device name	Symbol	Number of points (default)	Changeability with parameter settings	Notation
Safety user devices	Bit	Safety input	SA\X	8K points	Selectable either 8K or 12K points*1	Hexadecimal
	Bit	Safety output	SA\Y	8K points		Hexadecimal
	Bit	Safety internal relay	SA\M	6K points	Changeable (☞ Page 636 Safety device area setting range)	Decimal
	Bit	Safety link relay	SA\B	4K points		Hexadecimal
	Bit/word	Safety timer	SA\T	512 points		Decimal
	Bit/word	Safety retentive timer	SA\ST	0 points		Decimal
	Bit/word	Safety counter	SA\C	512 points		Decimal
	Word	Safety data register	SA\D	12K points		Decimal
	Word	Safety link register	SA\W	4K points		Hexadecimal
Safety system devices	Bit	Safety special relay	SA\SM	4K points	Unchangeable	Decimal
	Word	Safety special register	SA\SD	4K points		Decimal

*1 If selecting 12K points, check the version of the CPU module and engineering tool. (☞ Page 1139 Added and Enhanced Functions)

Point

For details on standard devices used for the Safety CPU, refer to the following.

☞ Page 374 DEVICES

Safety user devices

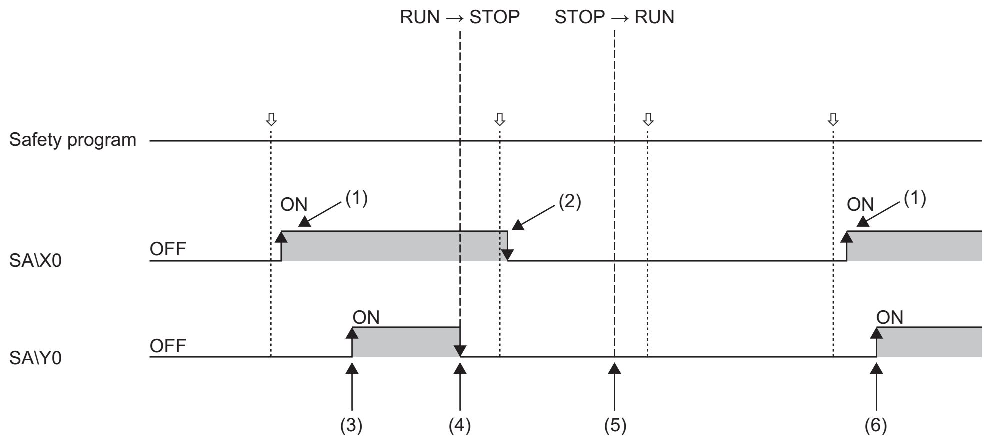
This section describes the safety user devices.

Safety input (SA\X)

Safety inputs are used to give instructions or data to the Safety CPU with external devices such as emergency stop buttons, safety plugs, door switches, and light curtains.

Safety output (SA\Y)

Safety outputs are used to output safety program control results to external relays, contactors, robots, and motion, etc. When the status changes from STOP → RUN, the value immediately before the RUN status is retained, and the value is updated the moment safety cycle processing is executed for the first time after changing to the RUN status.



⇩: Safety cycle processing start
(1) Turns on when safety communications are received.
(2) Turns off when safety communications are received.
(3) SA\Y0 turns on when SA\X0 turns on in the safety program.
(4) SA\Y0 turns off when the status of the CPU module is switched to STOP.
(5) The value is maintained when the status changes from STOP → RUN.
(6) Turns on when a safety program is executed.

Safety internal relay (SA\M)

This device is used as an auxiliary relay within the Safety CPU. The following operations for the Safety CPU turn off all safety internal relays.

- Powering off and on
- Reset

Safety link relay (SA\B)

The safety link relay is used as a device when sending and receiving safety data between Safety CPUs over CC-Link IE Field Network.

■Refreshing network modules using safety link relay

Data is sent and received between Safety CPUs. The safety communications send/receive device range is set in the CC-Link IE Field Network safety communication settings. Locations that are not used for safety communications send/receive devices can be used for other applications.

Safety timer (SA\T)/safety retentive timer (SA\ST)

This device starts measurement when the safety timer coil is turned on. When the current value reaches a setting value, time is up and the contact is turned on. This safety timer is an up-timing type device and therefore the current value matches a setting value when the safety timer time is up. Operations other than the following are the same as those for the timer. (☞ Page 383 Timer)

■Safety timer types

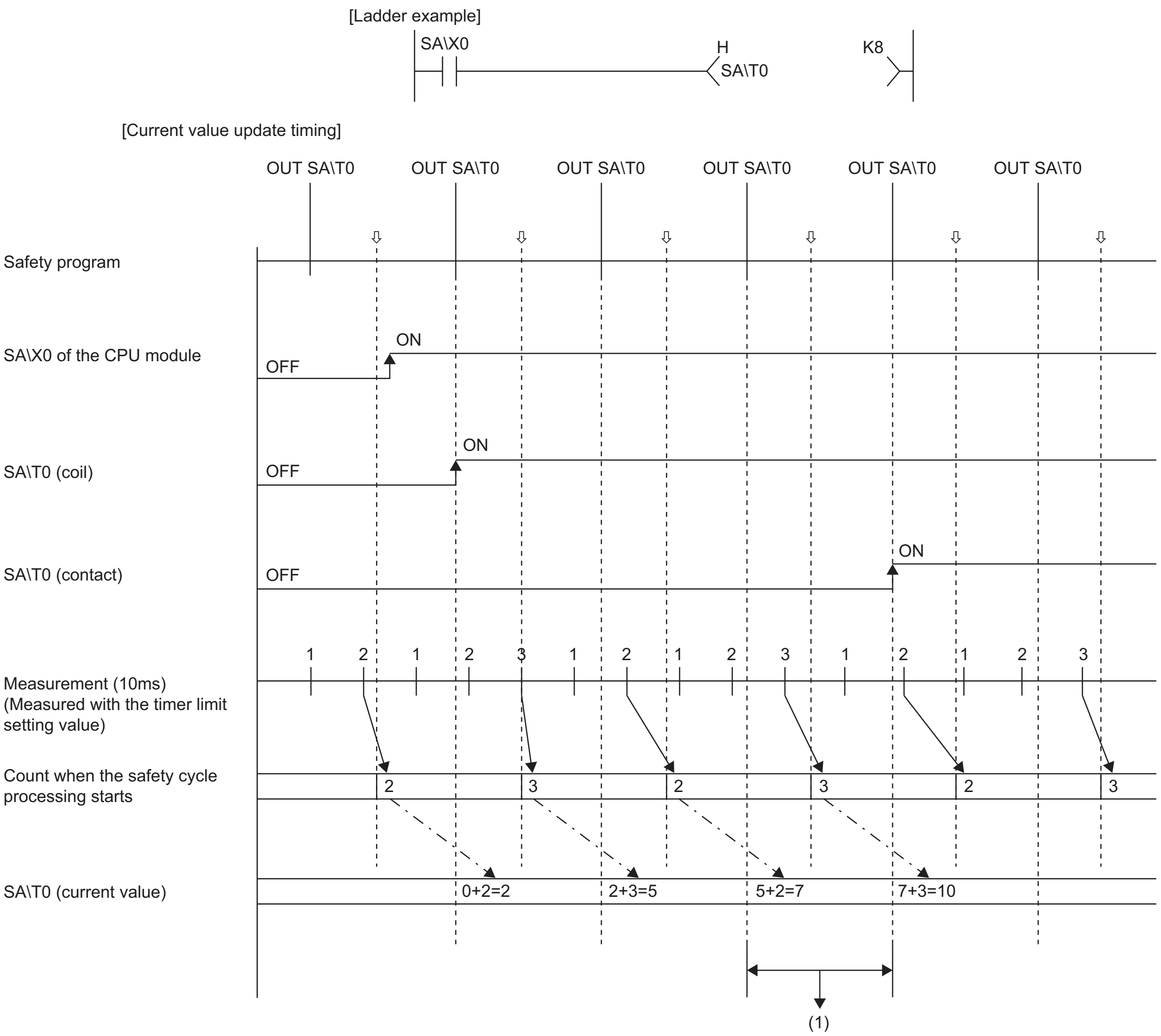
There is a safety timer (SA\T) that retains the current value in 16-bit units. In addition, there is a safety retentive timer (SA\ST) that retains the current value even if the coil turns off.

■Safety timer accuracy

The current value is measured when starting safety cycle processing. The value for the elapsed time since the previous safety cycle processing was started until the present is added to the current value when the OUT SA\T☐ instruction is executed. If the safety timer coil is off at the execution of the OUT SA\T☐ instruction, the current value is not updated. The maximum response accuracy of the timer is the "elapsed time since previous safety cycle processing was started until the present + timer limit setting".

Ex.

Timer limit setting = 10ms, SA\T0 setting value = 8



⇩: Safety cycle processing start

(1) Time accuracy - (Elapsed time since the previous safety cycle processing was started until the present + Timer limit setting) to (Elapsed time since the previous safety cycle processing was started until the present)

(2) Sets the coefficient when starting safety cycle processing.

Safety counter (SA\C)

This device counts the number of rising operation of the input condition in the program. The safety counter is an up-timing type device and therefore when the count value matches a setting value, the count reaches its upper limit and the contact is turned on. Operations other than the following are the same as those for the counter. (☞ Page 391 Counter)

■Safety counter types

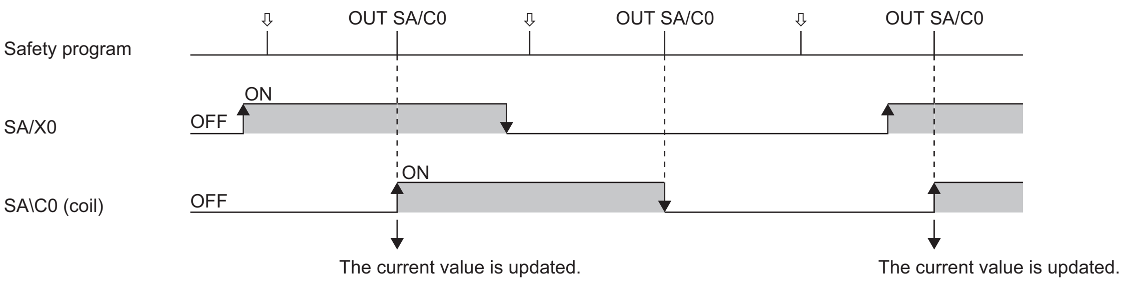
There is a safety counter (SA\C) that retains the counter value in 16-bit units.

■Counting process

When executing the safety counter coil, the safety counter coil is turned on/off, the current value is updated (count value +1), and the contact is turned on/off. The current value is updated (count value +1) when the safety counter coil input is rising (off to on). The current value is not updated when the coil input is off, on to on, and on to off.

[Ladder example]

[Current value update timing]



⇩: Safety cycle processing start

■Resetting the safety counter

The safety counter current value is not cleared even when the counter coil input is turned off. To clear (reset) the safety counter current value and turn off the contact, issue the RST SA\C□ instruction. When the RST SA\C□ instruction is executed, the counter value is cleared and the contact is turned off.

Safety data register (SA\D)

This device can store numerical values.

Safety link register (SA\W)

This register is used as a device when sending and receiving safety data between Safety CPUs over CC-Link IE Field Network.

■Refreshing network modules using safety link register

Data is sent and received between Safety CPUs. The safety communications send/receive device range is set in the CC-Link IE Field Network safety communication settings. Locations that are not used for safety communications send/receive devices can be used for other applications.

Safety system devices

Details of the safety system devices are described below.

Safety special relay (SA\SM)

This relay stores the Safety CPU status relating to safety control. (☞ Page 1023 List of Safety Special Relay Areas)

Safety special register (SA\SD)

This register stores the Safety CPU status relating to safety control. (☞ Page 1026 List of Safety Special Register Areas)

35.2 Safety Global Devices

A safety global device is a device that can be shared by all the safety programs.

35.3 Safety Local Devices

A safety local device is a device used individually by each safety program.

Devices that can be used as a safety local device

The following devices can be used as a safety local device.

- Safety internal relay (SA\M)
- Safety timer (SA\T)
- Safety retentive timer (SA\ST)
- Safety counter (SA\C)
- Safety data register (SA\D)

How to set safety local devices

The range and usability of the safety local devices can be set in the same way for the standard local devices. (☞ Page 417 Setting method for the local devices)

Point

Set safety local devices within the number of device points set in the CPU parameter. The number of safety local device points used is calculated by the following calculation formula. Set the number of safety local device points used so that the number is equal to or less than the capacity of the safety local device area.

Total number of safety local device points used = $((A \div 16) + B + (C \times 2) \div 16)) \times D$

- A: Number of points of the safety local device (SA\M)
- B: Number of points of the safety local devices (SA\D, SA\T (current value), SA\ST (current value), and SA\C (current value))
- C: Number of points of the safety local devices (SA\T (contact/coil), SA\ST (contact/coil), and SA\C (contact/coil))
- D: Number of programs using the safety local device

How to specify safety local devices

To specify safety local devices in a safety program, add "#".

Ex.

SA\#D100, SA\K4#M0

Point

In the program, safety local devices are displayed with "#" in front of the device name. This helps users to distinguish local devices from global devices.

SM777 setting

Regardless of the SM777 (Local device setting in interrupt programs) setting, local devices/safety local devices of a program file in the storage location are always used both in the standard program and safety program.

35.4 Safety Label

A label used in safety programs is called a safety label. Information not described in this section is same as that of standard labels. (☞ Page 420 LABELS)

Safety label types

There are three safety label types. Only the following labels can be used in safety programs.

- Safety global label*1
- Standard/safety shared label*2
- Safety local label

*1 Safety devices can be assigned.

*2 Can be used in standard programs and standard function blocks as well.

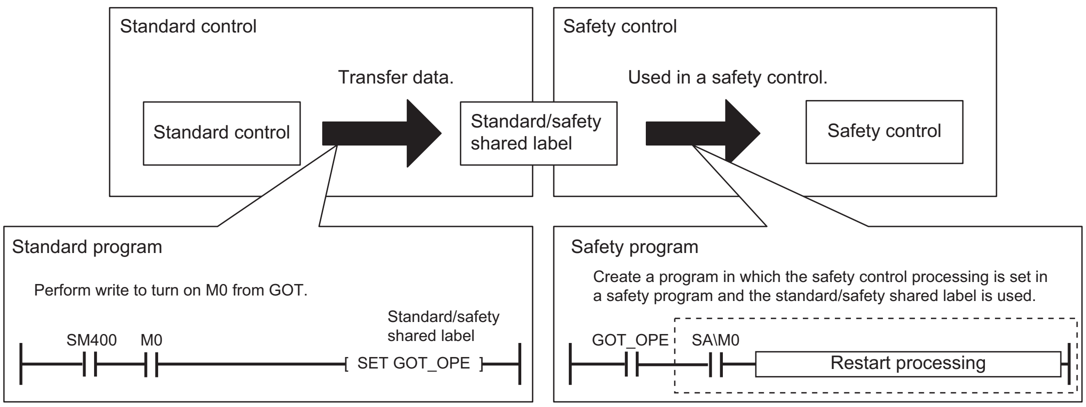
Restriction

- An initial value cannot be set for safety labels and standard/safety shared labels.
- Only safety labels can be used as alias source of a safety label. Only standard/safety shared labels can be used as alias source of a standard/safety shared label. Safety labels and standard/safety shared labels cannot be used as alias source of a standard label. (GX Works3 Operating Manual)

How to use standard/safety shared labels

A standard/safety shared label is used to pass device data from a safety program to a standard program, and vice versa. When a standard/safety shared label is used in a safety program as shown in the examples below, the program needs to be created so that the safety state is secured.

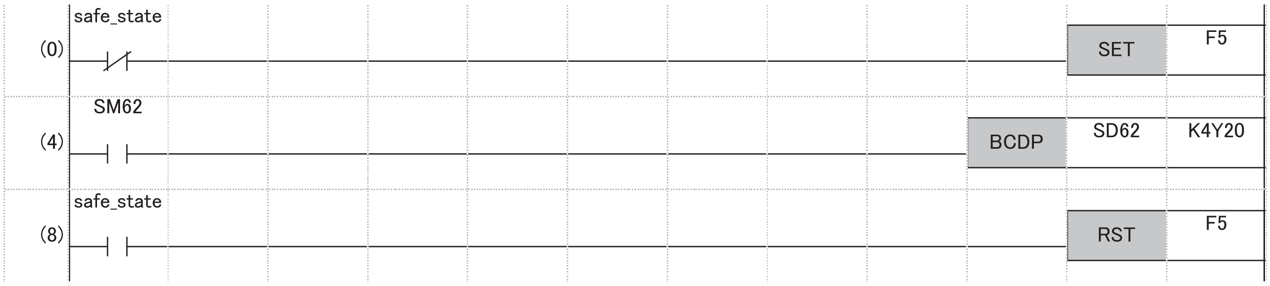
■To restart safety control by the command from the GOT



■To use the annunciator (F)

The safe state signal status can be controlled using the annunciator (F) in the standard program. The safe state signal status is passed from the safety program to the standard program via the standard/safety shared label (safe_state), and the status is controlled with the annunciator No.5. If an error is detected with the annunciator, the corresponding annunciator number is output to Y20.

- Standard program



(0) When the safe state signal turns off, the annunciator No.5 turns on.
(4) The annunciator number detected by SM62 (Annunciator) is output to Y20.
(8) When the safe state signal turns on, the annunciator No.5 turns off.

Classes

The following table lists the availability of the classes of safety global labels and standard/safety shared labels.

○: Applicable, ×: Not applicable

Class	Availability	
	Safety global label	Standard/safety shared label
VAR_GLOBAL	○	○
VAR_GLOBAL_CONSTANT	○	○
VAR_GLOBAL_RETAIN	×	×

The following table lists the availability of the classes of safety local labels.

○: Applicable, ×: Not applicable

Class	Availability		
	Safety program	Safety function	Safety function block
VAR	○	○	○
VAR_CONSTANT	○	○	○
VAR_RETAIN	×	×	×
VAR_INPUT	×	○	○
VAR_OUTPUT	×	○	○
VAR_OUTPUT_RETAIN	×	×	×
VAR_IN_OUT	×	×	○
VAR_PUBLIC	×	×	○
VAR_PUBLIC_RETAIN	×	×	×

Data types

Primitive data type

The following table lists the availability of primitive data types.

○: Applicable, ×: Not applicable

Data type		Availability
Bit	BOOL	○
Word [unsigned]/bit string [16 bits]	WORD	○
Double word [unsigned]/bit string [32 bits]	DWORD	○
Word [signed]	INT	○
Double word [signed]	DINT	○
Single-precision real number	REAL	×
Double-precision real number	LREAL	×
Time	TIME	○
String	STRING	×
String [Unicode]	WSTRING	×
Timer	TIMER	○
Retentive timer	RETENTIVETIMER	○
Long timer	LTIMER	×
Long retentive timer	LRETENTIVETIMER	×
Counter	COUNTER	○
Long counter	LCOUNTER	×
Pointer	POINTER	×

Structures

The structure definition is shared by standard programs and safety programs. However, it cannot be used in the following cases.

- A member of the primitive data type which cannot be used in safety programs exists.
- An initial value is set in the structure definition.

35.5 Constants

The decimal constant (K) and hexadecimal constant (H) can be used in safety programs. The specification method is the same as that for standard programs. (Example: K1234, H1FFF) (☞ Page 450 CONSTANTS)

PART 6 WHEN USING THE SIL2 PROCESS CPU

This part consists of the following chapters. Please read these chapters when using the SIL2 Process CPU. Since information same as that of the standard CPU is not described in these chapters, refer to Part 1 to Part 3.

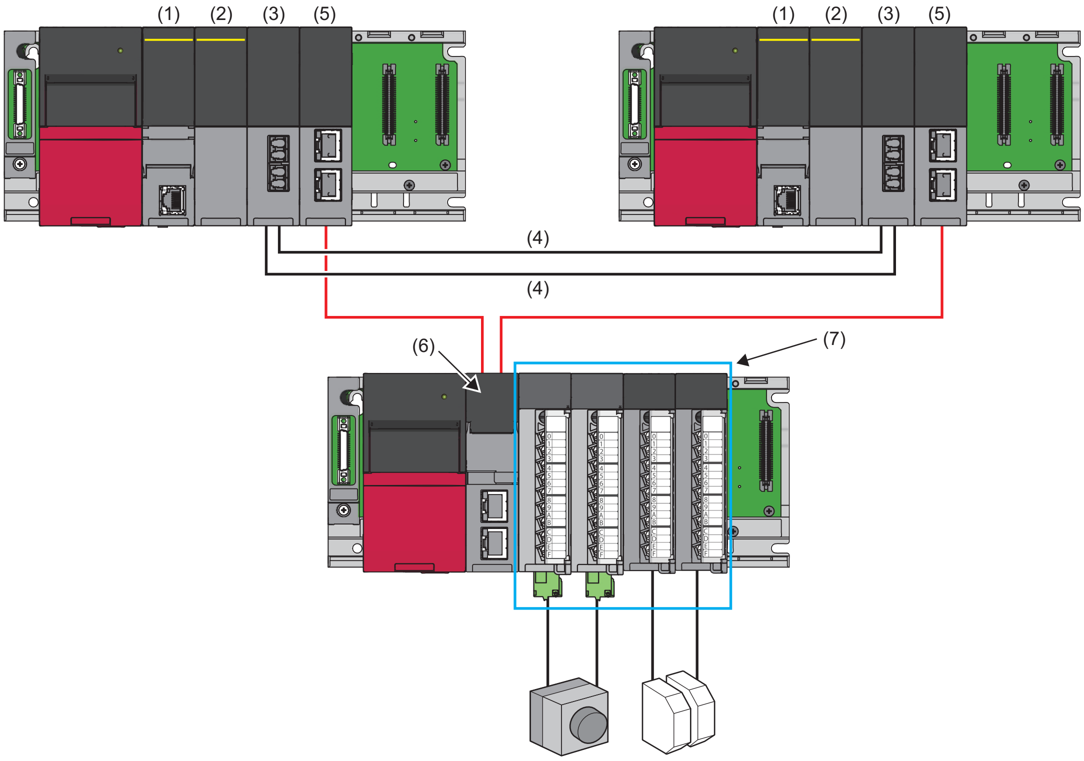
36 BASIC CONCEPT

Use the SIL2 Process CPU with the SIL2 function module as a pair. Programs for safety control and for standard control can be executed simultaneously in one system. These products can be used to construct a safety function for general industrial machinery.
Use the SIL2 Process CPU with the redundant function module as a pair to configure a redundant system.
Mount the modules on the slots from the CPU slot to slot number 1 in the order of the SIL2 Process CPU, SIL2 function module, and redundant function module.

Point

The mounting positions and order are as described above. They cannot be changed.

I/O modules and intelligent function modules set to SIL2 mode are mounted to the intelligent device station (remote head module) on CC-Link IE Field Network.



(1) SIL2 Process CPU
(2) SIL2 function module
(3) Redundant function module
(4) Tracking cable
(5) CC-Link IE Field Network master/local module
(6) Remote head module
(7) Modules set to SIL2 mode*1

*1 For details on modules set to SIL2 mode, refer to the following.
MELSEC iQ-R Module Configuration Manual
Manuals for the I/O module or intelligent function module used

36.1 System

System A and system B

One system is specified as a system A and the other is specified as a system B to distinguish between two systems connected with tracking cables. Set the system A or B with the engineering tool. (☞ Page 700 System A/B Settings)

Point

When one system is set to system A, the other system is automatically set to system B, and vice versa.

36

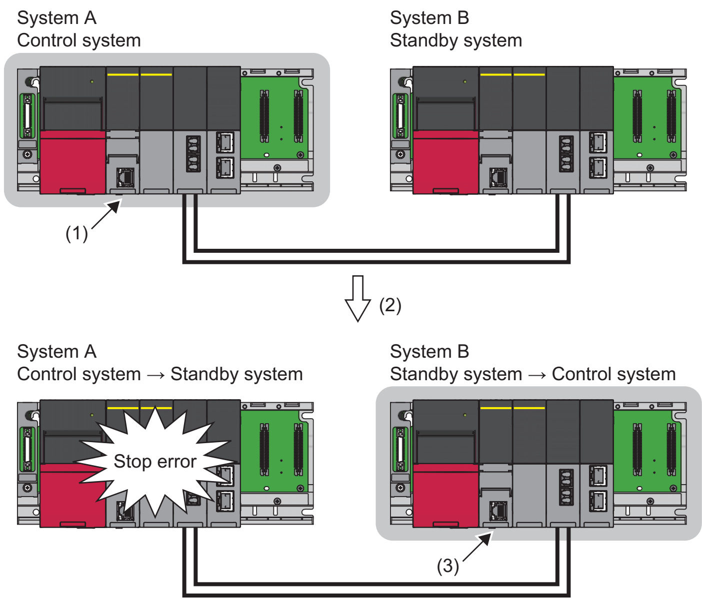
Control system and standby system

The CPU module in one system executes programs to perform controls. The other system is in the standby state and does not perform controls. The system that performs controls is called a control system and the system in the standby state is called a standby system.

A control system or standby system is determined when both the systems are started and ready for tracking communications. (☞ Page 669 Determination of Control System/Standby System)

36.2 System Switching Between the Control System and Standby System

Data link is performed between the redundant function modules connected with tracking cables and data required for operation is transferred (tracking transfer) at every scan from the control system to the standby system. If an error occurs in the control system, the standby system will function as the new control system and continue the control using the data that the system has received.



(1) The control system is operating without errors. (Data is transferred to the standby system from the control system at every scan.)
(2) A stop error occurs on the control system.
(3) The standby system now functions as the new control system and continues the control using the data that the system has received.

For details on the system switching, refer to the following.

☞ Page 708 System Switching

36.3 System Consistency Check

Whether both systems have the same configuration is checked to switch the system and continue the operation without causing a system failure. (☞ Page 742 System Consistency Check)

36.4 Operation Mode of the SIL2 Process CPU

This mode determines the operation of the SIL2 Process CPU and is fixed to redundant mode[*1]. (☞ Page 698 Creating a Project)

*1 Use this mode when a redundant system is built.

36.5 Operation Modes of the System

This mode determines the operation of the system and is fixed to backup mode.

Operation mode	Description
Backup mode	When an error or failure has occurred in the control system, the standby system is switched to the control system to continue the operation. Even if a system failure has occurred in the control system, the standby system continues the operation because data is transferred from the control system to the standby system at every scan.

36.6 Safety Operation Mode

There are two safety operation modes: TEST MODE and SAFETY MODE. Running normally as a safety system or changing a safety program and safety parameters can be selected.

Safety operation mode	Description
SAFETY MODE	Mode for operating the safety system controlled by the SIL2 Process CPU. In this mode, safety programs and safety parameters of the SIL2 Process CPU cannot be changed. Only device data values in safety programs can be changed.
TEST MODE	Mode for performing maintenance (such as setting changes and tests) of the safety system controlled by the SIL2 Process CPU. In this mode, safety programs and safety parameters of the SIL2 Process CPU can be changed. Device data can be changed by performing a device test.
SAFETY MODE (wait-for-restart)[*1]	The system switches to this mode during the period of time from immediately after safety operation mode is changed from TEST MODE into SAFETY MODE until the CPU module is powered off or is reset. Even if safety operation mode is switched to SAFETY MODE, safety operation mode remains in this mode until the CPU module is powered off and on or is reset. Programs are not executed in this mode. An error occurs if the CPU module is in the RUN state.

*1 The same restrictions apply to this mode as those for SAFETY MODE.

Checking the safety operation mode

The safety operation mode can be checked in the following ways.

- TEST LED of the SIL2 function module (📖 MELSEC iQ-R CPU Module User's Manual (Startup))
- "Safety Operation Mode Switch" window of the engineering tool (📖 GX Works3 Operating Manual)
- Module information list of the engineering tool (📖 GX Works3 Operating Manual)
- Safety special register of the SIL2 Process CPU (SA\SD205) (☞ Page 1026 List of Safety Special Register Areas)

Operations restricted in SAFETY MODE

When the safety operation mode is set to SAFETY MODE, the following operations cannot be performed.

Operation		Restrictions
Writing data to the programmable controller		Data cannot be written to files relating to safety control such as safety programs or safety CPU parameters. (☞ Page 686 File operation available)
Device/label test (value changes)		Device/label tests (value changes) cannot be conducted from external devices for the following. (This includes value changes from the test function or monitor windows, and change requests from other external devices.) • Safety global devices • Safety local devices • Safety global labels • Safety local labels • Standard/safety shared labels
Memory operation		Memory operations for the CPU built-in memory (including extended SRAM cassette) cannot be performed from the engineering tool. (☞ Page 684 Memory operation) However, SD memory card initialization is possible.
Security function	Security key authentication function	The following operations cannot be performed for the SIL2 Process CPU or extended SRAM cassette. • Registration • Deletion
	File password function	The following operations cannot be performed for the SIL2 Process CPU. • Registration • Change • Deletion

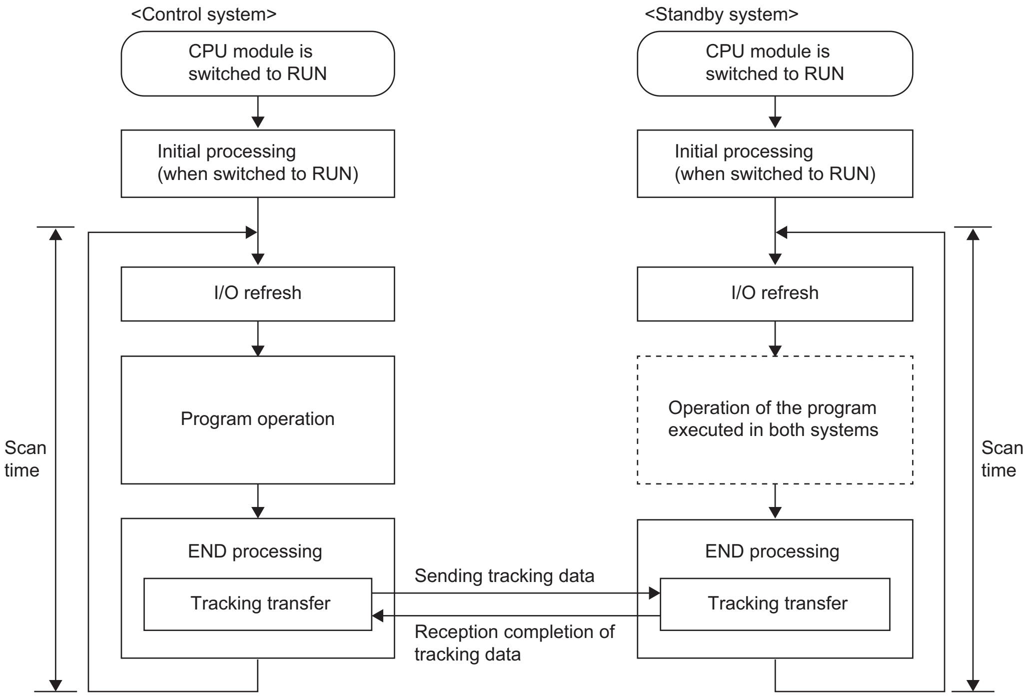
36.7 Scan Configuration

This section describes scan configurations of the CPU modules in a system using the SIL2 Process CPU.

Tracking transfer is performed during the END processing. (☞ Page 721 Tracking Transfer)

For details on the scan configuration other than tracking transfer, refer to Chapter 1. (☞ Page 40 Scan Configuration)

The following figure shows scan configurations of when both systems are started up simultaneously.



Both systems perform different processing.

○: Performed, ×: Not performed

Processing	Control system	Standby system
Initial processing/initial processing (when switched to RUN)	○	○
I/O refresh	○	○*1
Program operation	○	×*2
Tracking transfer	○	○
END processing	○	○

*1 Only the input refresh is performed. Whether or not to perform the output refresh depends on "Standby System Output Setting" of "CPU Parameter". (☞ Page 752 Redundant System Operation Setting)

*2 The program operation is not performed with the default setting. Programs are executed according to "Both Systems Program Executions Setting" in "Program Setting" of "CPU Parameter". (☞ Page 747 Program Execution in Both Systems)

36.8 Determination of Control System/Standby System

This section describes how to determine which system is the control system and the other is the standby system.

When starting up both systems simultaneously

The following describes the method of determining the system types of when both systems are started up simultaneously.

How to determine the system types

Whether each system is the control system or standby system is determined when the both systems are started up by powering off and on or reset of the CPU module and then ready for tracking communications. The safety operation mode (TEST MODE or SAFETY MODE) of the system does not affect the determination of the system types.

■When both systems are started up simultaneously

The system A operates as the control system and the system B as the standby system.
"Simultaneously" here means that one system is started up within three seconds after the other system is started.

■When one of the system A and system B is started up first

The system will wait for the start-up of the other system in three seconds. (☞ Page 671 The system waiting for the start-up of the other system)
When communications for both systems are established, system A operates as the control system and system B as the standby system.

Checking method

Check the LED of each redundant function module to check the status of the control system/standby system.

System status	LED of the redundant function module
Control system	
Standby system	

Point

Users can check the status of the control system/standby system with the engineering tool. (📖 GX Works3 Operating Manual)
- System monitor
- Monitor status bar

Precautions

■If the system A/B settings have not been set for both systems or the same system setting has been set for both systems

A stop error occurs. To determine the control system and standby system, properly set the system A/B settings for both systems.

■If both systems are restarted when a stop error has occurred on the standby system (system A)

If both systems are restarted simultaneously in a system where the system A operates as the standby system due to a stop error and the system B normally operates as the control system, a stop error may occur on both systems.
If the stop error cause of the system A is a program error, a stop error occurs on the system A again after the systems are restarted. Even if the system B normally operates as the control system before the restart, a stop error also occurs on the system B if a mismatch between files is detected in the system consistency check after the restart. As a result, a stop error occurs on both systems.
In this case, eliminate the cause of the error that has occurred on the system A and restart and restore both systems.

■When the READY LED of the CPU module in one of the systems is flashing

Do not power off the CPU module in the other system.

The system may start up without checking the system consistency even when the conditions between the both systems do not match. (☞ Page 743 Execution timing)

■When the configurations between the both systems do not match

When the configurations (such as insertion of an SD memory card) between the both systems do not match, it may take time to start communications between the both systems.

Check that the both system configurations match in the process of starting up both systems at a time, and then start up a system. (☞ Page 689 PROCEDURE FOR STARTING UP A SYSTEM USING THE SIL2 PROCESS CPU)

■Using an SD memory card

Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card. A stop error may occur on both systems and the systems may not be started as a system.

If the CPU module is powered off or reset or an SD memory card is removed during an access to the SD memory card, the data in the SD memory card may be corrupted. In this case, diagnostics of the SD memory card (such as a file system check or restoration processing) is performed when the CPU module is powered off and on or is reset.

During diagnostics of the SD memory card, tracking communications are disabled. If diagnostics of the SD memory card on the other system are not completed in the timeout period set in "Other system Start-up Timeout Setting", a stop error occurs on the own system. If diagnostics of the SD memory card takes time on the other system, a stop error occurs on the other system as well. In this case, a stop error occurs on both systems and the systems cannot be started as a redundant system. Restart and restore both systems.

When starting up one system first

One of the system A and system B can be started up first.

How to start one system first

Perform one of the following operations to the CPU module that is waiting for the start-up of the other system. The system of the CPU module will be started as the control system.

■Online operation

Perform the following operation on the engineering tool.

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

Select "Forced Start of Control System while Waiting for Other System to Start" and click the [Execute] button.

■Switch operation

Set the RUN/STOP/RESET switch of the CPU module to the RUN, STOP, and then the RUN position again.

Set "Control System Start-up Setting (Switch Operation)" to "Enable" in the CPU parameter in advance. (☞ Page 752 Redundant System Operation Setting)

■Operation with the input (X)

Turn on the input (X) set with the parameters.

Set "Control System Start-up Setting (Input (X))" to "Enable" in the CPU parameter in advance. (☞ Page 752 Redundant System Operation Setting)

The system waiting for the start-up of the other system

When the CPU module is started up and tracking communications cannot be established with the other system, the CPU module will start waiting for the start-up of the other system in three seconds.
Both of the CTRL LED and SBY LED of the redundant function module turn off because the system of the CPU module is not the control system or standby system yet. The BACKUP LED flashes because a system switching disable cause has been generated.
After tracking communications are established with the other system, the control and standby systems are determined and the system switching disable cause is eliminated.

■Operation while the system is waiting for the start-up of the other system

Even though the RUN/STOP/RESET switch of a CPU module is set to RUN while the system is waiting for the start-up of the other system, the CPU module is in the STOP state because the system has not yet been set to control system or standby system. Thus, the CPU module does not execute programs.
The following table lists the refresh operations to be performed while the system is waiting for the start-up of the other system.

Type	Operation
I/O refresh	Only the input refresh is performed. The output refresh is not performed.
Network module link refresh	Data in SB/SW is transferred and data in devices other than SB/SW is not transferred from the network module to the CPU module, and from the CPU module to the network module.
Intelligent function module refresh	Data is transferred between the intelligent function module and the CPU module.

■Timeout of waiting for the start-up of the other system

Measuring the length of the waiting time for the start-up of the other system is started upon completion of the initial processing.
Set the waiting time for the start-up of the other system in "Other system Start-up Timeout Setting" of "CPU Parameter". (☞ Page 752 Redundant System Operation Setting)
When a timeout occurs, a stop error will occur. To prevent occurrence of a stop error, set "Other system Start-up Timeout Setting" of "CPU Parameter" to "Not Set".

Precautions

- To start up the system that is waiting for the start-up of the other system as the control system, check that the other system is not operating as the control system.
- When tracking communications are established by connecting the other system with a tracking cable or powering on or resetting the CPU module in the other system, the CPU module that has been waiting for the start-up of the other system enters the RUN state and executes programs. Thus, always check that program execution by the CPU module does not cause any problems before powering on the other system or connecting the other system with a tracking cable.
- When the CPU module that has been waiting for the start-up of the other system becomes ready for tracking communications and a stop error has occurred in the other system, a stop error occurs in that CPU module as well. In this case, take a corrective action against the error of the other system, and turn off and on or reset the CPU module in both systems.
- When the system waits for the start-up of the other system, tracking communications cannot be made because the other system is off or either of tracking cables has problems. Check that the other system is on or tracking cables have no problems. (☞ Page 925 When the L ERR LED turns on)
- Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card. (☞ Page 670 Using an SD memory card)

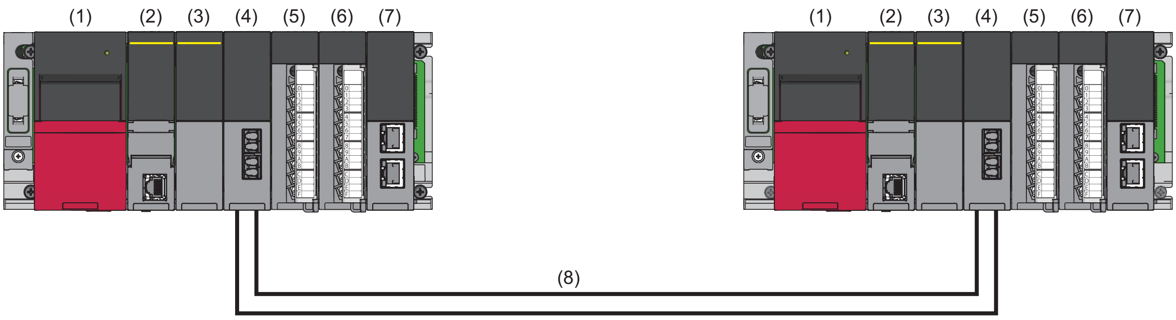
When one system is started automatically even though a tracking communication error has occurred

When the other system is powered off[*1] or there is an error with tracking cables at a system start, the CPU module enters the state that waits for the other system to start. The following shows examples, such as a system configuration and a program, to start up either of two systems using external signals without waiting for the other system to start, and prevent both systems from operating as control systems.

*1 Instead of configuring the system described below, configuring a redundant power supply system is recommended when taking a measure against failure of the power supply module in the other system.

System configuration

The following figure shows a configuration example of the system.

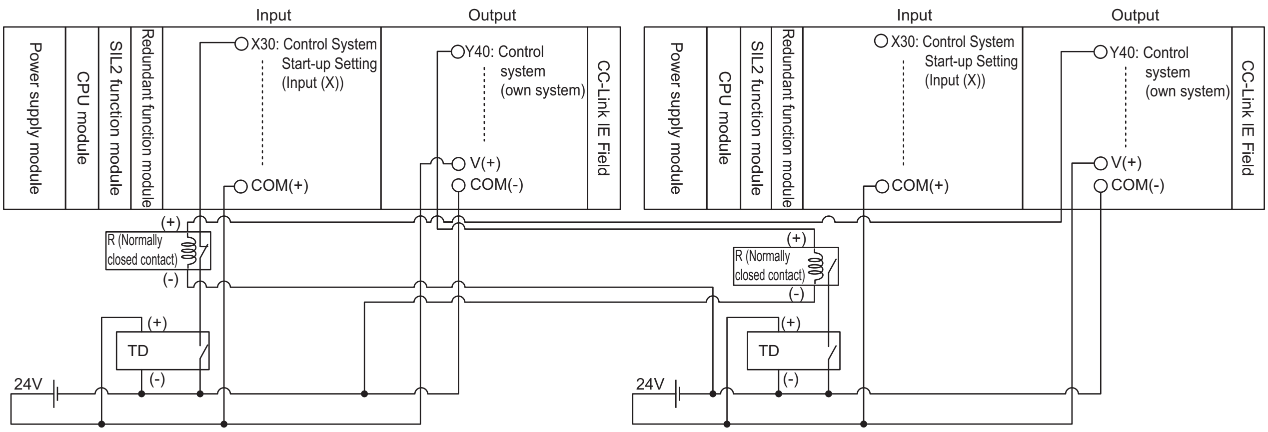


(1) Power supply module (R61P)
(2) CPU module (RnPSFCPU)
(3) SIL2 function module (R6PSFM)
(4) Redundant function module (R6RFM)
(5) Input module (RX40C7)
(6) Output module (RY40NT5P)
(7) Tracking cable
(8) CC-Link IE Field Network master/local module (RJ71GF11-T2)[*1]

*1 Starting up one system is allowed without the CC-Link IE Field Network master/local module.

Wiring example

The following figure shows a wiring example.



- Supply the 24V power using the 24V external power supply. Use the same power source as the one that supplies power to the power supply module in each system.
- TD is an on delay timer wired externally. Connect the output signal wire of the on delay timer to a relay (normally closed contact). To prevent both systems from starting up simultaneously, configure different timer settings for system A and system B.
- R is a relay (normally closed contact) wired externally. This relay connects the output signal wire of the on delay timer and the output device (Y40: Control system (own system)). The output signal wire of the relay is input to X30.

■I/O signals

The following table lists the details on the I/O signals.

Device No.	Signal name
X30	Control System Start-up Setting (Input (X)). With the timer wired externally, this bit turns on after a certain time. When the output Y of the other system is off (control system (own system)) at that time, the system starts as the control system.
Y40	Control system (own system)

■Setting time of the external on delay timer

For the external timer, with the following equation as a guide, set a longer time than the time until both systems start up so that this function (Automatic start-up at tracking communication error) is not executed when tracking communications are normally performed. In addition, set different times to system A and system B so that the times of both systems are not up simultaneously.

- Time set for the external timer*1 = ❶ Start-up time of the CPU module (Time from power-on to RUN) + ❷ One scan time + ❸ Time lag of power-on + ❹ α + ❺ β

❶ Start-up time of the CPU module: Time from when the CPU module is powered on until when the CPU module enters to the RUN state
❷ One scan time: Time until when Y40 (Control system (own system)) is refreshed
❸ Time lag of power-on: Time to add to the external timer of the system that has started up first when two systems start up one by one. It adjusts the activation timing of the external timer.
❹ α: Margin for variation in start-up time of the CPU module. Set a sufficient margin to accommodate the variation.
❺ β: Time to add to the timer of either of two systems so that the times of both systems are not up simultaneously

*1 If the time set for the external timer is shorter than the time determined by the above equation or an identical time is set to system A and system B, one system cannot recognize whether the other system has started up as the control system and both systems may start up as control systems.

Parameter settings

The following shows parameter settings.

■System parameter

Set the system parameter according to the system configuration in "I/O Assignment Setting".

Slot	Module Name	Module Status Setting	Points	Start XY
Base				
CPU	R32PSFCPU(Host Station)			3E00
0(*-0)	R6PSFM	No Setting	16 Points	0000
1(*-1)	R6RFM	No Setting	32 Points	0010
2(*-2)	RX40C7	No Setting	16 Points	0030
3(*-3)	RY40NT5P	No Setting	16 Points	0040

■CPU parameter (program settings)

Set this program example (MAIN in this example) in "Program Setting" as follows.

Execute Order	Program Name	Execution Type		Refresh Group Setting	Device/File Use or not	Both Systems Program Executions Setting
		Type	Detailed Setting Information			
1	MAIN	Scan		(Do not Set)	<Detailed Setting>	Both Systems Executions

- Set "Execution Type" to "Scan".
- Set "Both Systems Program Executions Setting" to "Both Systems Executions".

■CPU parameter (redundant settings)

Set the CPU parameter in "Redundant System Settings" as follows.

Item	Setting	
Redundant Behavior Setting		
Watching Standby System Setting	Enable	
Standby System Output Setting	Disable	
Backup Mode Setting		
Operating Status Check	Check	
Safety Operation Mode Check	Check	
Auto Memory Copy Setting	Disable	
Control/Standby System Start-up Setting		
Other system Start-up Timeout Setting	Not Set	(1)
Time-out Period	60 Second	
Control System Start-up Setting(Switch Operation)	Disable	
Control System Start-up Setting(Input(X))	Enable	(2)
Input(X)	X30	(3)
Tracking Setting		
Signal Flow Memory Tracking Setting	Transfer	
Tracking Device/Label Setting	Detailed setting	(4)
Tracking Block No.1 Auto Transfer Setting	Transfer Automatically	
Device/Label Detailed Setting	<Detailed Setting>	

(1) Set "Not Set" in "Other system Start-up Timeout Setting".
(2) Set "Enable" in "Control System Start-up Setting (Input (X))".
(3) Set "X30" in "Input (X)".
(4) Set "Detailed setting" in "Tracking Device/Label Setting".

Set "Global Device Setting" in "Device/Label Detailed Setting" as follows.

Tracking Block No. 1
Total Global Device 1.5K Word
File Register Setting
File Name

No.	Device	Points (Decimal)	Start	End	
1	X	12224	40	2FFF	(5)
2	Y	12208	50	2FFF	(5)

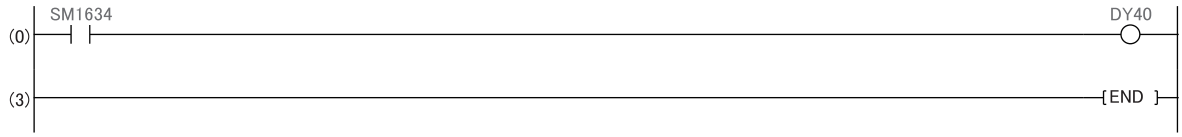
(5) Do not include X30 to X3F and Y40 to Y4F, which are used in the program example, in the tracking transfer range.

Point

- To switch enabling and disabling "Control System Start-up Setting (Input (X))", set a switch for the input to X30 so that "Control System Start-up Setting (Input (X))" is enabled only when the switch is on.
- When using this system, set the RUN/STOP/RESET switch to RUN to operate the ladder program.

Program example

The following figure shows a program example and the overview of the operation.



■Output of the control system (own system)

(0) The other system is notified of the start-up of the own system as the control system by turning on Y40 (Control system (own system)) using the direct access output when the own system operates as the control system (SM1634 is on), or by turning off Y40 when the own system does not operate as the control system.

Point

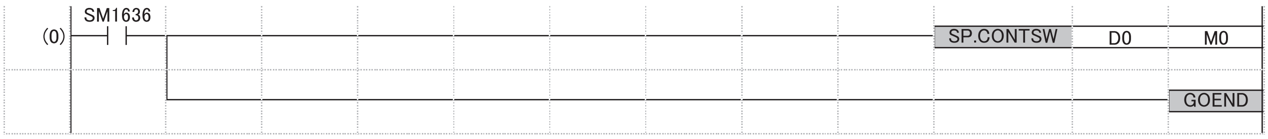
After the automatic start-up, to enable the system switching, eliminate the cause of tracking communication error and restart the CPU module in the control system or in the state that waits for the other system to start.

When starting up the previous control system as the control system

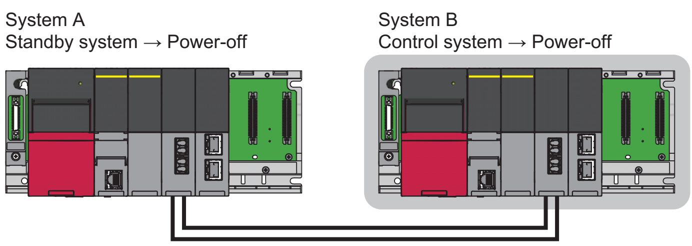
The system A is always specified as the control system when both systems are started up simultaneously. Even though both systems are temporarily powered off due to a power failure or other causes while the system B is operating as the control system, the system A is started up as the control system when both systems are powered on again.
To start up the system B that was the previous control system as the control system again, create a program that uses SM1636 (Previous control system identification flag). For a system where a network module has been mounted, execute the SP.CONTSW instruction after a network module on the other system is started up.

Program example

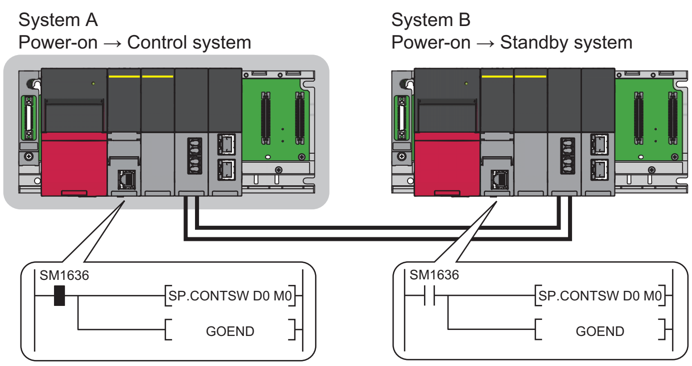
Executing a system switching instruction when the CPU module in the system A is set to the RUN state for the first time switches the system B to the control system. Turn on SM1646 (System switching by a user) in advance.



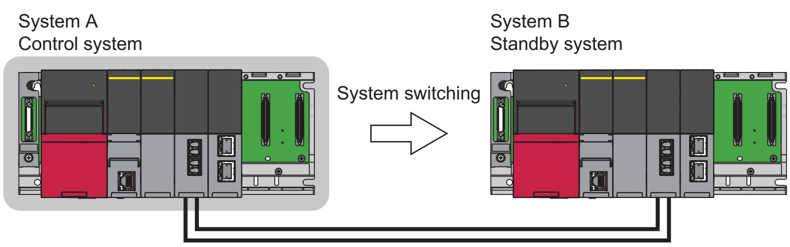
Operation example



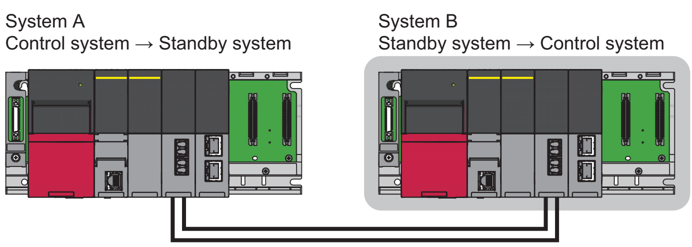
1. Both systems are temporarily powered off due to a power failure or other causes while the system B is operating as the control system.



2. The system A is started as the control system when both systems are powered on. SM1636 turns on for only one scan after the CPU module in the system A is set to the RUN state.



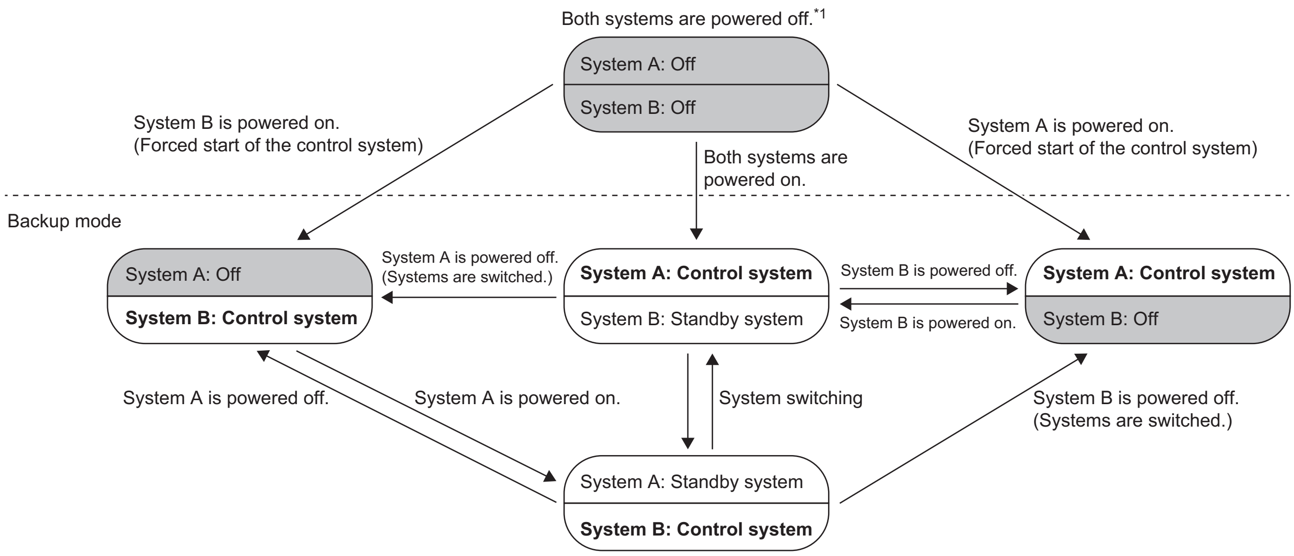
3. The system switching is performed by the SP.CONTSW instruction.



4. The system B is switched from a standby system to a control system.

36.9 State Transition of a System Using the SIL2 Process CPU

The following figure shows the state transition of the system after start-up.



*1 When both systems are powered off, both systems enter the power-off state regardless of the system type.

36.10 Running a Safety Program

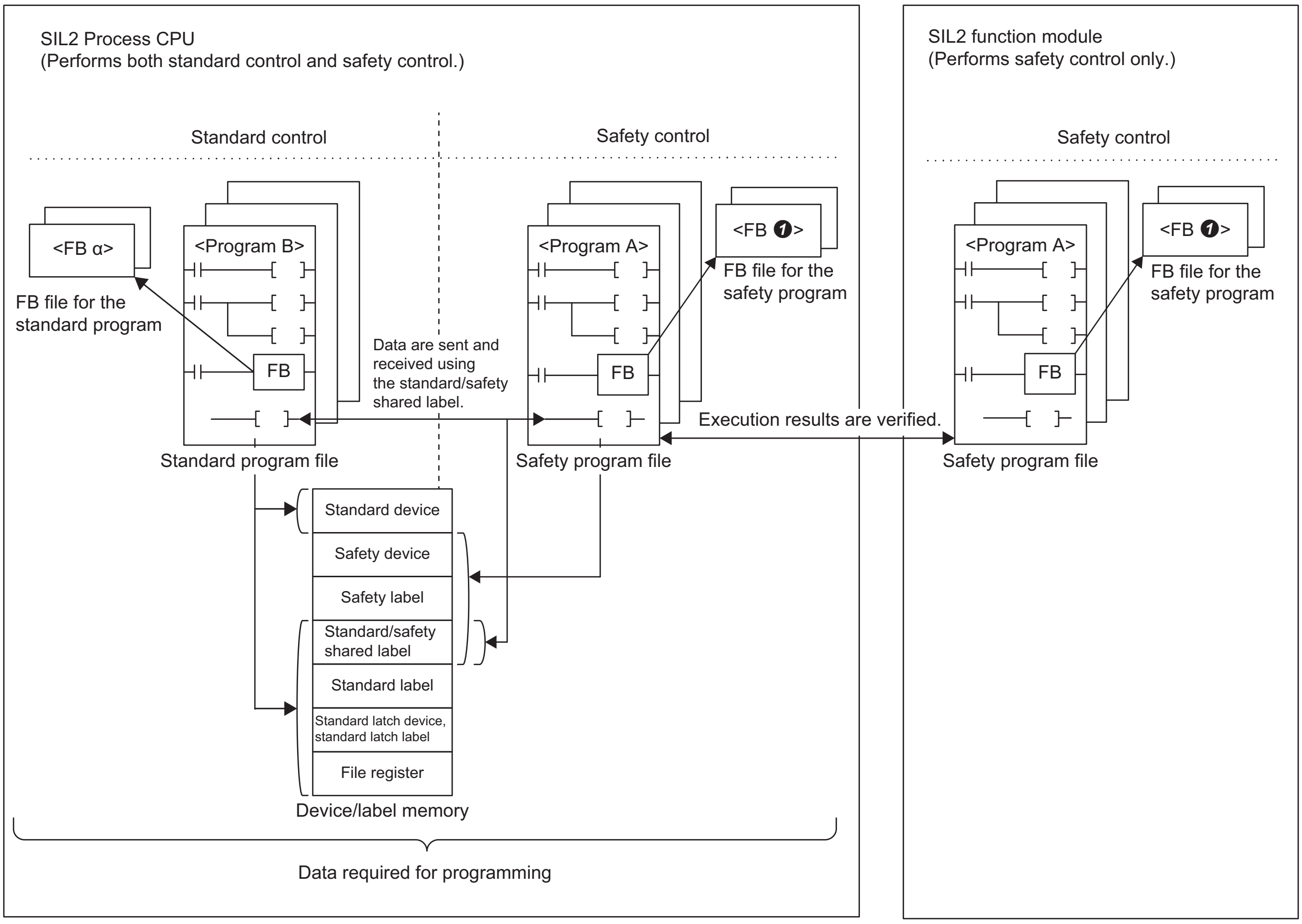
This section describes how to execute a safety program. Information not described in this chapter is same as that of the standard CPU. (☞ Page 40 RUNNING A PROGRAM to ☞ Page 100 MEMORY CONFIGURATION OF THE CPU MODULE)

Point

For details on the setting method (registration procedure) of safety programs, refer to the following.
📖 GX Works3 Operating Manual

Overview

The SIL2 Process CPU executes standard programs and safety programs, and the SIL2 function module executes safety programs only. During operations, the SIL2 Process CPU verifies the execution results of safety programs with those of the SIL2 function module.



Devices/labels that can be used in safety programs

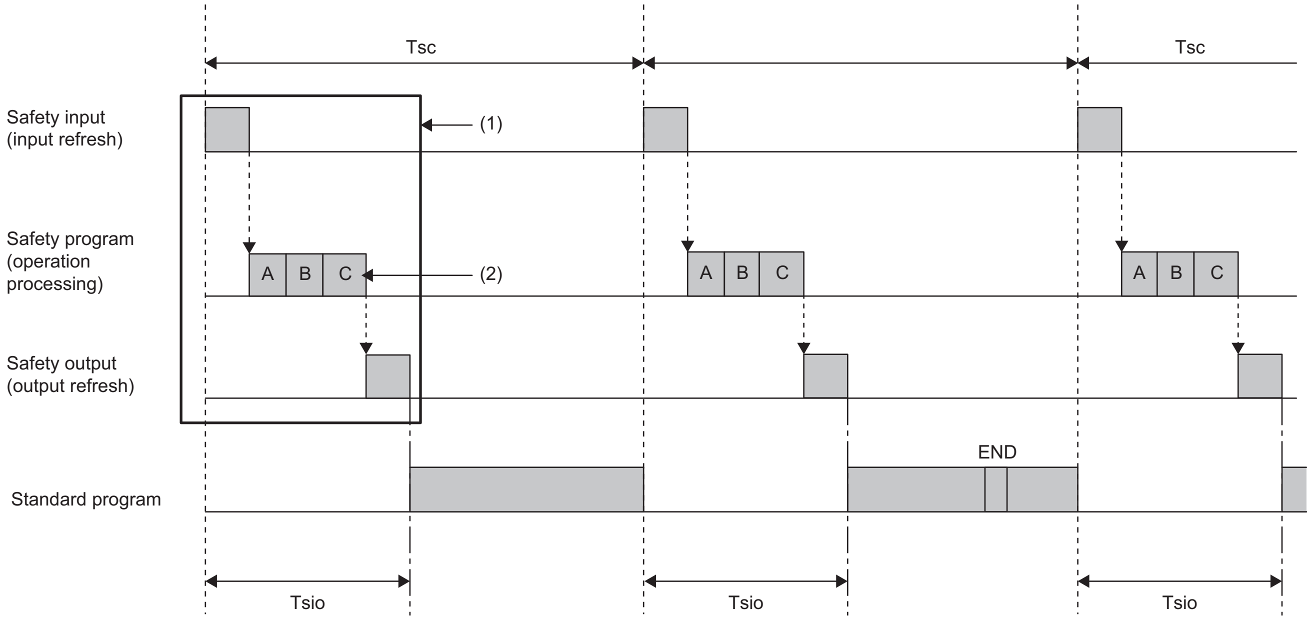
Only safety devices and safety labels can be used in safety programs.

- Safety device: ☞ Page 654 Safety Devices
- Safety Label: ☞ Page 783 Safety Label

Safety program

Safety programs are executed at every safety cycle. Safety cycle processing is performed in the order of safety input (refresh) processing → safety program → safety output (refresh) processing. (☞ Page 681 Safety cycle time)

Standard programs (+ END processing) are executed within the remaining time of the safety cycle time after safety programs are executed. (Standard programs are executed until next safety cycle time starts.)



Tsc: Safety cycle time

Tsio: Safety program + Safety input/output processing time

(1): Store the processing times below.

- SIL2 Process CPU: SD1890, SD1891
- SIL2 function module: Un\G62, Un\G63

(2): Execute all safety programs. (Execute safety programs A, B, and C.)

Point

- After executing a rising/falling instruction, the instruction will be valid until the safety program for the next safety cycle processing is executed. (📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))
- In safety programs, the block number of the file register (R) is not saved or restored. Data in the index register (Z, LZ) are not saved or restored, either.
- Even if the operating status of SIL2 Process CPUs for which no safety program has been registered is STOP or PAUSE, safety program operations other than operation processing are performed. Consequently, the standard control processing time and device/label access service processing time will be longer.

Safety program execution type

The following table lists the standard program and safety program execution types.

Program	Execution type
Standard program	• Initial • Scan • Fixed scan • Event • Standby
Safety program	Fixed scan

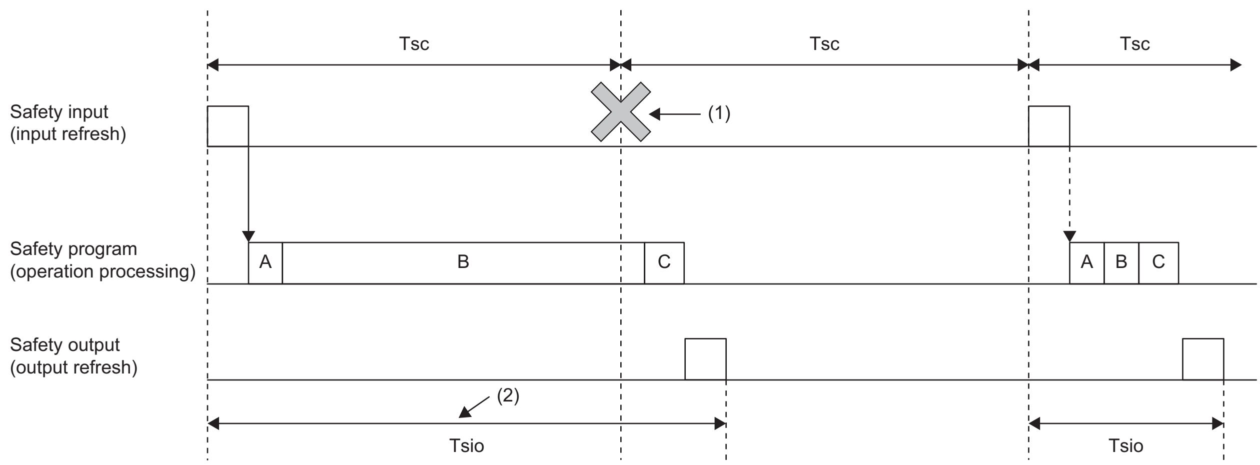
Safety programs are executed as a fixed scan execution type program. However, safety programs perform safety control, and therefore operation differs from standard fixed scan execution type programs in the following ways.

- They are unaffected by disabling interrupt instructions (EI instruction, DI instruction, and IMASK instruction), and are executed based on the safety cycle time.
- Of all interrupt programs that can be created by customers, the execution of safety programs is given highest priority.
- Safety cycle processing including safety programs prioritizes fixed periodicity. If the safety cycle time is exceeded, and safety cycle processing is executed, processing is not performed in succession, but based on the interrupt timing of the next cycle time.

When the program does not complete within the safety cycle time

Whether execution of the safety program is completed within the safety cycle time is monitored, and if the program execution time exceeds the next safety cycle time, an error occurs at the SIL2 Process CPU and SIL2 function module. Note that if the next safety cycle time is exceeded, the safety input → safety program is not executed following program completion, a safety cycle processing error (error code: 1A01H) is detected at the SIL2 Process CPU, and the safety program is executed at the next safety cycle time. Also, if execution of the program is completed after the safety cycle time has been exceeded, the safety cycle processing time execution cycle error flag turns on, and the safety cycle processing time execution cycle error count is incremented by one as follows.

Item	Safety cycle processing time execution cycle error flag	Safety cycle processing time execution cycle error count	Cause of increments of safety cycle processing time execution cycle error count
SIL2 Process CPU	SM1888	SD1888	Occurrence of an error code: 1A00H or 1A01H
SIL2 function module	—	Un\G54	Occurrence of an error code: 1A00H



Tsc: Safety cycle time

Tsio: Safety program + Safety input/output processing time

(1): If safety input/output and safety program execution are not completed within the safety cycle time, interrupts during this period are ignored. Execute safety input/output and safety programs from the next interrupt.

(2): If the time from safety input to safety output exceeds the safety cycle time, a continuation error will occur after safety processing is complete.

- SIL2 Process CPU: SM1888 turns ON, and the value in SD1888 is incremented by one.
- SIL2 function module: The value in Un\G54 is incremented by one.

Safety cycle time

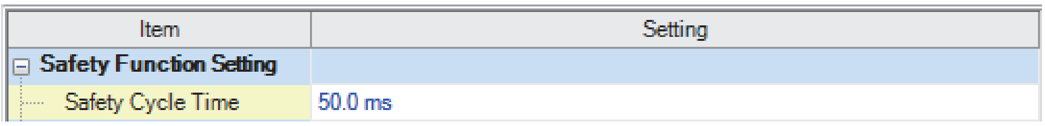
A safety cycle time is a timing for executing safety programs and performing safety input/output processing.

Setting method

Set a safety cycle time in the CPU parameter.

[CPU parameter] ⇨ [Safety Function Setting]

Window



Displayed items

Item	Description	Setting range	Default
Safety Cycle Time	Sets a timing (safety cycle time) for executing safety programs and safety input/output processing.	10.0 to 1000.0ms (unit: 0.1ms)	50.0ms

Point

If an instruction with long processing time is executed in the standard program, the start of safety cycle processing is delayed. To prioritize safety cycle processing, set "Enable" for the enabling interrupt setting during command execution. (☞ Page 85 Interrupt enabled during instruction execution)

SIL2 Process CPU operating status

The following is a list of SIL2 Process CPU operating statuses.

- RUN state
- STOP state
- PAUSE state

Operation processing based on SIL2 Process CPU operating status

Operation processing based on the SIL2 Process CPU operating status is the same as that of the standard CPU. (☞ Page 95 Operation Processing by Operating Status)

■Safety communications processing based on SIL2 Process CPU operating status

The following table shows safety communications processing based on the SIL2 Process CPU operating status.

SIL2 Process CPU operating status	Safety communications processing
RUN state	Executes the set safety communications.
STOP state	Executes the set safety communications, however, send data is turned off (0).[*1*2]
PAUSE state	Executes the set safety communications, however, send data is turned off (0).[*1]

*1 If the safety operation mode is TEST MODE, output is based on the output hold and clear settings when the status of network modules (RJ71GF11-T2) for which safety communication settings are specified is CPU STOP. (📖 Manual for network module used)

*2 Safety communications are not performed if a stop error occurs at the SIL2 Process CPU.

■Operation processing when operating status of the SIL2 Process CPU changes

The following table shows operation processing in the safety program when the operating status of the SIL2 Process CPU is changed.

SIL2 Process CPU operating status	SIL2 Process CPU processing			
	Safety program	External output	Safety device data[*1]	
			Other than SA\Y	SA\Y
STOP → RUN	Executes safety programs.	Outputs the value after safety program execution.	Retains the safety device data status immediately before the RUN state. Clears safety local devices.	Retains the SA\Y value immediately before the RUN state.
RUN → STOP	Executes up to the END instruction and then stops.	Turns off all safety communication outputs.[*3]	Retains the device data status immediately before the STOP state.	Turns off all SA\Y values.[*2]
RUN → PAUSE	Stops operation after the execution of one scan.	Turns off all safety communication outputs.[*3]	Retains the device memory status immediately before the PAUSE state.	Turns off all SA\Y values.[*2]
PAUSE → RUN	Executes the program from the start.	Outputs the value after safety program execution.	Retains the device data status immediately before the RUN state. Clears safety local devices.[*2]	Retains the device data status immediately before the RUN state.
PAUSE → STOP	Operation remains stopped.	All safety communication outputs remain off.[*3]	Retains the device data status immediately before the STOP state.	Retains the SA\Y value immediately before the STOP state.
STOP → PAUSE	Operation remains stopped.	All safety communication outputs remain off.[*3]	Retains the device data status immediately before the PAUSE state.	Retains the SA\Y value immediately before the PAUSE state.

*1 Assignment device data for standard/safety shared labels is processed in the same way as operation processing with standard programs.

*2 Values for safety devices when the SIL2 Process CPU operating status changes are cleared or turn off, regardless of the safety operation mode.

*3 If the safety operation mode is TEST MODE, output is based on the output hold and clear settings when the status of network modules (RJ71GF11-T2) for which safety communication settings are specified is CPU STOP. (📖 Manual for network module used)

Operation processing with standard programs is the same as the following.

☞ Page 96 Operation Processing When Operating Status Is Changed

36.11 Memory Specifications

This section describes the memory specifications of the SIL2 Process CPU. Information not described in this chapter is same as that of the standard CPU. (☞ Page 100 MEMORY CONFIGURATION OF THE CPU MODULE)

Memory configuration

Specifications of the device/label memory differ from the standard CPU in the memory configuration of the SIL2 Process CPU.

Device/label memory

In addition to standard devices and standard labels, the data such as safety devices and safety labels are allocated to each data area of the device/label memory.
The safety device area, safety label area, safety local device area, and standard/safety shared label area are allocated between the standard device area and standard label area. The capacities for each area can be changed in the device/label memory area setting by using the engineering tool. (☞ Page 767 Device/Label Memory Area Setting)

<table>
<tr><td colspan="2">Standard device area</td></tr>
<tr><td rowspan="3">Safety device/label area</td><td>Safety device area</td></tr>
<tr><td>Safety label area</td></tr>
<tr><td>Safety local device area</td></tr>
<tr><td colspan="2">Standard/safety shared label area</td></tr>
<tr><td rowspan="2">Standard label area</td><td>Standard label area</td></tr>
<tr><td>Standard latch label area</td></tr>
<tr><td colspan="2">Standard local device area</td></tr>
<tr><td colspan="2">File storage area</td></tr>
</table>

■Data to be allocated

The following table lists the data allocated to each area.

Area		Application
Standard device area		Standard user devices
Safety device/label area	Safety device area	Safety user devices
	Safety label area	Safety global labels and safety local labels
	Safety local device area	Safety local devices
Standard/safety shared label area		Standard/safety shared labels
Standard label area	Standard label area	Standard global labels and standard local labels
	Standard latch label area	Standard global labels and standard local labels with latch specified
Standard local device area		Standard local devices (excluding index register)
File storage area		File register files and other data

Point

Safety devices and safety labels cannot be latched, and therefore there is no latch area for safety devices and safety labels.

File size unit in memory

The following table lists the unit of the file size (cluster size) of the SIL2 Process CPU memory.

SIL2 Process CPU	File size unit		
	Program memory	Device/label memory	Data memory
R08PSFCPU	128 bytes	512 bytes	2048 bytes
R16PSFCPU			4096 bytes
R32PSFCPU			8192 bytes
R120PSFCPU			16384 bytes

Point

Data is written in the unit of the file size (cluster size). For example, when 464 bytes of CPU parameter is written to the data memory on the R08PSFCPU, it is written as 2048 bytes of data because the file size unit of the data memory is 2048 bytes.

Memory operation

Using an engineering tool, each memory can be initialized and cleared to zero. For details on the operation method, refer to the following.

📖 GX Works3 Operating Manual

Items to be specified in the engineering tool				Target
Initialization	Data memory			Deletes all the folders and files in the program memory and data memory.
	Device/label memory			Deletes all the files in the file storage areas in the device/label memory.
	SD memory card			Deletes all the folders and files in the SD memory card.
Value clear	Device, label	Zero clear		Excluding devices and labels with latch specified, clears the following to zero: X, Y, M, B, F, SB, V, T, ST, LT, LST, C, LC, D, W, SW, FX, FY, FD, Z, LZ, RD, SA\X, SA\Y, SA\M, SA\B, SA\T, SA\ST, SA\C, SA\D, SA\W, and all labels (including module labels).
		Zero clear (including Latches (1) and (2))		Including devices and labels with latch specified, clears the following to zero: X, Y, M, B, F, SB, V, T, ST, LT, LST, C, LC, D, W, SW, FX, FY, FD, Z, LZ, RD, SA\X, SA\Y, SA\M, SA\B, SA\T, SA\ST, SA\C, SA\D, SA\W, and all labels (including module labels).
	File register	Zero clear	All files	Clears the contents of all the file register files to zero.
			File specification	Clears only the contents of the specified file register file to zero.
		Zero clear excluding Latch (2)		Clears the file register files other than Latch (2) to zero.
	Device/label/file register latch clear			Clears devices, labels, and file register files other than Latch (2) to zero.

Point

- If the power goes off during initialization or zero clear, the memory is left in the state of that point, and it is necessary to re-execute the memory operation.
- Following initialization, write the file required to run the SIL2 Process CPU. (📖 GX Works3 Operating Manual)
- Following data memory initialization, if the power is turned off and on or reset, the device/label memory will revert to the default status (default capacity for each area).

Files

This section lists the files used by the SIL2 Process CPU.

File types and storage memory

The following table summarizes the types of files stored in the SIL2 Process CPU and storage memory.

◎: Required, ○: Can be stored, ×: Cannot be stored

File type		CPU built-in memory			SD memory card	File name and extension
		Program memory	Device/label memory	Data memory		
		Drive 0	Drive 3	Drive 4	Drive 2	
Standard program		◎*3*5	×	◎*3*5	×	ANY_STRING.PRG
Standard FB file		○*3	×	○*3	×	ANY_STRING.PFB
Standard CPU parameter		×	×	◎	×	CPU.PRM
System parameter		×	×	◎	×	SYSTEM.PRM
Standard module parameter		×	×	○	×	UNIT.PRM
Module extension parameter		×	×	○	○	• UEXmmmnn.PRM*1 • UEXmmm00.PPR*4
Module-specific backup parameter*8		×	×	○	×	UBPmmmnn.BPR
Memory card parameter		×	×	×	○	MEMCARD.PRM
Device comment		×	×	○	○	ANY_STRING.DCM
Device initial value		×	×	○	×	ANY_STRING.DID
Standard global label setting file		×	×	○	×	GLBLINF.IFG
Initial label value file	Initial global label value file	×	×	○	×	GLBLINF.LID
	Initial local label value file	×	×	○	×	PROGRAM_NAME.LID
File register		×	○	×	○*2	ANY_STRING.QDR
Event history		×	×	○	○	EVENT.LOG
Device data storage file		×	×	○	○*2	DEVSTORE.QST
General-purpose data		×	×	○	○	ANY_STRING.CSV/BIN
Remote password		×	×	○	○	00000001.SYP
Safety program*6		◎*3*5	×	◎*3*5	×	ANY_STRING.SPG
Safety FB file*6		○*3	×	○*3	×	ANY_STRING.SPB
Safety CPU parameter*6*7		×	×	◎	×	CPU.SPR
Safety module parameter*6		×	×	○	×	UNIT.SPR
Safety global label setting file*6		×	×	○	×	GLBLINF.SIF
Standard/safety shared label setting file*6		×	×	○	×	S_GLBLINF.SIF

*1 mmm represents the start I/O number (first three digits in four-digit hexadecimal representation) of each module. For the SIL2 Process CPU, it will be 3FFH. Also, nn represents the consecutive number (two-digit hexadecimal representation) of module extension parameter files of each module.
*2 Can be stored but cannot operate as a function.
*3 When this file is stored in the built-in memory of the SIL2 Process CPU, it is divided into program memory and data memory and stored. (☞ Page 116 Configuration of a program file)
*4 Module extension parameter for the protocol setting, storing protocol setting information in the predefined protocol support function
*5 One or more of either standard program or safety program is required.
*6 The file is also stored in the SIL2 function module.
*7 This parameter relates to safety control inside the CPU parameters. (☞ Page 1091 CPU Parameters)
This item is yellow in the engineering tool CPU parameters.
*8 The module-specific backup parameter is a file for storing the save or restore data of the module to be replaced by the online module change function. For details, refer to the manual for the module used.

File operation available

This section describes file operations which are available for each file type.

■In TEST MODE

The following table lists file operations which are available for each file type in TEST MODE.

○: Available, ×: Not available, —: N/A

File type		Operation using engineering tool			Operation with SLMP and FTP server function			Operation with instructions in standard programs	
		Write	Read	Delete	Write	Read	Delete	Write	Read
Standard program		○*1	○	○*2	×	×	×	—	—
Standard FB file		○*2	○	○*2	×	×	×	—	—
Standard CPU parameter		○*2	○	○*2	×	×	×	—	—
System parameter		○*2	○	○*2	×	×	×	—	—
Standard module parameter		○*2	○	○*2	×	×	×	—	—
Module extension parameter		○*2	○	○*2	×	×	×	—	—
Module-specific backup parameter		—	—	○*4	—	—	×	—	—
Memory card parameter		○*2	○	○*2	×	×	×	—	—
Device comment		○	○	○*2	×	×	×	—	—
Device initial value		○	○	○*2	×	×	×	—	—
Standard global label setting file		○*2	○	○*2	×	×	×	—	—
Initial label value file	Initial global label value file	○	○	○*2	×	×	×	—	—
	Initial local label value file	○	○	○*2	×	×	×	—	—
File register		○	○	○*2	×	×	×	○	○
Event history		—	—	—	×	×	×	—	—
Device data storage file		—	—	—	×	×	×	○	○
General-purpose data		○	○	○	○	○	○	○	○
Remote password		○*2	○	○*2	×	×	×	—	—
Safety program		○*1	○	○*2	×	×	×	—	—
Safety FB file		○*2	○	○*2	×	×	×	—	—
Safety CPU parameter*3		○*2	○	○*2	×	×	×	—	—
Safety module parameter		○*2	○	○*2	×	×	×	—	—
Safety global label setting file		○*2	○	○*2	×	×	×	—	—
Standard/safety shared label setting file		○*2	○	○*2	×	×	×	—	—

*1 To write files to the programmable controller, use the write function when the operating status of the SIL2 Process CPU is STOP. When the operating status is RUN, use the online change function. When the SIL2 Process CPU operating status is RUN, only the standard programs registered in the CPU parameter can be written to the programmable controller.

*2 Operation is possible only when the operating status of the SIL2 Process CPU is STOP/PAUSE. If any file operation is performed in the RUN state, the operating status of the SIL2 Process CPU is changed by the remote STOP function, and then the operation continues.

*3 This parameter relates to safety control inside the CPU parameters. (☞ Page 1091 CPU Parameters)
This item is yellow in the engineering tool CPU parameters.

*4 Operation is possible only when the operating status of the SIL2 Process CPU is STOP/PAUSE. A communication error occurs when the file is operated in the RUN state.

■In SAFETY MODE

The following table lists file operations which are available for each file type in SAFETY MODE.

○: Available, ×: Not available, —: N/A

File type		Operation using engineering tool			Operation with SLMP and FTP server function			Operation with instructions in standard programs	
		Write	Read	Delete	Write	Read	Delete	Write	Read
Standard program		○*1*4	○	○*2	×	×	×	—	—
Standard FB file		○*2*4	○	○*2	×	×	×	—	—
Standard CPU parameter		×	○	×	×	×	×	—	—
System parameter		×	○	×	×	×	×	—	—
Standard module parameter		×	○	×	×	×	×	—	—
Module extension parameter		○*2	○	○*2	×	×	×	—	—
Module-specific backup parameter		—	—	○*5	—	—	×	—	—
Memory card parameter		○*2	○	○*2	×	×	×	—	—
Device comment		○	○	○*2	×	×	×	—	—
Device initial value		○	○	○*2	×	×	×	—	—
Standard global label setting file		○*2*4	○	○*2	×	×	×	—	—
Initial label value file	Initial global label value file	○	○	○*2	×	×	×	—	—
	Initial local label value file	○	○	○*2	×	×	×	—	—
File register		○	○	○*2	×	×	×	○	○
Event history		—	—	—	×	×	×	—	—
Device data storage file		—	—	—	×	×	×	○	○
General-purpose data		○	○	○	○	○	○	○	○
Remote password		○*2	○	○*2	×	×	×	—	—
Safety program		×	○	×	×	×	×	—	—
Safety FB file		×	○	×	×	×	×	—	—
Safety CPU parameter*3		×	○	×	×	×	×	—	—
Safety module parameter		×	○	×	×	×	×	—	—
Safety global label setting file		×	○	×	×	×	×	—	—
Standard/safety shared label setting file		×	○	×	×	×	×	—	—

*1 To write files to the programmable controller, use the write function when the operating status of the SIL2 Process CPU is STOP. When the operating status is RUN, use the online change function. When the SIL2 Process CPU operating status is RUN, only the standard programs registered in the CPU parameter can be written to the programmable controller.

*2 Operation is possible only when the operating status of the SIL2 Process CPU is STOP/PAUSE. If any file operation is performed in the RUN state, the operating status of the SIL2 Process CPU is changed by the remote STOP function, and then the operation continues.

*3 This parameter relates to safety control inside the CPU parameters. (☞ Page 1091 CPU Parameters)
This item is yellow in the engineering tool CPU parameters.

*4 Writing is possible in file units, however, it is not possible to select files together with parameter files or files relating to safety and then write to the programmable controller CPU. If writing in file units, write after clearing the selection of other parameter files or files relating to safety, or write after switching to TEST MODE.

*5 Operation is possible only when the operating status of the SIL2 Process CPU is STOP/PAUSE. A communication error occurs when the file is operated in the RUN state.

File size

The following table lists the size of files that can be stored in the SIL2 Process CPU.

File type		File size
Standard program		Approx. 4248 bytes minimum (only END instruction + 500 steps reserved for online program change)
Standard FB file		Approx. 4984 bytes minimum (non-processing FB + 500 steps reserved for online program change)
Standard CPU parameter		824 bytes minimum
System parameter		112 bytes minimum
Standard module parameter		The size differs depending on the module used. For example, the size is 1036 bytes at minimum when the Ethernet function is used.
Module extension parameter		The size differs depending on the module used. For example, the size is 65572 bytes when the predefined protocol support function is used with the Ethernet function.
Module-specific backup parameter		Manual for the module used
Memory card parameter		124 bytes minimum
Device comment		220 bytes minimum (when a device comment with 20 characters is set in a device)
Device initial value		Approx. 192 bytes minimum (when the initial value of a word device is set)
Standard global label setting file		Approx. 716 bytes minimum (when a word-type global label is set)
Initial label value file	Initial global label value file	Approx. 140 bytes minimum (when the initial value of a word-type global label is set)
	Initial local label value file	
File register		2048 bytes minimum (when the setting capacity is 1K words)
Event history		1024 bytes minimum (when the setting capacity is 1K words (default setting is 131072 bytes))
Device data storage file		2048 bytes minimum (when the setting capacity is 1K words)
General-purpose data		The size differs depending on the size of the file to be written.
Remote password		224 bytes minimum
Safety program		Approx. 4288 bytes minimum (only END instruction + 500 steps reserved for online change)
Safety FB file		Approx. 5028 bytes minimum (non-processing FB + 500 steps reserved for online change)
Safety CPU parameter		292 bytes minimum
Safety module parameter		The size differs depending on the module used. (Example: When the safety communication setting is used with a remote I/O station: 860 bytes)
Safety global label setting file		Approx. 752 bytes minimum (when a word-type safety global label is set)
Standard/safety shared label setting file		Approx. 748 bytes minimum (when a word-type standard/safety shared label is set)

37 PROCEDURE FOR STARTING UP A SYSTEM USING THE SIL2 PROCESS CPU

This chapter describes the procedures for starting up a system using the SIL2 Process CPU.

37.1 Overview

This section describes the overview of the procedures for starting up a system using the SIL2 Process CPU.

Point

The procedures for starting up both systems at a time are shown in this section.
In the system using the SIL2 Process CPU, both system are started up at a time. However, starting up only one system is possible when:
- Debugging is required for only one system.
- One system is powered off before module replacement because the system is failed.

When starting up the systems one by one, apply the following procedures but replace the term "both systems" with "one system" to start up the control system. For start-up of the standby system, refer to the following.
☞ Page 694 Starting up the standby system

Installing/mounting/inserting procedure

1. Installing batteries

Install a battery to the SIL2 Process CPU of each system. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

2. Inserting extended SRAM cassettes and SD memory cards

Insert an extended SRAM cassette or an SD memory card or both to the SIL2 Process CPU of each system as needed. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))
Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card. (📖 MELSEC iQ-R CPU Module User's Manual (Application))

3. Mounting modules

Mount each module to base units. (📖 MELSEC iQ-R Module Configuration Manual)

Wiring procedure

1. Wiring

Wire each module and external device.

Wiring target	Reference
Power supply module	📖 MELSEC iQ-R Module Configuration Manual
Redundant function module	☞ Page 696 Wiring the redundant function modules
From the master/local module to the remote head module	• 📖 MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup) • 📖 MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Startup)
Intelligent function module	📖 Manual for the intelligent function module used
I/O module	📖 Manual for the I/O module used

Procedure for the remote head module side

1. Powering on the system

Check the system for the following and power on the system.

- Wiring to the power supply module is correct.
- The power supply voltage is within the range of specifications.
- The remote head module is in the STOP state.

2. Creating a project

Start the engineering tool and create a project. (📖 MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Startup))

3. Connecting the personal computer and the remote head module

Connect the personal computer on which the engineering tool has been installed and the remote head module. (📖 MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Startup))

4. Initializing the remote head module

Initialize the remote head module with the engineering tool. (📖 MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Startup))

5. Setting parameters for the remote head module side

Set system parameters, CPU parameters, and module parameters of each module. (📖 MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Application))

- Set system parameters and CPU parameters.
- Set module parameters of each module.

6. Writing the parameters to the remote head module

Write the parameters set using the engineering tool to the remote head module. (📖 MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Startup))

7. Resetting the remote head module

Restart the system on the remote head module side with either of the following methods.

- Powering off and on the system
- Resetting the remote head module (📖 MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Application))

Procedure for the SIL2 Process CPU side

1. Powering on the systems

Check each system for the following and power on the systems.

- Wiring to the power supply module is correct.
- The power supply voltage is within the range of specifications.
- The SIL2 Process CPUs are in the STOP state.

Check that the following LEDs turn on after the systems are powered on.

- Power supply module: POWER LED
- SIL2 Process CPU: READY LED
- SIL2 function module: READY LED
- Redundant function module: RUN LED
- Master/local module: RUN LED

Although the ERROR LED of each module flashes or turns on as follows in step 1, proceed to the next step.

- SIL2 Process CPU: The ERROR LED flashes.
- SIL2 function module: The ERROR LED flashes.
- Redundant function module: The ERR LED turns on.
- Master/local module: The ERR LED turns on.

2. Creating a project

Start the engineering tool and create a project. (☞ Page 698 Creating a Project)

3. Connecting the personal computer and the SIL2 Process CPUs

Connect the personal computer on which the engineering tool has been installed and the SIL2 Process CPUs.

4. Initializing the SIL2 Process CPU

Initialize the SIL2 Process CPUs with the engineering tool.

After initializing a SIL2 Process CPU, connect the other SIL2 Process CPU and the personal computer. Initialize the other SIL2 Process CPU in the same way.

5. Setting parameters for the SIL2 Process CPU side

Set system parameters, CPU parameters, and module parameters of each module.

Point

System parameters are automatically set by loading an actual system configuration to the Module Configuration window of the engineering tool.

6. Safety communication settings

Set the safety communication settings. (📖 MELSEC iQ-R CC-Link IE Field Network User's Manual (Application))

7. Writing the system A/B setting

Set the system A or B with the engineering tool.

8. Resetting the SIL2 Process CPUs

Restart both systems with either of the following methods.

- Powering off and on the systems
- Resetting the SIL2 Process CPUs

9. Setting user information

Set the user information for the SIL2 Process CPUs in both systems and the project. (📖 GX Works3 Operating Manual)

10. Creating programs

Create a safety program and standard program.

11. Writing data to the programmable controller

Write the set parameters and created programs to both systems with the engineering tool.

12. Resetting the SIL2 Process CPUs

Restart both systems with either of the following methods.

- Powering off and on the systems
- Resetting the SIL2 Process CPUs

Point

For starting up only one system, perform the following operation on the engineering tool to start up the system as the control system within the time period set in "Other system Start-up Timeout Setting".

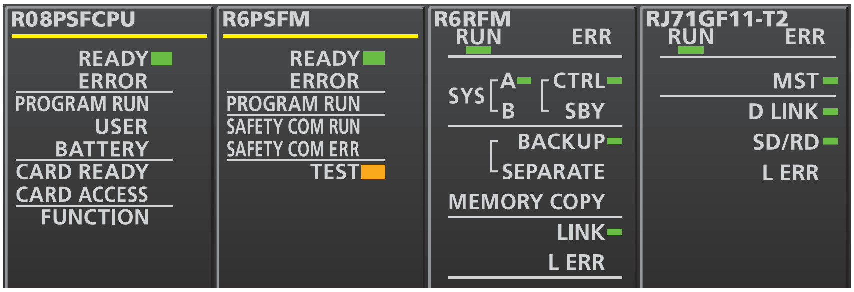
[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

Select "Forced Start of Control System while Waiting for Other System to Start" and click the [Execute] button. (When "Other system Start-up Timeout Setting" of "CPU Parameter" has not been changed, a stop error occurs in 60 seconds.)

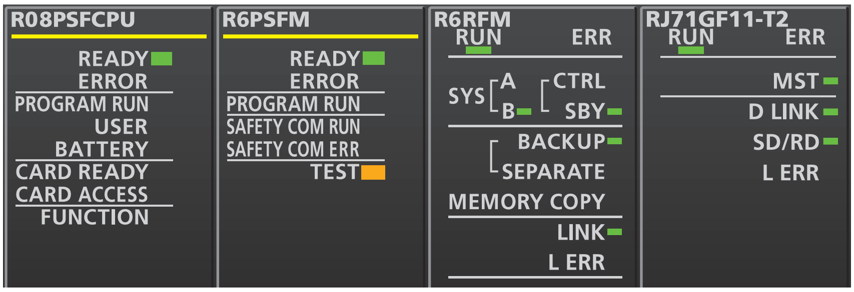
13. Checking the LEDs on the SIL2 Process CPU side

Check that the LEDs of each module are in the following states. The CARD READY LED status depends on whether an SD memory card has been installed to each CPU module or not.

- Control system



- Standby system[*1]



*1 In a redundant master station system, the MST LED of the master/local module in the standby system flashes.

Point

When one system has been started up, a continuation error occurs since only the control system has been started up. Thus, the ERROR LED of the SIL2 Process CPU and the L ERR LED of the redundant function module turn on.

The BACKUP LED flashes because a system switching disable cause has been generated.

When an error has occurred, the following LEDs turn on. Check details on the error with the engineering tool and eliminate the error cause.

- SIL2 Process CPU: ERROR LED (MELSEC iQ-R CPU Module User's Manual (Startup))
- SIL2 function module: ERROR LED (MELSEC iQ-R CPU Module User's Manual (Startup))
- Redundant function module: ERR LED, L ERR LED (MELSEC iQ-R CPU Module User's Manual (Startup))
- Master/local module: ERR LED, L ERR LED (MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup))

Enabling the modules

1. Safety module operation

Check that the systems on the SIL2 Process CPU side and remote head module side are powered on, and then enable the modules set to SIL2 mode with the "Safety Module Operation" of the engineering tool. (📖 Manual for the intelligent function module used, 📖 Manual for the I/O module used)

2. Powering off the systems

After enabling the modules, power off the systems on the SIL2 Process CPU side and remote head module side.

3. Restarting the systems

Set the RUN/STOP/RESET switch of the SIL2 Process CPU and remote head module in both systems to the RUN position and power on the systems.

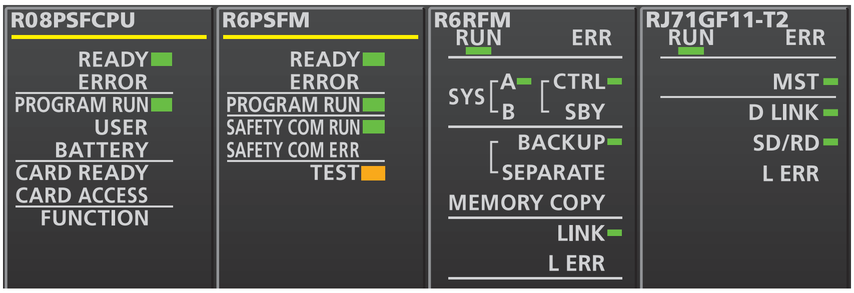
Procedure for checking the operation

1. Checking the operation

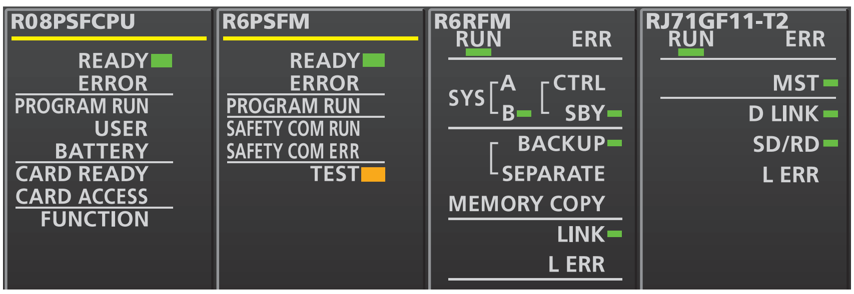
Check each module status and program operation used in the systems.

- Check if an error has occurred in each module.
- Check that the LEDs of each module are in the following states.

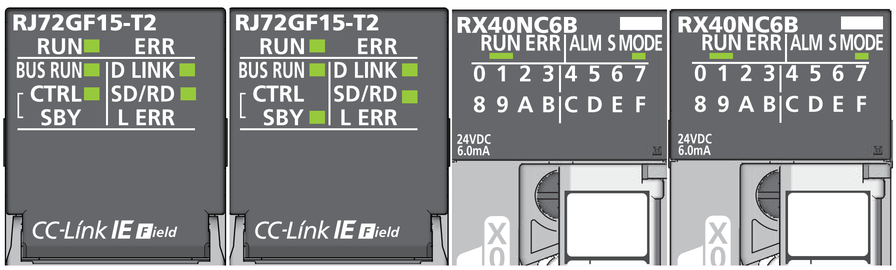
• Control system



• Standby system[*1]



• Remote head module side[*2*3]



*1 In a redundant master station system, the MST LED of the master/local module in the standby system flashes.

*2 In a redundant master station system, the following LEDs are always off since a redundant configuration is not made for the remote head module.
· CTRL LED
· SBY LED

*3 The input module with diagnostic functions is used as an example. For the LEDs of other modules, refer to the manual for the module used.

- Check if an error occurs in the CC-Link IE Field Network diagnostics. (MELSEC iQ-R CC-Link IE Field Network User's Manual (Application))
- Check the operation of safety programs and standard programs.

2. Switching the safety operation mode

Switch the safety operation mode to SAFETY MODE to run the system normally as a system using the SIL2 Process CPU. Set the SIL2 Process CPU to the STOP state before switching the mode. (Page 756 Switching Safety Operation Mode)

Point

Setting the SIL2 Process CPU to the STOP state causes a continuation error due to the operating status mismatch and the error is detected in the standby system.

3. Executing a program

Power off the SIL2 Process CPUs and remote head module in both systems. After powering off, set the RUN/STOP/RESET switch of the SIL2 Process CPU and remote head module in both systems to the RUN position and power on the systems. Check that the PROGRAM RUN LED on the SIL2 Process CPU in the control system turns on.

Point

Individually setting the RUN/STOP/RESET switch of each CPU module to the RUN position with the CPU module powered on causes a continuation error due to the operating status mismatch and the error is detected in the standby system.

4. Monitoring the program

Check that the program is normally running on the engineering tool.

Starting up the standby system

Point

When only the control system is started up first, start up the standby system subsequently.
When both systems are started up simultaneously, this procedure is not required.

Start up the standby system while the control system keeps operating.
Perform the same procedures as those for starting up the control system.

Page 689 Installing/mounting/inserting procedure
Page 689 Wiring procedure

Perform step 1, 3, and 4 for the following procedure.

Page 691 Procedure for the SIL2 Process CPU side

1. Powering off the system

Power off the standby system.

2. Wiring the redundant function modules

Connect the redundant function modules of both systems with tracking cables. (Page 696 Wiring the redundant function modules)

3. Starting up the system

Power on the standby system.

Connect the personal computer to the control system and copy the memory of the control system to the standby system with the engineering tool by following the procedure described below. (Page 736 Memory Copy from Control System to Standby System)

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

Select "Memory Copy" and click the [Execute] button.

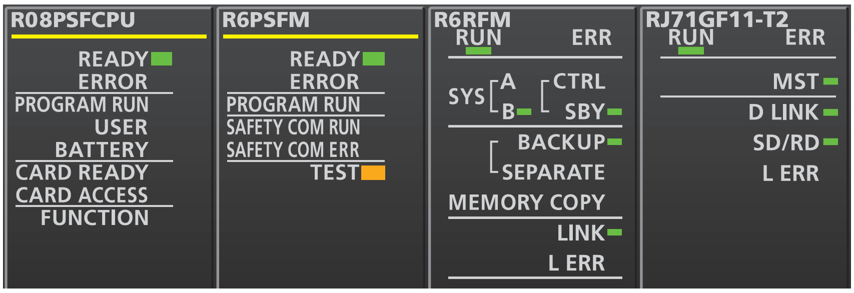
When copying the memory is completed, the MEMORY COPY LED of the redundant function module of the standby system turns on.

After that, reset the SIL2 Process CPU of the standby system and set it to the RUN state.

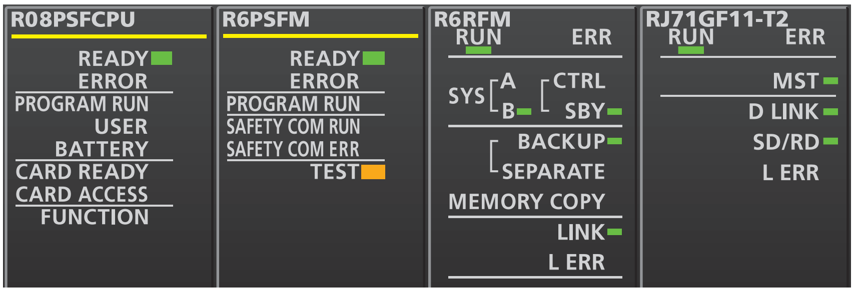
4. Checking the LEDs on the SIL2 Process CPU side

Check that the LEDs of each module are in the following states. The CARD READY LED status depends on whether an SD memory card has been installed to each CPU module or not.

- Control system



- Standby system



5. Clearing errors

When the ERROR LED of the SIL2 Process CPU of the control system is on, clear the error with the engineering tool. (☞ Page 771 Using the engineering tool)

6. Checking the operation

☞ Page 693 Procedure for checking the operation

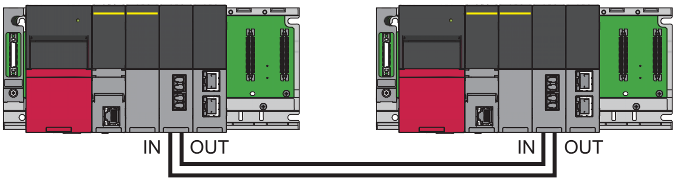
37.2 Wiring

Wiring the redundant function modules

This section describes how to wire the redundant function modules.

Wiring method

Connect the tracking cables from the OUT connector of a redundant function module to the IN connector of the other redundant function module.



Point

For the specifications of the tracking cables connected to redundant function modules, refer to the following.
MELSEC iQ-R CPU Module User's Manual (Startup)

Connecting/disconnecting tracking cables

■Connection procedure

1. Pay attention to the direction of a tracking cable connector and insert the cable to a redundant function module until it clicks. A redundant function module has one IN connector and one OUT connector. Connect the IN connector of the system A and the OUT connector of the system B with a tracking cable and connect the OUT connector of the system A and the IN connector of the system B.
2. Pull each cable lightly and check that it has been connected securely.

■Disconnection procedure

1. Disconnect the tracking cables while pressing the connector hook.

Precautions

- There are restrictions on the cable bending radius. For details, refer to the specifications of the tracking cables used.
- Place the cables in a duct or clamp them. If not, dangling cables may swing or inadvertently be pulled, resulting in damage to the module or cables or malfunction due to poor contact.
- When connecting tracking cables, pay attention not to touch optical fiber cores of the connectors on the cables and module sides and prevent dirt and dust from adhering to them. If oil on hands, dirt, and dust adhere to the optical fiber cores, the transmission loss increases and tracking may not work properly.
- Hold the connector of the tracking cables to connect or disconnect the cable. Pulling the cable connected to the module may result in malfunction or damage to the module or cable or malfunction due to poor contact.

Wiring to the power supply modules in a system using the SIL2 Process CPU

This section describes the wiring to the power supply modules.

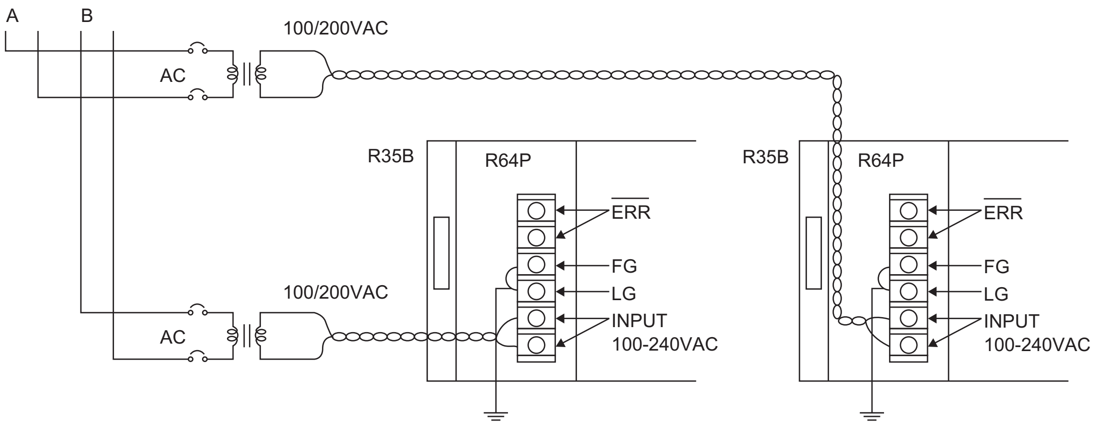
The terminal block of each power supply module has a screw size of M4. Wire cables to the terminal block with the applicable solderless terminal RAV1.25-4 or RAV2-4.

Separately supply power to the system A and B.

Wiring example

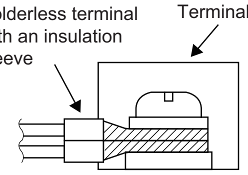
The following figure shows a wiring example of power cables to each main base unit and ground cables.

For wiring examples for each power supply module, refer to the manual included with the power supply module. (📖 Before Using the Product)



Point

- 100VAC, 200VAC and 24VDC wires must be twisted starting from the terminal connected, and connect modules at the shortest distance. Also, use the thickest wire (maximum $2mm^2$) to reduce the voltage drop.
- For the wiring to a terminal block, use a solderless terminal.
- To prevent the short-circuit because of loosening screws, use the solderless terminal with an insulation sleeve of 0.8mm or less. Note that up to two solderless terminals can be connected per terminal block.

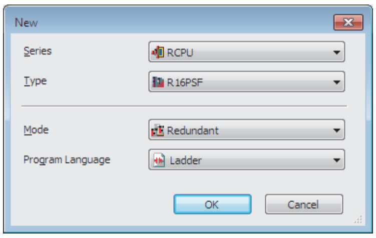


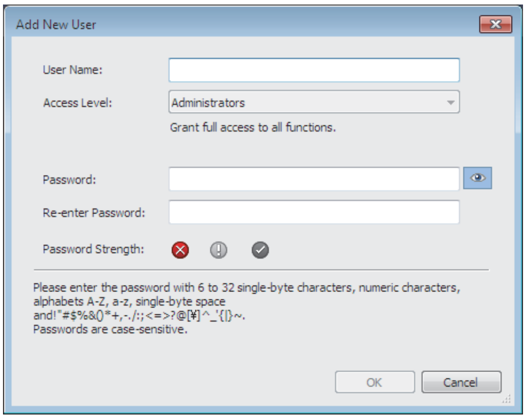
- Ground the LG and FG terminals after short-circuiting them. Failure to do so may be susceptible to the noise. The LG terminal has a half potential of the input voltage.
- When two redundant power supply modules operate in parallel as a redundant power supply system, it is recommended to connect the one redundant power supply module to an AC power supply and the other one to an uninterruptible power supply (UPS).

37.3 Creating a Project

Start the engineering tool and create a project.

[Project] ⇨ [New]





1. Select a SIL2 Process CPU to be used for "Type". The setting for "Mode" is fixed to "Redundant". Select a programming language to be used for "Programming Language" and click the [OK] button.

2. When creating a new project, registration of user information for the user authentication function is required. Set a user name and password.
(📖 GX Works3 Operating Manual)

Point

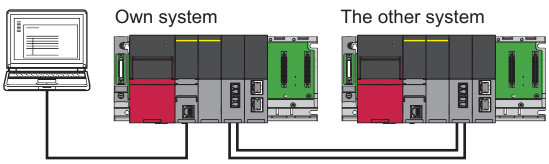
Users with an access level of "Developers", "Users", and "Assistant Developers" can be added. For details on the access level, refer to the following.
📖 GX Works3 Operating Manual

37.4 Connecting a Personal Computer and a CPU Module

Connect the personal computer on which the engineering tool has been installed and a CPU module.

Connection procedure

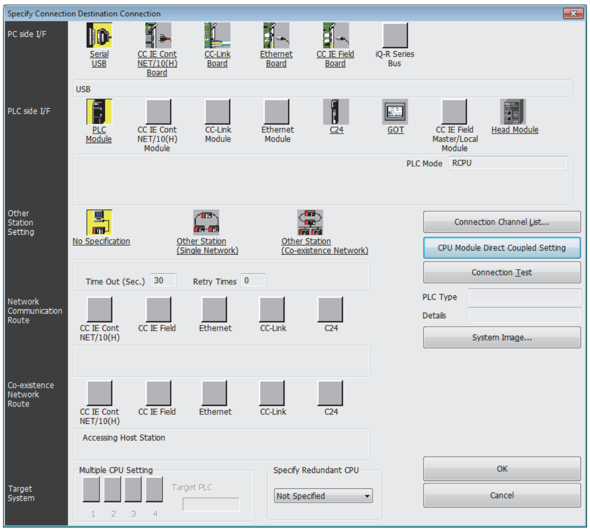
The following describes the procedure for directly connecting a CPU module and the personal computer.

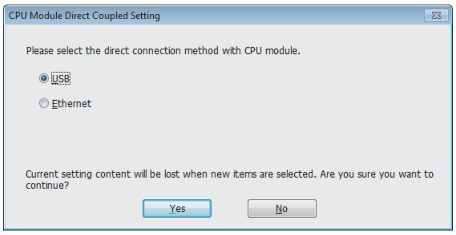


1. Connect a CPU module to the personal computer with a USB cable or an Ethernet cable. The CPU module connected with a USB cable or an Ethernet cable becomes own system.

2. Select [Online] ⇨ [Current Connection Destination] from the menu of the engineering tool.

3. Click the [CPU Module Direct Coupled Setting] button on the "Specify Connection Destination Connection" window.





4. Select a method of connection with the CPU module and click the [Yes] button.

5. Set "Specify Redundant CPU" to "Not Specified" on the "Specify Connection Destination Connection" window and click the [Connection Test] button to check that the personal computer has been connected with the CPU module.

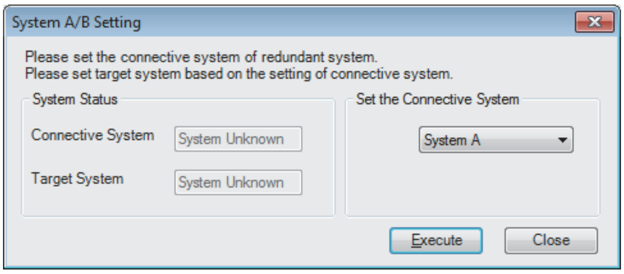
When connecting the personal computer and the CPU module with a USB cable for the first time, install a USB driver. (📖 GX Works3 Operating Manual)

37.5 System A/B Settings

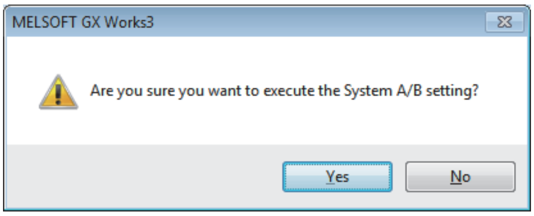
Set the system A or B with the engineering tool and write the system settings to the CPU module.

[Online] ⇨ [Redundant PLC Operation] ⇨ [System A/B Setting]

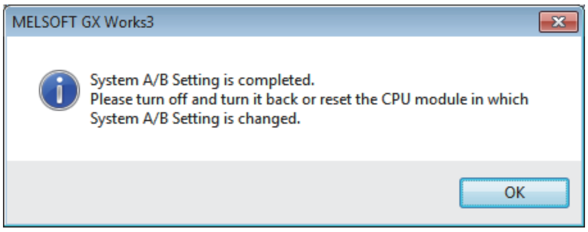
Setting procedure



1. Set the CPU module to the STOP state.
2. Select the setting to be written to the own system on the "System A/B Setting" window.
3. Click the [Execute] button.



4. Click the [Yes] button on the window on the left. Check that the SYS A LED or SYS B LED of the redundant function module flashes in accordance with the setting.



5. To change the system A/B settings, power off and on the system or reset the CPU module, and then click the [OK] button. Check that the SYS A LED or SYS B LED of the redundant function module turns on in accordance with the new setting.

Point

Besides the method that uses the engineering tool for setting system A and B, system A and B can be automatically set by using the systems.
When one system is set to system A and gets ready for tracking communications, the other system is automatically set to system B, and vice versa.
For the system to which system A or system B has been assigned automatically, the system setting will be overwritten automatically. In that case, the system setting change can be checked in the event history. (Event code: 00700)

- If a system that has no system A/B setting yet is connected to the system specified as system A, the system that has no system A/B setting will be automatically set to system B. (This event is not recorded in the event history.)
- If a system specified as system A is connected to another system specified as system A, the system where the CPU module is turned off and on or reset will be automatically set to system B. (This event is recorded in the event history.)

Check method

Check the LED of each redundant function module to check the system status.

Setting on the engineering tool	LED of the redundant function module
System A	
System B	
When the system setting is switched from system A to B	 When the system setting is switched to the system B with the engineering tool, the SYS B LED will flash. Powering off and on or resetting the CPU module turns off the SYS A LED and turns on the SYS B LED.

Point

Users can check the system A/B setting with the engineering tool. (GX Works3 Operating Manual)
- System monitor
- "System A/B Setting" window
- Monitor status bar

Precautions

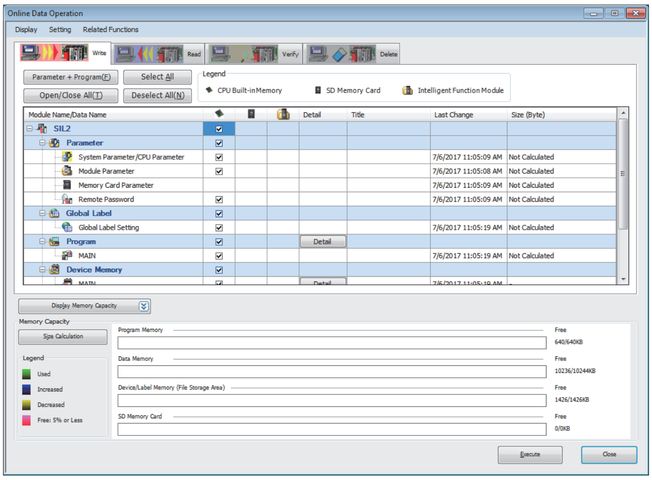
- Do not power off or reset the CPU module during system A/B setting. Doing so may not properly reflect the system A/B setting. If the system A/B setting has not been properly reflected, set the system again.
- For the system A/B setting, use the CPU module that has been connected with a USB cable or an Ethernet cable. Users cannot set a system if they have changed the connection destination on the engineering tool.
- The set system A/B information cannot be deleted. Only changing the set system A/B information is supported.
- When starting up both systems, if both systems have no system A/B setting yet or the same system A/B setting, a stop error will occur when tracking communications are established. Set a different system A/B setting for each system.
- If the system that has no system A/B setting yet is started first, a stop error will occur when tracking communications are established. Set a different system A/B setting for each system.
- Be careful not to connect between running system As or between system Bs with tracking cables. Otherwise, a continuation error will occur.

37.6 Writing Data to the Programmable Controller

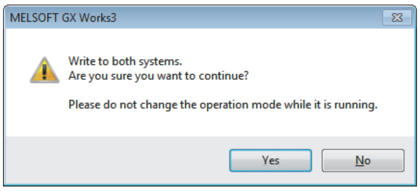
Write the set parameters and created programs to the CPU module.

[Online] ⇨ [Write to PLC]

Operating procedure



1. Select system parameters, CPU parameters (standard/safety), module parameters, and program files (standard/safety) on the "Online Data Operation" window. When FBs are used, select the corresponding FB/FUN files (standard/safety).
2. Click the [Execute] button.



3. Click the [Yes] button.*1
4. When writing the data to the programmable controller is completed, click the [Close] button.

*1 When one of both systems is started up first, the window asking for writing the data to only the connected system appears.

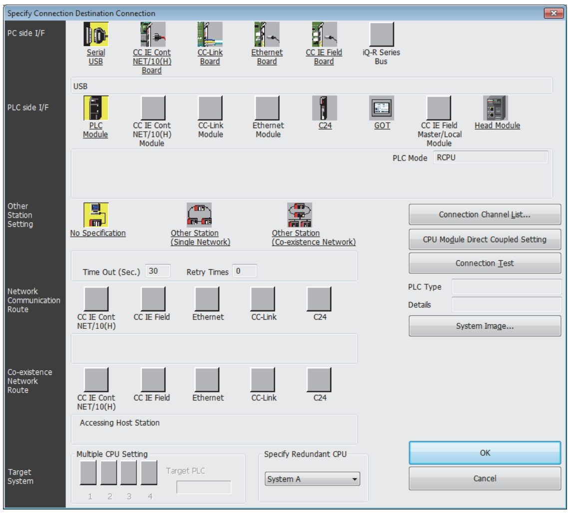
Point

- To operate CPU modules, write system parameters, CPU parameters (standard/safety), and program files (standard/safety). To operate I/O modules and intelligent function modules, write module parameters or module extension parameters.
- When new parameters have been set or the set parameters have been changed, reset the CPU module. (MELSEC iQ-R CPU Module User's Manual (Startup))
- Use the [Select Favorites] button to register frequently used items such as system parameters, CPU parameters, and programs. Select [Setting] ⇨ [Register Favorites Selection] from the menu on the "Online Data Operation" window and set the items for the [Select Favorites] button.

37.7 Monitoring the Program

Check the operation of a program on the engineering tool.
Change the connection destination with the engineering tool and check the operating status of the system A or B.

[Online] ⇨ [Current Connection Destination]



1. Select a system in "Specify Redundant CPU" on the "Specify Connection Destination Connection" window.
2. Click the [Connection Test] button to check whether the CPU module of the selected system has been connected.

For how to check the operation, refer to the following.

GX Works3 Operating Manual

37

38 FUNCTIONS

This chapter describes the redundant functions and safety functions that can be used with the SIL2 Process CPU and the functions that are different (modified or restricted) from those described in Part 2. Functions not described in this chapter are the same as those described in Part 2.

For the availability of SIL2 Process CPU functions, refer to the following.

MELSEC iQ-R CPU Module User's Manual (Startup)

Function		Description	Reference
Redundant function	System switching	Switches the systems between the control system and the standby system to continue operation of the redundant system when a failure or an error occurs in the control system. The systems can also be switched manually by a user for debugging or maintenance.	Page 708 System Switching
	Tracking transfer	Transfers the control data from the control system to the standby system and maintains the consistency of the data in the two systems to continue operation of the redundant system when a failure or an error occurs in the control system.	Page 721 Tracking Transfer
	Memory copy from control system to standby system	Transfers data such as parameters and programs in the CPU module of the control system to the CPU module of the standby system to maintain the consistency of the memory in the two CPU modules.	Page 736 Memory Copy from Control System to Standby System
	System consistency check	Checks whether the system configurations and files in the CPU modules are the same between the control system and the standby system.	Page 742 System Consistency Check
	Program execution in both systems	Detects an error in the external device or network of the systems (control system and standby system) by executing a program that diagnoses external devices or networks of both systems.	Page 747 Program Execution in Both Systems
	Redundant function module communication test	Checks a redundant function module for an error when its communications are unstable.	Page 755 Redundant Function Module Communication Test
Safety function	Switching safety operation mode	There are two safety operation modes: TEST MODE and SAFETY MODE. One mode is for changing the safety program or safety parameters, and the other is for operating the system as a safety system.	Page 756 Switching Safety Operation Mode
	Safety diagnostic function	Self-diagnostic function of the SIL2 Process CPU	Page 759 Safety Diagnostic Function
	Identification check for safety data	Checks if the project data created using the engineering tool and the data in the SIL2 Process CPU are the same, and confirms that the program executed in SAFETY MODE is the one written by the user.	Page 759 Identification Check for Safety Data
	Safety communication function	Communicates data between the SIL2 Process CPU of the control system and modules supporting safety functions using safety protocols. Safety communication processing is not performed in the SIL2 Process CPU of the standby system.	Page 760 Safety Communication Function
	User authentication function of CPU modules	Prevents an unauthorized access to data (such as a program or parameters) written to a CPU module. Operations are restricted by registering a user name/password for a CPU module.	Page 763 User Authentication Function of CPU Modules

Function			Description	Reference
Functions different (modified or restricted) from Part 2	Constant scan		• In the standby system, system switching when the system is powered off, a hardware failure has occurred, or a tracking cable has a failure may generate a continuation error due to the excess of constant scan time. • The constant scan function is invalid for the standby system.	Page 764 Constant Scan
	Output mode setting of STOP to RUN		The safety output (SA\Y) for the safety output device (SA\Y) is off, regardless of this setting. If the value is changed with a device test or others while the CPU module is in the STOP state, the value will be applied when the status of the CPU module changes from STOP → RUN.	Page 778 Safety output (SA\Y)
	Interrupt function	Internal timer interrupt	When the systems are switched during time measurement of I28 to I31, the time measurement is interrupted and the new control system starts the time measurement from 0.	—
		Interrupt period setting	The interrupt pointer (I) cannot be used in safety programs. Also, the following cannot be used in standard programs. • Inter-module synchronous interrupt (I44) • Multiple CPU synchronous interrupt (I45) • High-speed internal timer interrupts (I48, I49)	MELSEC iQ-R CPU Module User's Manual (Startup)
		Multiple interrupt function	Safety cycle processing is performed as a higher priority interrupt than other processing, and therefore even when set to "Enable" in the multiple interrupt settings, interrupts will never occur during safety cycle processing.	—
	Clock function	Clock setting	When setting clock data for the Safety CPU at the "Set Clock" window of the engineering tool, "All Stations Specified" cannot be selected as an execution target.	—
		Time synchronization	The time of the standby system is synchronized with the time of the control system. The time setting using the time setting function (SNTP client) is available only for the control system.	—
		System clock	After system switching, SM420 (User timing clock No.0) to SM424 (User timing clock No.4) remain off on the new control system CPU module. To use SM420 to SM424 on the new control system CPU module, execute the DUTY instruction again.	—
	Writing data to the CPU module	Online change (ladder block)	• When the online change (ladder block) is performed on the CPU module in one system, the change is also reflected on the CPU module in the other system. • Online changes (ladder block) are not possible in safety programs. • Safety program files, safety global label setting files, and standard/safety shared global label setting files cannot be changed online. However, if standard/safety shared labels inside standard programs are edited, standard programs can be changed online. • If a disconnection, power-off, or reset of the CPU module is detected while the program restoration information is being written, the processing will be suspended and a message will be displayed. If the processing is suspended by a disconnection, click the [Retry] button in the message after reconnecting cables. If the processing is suspended by power-off or reset, login to the user authentication function is disabled. Log in to the function and write data again to the programmable controller.	Page 764 Online Change

Function			Description	Reference
Functions different (modified or restricted) from Part 2	RAS function	Scan monitoring function	• During system switching, scan time monitoring with the watchdog timer is interrupted. Thus, no error is detected even if the scan time monitoring time has elapsed. Therefore, a time taken for system switching does not need to be considered in the scan time monitoring time setting. • Scan time monitoring with the watchdog timer is interrupted while the standby system is waiting for tracking data reception. Thus, no error is detected even if the scan time monitoring time has elapsed while the standby system is waiting for tracking data reception. Therefore, a time taken for tracking data reception does not need to be considered in the scan time monitoring time setting.	—
		Self-diagnostic function	The following items are added to the contents that can be checked with the system monitor. • Tracking cable status • Display of the main base units of systems A and B • Module configuration on the base unit of the selected system • Event history of own system • Systems to which a CPU module belongs (control system/standby system, system A/B)	GX Works3 Operating Manual
			• The behavior for each setting differs. • Applicable error codes to relevant parameter settings of the SIL2 Process CPU are listed in the following chapter.	Page 771 Self-diagnostic function
			The "Instruction Execution Error" setting under the "CPU Module Operation Setting at Error Detected" is applied only for standard programs. A stop error always occurs in safety programs. (Even if "Continue" is selected, a stop error occurs.)	—
		Error clear	• Errors that can be cleared are added. Clearing the errors of the SIL2 function module is explained in the following chapter. • A continuation error that occurred in the own system can be cleared by using the special relay (SM50) or the connected engineering tool. • An error on the standby system CPU module can be cleared by using the special relay (SM1679) of the control system CPU module.	Page 773 Error clear, Page 775 Clearing errors on the standby system CPU module from the control system CPU module
		Event history function	Events in the SIL2 Process CPU are added. Note that events that occur on the SIL2 function module are not retained in the event history.	Page 908 Event list
	Remote operation	Using a contact	Both systems may be recognized as being mismatched in the system consistency check even if the remote operation is simultaneously performed on the systems because their operating statuses are mismatched depending on the timing.	—
		Using an engineering tool	When specifying the execution destination at the "Remote Operation" window of the engineering tool, the specification of all stations or group No. is not supported, and the CPU module cannot issue the command if being issued by the own station. All stations or group No. specification is not executed by the CPU module if those in which all stations or group No. have been specified are received. Also, if the Safety CPU is in SAFETY MODE when clearing device/label memory when performing remote RUN, safety global devices, safety global devices, safety global labels, safety local labels, and standard/safety shared labels are not cleared.	Page 765 Remote Operation
	Test function		Changes in values of devices and labels are transferred (tracking) from the control system to the standby system before the END processing. If the systems are switched between the change and the tracking transfer, the change is not reflected on the new control system.	—
	Security function		The security key setting is written or deleted on both systems individually.	—
	PID control function		PID control instructions can be used only in standard programs.	MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

Function			Description	Reference
Functions different (modified or restricted) from Part 2	Process control function		Process control function blocks and process control instructions can be used only in standard programs.	MELSEC iQ-R Programming Manual (Process Control Function Blocks/ Instructions)
	Label initialization function	Label initialization after converting all programs	Safety labels and standard/safety shared labels are not applicable as only standard labels are supported.	—
		Label initialization with label initial values		
	Label access setting from external device		• In the transfer setup of external devices that communicate with the CPU module through GOT, SLMP, or other methods, select "Control System/ Standby System", "System A/System B", or "Not Specified" to start a communication with a global label name specified. • Safety global labels or standard/safety shared labels cannot be specified in the global label settings.	—
	Latch function		• When the systems are switched, the new control system and the new standby system start the value in "Interval Setting at Time Setting" in "Device Latch Interval Setting" from 0. • Safety devices, safety labels, and standard/safety shared labels cannot be latched.	—
	Device/label initial value setting		An initial value cannot be set for safety global devices, safety local devices, safety global labels, safety local labels, and standard/safety shared labels. To set an initial value for those devices or labels, create a program so that the safety state can be secured and the value is set only at start-up.	—
	Device/label access service processing setting		Even though this function is set, the scan time of some functions (operations) becomes longer than the specified time. The following are added to those functions. • Safety local devices (label) batch/registration monitor • Current value change of the safety labels/safety devices • Switching safety operation mode • Identification check for safety data	—
	SLMP communications		• There are notes for system switching when the system IP address matching function is not used. • If a remote operation command is executed, the CPU modules enter different operating statuses and thus systems cannot be switched. • For SLMP communications via the built-in Ethernet port of the CPU module, when communications are performed to the other system that cannot respond (power-off, reset, or tracking cable disconnection), a timeout error may occur. • Safety devices cannot be accessed. Also, the command execution status differs.	Page 776 SLMP Communications
	Ethernet function	File transfer function (FTP server)	Safety devices cannot be accessed. Also, the command execution status differs.	Page 686 File operation available

38.1 System Switching

This function switches the systems between the control system and the standby system to continue operation of the redundant system when a failure or an error occurs in the control system. The systems can also be switched manually by a user for debugging or maintenance.

System switching method

The following two methods are available for system switching: automatic system switching (automatically performed by a system) and manual system switching (performed by a user).
The following table lists the system switching types, causes, and priority of when multiple causes are simultaneously generated.

System switching type	System switching cause	Priority	
Automatic system switching	Power-off, reset, hardware failure of the CPU module	High ↑ ↓ Low	1
	Stop error of the CPU module		2
	System switching request from a network module		3
Manual system switching	System switching request by the SP.CONTSW instruction		4
	System switching request from the engineering tool		5

- When multiple system switching causes are simultaneously generated, the systems are switched according to the cause with higher priority. The switching cause determined according to the priority is stored in the event history and SD1643 (System switching cause).
- The manual system switching is requested for the control system CPU module.

Point

The systems are not switched even when the CPU module is set to the STOP state. They are switched when any of the above system switching causes is generated.

Automatic system switching

A system judges whether system switching is required or not, and automatically switches the systems between the control system and the standby system as required.

■System switching due to power-off, reset, or hardware failure of the CPU module

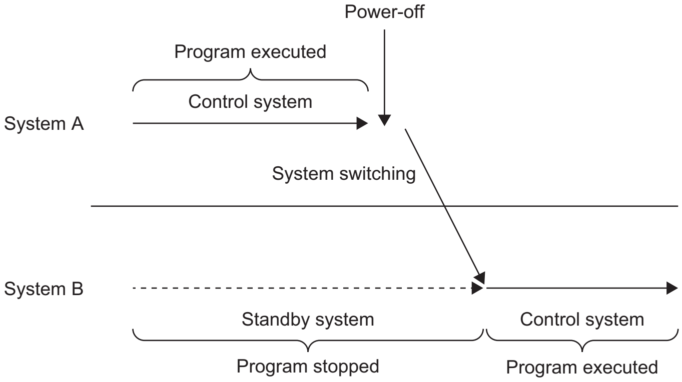
The standby system CPU module monitors the control system status. If the current control system is unable to control the system in the following situations, the standby system CPU module is switched to the control system and the new control system CPU module continues the control over the system.

- The control system CPU module has been turned off.
- The control system CPU module has been reset.
- A hardware failure has occurred on the control system CPU module.*1

*1 The control system CPU module is switched to the standby system even when the existing standby system CPU module is not ready to switch. (☞ Page 715 Execution availability of system switching)

Ex.

Operation of when the control system (system A) CPU module is turned off



Point

When a network is established using the CC-Link IE Field Network module, the systems can be switched even when a redundant function module is removed from the base unit or a failure has occurred in the base unit.

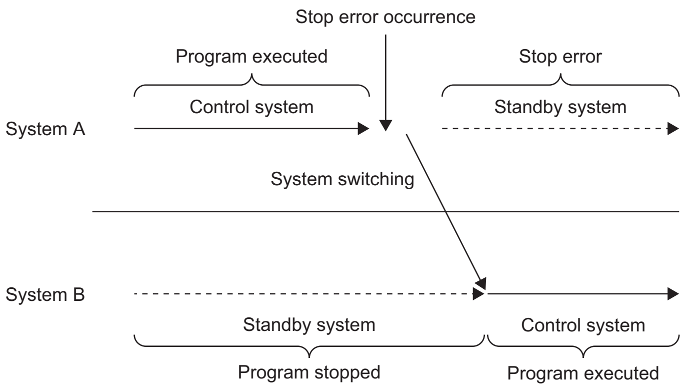
■Stop error of the CPU module

When a stop error occurs on the control system CPU module, the standby system is notified of a system switching due to the stop error, and the standby system CPU module is switched to the control system. The CPU module of the control system where the stop error has occurred is switched to the standby system.*1

*1 If a WDT error has occurred, the control system CPU module is switched to the standby system even when the existing standby system CPU module is not ready to switch. (☞ Page 715 Execution availability of system switching)

Ex.

Operation of when a stop error occurs on the control system (system A) CPU module

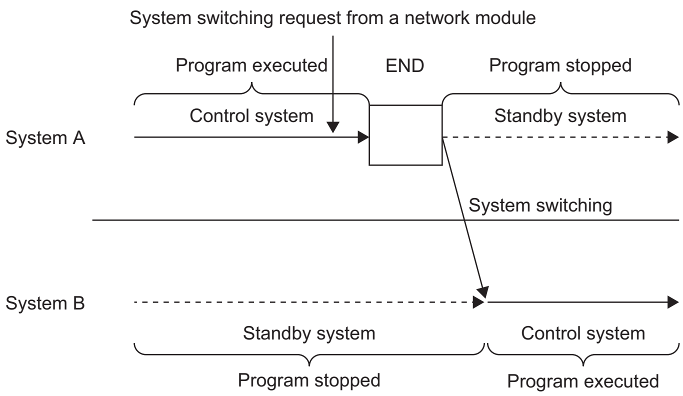


■System switching request from a network module

A control system network module requests the CPU module to switch systems when a communication error or disconnection is detected. When the control system CPU module receives the system switching request from the network module, the systems are switched in the END processing.

Ex.

Operation of when the CPU module receives a system switching request from a network module



The following network modules send a system switching request.

- CC-Link IE Controller Network module
- CC-Link IE Field Network module
- Ethernet interface module with built-in CC-Link IE

If a network module cable is disconnected, the systems may not be switched depending on the timing of error detection on the control system and the timing on the standby system. (☞ Page 720 When the cable for the network module is disconnected)

Manual system switching

The user can manually switch the systems between the control system and the standby system.

Point

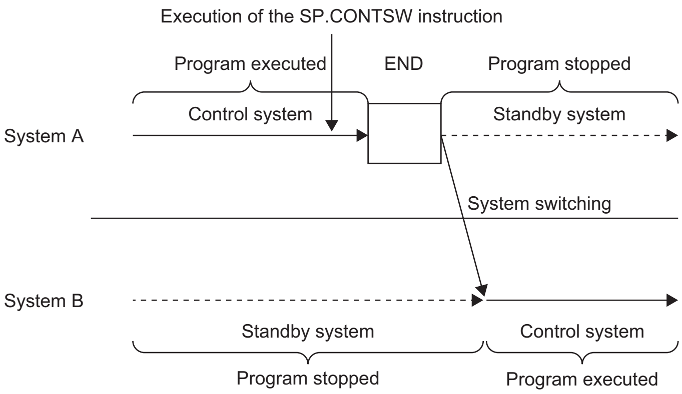
- After turning on SM1646 (System switching by a user), perform the manual system switching in the control system.
- When the manual system switching is disabled by the DCONTSW instruction, execute the ECONTSW instruction. The system switching is enabled in the initial status.

■System switching by the SP.CONTSW instruction

When the SP.CONTSW instruction is executed on the control system CPU module, the systems are switched at the END processing after the instruction execution.

Ex.

Operation of system switching by using the SP.CONTSW instruction

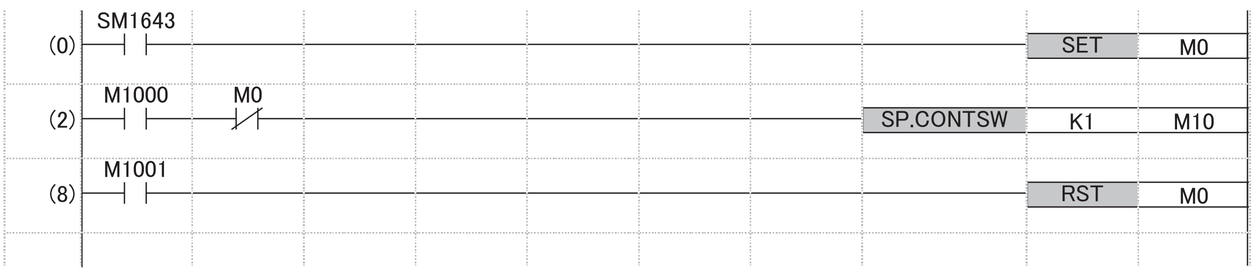


Point

Once the devices and labels specified in "Tracking Setting" are transferred, the CPU modules in both of the control system and standby system have the same specified devices and labels. Thus, if the systems are switched by using the SP.CONTSW instruction on the control system CPU module, switching may also be performed on the new control system CPU module.

When the SP.CONTSW instruction is used, create a program that does not execute the SP.CONTSW instruction again on the new control system CPU module by using SM1643 (ON for only one scan after system switching (standby system to control system)), as shown below.

M1000: System switching command, M1001: Clear signal



For details on the SP.CONTSW instruction, refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

■System switching request from the engineering tool

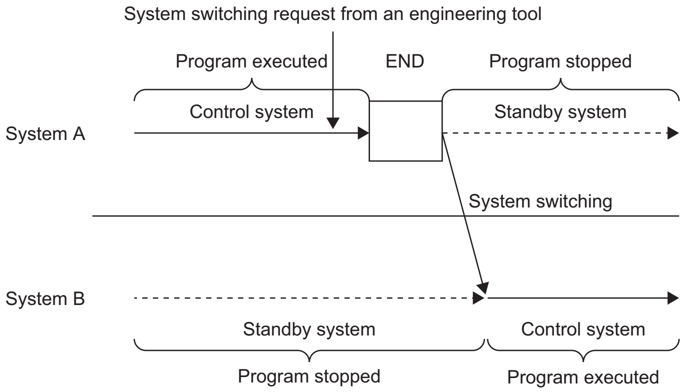
When the engineering tool sends a system switching request to the control system CPU module, the systems are switched in the END processing.

Switch the systems from the "Redundant Operation" window of the engineering tool.

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

Ex.

Operation of system switching from the engineering tool



Operation at system switching

The following table shows the operations of the CPU modules of when the control system and the standby system are switched.
These operations are for when both systems are operating and the operating statuses of the CPU modules are the same.

Item			New control system CPU module	New standby system CPU module
Program execution	Standard program	Initial execution type program	This type of program is not executed. However, when an initial execution type program has not been completed on the old control system at system switching, it is executed again from its head.	The program execution stops.[*1]
		Scan execution type program	This type of program is executed from the step 0.	
		Fixed scan execution type program	The fixed scan execution interval is measured from 0.	The program execution stops.
		Standby type program	This type of program is not executed.	
		Event execution type program	• Interrupt program: The program is executed when an interrupt factor is generated. • ON of bit data (TRUE): The program is executed when a specified factor is generated. • Passing time: A specified time is measured from 0.	
	Safety program	Fixed scan execution type program	The fixed scan execution interval is measured from 0.	
Tracking transfer			Tracking data is transferred to the new standby system.	Tracking data is received. However, when a stop error occurs on the new standby system, the new standby system does not receive the tracking data.
Online change	Online change (ladder block)	Standard program	Write operation at system switching continues.	
		Safety program	Safety programs cannot be changed online.	
	File batch online change	Standard program	Write operation at system switching continues.	
		Safety program	Safety programs cannot be changed online.	
Device/label memory (standard label, safety label, standard/safety shared label)			The state before system switching is held.	
Signal flow memory (standard/safety program)		With tracking setting	The signal flow of the old control system is reflected.	The state before system switching is held.
		Without tracking setting	The signal flow of the tracking transfer targets is turned on.	
Setting the device/label initial value			The values are not set.	
Special relay (SM), special register (SD)			The state before system switching is held. However, when the CPU module is in the RUN state, SD520 to SD531 and SD1890 to SD1895 are cleared.	
Safety special relay (SA\SM), safety special register (SA\SD)			The state before system switching is held. The safety special relay areas/safety special register areas related to safety communications (SA\SM1008, SA\SM1016, SA\SM1024, SA\SM1032, SA\SM1040, SA\SM1048, SA\SM1056, SA\SM1064, SA\SM1088, SA\SD1008 to SA\SD1071, SA\SD1088, SA\SD1089, SA\SD1090 to SA\SD1097, SA\SD1104 to SA\SD1223, SA\SD1232 to SA\SD1359, and SA\SD1600 to SA\SD1663) are cleared.	
Output (Y)			The state before system switching is held.	The state is held when the standby system output setting of the redundant system operation setting is enabled. If the setting is disabled, the output (Y) is turned off and then the output refresh stops.
Safety output device (SA\Y)				The safety output device (SA\Y) is turned off regardless of the standby system output setting.
Safety output	CC-Link IE Field Network (CPU module → Master/local module)		After a safety program which is the fixed scan execution type is operated using transferred device data, data is output.[*2]	Refresh processing is stopped because the safety connection is disconnected.

Item	New control system CPU module	New standby system CPU module
Local device setting (standard/safety program)	This setting is in accordance with the parameter settings.	No operation is performed because the program does not operate.[*1]
File register setting	The file register setting before system switching is held.	
Direct access input (DX)	In the program execution after system switching, data is fetched when an instruction using the direct access input (DX) is executed.	
Direct access output (DY)	In the program execution after system switching, data is output when an instruction using the direct access output (DY) is executed.	
FROM/TO instructions	In the program execution after system switching, this instruction is executed when the instruction execution condition is satisfied.	No operation is performed because the program does not operate.[*1]
Instruction that requires several scans	• This instruction is executed in the program execution after system switching. • When the systems are switched while the instruction is being executed in a program executed in both systems, the instruction execution continues and the completion device turns on at a completion of the instruction execution.	• No operation is performed because the program does not operate.[*1] • When the systems are switched while the instruction is being executed, the instruction execution continues. However, the completion device does not turn on in the new standby system, and it turns on in the control system after the next system switching.[*1]
Constant scan	Constant scan is enabled.	Constant scan is disabled.

*1 The operation varies in a program executed in both systems. (Page 749 Operation at system switching)

*2 The safety program may be executed multiple times up until safety output is performed from the CPU module to the network module.

Execution availability of system switching

The following tables show the execution availability of system switching.

○: Switching possible, ×: Switching not possible

Status of system using the SIL2 Process CPU		Execution availability of system switching						
		Automatic system switching					Manual system switching	
		Power-off or reset of the CPU module	Hardware failure of the CPU module	Stop error of the CPU module		System switching request from a network module*3	System switching request by the SP.CONTSW instruction*3	System switching request from the engineering tool*4
				WDT error	Error other than WDT error			
Normal operation or continuation error in the standby system		○	○	○	○	○	○	○
Tracking communications disabled (cable disconnection)		×	○*6	○*6	×	×	×	×
Power-off, reset, hardware failure of the standby system CPU module		×	○*6	○*6	×	×	×	×
Stop error in the standby system	WDT error	○	○*6	○*6	×	×	×	×
	Error other than WDT error	○	○	○*6	×	×	×	×
Network error detected in the standby system*1		○	○	○	○	×	×	×
During memory copy from control system to standby system		○	○	○*6	×	×	×	×
During online change		○	○	○	○	○*2*5	×	×
Mismatch between the CPU module operating status of both systems		○	○	○	○	×	×	×
Mismatch between the safety operation mode of both systems		○	○	○	○	×	×	×
Error in safety tracking data detected		×	○*6	○*6	×	×	×	×
During system switching*7		○	○	○	○	○	×	×
During online module change	Redundant function module	×	○*6	○*6	×	×	×	×
	Other modules	○	○	○	○	○	○	○
System switching disabled by the DCONTSW instruction		○	○	○	○	○	×	×

*1 When the group specification has been set in the standby system, a network error is not detected if communication is available with the line of an Ethernet-equipped module after a communication error has occurred on the other Ethernet-equipped module. (MELSEC iQ-R Ethernet User's Manual (Application))

*2 After the online change has been completed, the system switching cause is detected and the systems are switched. However, when an online change is being executed only in the standby system, the systems cannot be switched.

*3 If system switching is disabled when a system switching request is sent, a continuation error occurs and the cause of the system switching failure is stored in SD1644 (Cause of system switching failure).

*4 If system switching is disabled when a system switching request is sent, the error code corresponding to the cause of the system switching failure is returned.

*5 When the systems are switched during an online change, a mismatch between the files is detected in the system consistency check and a stop error may occur on the new standby system. (Page 744 File)

*6 Although the control system CPU module is switched to the standby system CPU module, the existing standby system is not switched.

*7 The initial processing of the safety function after system switching completes is also included.

Occurrence of a cause of the system switching failure

The BACKUP LED flashes when a cause of the system switching failure has been generated.

Point

The cause of the BACKUP LED flashing can be checked in SD1642 (BACKUP/SEPARATE LED flashing cause). Check SD1642 and eliminate the cause to flash the LED.

Check method of system switching information

The following table lists the check methods of system switching information at system switching (automatic system switching and manual system switching).

Check method	Information	Reference
Event history	System switching result, system switching cause, and control system/standby system transition	Page 716 Event history
Special relay (SM)/Special register (SD)	System switching result and detailed information	Page 717 Special relay/Special register
CTRL LED and SBY LED of the redundant function module	System switching result	Page 717 CTRL LED and SBY LED of the redundant function module

When the systems have been switched, check the switching cause or detailed information and take corrective action to restore the system to a normal state as required.

Event history

The information related to system switching results, system switching cause, and control system/standby system transitions can be checked with the event history of the engineering tool.

When the systems are switched, the following items are stored in the event history of both systems.

- Automatic system switching: "System switching (system)" (event code: 00F00)
- Manual system switching: "System switching (user)" (event code: 2B000)

When the systems are switched by the SP.CONTSW instruction, the system switching instruction ID number specified with the SP.CONTSW instruction is also stored.

Special relay/Special register

System switching results and detailed information can be checked with the special relay and special register.

- Whether the systems have been switched or not can be checked by checking SD1649 (System switching cause (when the systems are successfully switched)). When the systems have been switched, the system switching cause is stored in SD1649 of the control system and standby system.
- If the systems have not been switched even after a system switching cause is generated, the cause why the systems have not been switched can be checked by checking SD1644 (Cause of system switching failure). The switching cause is stored in SD1643 (System switching cause).

For the values stored in the special relay and special register, refer to the following.

- Special relay (☞ Page 961 Redundant function)
- Special register (☞ Page 1012 Redundant function)

■Special relay

The following table shows the special relay for system switching and the storage status of the CPU modules in the control system and standby system.

○: Stored, ×: Not stored

SM number	Name	Storage status at system switching	
		New control system CPU module	New standby system CPU module
SM1637	System switching detection (standby system to control system)	○	○
SM1643	ON for only one scan after system switching (standby system to control system)	○	×
SM1644	ON for only one scan after system switching (control system to standby system)	×	○
SM1645	System switching request from a network module	○	○
SM1646	System switching by a user	×	×

■Special register

The following table shows the special register for system switching and the storage status of the CPU modules in the control system and standby system.

○: Stored, ×: Not stored

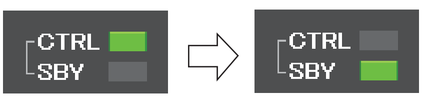
SD number	Name	Storage status at system switching	
		New control system CPU module	New standby system CPU module
SD1642	BACKUP/SEPARATE LED flashing cause	×	×
SD1643	System switching cause	×	○
SD1644	Cause of system switching failure*1	×	×
SD1645	System switching request status from a network module of own system	○	○
SD1646	System switching request status from a network module of the other system	○	○
SD1648	Cause of the other system monitoring error	○	○
SD1649	System switching cause (when the systems are successfully switched)	○	○
SD1650	System switching instruction ID number	○	○

*1 When the system switching has not been normally completed, a value is stored in this special register area of the control system.

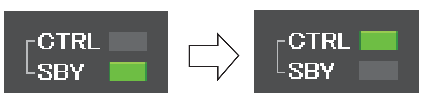
CTRL LED and SBY LED of the redundant function module

System switching results can be checked with the CTRL LED and SBY LED on the front of the redundant function module.

Switching from the standby system to the control system



Switching from the control system to the standby system



Precautions

The following table lists the precautions on system switching.

Item	Description	Reference
Error in the redundant function module	• When an error has been detected on the redundant function module, the control system and standby system continue operating without being switched. • When a communication error has been detected in the communication between the redundant function module and a CPU module, the systems are switched.	Page 719 Error in the redundant function module
When both systems operate as standby systems	• If a communication error is generated due to a tracking cable error during system switching, both systems may operate as standby systems. • If a WDT error or a hardware failure of the CPU module has occurred in the control system while a stop error exists in the standby system, both systems operate as standby systems.	Page 719 When both systems operate as standby systems
When both systems operate as control systems	If the tracking cable and the network cable that is connected to the control system network module are simultaneously pulled out or disconnected, both systems may operate as control systems. In this case, connect the tracking cable again. After a stop error has occurred in the system B, power off and on or reset the system B.	—
Scan time monitoring function	During system switching, scan time monitoring with the watchdog timer is interrupted. Thus, in a scan in which the systems are switched, no error is detected even if the scan time monitoring time has elapsed. A time taken for system switching does not need to be considered in the scan time monitoring time (WDT) setting.	—
System switching disabled by online change	Manual system switching is disabled during an online change.	Page 719 System switching disabled by online change
Operation of an event execution type program	When "ON of bit data (TRUE)" has been specified in the trigger type and the systems are switched before tracking transfer of device data, an event execution type program does not operate in the new control system.	—
When the cable for the network module is disconnected	If a network module cable is disconnected, the systems may not be switched depending on the timing of error detection on the control system and the timing on the standby system.	Page 720 When the cable for the network module is disconnected
System switching during initial processing/initial processing (when switched to RUN)	• When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the control system, the systems are switched after the initial processing or initial processing (when switched to RUN). • When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the standby system, whether the systems are switched or not depends on the system switching cause.	Page 720 System switching during initial processing/initial processing (when switched to RUN)
CC-Link control at system switching	When the master operating station is switched with a program by using the CC-Link standby master function, the CC-Link control can be continued even after system switching.	MELSEC iQ-R CC-Link System Master/Local Module User's Manual (Application)
When a stop error is generated in a new standby system at system switching	When executing a system switching, if there is a hardware failure in the SIL2 Process CPU or SIL2 function module, the startup of the safety control after switching to the new standby system may take two seconds longer compared to the normal system switching.	—

Error in the redundant function module

- When an error has been detected on a redundant function module, a continuation error occurs on the CPU module, and the control system and standby system continue operating without being switched. Check the error code, and perform online module change if the redundant function module has failed. (MELSEC iQ-R Online Module Change Manual)
- When the redundant function module was removed from the base unit or a failure has occurred in the base unit and a communication error has been detected in the communication between the redundant function module and a CPU module, a stop error occurs on the CPU module and the systems are switched. In this case, check the error code and eliminate the error cause.

Restriction

When the CC-Link IE Field Network module has not been mounted and a communication error has been detected in the communications between the redundant function module and the CPU module, the systems may not be switched.

To switch the systems even though the redundant function module has been removed from the base unit or a failure has occurred in the base unit, mount the CC-Link IE Field Network module.

When both systems operate as standby systems

- If a communication error is generated due to a tracking cable error during system switching, both systems may operate as standby systems. When an error occurs with the tracking cable, each L ERR LED of the redundant function modules turns on. In this case, replace the tracking cable with a new one as soon as possible. If both systems operate as standby systems, connect a new tracking cable properly and turn off and on or reset the CPU module of one system so that the other system will operate as the control system.
- If a WDT error or a hardware failure of the CPU module has occurred in the control system while a stop error exists in the standby system, the control system is switched to the standby system and the both systems operate as standby systems. In this case, eliminate error causes, and then power off and on or reset the CPU modules in the both systems.

The following table lists the operations of when both systems operate as standby systems.

Item		Operation of both systems
LED of the redundant function module	BACKUP LED	Flashing
	CTRL LED	Off
	SBY LED	On
Special relay	SM1635 (Standby system judgment flag)	On
System switching request	System switching request from a network module	System switching disabled
	System switching request by the SP.CONTSW instruction	
	System switching request from the engineering tool	

System switching disabled by online change

Manual system switching is disabled during an online change.

To disable the manual system switching during an online change, the CPU module is set to the manual system switching disabled state before an online change starts. After the online change is completed, it is set to the manual system switching enabled state.

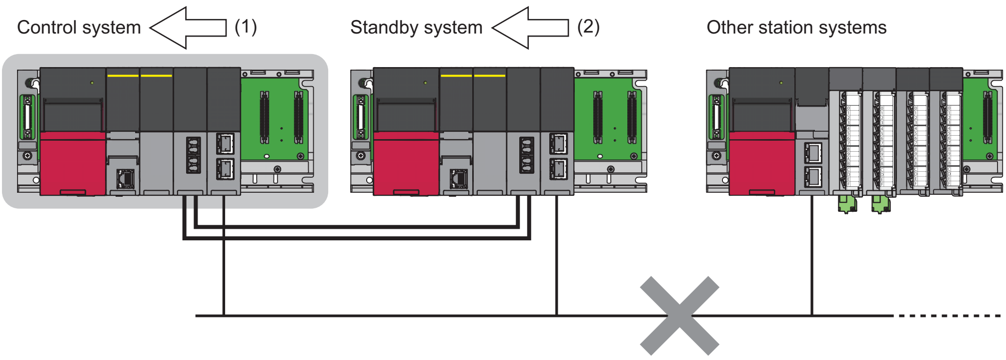
If a communication with the engineering tool is interrupted due to communication cable disconnection or other causes during an online change, the online change cannot be completed, and the CPU module remains in the manual system switching disabled state. When the CPU module is in the manual system switching disabled state, the systems cannot be switched by manual system switching or a system switching request from a network module.

If an online change has failed, refer to the following and take actions.

☞ Page 764 Action for when an online change has failed

When the cable for the network module is disconnected

If a network module cable is disconnected, the systems may not be switched depending on the timing of error detection on the control system and the timing on the standby system.



(1) When the control system detects cable disconnection first, the systems are switched.
(2) When the standby system detects cable disconnection first, the systems are not switched. At this time, a continuation error occurs on the control system CPU module and the BACKUP LED flashes since a cause of the system switching failure has been generated.

In both case, replace the network module cable with a new one and clear the network error.

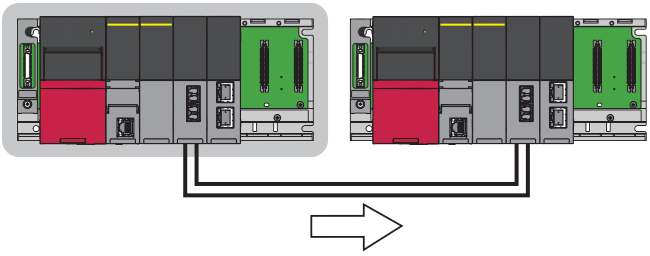
System switching during initial processing/initial processing (when switched to RUN)

When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the control system, the systems are switched after the initial processing or initial processing (when switched to RUN). Even if the initial processing or initial processing (when switched to RUN) takes time and a communication error occurs, the systems are switched after the initial processing or initial processing (when switched to RUN).
When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the standby system, whether the systems are switched or not depends on the system switching cause.

System switching cause	Whether to switch the systems
Power-off, reset, hardware failure of the CPU module	The systems are switched after the initial processing or initial processing (when switched to RUN).
Stop error of the CPU module	
System switching request from a network module	The systems are not switched.
System switching request by the SP.CONTSW instruction	
System switching request from the engineering tool	

38.2 Tracking Transfer

This function transfers the control data from the control system to the standby system and maintains the consistency of the data in the two systems to continue operation of the redundant system when a failure or an error occurs in the control system.



- The control system CPU module sends tracking data to the standby system CPU module before the END processing. The tracking data includes special relay (SM), special register (SD), PID control instruction information, and system data in addition to device (standard/safety device)/label (standard label, safety label, and standard/safety shared label) data. (☞ Page 722 Tracking data)
- The standby system CPU module receives the tracking data, and then reflects the received data on the device/label memory.
- Standard device/standard label data can be divided and transferred as required. To transfer device/label data within a desired range, set a range of standard global devices to be transferred or whether to transfer standard local devices/ standard global labels/standard local labels for each tracking block and turn on the tracking triggers (SD1667 to SD1670) assigned for each tracking block.
- Up to 1M-word standard device/standard label data can be transferred in one scan.
- When a stop error has occurred, only system data is transferred. (☞ Page 724 System data)

Point

- In the CPU parameters, "Tracking Device/Label Setting" under "Tracking Setting" in "Redundant System Settings" is set to "Transfer collectively" by default. Thus, tracking transfer is performed without setting the parameter. (☞ Page 728 Batch transfer)
- When a program file is converted with the process control extension setting enabled, "Tracking Device/ Label Setting" is automatically set to "Detailed setting". In the tracking block No.64, the range of the file register specified in the system resource of the process control extension setting is registered. When the system is operating in backup mode, the tracking transfer trigger (bit 15 of SD1670) corresponding to the tracking block No.64 automatically turns on at every scan. (☞ Page 727 When "Process Control Extension Setting" is enabled)

Restriction

When a tracking transfer is not performed, the possible causes are the following:

- The standby system CPU module has been powered off or reset.
- Hardware failure of the CPU module has occurred.*1
- An error has occurred in the redundant function module.*1
- A WDT error has occurred.*1
- Tracking cables has been pulled out or disconnected.

*1 When the tracking communication line properly works, "Link-up" (event code: 00100) may be stored in the event history.

38

Tracking data

The following table lists the tracking data that can be transferred from the control system to the standby system.

○: Transfer possible, ×: Transfer not possible

Item		Transfer*4	Reference
Device data	User device	○	Page 723 Devices that can be specified
	Safety user device	○*3	
	Special relay	○ (Auto transfer)	
	Safety special relay	×	
	Special register	○ (Auto transfer)	
	Safety special register	×	
	Index register	○	
	File register	○	
	Refresh data register	○	
Global label*2 (Except module labels)	Standard global label	○	Page 728 Tracking device/label setting
	Safety global label	○*3	Page 730 Safety tracking transfer setting
	Standard/safety shared label	○*3	
Local device	Standard local device	○*1	Page 728 Tracking device/label setting
	Safety local device	○*3	Page 730 Safety tracking transfer setting
Local label	Standard local label	○*1	Page 728 Tracking device/label setting
	Safety local label	○*3	Page 730 Safety tracking transfer setting
Signal flow memory	Standard signal flow memory	○*1	Page 728 Tracking transfer setting for the signal flow memory
	Safety signal flow memory	○*3	Page 730 Safety tracking transfer setting
PID control instruction information		○ (Auto transfer)	Page 724 PID control instruction information
CPU buffer memory		×	—
System data		○ (Auto transfer)	Page 724 System data

*1 The data that is used in a program executed in both systems is not transferred. (☞ Page 747 Program Execution in Both Systems)
*2 Global labels/safety global labels with devices assigned are not transferred as global labels/safety global labels. They are transferred according to the tracking settings/safety tracking settings of the assigned devices.
*3 Tracking transfer is performed when "Transfer" is set in "Safety Tracking Setting" of "Redundant System Settings" in the CPU parameters. It is performed when the safety operation mode is either TEST MODE or SAFETY MODE.
*4 During online change on both systems or on the control system only, only system data is transferred. During online change on the standby system only, data is transferred according to the table.

Devices that can be specified

The following table lists the data that can be specified for tracking transfer.

○: Specifiable, ×: Not specifiable, —: Not settable as a local device

Classification	Device name	Transfer	
		Global device	Local device
User device	Input (X)	○	—
	Output (Y)	○	—
	Internal relay (M)	○	○
	Link relay (B)	○	—
	Annunciator (F)	○	—
	Link special relay (SB)	○	—
	Edge relay (V)	○	○
	Step relay (S)	×	—
	Timer (T)	○	○
	Retentive timer (ST)	○	○
	Long timer (LT)	○	○
	Long retentive timer (LST)	○	○
	Counter (C)	○	○
	Long counter (LC)	○	○
	Data register (D)	○	○
	Link register (W)	○	—
	Link special register (SW)	○	—
	Latch relay (L)	○	—
Safety user device	Safety input (SA\X)	○	—
	Safety output (SA\Y)	○	—
	Safety internal relay (SA\M)	○	○
	Safety link relay (SA\B)	○	—
	Safety timer (SA\T)	○	○
	Safety retentive timer (SA\ST)	○	○
	Safety counter (SA\C)	○	○
	Safety data register (SA\D)	○	○
	Safety link register (SA\W)	○	—
System device	Function input (FX)	×	—
	Function output (FY)	×	—
	Function register (FD)	×	—
	Special relay (SM)	○*1	—
	Special register (SD)	○*1	—
Safety system device	Safety special relay (SA\SM)	×	—
	Safety special register (SA\SD)	×	—
Index register	Index register (Z)	○	○
	Long index register (LZ)	○	○
File register	File register (R)	×	—
	File register (ZR)	○	—
Refresh data register	Refresh data register (RD)	○	—

*1 The data is automatically transferred regardless of parameter settings. (☞ Page 724 Auto tracking data)

Auto tracking data

The following tables list the data that is automatically transferred by the system regardless of parameter settings of tracking transfer.

■Special relay

The following table lists the special relay areas that are automatically transferred by the system.

SM number	Name
SM752	Dedicated instruction End bit control flag
SM754	BIN/DBIN instruction error control flag
SM755	Scaling data check settings
SM756	Module access completion wait control flag
SM775	Selection of link refresh processing during the COM instruction execution
SM776	Local device setting at CALL
SM777	Local device setting in interrupt programs
SM792	PID bumpless processing (for the complete differentiation PIDCONT instruction)
SM794	PID bumpless processing (for the inexact differential S.PIDCONT instruction)
SM816	Hold mode (S.IN instruction)
SM817	Hold mode (S.OUT instruction)
SM1646	System switching by a user

For details on the special relay, refer to the following.

☞ Page 930 List of Special Relay Areas

■Special register

The following table lists the special register areas that are automatically transferred by the system.

SD number	Name
SD250	Loaded maximum I/O
SD414	2n second clock setting
SD415	2n ms clock setting
SD771	Specification of the number of write instruction executions to data memory
SD775	Selection of refresh processing during the COM instruction execution
SD792, SD793	PID limit setting (for complete derivative)
SD794, SD795	PID limit setting (for incomplete derivative)
SD816, SD817	Basic period
SD818	Bumpless function availability setting for the S.PIDP instruction
SD1662	Tracking transfer data receive completion wait time
SD1667 to SD1670	Tracking trigger

For details on the special register, refer to the following.

☞ Page 966 List of Special Register Areas

■PID control instruction information

The PID control data that is specified with the PIDINIT or S.PIDINIT instruction is transferred. (📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))

■System data

The data related to the system, such as system switching and operation mode change, is transferred.

Tracking block and tracking trigger

The devices or labels of a specified range is transferred by setting a range of devices or labels to be transferred for each tracking block and turning on the tracking trigger which is assigned for each tracking block. Note that only standard devices/labels can be transferred for each tracking block. (Safety devices/labels cannot be transferred for each tracking block.)

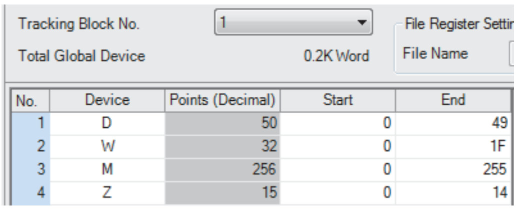
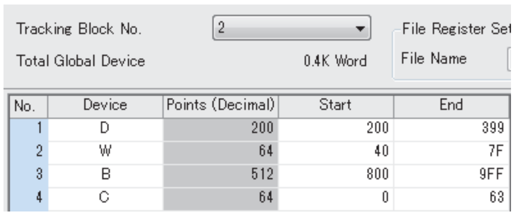
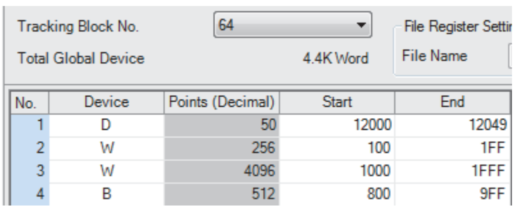
Tracking block

A tracking block is used for setting a tracking transfer range of global devices and whether or not to transfer local devices/global labels/local labels.

- Up to 64 blocks (No.1 to 64) are available.
- Up to 2048 devices can be set in a block. Up to 2048 devices can be transferred in a single tracking.
- The available capacity of tracking devices or labels is 1M words per block.

Ex.

When only the tracking block No.3 has no setting

Tracking block No.	Setting example
1	
2	
3	Not set
⋮	⋮
64	

Point

Global labels, local labels, and local devices are transferred only once in one scan.
Even when the same global labels, local labels, and local devices are specified in multiple tracking blocks, the overlap does not affect the data capacity to be transferred.

Tracking trigger

By turning on a tracking trigger, the devices or labels of a range specified in the corresponding tracking block are transferred. Bits used as tracking triggers change depending on the CPU parameter setting, as shown below.

■When "Tracking Device/Label Setting" is set to "Transfer collectively"

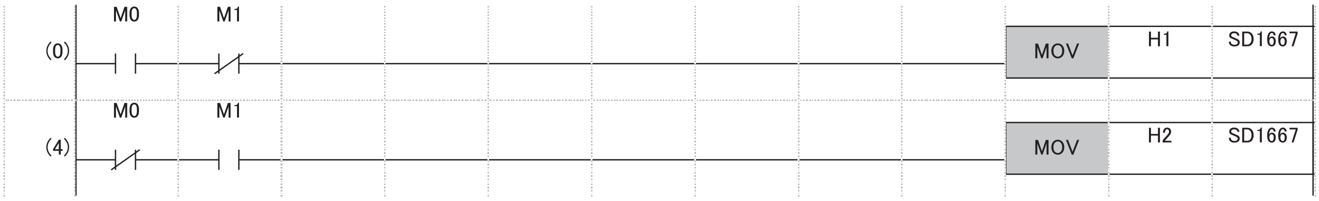
The bit 0 of SD1667 is used as a tracking trigger. The bit is automatically turned on by the system at an initial processing or operating status change (STOP to RUN), and the tracking transfer is started. To stop the tracking transfer, turn off the bit. To restart, turn on the bit.

■When "Tracking Device/Label Setting" is set to "Detailed setting"

SD1667 to SD1670 (64 bits) are used as tracking triggers. The bit 0 of SD1667 to the bit 15 of SD1670 correspond to the tracking blocks No.1 to 64. To start the tracking transfer of a tracking block, turn on the corresponding bit. To stop the tracking transfer, turn off the bit. To restart the tracking transfer, turn on the bit again.
When "Tracking Block No.1 Auto Transfer Setting" is set to "Transfer Automatically" in the CPU parameter, the bit 0 of SD1667 is automatically turned on by the system at initial processing or operating status change from STOP to RUN, and the tracking transfer is started.

Ex.

Program example for changing the status of tracking transfer triggers according to conditions



- When only M0 (Trigger switching condition 1) is on, the tracking transfer trigger of the tracking block No.1 is turned on. (Tracking blocks No.2 to No.64 are not transferred.)
- When only M1 (Trigger switching condition 2) is on, the tracking transfer trigger of the tracking block No.2 is turned on. (Tracking blocks No.1 and No.3 to No.64 are not transferred.)

Setting procedure for tracking transfer

The following describes the setting procedure for tracking transfer.

1. Set "Tracking Setting" and "Safety Tracking Setting" in the CPU parameters. (☞ Page 727 Tracking transfer setting, ☞ Page 730 Safety tracking transfer setting)
2. Add a program that controls tracking triggers. (☞ Page 726 Tracking trigger)
3. Write the set parameters and program to the CPU module with the engineering tool.
4. Start up the system again.
5. Turn on the tracking trigger with the program to start the tracking transfer.

When "Tracking Device/Label Setting" is set to "Transfer collectively", the above steps 2 and 5 are not required.

Point

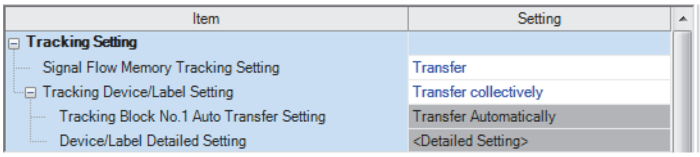
To continue the same control as before even after system switching, configure the settings in "Tracking Setting" so that all data necessary for the program is transferred.

Tracking transfer setting

The following describes the CPU parameters related to tracking transfer.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Tracking Setting]

Window



Displayed items

Item	Description	Setting range	Default
Signal Flow Memory Tracking Setting	Sets whether to transfer the signal flow memory or not. (Page 728 Tracking transfer setting for the signal flow memory)	• Do Not Transfer • Transfer	Transfer
Tracking Device/Label Setting	Sets "Transfer collectively" or "Detailed setting" for devices or labels to be transferred. • When "Transfer collectively" is set, global devices, local devices, global labels, and local labels are assigned to the tracking block No.1 and automatically transferred. (Page 728 Batch transfer) • When "Detailed setting" is set, detailed settings can be configured in "Tracking Block No.1 Auto Transfer Setting" and "Device/Label Detailed Setting".	• Transfer collectively • Detailed setting	Transfer collectively
Tracking Block No.1 Auto Transfer Setting	Sets whether to automatically transfer the tracking block No.1 or not. (Page 725 Tracking block and tracking trigger)	• Do Not Transfer Automatically • Transfer Automatically	Transfer Automatically
Device/Label Detailed Setting	Sets devices and labels to be transferred. (Page 729 Detailed setting)	—	—

Point

Set "Detailed setting" of "Tracking Device/Label Setting" for the following purposes.
- To shorten a tracking transfer time
- To add tracking data, such as the file register (ZR)
- To transfer each data set individually

When "Process Control Extension Setting" is enabled

Settings and operations are as follows:
- When a program file is converted, "Tracking Device/Label Setting" is automatically set to "Detailed setting".
- In the tracking block No.64, the range of the file register specified in the system resource of the process control extension setting is registered. Since the other devices are not automatically registered, specify the other tracking block numbers for those devices. The devices and their ranges set in "Device/Label Memory Area Setting" can be input in a batch by clicking the [Device Setting Reflection] button in the "Global Device Setting" window. (Page 730 Global device setting)
- To transfer the tracking block No.64, the bit 15 of SD1670 automatically turns on at every scan when the system is operating in backup mode. Do not turn off the bit 15 of SD1670.

Tracking transfer setting for the signal flow memory

By transferring the signal flow memory, operations of rising/falling instructions in the old control system are taken over to the new control system even after system switching.

Point

"Signal Flow Memory Tracking Setting" is set to "Transfer" by default. Transferring the signal flow memory is recommended. For the operation not to transfer it, refer to the following.

☞ Page 790 Instructions whose operations vary depending on tracking of the signal flow memory

■Tracking of individual POUs

The following table shows whether the signal flow memory is transferred or not for each POU.

○: Transferred, ×: Not transferred, —: No signal flow memory

POU			Both systems program executions setting	
			Control system execution	Both systems executions
Program block			○	×
Function block	Macro type		○	×
	Subroutine type	Global FB	○	○*1
		Local FB	○	×
Function			—	

*1 To prevent the signal flow memory in the standby system from being overwritten in a both systems execution program, use a local FB. When a global FB is used, the signal flow memory in the standby system is overwritten with the signal flow memory in the control system.

Tracking device/label setting

The following two methods are available for transferring devices and labels: automatically transferring all the devices and labels in a batch or transferring specified devices and labels of a specified tracking block.

■Batch transfer

When "Tracking Device/Label Setting" is set to "Transfer collectively", the following devices and labels are assigned to the tracking block No.1 and automatically transferred.

Type	Description
Global device	■Bit device • Input (X) • Output (Y) • Internal relay (M) • Link relay (B) • Edge relay (V) • Latch relay (L) ■Word device • Timer (T) • Long timer (LT) • Retentive timer (ST) • Long retentive timer (LST) • Counter (C) • Long counter (LC) • Data register (D) • Link register (W) • Index register (Z) • Long index register (LZ)
Local device*1	All local devices
Global label*2	All the global labels assigned to the device/label memory
Local label*1	All local labels

*1 The data used in both systems execution programs is not transferred.

*2 Global labels with devices assigned are not transferred as global labels. They are transferred according to the tracking settings of the assigned devices. To transfer global labels with devices assigned, specify the assigned global devices in the global device setting. (☞ Page 730 Global device setting)

Point

To transfer the annunciator (F), link special relay (SB), link special register (SW), file register (ZR), or refresh data register (RD), specify the corresponding data in "Device/Label Detailed Setting" of "Tracking Device/Label Setting". (☞ Page 729 Detailed setting)

Restriction

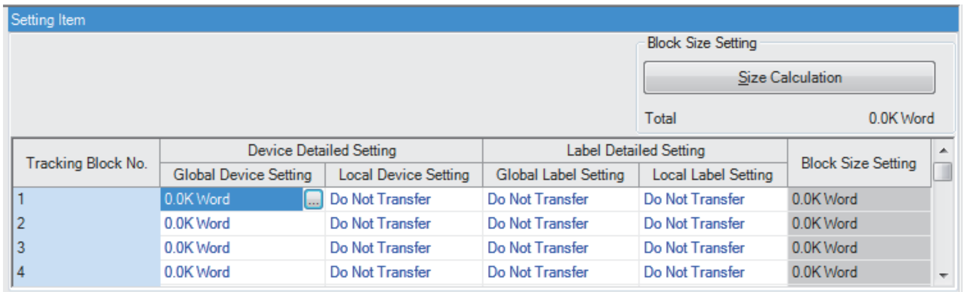
After setting "Transfer collectively", perform a test operation in the system design phase and check if the size of tracking data is 1M words or smaller. If the size of the tracking data is larger than 1M words, a stop error occurs when the CPU module is powered off and on or reset.

■Detailed setting

Set devices and labels to be transferred for each tracking block (No.1 to 64) to be used.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Tracking Setting] ⇨ [Device/Label Detailed Setting]

Window



Displayed items

Item	Item	Description	Setting range	Default
Device Detailed Setting	Global Device Setting	Sets global devices to be transferred. (☞ Page 730 Global device setting)	—	0.0K Word
	Local Device Setting	Sets whether to transfer local devices or not. When "Transfer" is set, all the local devices are transferred. For the devices that can be set as local devices, refer to the following. ☞ Page 723 Devices that can be specified	• Do Not Transfer • Transfer	Do Not Transfer
Label Detailed Setting	Global Label Setting*1	Sets whether to transfer global labels or not. When "Transfer" is set, all the global labels assigned to the device/label memory are transferred.	• Do Not Transfer • Transfer	Do Not Transfer
	Local Label Setting	Sets whether to transfer local labels or not. When "Transfer" is set, all the local labels are transferred.	• Do Not Transfer • Transfer	Do Not Transfer
Size Calculation		Calculates "Block Size Setting" of each tracking block and "Total" of the setting capacity.	—	—

*1 Global labels with devices assigned are not transferred as global labels. They are transferred according to the tracking settings of the assigned devices.

Point

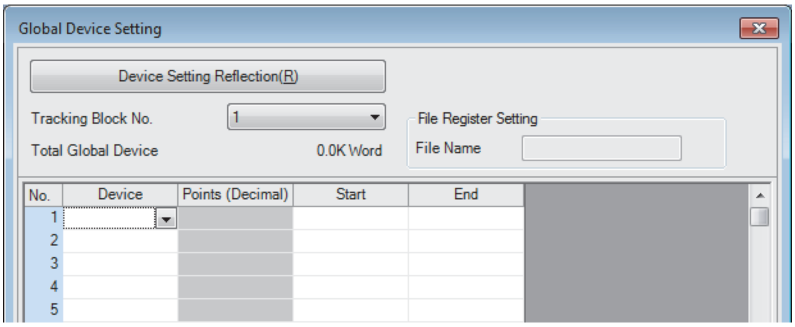
When setting devices and labels for tracking transfer, click the [Size Calculation] button to check if the tracking data capacity in one scan is equal to or less than 1M words.

■Global device setting

Set devices and their ranges for each tracking block No.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Tracking Setting] ⇨ [Device/Label Detailed Setting] ⇨ [Global Device Setting]

Window



Displayed items

Item	Description
Device Setting Reflection	Reflects the device setting of "Device/Label Memory Area Setting" in the CPU parameter. (Except for the annunciator (F), link special relay (SB), and link special register (SW))
Tracking block No.	Selects the number of a tracking block to be set.
File Register Setting	Enter a file name of the file register file. This item is valid only when the file register (ZR) is selected in the device name field.
Device	Selects a device to be transferred. (☞ Page 723 Devices that can be specified)
Start/End	Specifies a range of devices to be transferred.

Point

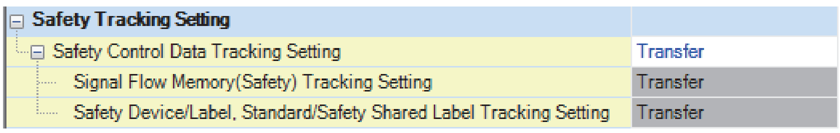
To input devices and their ranges set in "Device/Label Memory Area Setting" in a batch, click the [Device Setting Reflection] button.

Safety tracking transfer setting

Set tracking transfer of the information required for safety control.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Safety Tracking Setting]

Window



Displayed items

Item		Description	Setting range	Default
Safety Control Data Tracking Setting	Signal Flow Memory (Safety) Tracking Setting	Set whether to transfer the data to be used for safety control or not. "Signal Flow Memory (Safety) Tracking Setting" and "Safety Device/Label, Standard/Safety Label Tracking Setting" are set according to "Safety Control Data Tracking Setting".	• Do Not Transfer • Transfer	Transfer
	Safety Device/Label, Standard/Safety Label Tracking Setting			

Tracking mode

The following two modes are available for tracking.

Item	Description
Synchronous tracking mode	Tracking data is always transferred to the standby system once every scan of the control system. During a tracking transfer from the control system to the standby system, the next scan does not start in the control system.
Asynchronous tracking mode	When a tracking transfer from the control system is to be performed and the previous tracking is still in progress, the tracking transfer from the control system is canceled and the previous tracking continues. The control system starts the next scan without waiting for notifications of data reception/reflection completion from the standby system.

The tracking mode depends on the CPU module operating status.

CPU module operating status of standby system and control system		Tracking mode
Control system	**Standby system**	
RUN	RUN	Synchronous tracking mode[*1]
	STOP, PAUSE	Asynchronous tracking mode
STOP, PAUSE	RUN	
	STOP, PAUSE	

*1 When the RUN/STOP/RESET switch of the CPU module of each system is set to the RUN position and both systems are powered on, data is transferred in the asynchronous tracking mode at first. After the tracking data is reflected to the standby system, the mode is switched to the synchronous tracking mode.

In the following conditions, the tracking mode is the asynchronous tracking mode.

- During an online change
- When the systems are switched
- When tracking communications disabled is detected

Point

Even if the safety operation mode is changed, the tracking mode will not change.

Effect on the scan time

The following describes the effect on the scan time depending on the tracking mode.

Point

For the calculation method for an increase in the scan time due to tracking transfer, refer to the following.

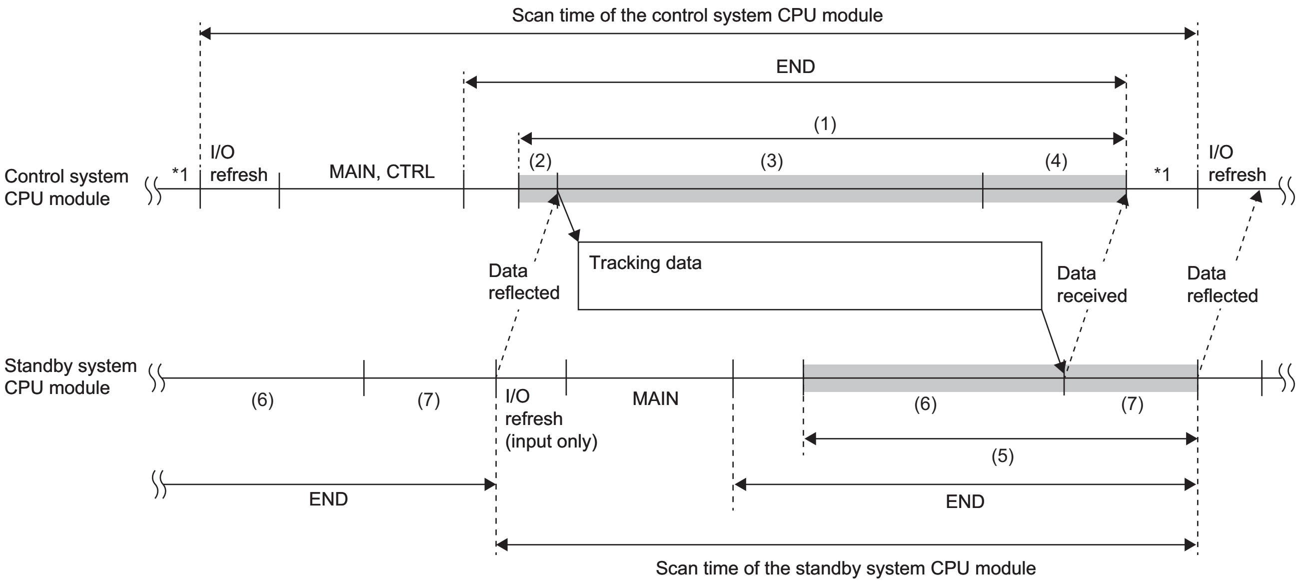
☞ Page 1078 Increase in the scan time due to tracking transfer

■Synchronous tracking mode

In the synchronous tracking mode, tracking transfer is always performed once every scan during the END processing. Until the standby system receives the tracking data sent from the control system, the next scan is not started. Thus, the scan time of the control system and standby system increases by the time for tracking data send/receive processing.

Ex.

CTRL: Program executed in the control system, MAIN: Program executed in both systems



*1 If the constant scan is used, waiting time for the constant scan generated.

In the control system, the scan time is extended by the following tracking send processing time.

Item		Description
(1) Tracking send processing	(2) Waiting for completion of tracking data reflection	After receiving a notification of the reflection completion from the standby system, the control system sends the tracking data.
	(3) Sending tracking data	The control system sends the tracking data.
	(4) Waiting for completion of tracking data reception	The control system waits for a notification of receive completion from the standby system. After receiving the notification of the receive completion from the standby system, the control system starts another END processing.

In the standby system, the scan time increases by the following tracking receive processing time.

Item		Description
(5) Tracking receive processing	(6) Waiting for tracking data reception	The standby system receives the tracking data from the control system. After receiving the tracking data, the standby system notifies the control system of the receive completion and reflects the tracking data.
	(7) Reflecting tracking data	The standby system reflects the tracking data. After the reflection completion, the standby system notifies the control system of the reflection completion and starts another END processing.

■Asynchronous tracking mode

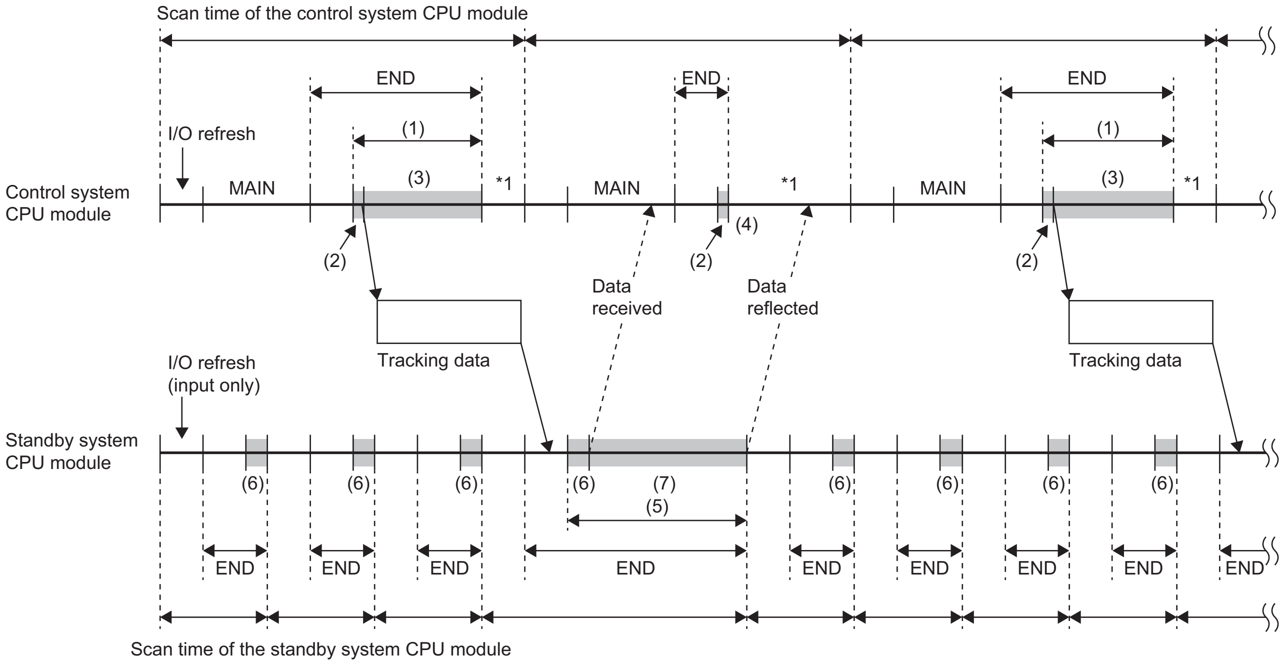
In the asynchronous tracking mode, the control system starts the next scan without waiting for notifications of data reception/ reflection completion from the standby system.

Unlike the synchronous tracking mode, the scan time is not affected by waiting time for data reception/reflection completion.

When the standby system does not receive the tracking data from the control system, the standby system starts the next scan.

Ex.

When the control system CPU module is in the RUN state and the standby system CPU module is in the STOP state



*1 If the constant scan is used, waiting time for the constant scan generated.

In the control system, the scan time is extended by the following tracking send processing time.

Item		Description
(1) Tracking send processing	(2) Waiting for completion of tracking data reflection	After receiving a notification of the reflection completion from the standby system, the control system sends the tracking data. When a notification of reflection completion is not arrived (4), the control system does not send tracking data in the scan.
	(3) Sending tracking data	The control system sends the tracking data. After completing the send, the control system starts another END processing without waiting for a notification of receive completion from the standby system.

In the standby system, the scan time increases by the following tracking receive processing time.

Item		Description
(5) Tracking receive processing	(6) Waiting for tracking data reception	The standby system receives the tracking data from the control system. When the standby system does not receive the tracking data, the standby system starts the next scan. After receiving the tracking data, the standby system notifies the control system of the receive completion and reflects the tracking data.
	(7) Reflecting tracking data	The standby system reflects the tracking data. After the reflection completion, the standby system notifies the control system of the reflection completion and starts the next scan.

■When the mode is switched from the asynchronous tracking mode to the synchronous tracking mode

When the mode is switched from the asynchronous tracking mode to the synchronous tracking mode, the standby system receives the tracking data twice in one scan. Therefore, the scan time of the standby system is extended by the following time.

Scan time of the standby system: Standby system scan time × 2 + Control system scan time

Precautions

Operation at power-on

When the RUN/STOP/RESET switch of the CPU module of each system is set to the RUN position and both systems are powered on, the control system CPU module starts in the STOP state and switched to the RUN state after reflecting the tracking data is completed in the standby system CPU module. (The same operation is performed when one system is powered on while the other system is waiting for the start-up of the other system.)
Since the operating status of the CPU modules between the control system and the standby system do not march until the control system is switched from the STOP state to RUN state, the BACKUP LED flashes.
When the time of the initial processing and that of the initial processing (when switched to RUN) of the standby system CPU module are longer than those of the control system CPU module, the control system CPU module may not be immediately switched to the RUN state after power-on.

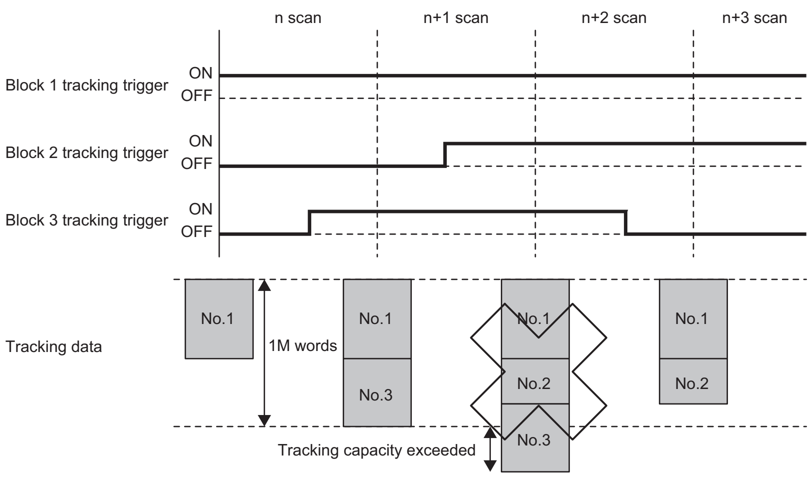
Device data used by the new control system CPU module

After system switching, the new control system CPU module starts operations by using the device/label data received in tracking transfer. The following table shows device/label data used by the new control system CPU module by tracking data receive status at system switching.

Device/label initial value	No tracking data received	Tracking data received once at least
Not set	Operation starts based on the latched device/label data.	Operation starts based on the device/label data that is transferred from the old control system. When the synchronous tracking mode is used as a tracking mode, the data in the old control system at the final scan start is used by the new control system.
Set	Operation starts based on the device/label data that is set with device/label initial values.	

Data capacity for tracking transfer

Up to 1M words of device/label data can be transferred in one scan. Set the tracking data capacity within 1M words.
If the data capacity exceeds 1M words, standard global devices/standard local devices/standard global labels/standard local labels are not transferred in the scan. In this case, check which tracking trigger turned on in the event history of the engineering tool and set the tracking data capacity within 1M words.

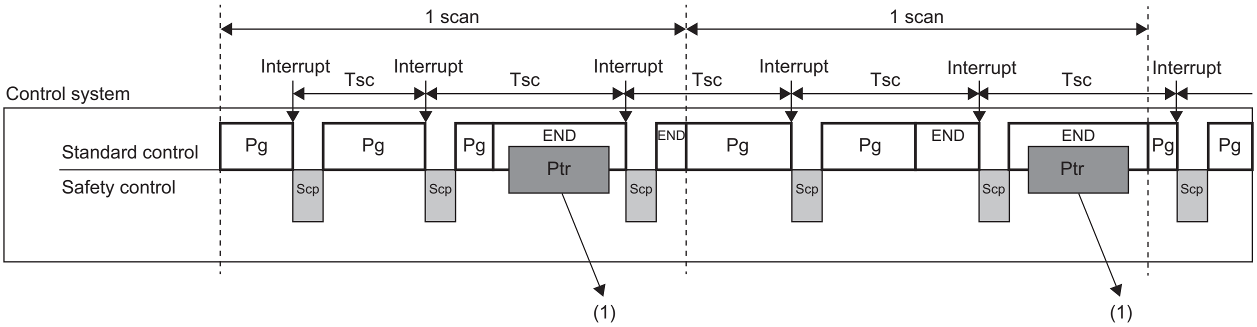


Even though the size of the tracking data to be transferred is 1M words or less, the size of the data may become larger than 1M words depending on the label type or data type to be used after the data or all the data is converted with the engineering tool. When configuring "Tracking Setting" with the CPU parameter, click the [Size Calculation] button in "Detailed setting" of "Device/Label Detailed Setting" to check if the size of the tracking data is 1M words or less and transferred in one scan. (☞ Page 729 Detailed setting)
To reduce the size of the tracking data, consider the following.

- Exclude devices/labels that are not required to continue the system operation.
- Divide the tracking data into multiple blocks to transfer the data in multiple scans.

Influence on tracking data due to safety cycle time

Safety cycle processing is performed at the interval of the specified safety cycle time, and the tracking transfer is performed during the END processing. If the safety cycle time set in the parameter is shorter than the scan time, safety cycle processing may be performed multiple times in a single scan. In such case, safety devices and safety labels immediately before the tracking transfer are transferred to the other system as tracking data.



Tsc: Safety cycle time
Pg: Standard program
Scp: Safety cycle processing
Ptr: Tracking
END: END processing
(1) Tracking transfer to the standby system is performed.

To transfer all data of safety cycle processing to the other system as tracking data, set a longer time than the scan time for the safety cycle time. (☞ Page 681 Setting method)

When data is different between the control system and the standby system

When executing the tracking transfer, files such as program (standard/safety), FB file (standard/safety), CPU parameter (standard/safety), global label setting (standard/safety), standard/safety shared label setting must be the same between the CPU modules of the both systems. (☞ Page 736 Memory Copy from Control System to Standby System)

If there is any difference, standard global devices, safety global devices, system data, and PID control instruction information are transferred.*1

*1 Standard global devices and safety global devices are transferred even when the tracking transfer settings differ between the both systems. However, devices that are included in the device range settings of the control system CPU module and not included in the device range settings of the standby system CPU module are not reflected to the standby system CPU module. (The standby system CPU module reflects values according to the device range settings of the standby system.)

If the files differ between the control system and the standby system, the following tracking transfer stops.

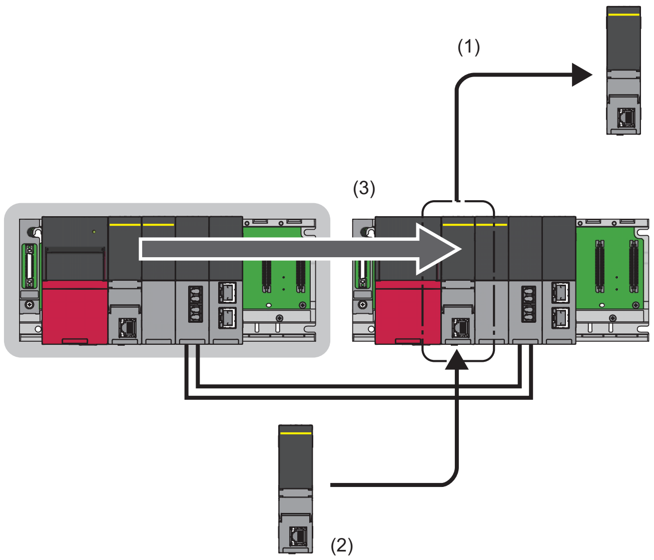
Condition	Data for which the tracking transfer is stopped
When there is a difference in any of standard CPU parameter file, standard program file, standard FB file, or standard global label setting file	• Standard local device • Standard global label • Standard local label • Standard signal flow memory
When there is a difference in any of safety CPU parameter file, safety program file, safety FB file, safety global label setting file, or standard/safety shared label setting file	• Safety local device • Safety global label • Safety local label • Standard/safety shared label • Safety signal flow memory

38.3 Memory Copy from Control System to Standby System

This function transfers data such as parameters and programs in the CPU module of the control system to the CPU module of the standby system to maintain the consistency of the memory in the two CPU modules.

Ex.

Replacement of the standby system CPU module using the memory copy



(1) Remove the standby system CPU module.
(2) Mount a new CPU module.
(3) Execute the memory copy to match the memory contents of the control system CPU module and the standby system CPU module.

The following memories are available for the memory copy.

- Program memory
- Device/label memory (copying only the file area of file register)
- Data memory (copying only the files in the system folder ($MELPRJ$))
- SD memory card (copying only the files in the system folder ($MELPRJ$))
- System memory (copying only the user management information file)

For the CPU built-in memory, the memory copy is executed regardless of whether or not there are differences between the control system and standby system. For the SD memory card, the memory copy is executed only when there are differences between the control system and standby system.

Point

For the module replacement method and maintenance procedure using the memory copy, refer to the following.

☞ Page 799 Module Replacement in a System Using the SIL2 Process CPU

Files copied by the memory copy function

The following table lists the files to be copied by the memory copy function.

○: Memory copy possible, ×: Memory copy not possible, —: Storage not possible

File type		Copy			
		CPU built-in memory			SD memory card
		Program memory	Device/label memory	Data memory	
Standard program		○	—	○	—
Standard FB file		○	—	○	—
Standard CPU parameter		—	—	○	—
System parameter		—	—	○	—
Standard module parameter		—	—	○	—
Module extension parameter		—	—	○	○
Module-specific backup parameter		—	—	×	×
Memory card parameter		—	—	—	○
Device comment		—	—	○	○
Device initial value		—	—	○	—
Standard global label setting file		—	—	○	—
Initial label value file	Initial global label value file	—	—	○	—
	Initial local label value file	—	—	○	—
File register		—	○	—	×
Event history		—	—	×	×
Device data storage file		—	—	○	×
General-purpose data		—	—	×	×
Remote password		—	—	○	○
Safety program		○	—	○	—
Safety FB file		○	—	○	—
Safety CPU parameter		—	—	○	—
Safety module parameter		—	—	○	—
Safety global label setting file		—	—	○	—
Standard/safety shared label setting file		—	—	○	—

Point

When the memory copy is executed, files are copied after the memory of the standby system that has differences is initialized. (☞ Page 740 Automatic memory initialization)

The memory copy cannot be executed to the files such as event history and those files are deleted from the memory of the standby system CPU module. Back up the files before executing the memory copy as required.

■When the security key authentication function is used

The security key of the CPU module is not copied to the standby system. Write the security key from the personal computer where the security key is registered again after completion of the memory copy.

If the memory copy is executed when the security key is written in the CPU module, program files and their security keys are copied but the security key of the CPU module is not copied. Starting up the CPU module without re-writing of the security key causes an error of the CPU module because the security keys do not match between those program files and the CPU module.

Point

If the security key of the CPU module is written in an extended SRAM cassette, the replaced CPU module can take over the security key by simply replacing the extended SRAM cassette at replacement of the CPU module. In this case, re-writing of the security key after completion of the memory copy is not required.

Execution method of memory copy

The following methods are available for memory copy.

Item	Description	Reference
Memory copy with the engineering tool	Executes the memory copy with an online operation of the engineering tool that is connected to the standby system CPU module.	Page 739 Memory copy with the engineering tool

■Check method using the special relay/special register

The execution status and completion status of memory copy can be checked using the following special relay/special register areas.

SM/SD number	Name
SM1654	Memory copy being executed
SM1655	Memory copy completion
SD988*1	Memory copy completion status (latch)
SD1654	Memory copy completion status

*1 Even after the CPU module is turned off and on or reset, the result of the memory copy executed just before this reset operation can be checked because this register area is a latch area.

For details on each special relay and special register area, refer to the following.

☞ Page 930 List of Special Relay Areas, ☞ Page 966 List of Special Register Areas

Execution of memory copy

The memory copy can be executed regardless of the operating statuses (RUN/STOP/PAUSE, stop error) of the control system and standby system CPU modules. However, execution availability differs depending on the safety operation mode.

○: Can be executed, ×: Cannot be executed

Safety operation mode of each system		Execution availability
Control system	Standby system	
TEST MODE	TEST MODE	○
	SAFETY MODE	×
SAFETY MODE	TEST MODE	○
	SAFETY MODE	×

Memory copy with the engineering tool

The following describes the execution procedure of memory copy using the engineering tool.

Execution procedure

1. Connect the engineering tool to the control system CPU module.

2. Open the "Redundant Operation" window of the engineering tool.

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

3. On the control system CPU module of memory copy source, a user with the Administrator access level of the user authentication function logs on. (Execute the memory copy only in the logged on state.)

4. Select "Memory Copy" in the "Redundant Operation" window and click [Execute]. During the memory copy, the MEMORY COPY LEDs of the redundant function modules in both systems flash (at 200ms intervals).

Control system Standby system

5. When the memory copy is completed, the MEMORY COPY LED of the control system turns off and that of the standby system turns on.

Control system Standby system

6. Turn off and on or reset the standby system CPU module. The MEMORY COPY LED of the standby system turns off.

Control system Standby system

Precautions

The following describes precautions on the memory copy function.

Automatic memory initialization

The initialization operation for each memory differs.

Memory	Initialization operation
Program memory, device/label memory	Regardless of whether or not there are differences, execute the memory copy after the memory is initialized.
Data memory	Regardless of whether or not there are differences, the "$MELPRJ$" folder is deleted from the memory and then the memory copy is executed.
SD memory card	If there are differences, the "$MELPRJ$" folder is deleted from the SD memory card and then the memory copy is executed.

When memory copy is executed, the ERROR LED of the standby system CPU module may flash. This phenomenon is not an error.*1

*1 When a stop error has been occurred in the standby system, no error is detected.

Follow the procedure to turn off and on or reset the standby system CPU module.

Also, back up the data that is not supported by the memory copy and is stored in the "$MELPRJ$" folder, such as event history, before executing the memory copy.

Restrictions

■Restrictions on memory copy execution

Do not execute the memory copy in the following cases.

- The standby system CPU module is off or being reset.
- During initial processing (when switched to RUN)
- The tracking cable has an error or is pulled out.
- An error occurs on a redundant function module.
- A system configuration in which communications via a redundant function module are unavailable is used.
- Different models of CPU module are used in the control system and standby system.
- An online operation is performed from the engineering tool.
- The systems are being switched.
- An online module change is being performed (for replacing a redundant function module or modules of the standby system).
- The memory copy has been executed on the standby system.
- The safety operation mode of the standby system is SAFETY MODE.
- Memory copy cannot be executed with the access level of the operator. (It is executed by a user other than an Administrator.)
- A communication timeout occurs in tracking communications.

■Restrictions during memory copy execution

Do not execute the following functions during memory copy execution.

- Online operation from the engineering tool
- Operation of the RUN/STOP/RESET switch
- Removal of the SD memory card
- Online module change
- Powering off or resetting the CPU modules
- Removal of tracking cables

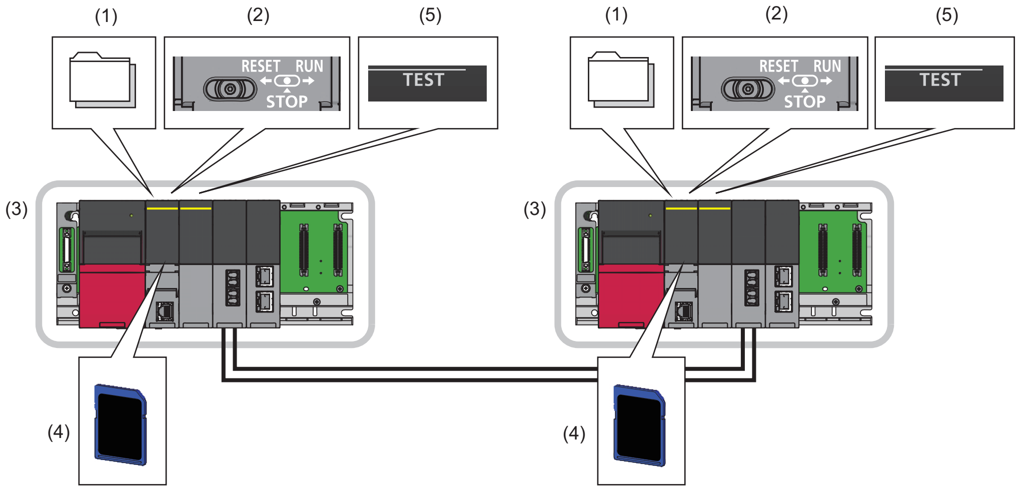
Errors during memory copy

When the memory copy is completed with an error, the MEMORY COPY LED of the control system turns off and that of the standby system flashes (at 1s intervals). In this case, the memory copy has not been normally executed on the standby system CPU module.

After checking the error code of memory copy stored in SD1654 (Memory copy completion status) and eliminating the error cause, execute the memory copy again. (☞ Page 966 List of Special Register Areas)

38.4 System Consistency Check

This function checks whether the system configurations and files in the CPU modules are the same between the control system and the standby system.



The following table lists the items to be checked in the system consistency check.

Number in figure	Item	Description	Reference
(1)	File	Parameter file, program file, and other files	Page 744 File
(2)	Operating status	Operating status of CPU module (RUN/STOP/PAUSE)	Page 745 Operating status
(3)	Mounting status of main base unit	CPU module and other modules mounted on the main base unit	Page 745 Mounting status of main base unit
(4)	SD memory card	Installing status of the SD memory card and status of the write protect switch	Page 746 SD memory card
(5)	Mode of safety control	Safety operation mode	Page 746 Safety operation mode

Execution timing

The following table shows the execution timing of the system consistency check.

Item	Execution timing
File	• When both systems are simultaneously turned on or reset • When one system is turned on or reset while waiting for the start-up of the other system • When the standby system is turned on or reset while the control system is operating • When the operating status of the CPU module is switched from STOP to RUN • At END processing • At online change completion • At system switching • When a tracking cable is reconnected
Operating status*1	• When the standby system is turned on or reset while the control system is operating • When the operating status of the CPU module is switched from STOP to RUN • At END processing • When a tracking cable is reconnected
Mounting status of main base unit	• When both systems are simultaneously turned on or reset • When one system is turned on or reset while waiting for the start-up of the other system • When the standby system is turned on or reset while the control system is operating • When a tracking cable is reconnected
SD memory card	• When both systems are simultaneously turned on or reset • When one system is turned on or reset while waiting for the start-up of the other system • When the standby system is turned on or reset while the control system is operating
Safety operation mode*1	• When the standby system is turned on or reset while the control system is operating • When the operating status of the CPU module is switched from STOP to RUN • At END processing • When a tracking cable is reconnected

*1 Checking this item can be disabled by setting "Backup Mode Setting" of "Redundant Behavior Setting". (☞ Page 752 Redundant System Operation Setting)

In the following conditions, the system consistency check is not performed.

- One of the systems has not started up (powered off, reset, or initial processing being performed).
- A stop error has occurred in the CPU module.
- The memory copy is being executed.

38

File

Whether both systems have the same files is checked.
The following table shows whether or not to perform the check on each file type.
○: Checked, ×: Not checked, —: Storage not possible

File type		Check target memory	
		Built-in memory of CPU module	SD memory card[*3]
Program[*1]		○	—
FB file[*1]		○	—
CPU parameter		○	—
System parameter		○	—
Module parameter		○	—
Module extension parameter		○	○
Module-specific backup parameter		×	×
Memory card parameter		—	○
Device comment		×	×
Device initial value		○	○
Global label setting file		○	—
Initial label value file	Initial global label value file	○	○
	Initial local label value file	○	○
File register file[*2]		○	×
Event history		×	×
Device data storage file		×	×
General-purpose data		×	×
Remote password		○	×
Safety program		○	—
Safety FB file		○	—
Safety CPU parameter		○	—
Safety module parameter		○	—
Safety global label setting file		○	—
Standard/safety shared label setting file		○	—

*1 Reserved area for online change is also checked. (☞ Page 116 Configuration of a program file)
*2 Whether a file is stored or not is checked. The file contents are not checked.
*3 The check is not performed at the point when SD memory cards are inserted during operation.

Operation of when a mismatch is detected

If a mismatch between the files is detected, a stop error occurs on the standby system CPU module.
Set the CPU modules of both systems to have the same files with either of the following methods.

- Writing the correct project to the standby system (☞ Page 702 Writing Data to the Programmable Controller)
- Executing the memory copy from the control system to the standby system for storing the same files in both systems (☞ Page 736 Memory Copy from Control System to Standby System)

Operating status

Whether the CPU modules of the control system and standby system are in the same operating status (RUN/STOP/PAUSE) is checked.

Operation of when a mismatch is detected

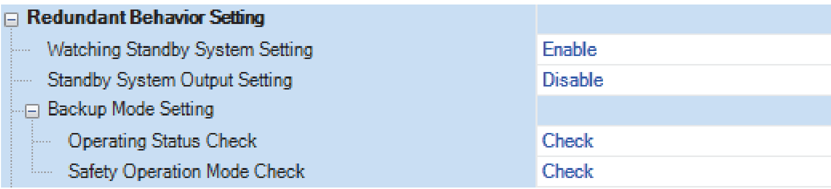
If a mismatch between the operating statuses is detected, a continuation error occurs on the standby system CPU module. The BACKUP LEDs of the redundant function modules of the control system and standby system flash because a cause of the system switching failure has occurred.

Backup mode setting

To prevent a continuation error from being detected when the operating status is changed during operation, set the system consistency check so that the consistency check on the operating status is not performed.
With this setting, a continuation error is not detected for the standby system CPU module even if the CPU modules of both systems are in different statuses.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Redundant Behavior Setting] ⇨ [Backup Mode Setting]

Window



Displayed items

Item	Description	Setting range	Default
Operating Status Check	Sets whether to perform the consistency check for the operating statuses of both systems or not.	• Do Not Check • Check	Check

Mounting status of main base unit

The system consistency check is performed for the following items.

- Models of CPU modules
- Models and types of modules mounted on each slot

Operation of when a mismatch is detected

If a mismatch between the mounting statuses of the main base units is detected, a stop error occurs on the standby system CPU module. If a mismatch between the mounting statuses of the main base units is detected when the CPU modules of both systems are simultaneously turned on or reset, a stop error occurs on the control system CPU module as well.
("Simultaneously" here means that one CPU module is started up within three seconds after the other CPU module is started.)

SD memory card

The system consistency check is performed to check the installation of the SD memory cards and the status of the write protect switch. The SD memory card type or capacity is not checked.

Point

The system checks if the SD memory card is inserted while the control system is running even in the case only the standby system is turned off and on or reset. When an SD memory card is used, removing it more than necessary is not recommended.

Operation of when a mismatch is detected

If a mismatch of the installation of the SD memory cards or the status of the write protect switch is detected, a stop error occurs on the standby system CPU module. If a mismatch is detected when the CPU modules of both systems are simultaneously turned on or reset, a stop error occurs on the control system CPU module as well.

Safety operation mode

Whether the safety operation modes (SAFETY MODE/TEST MODE) of the CPU modules of both systems are the same is checked.

Operation of when a mismatch is detected

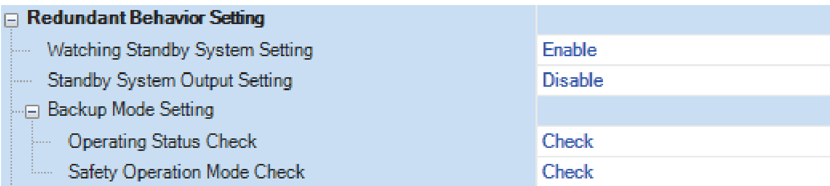
If a mismatch is detected between the safety operation modes, a continuation error occurs on the standby system CPU module.
The BACKUP LEDs of the redundant function modules of the control system and standby system flash because a cause of the system switching failure has occurred.

Backup mode setting

When "Do Not Check" is set in "Operating Status Check", a continuation error is not detected for the standby system CPU module even if the safety operation modes are different between the both systems.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Redundant Behavior Setting] ⇨ [Backup Mode Setting]

Window



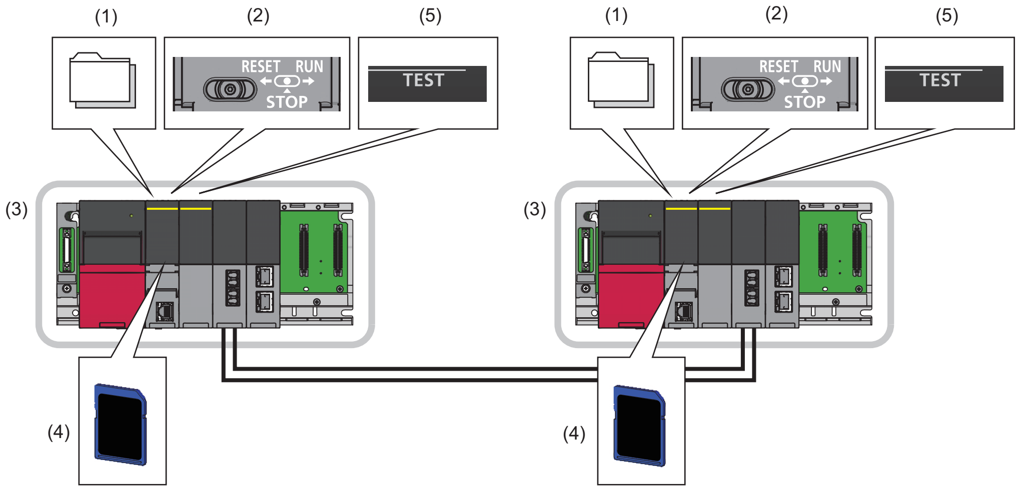
Displayed items

Item	Description	Setting range	Default
Safety Operation Mode Check	Sets whether to perform the consistency check for the safety operation modes of both systems or not.	• Do Not Check • Check	Check

38.5 Program Execution in Both Systems

This function detects an error in the external device or network of the systems (control system and standby system) by executing a program that diagnoses external devices or networks of both systems.
The program set in "Both Systems Program Executions Setting" is executed on the CPU modules of both systems.

Ex.
When an error in the external device of the standby system is notified through a continuation error



(1) The diagnostic program set as a both systems execution program is executed.
(2) An error is detected on the standby system external device by the diagnostic program.
(3) The PALERT instruction is executed, and the detailed information on a continuation error is displayed.

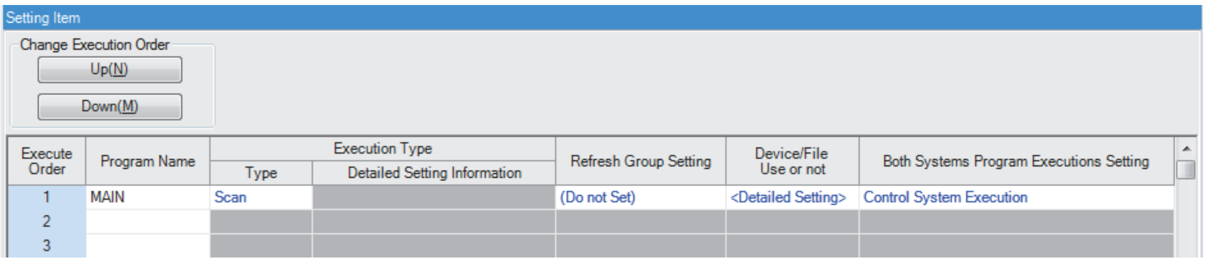
Programs whose execution type is the fixed scan execution type or event execution type and safety programs cannot be executed in both systems.

Setting for the program execution in both systems

Configure the setting for each program to be executed on both systems.

[CPU Parameter] ⇨ [Program Setting]

Window



Displayed items

Item	Description	Setting range	Default
Both Systems Program Executions Setting	Sets whether to execute a program only on the control system CPU module or on the CPU modules of both systems. • "Both Systems Executions" can be set for programs whose execution type is the initial execution type, scan execution type, or standby type program. • Only "Control System Execution" can be set for programs whose execution type is the fixed scan execution type or event execution type and safety programs.	• Control System Execution • Both Systems Executions	Control System Execution

Point

To enable the output (Y) from a standby system external device using a program executed in both systems, configure settings in "Standby System Output Setting" of the CPU parameter. (☞ Page 752 Redundant System Operation Setting)

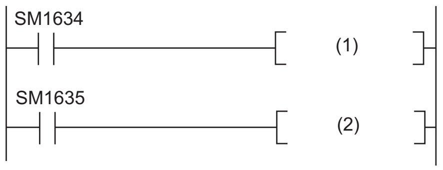
Operation of a program executed in both systems

The following table shows the operation of a program executed in both systems.

Control system/ Standby system	Operation
Control system	A program is executed according to its execution type.
Standby system	A program executed in both systems is executed according to its execution type. When an initial execution type program is set as a program executed in both systems, the program is executed in the first scan while the CPU modules are in the RUN state. When an initial execution type program is not set as a program executed in both systems, a program executed in both systems of scan execution type is executed in the first scan while the CPU modules are in the RUN state.

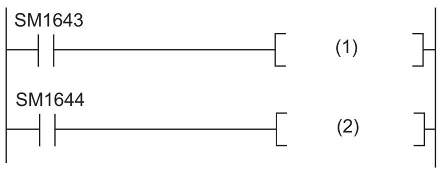
Point

- To perform different diagnostic processing in the control system and standby system by using a program executed in both systems, use the special relay. SM1634 (Control system judgment flag) turns on only in the control system and SM1635 (Standby system judgment flag) only in the standby system. The following shows a program example of when different processing is performed. For (1) and (2) in the following example, use the SET instruction or others to prevent duplication of a coil in the processing.



(1) Diagnostic processing in the control system, (2) Diagnostic processing in the standby system

- When a program executed in both systems is used, device/label values may differ between the control system and the standby system. If the systems are switched in this state, the program is started based on the different data, causing an unintended operation. The device/label values to be used can be initialized by using SM1643 (ON for only one scan after system switching (standby system to control system)) and SM1644 (ON for only one scan after system switching (control system to standby system)). For (1) and (2) in the following example, use the SET instruction or others to prevent duplication of a coil in the processing.



(1) Initialization processing (standby system → control system), (2) Initialization processing (control system → standby system)

Operation at system switching

The operation at system switching is different for a program executed in both systems. The following table shows the operation at system switching.

Item		New control system CPU module	New standby system CPU module
Program execution	Initial execution type program	When an initial execution type program has not been completed on the old control system at system switching, it is executed again from its head.	When an initial execution type program has not been completed on the old control system at system switching, it is not executed on the new standby system because system switching is performed after the program completion on the old control system.
	Scan execution type program	This type of program is executed from the step 0.	
Local device setting		This setting is in accordance with the parameter settings.	This setting is in accordance with the parameter settings.
File register setting		The file register setting before system switching is held.	The file register setting before system switching is held.
Direct access input (DX)		In the program execution after system switching, data is fetched when an instruction using the direct access input (DX) is executed.	
Direct access output (DY)		Data is not output. However, data is output when "Standby System Output Setting" of the CPU parameter is enabled and an instruction using the direct access output (DY) is executed in the program execution after system switching. (☞ Page 752 Redundant System Operation Setting)	
FROM/TO instructions		In the program execution after system switching, this instruction is executed when the instruction execution condition is satisfied.	
Instruction that requires several scans		• In the program execution after system switching, this instruction is executed when the instruction execution condition is satisfied. • When the systems are switched while the instruction is being executed, the instruction execution continues and the completion device turns on at a completion of the instruction execution.	

For the operation of programs which are not set as both systems execution programs at system switching, refer to the following.

☞ Page 713 Operation at system switching

38

Precautions

The following lists the precautions for using a program executed in both systems.

Item		Description	Reference
Program execution time		Set a program execution time of the standby system to be shorter than that of the control system.	Page 750 Program execution time
Constant scan		The constant scan function is invalid for the standby system.	—
Time required for system switching		If system switching is attempted to be performed while a program executed in both systems is being executed, the system is switched after the END instruction is completed. Thus, a time required for system switching may be extended.	—
Program execution type	Execution type change	The execution type of a program executed in both systems is not taken over at system switching. Even if the systems are switched after the execution type is changed by using a program control instruction (PSCAN(P), PSTOP(P), or POFF(P) instruction) on the old control system, the program is executed on the new control system with the execution type of the old standby system.	—
	Initial execution type	When the systems are switched while an initial execution type program set as a both systems execution program is being executed, the initial execution type program is executed twice on the new control system CPU module. The old standby system is switched to the new control system after the initial execution type program has been completed, and then the initial execution type program is executed again.	—
Interrupt disabled or enabled state		The control system and standby system have individual interrupt disabled or enabled state because the states are not transferred.	—
Tracking transfer		Do not set global devices used in a program executed in both systems as a tracking transfer target. When using labels in a program executed in both systems, use local labels. When using FBs in a program executed in both systems, use local FBs.	Page 751 Tracking transfer
Device	Timer (T)	When the control system is switched to the standby system, the current value of the timer is not updated in the first scan of the new standby system and a timeout does not occur. As a result, an error of the time required for system switching + one scan is produced at system switching.	—
	Long timer (LT) and long retentive timer (LST)	When the long timer (LT) or long retentive timer (LST) is used in the standby system, time is not measured and a timeout does not occur. After the standby system has been switched to the control system, the long timer (LT) or long retentive timer (LST) is started up. To measure time in the standby system, use the timer (T).	—
	Interrupt pointer (I)	The interrupt pointer (I) cannot be used for the standby system.	—
Restricted instructions		Some instructions have restrictions when they are used in a program executed in both systems.	Page 751 Restricted instructions
Standard function block	TIMER_□_M	When the control system is switched to the standby system, the current value of the timer is not updated in the first scan of the new standby system and a timeout does not occur. As a result, an error of the time required for system switching + one scan is produced at system switching.	—
	TP(_E), TON(_E), TOF(_E)	When these function blocks are used in the standby system, time is not measured and a timeout does not occur. After the standby system has been switched to the control system, TP(_E), TON(_E), or TOF(_E) is started up.	—
Ethernet	Socket communications	When data is sent to the Ethernet-equipped module of the standby system, the data receive processing is not performed because the Ethernet-equipped module discards the received data.	MELSEC iQ-R Ethernet User's Manual (Application)
	Communications using the fixed buffer		

Program execution time

Set a program execution time of the standby system to be shorter than that of the control system. When a program execution time of the standby system is longer than the total program execution time of the control system, the control system starts the next scan before the standby system has received tracking data, causing a continuation error. If the systems are switched in this state, the latest data may not be reflected on the new control system.
When a program execution time of the standby system cannot be shortened, set SD1662 (Tracking transfer data receive completion wait time) to extend a waiting time for tracking data receive completion of the control system. With this setting, the control system CPU module starts the next scan after the standby system has received tracking data. Even if the systems are switched, the control can be continued based on the latest data.

Tracking transfer

- Do not set global devices used in a program executed in both systems as a tracking transfer target. Due to tracking transfer, the standby system data is overwritten with that of the control system, and the standby system program may operate in an unintended way.
- When using a label in a program executed in both systems, use a local label.
- When using an FB in a program executed in both systems, use a local FB. If a global FB is used, a program execution result of the control system is transferred and written over a program execution result of the standby system, causing an unintended operation.

Restricted instructions

The following table lists the instructions that are restricted in a program executed in both systems.

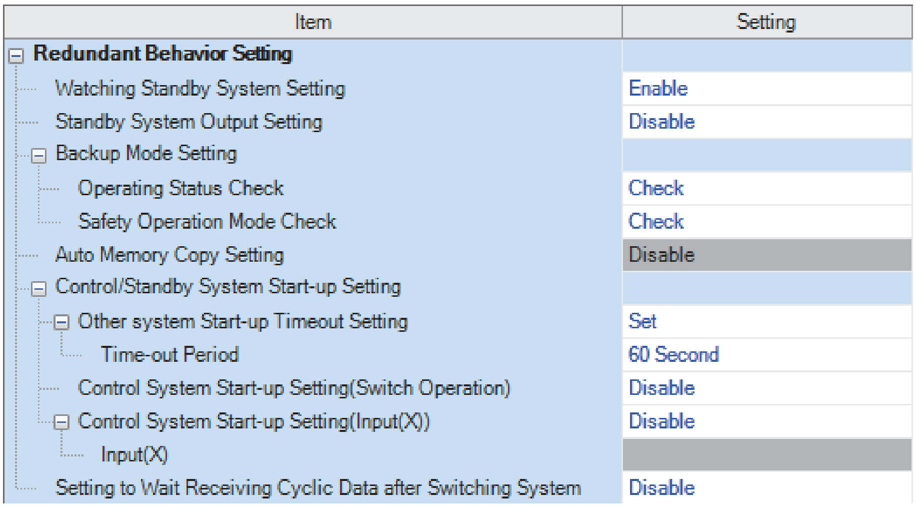
Classification	Notation	Description
Calling a subroutine program	CALL(P)	These instructions may not normally operate if a subroutine with any of a rising instruction, falling instruction, or SCJ instruction in a program which is not set as a both systems execution program is called in the standby system.
Calling a subroutine program and turning the output off	FCALL(P)	
Calling a subroutine program in the specified program file	ECALL(P)	
Calling a subroutine program in the specified program file and turning the output off	EFCALL(P)	
Calling a subroutine program with output off	XCALL	
Program control instruction	PSTOP(P)	No operation is performed if these instructions are executed on a program which is not set as a both systems execution program in the standby system.
	POFF(P)	
	PSCAN(P)	
Redundant system instruction	SP.CONTSW	No operation is performed if this instruction is executed in the standby system.
PID control instruction (Inexact differential)	S(P).PIDINIT	The PID control instruction information of the control system is transferred and written over the PID control instruction information of the standby system. When these instructions are executed in the standby system, a stop error may occur.
	S(P).PIDCONT	
	S(P).PIDSTOP	
	S(P).PIDRUN	
	S(P).PIDPRMW	
PID control instruction (Exact differential)	PIDINIT(P)	
	PIDCONT(P)	
	PIDSTOP(P)	
	PIDRUN(P)	
	PIDPRMW(P)	

38.6 Redundant System Operation Setting

Set the redundant system operation of a system using the SIL2 Process CPU in the redundant settings of the CPU parameter.

[CPU Parameter] ⇨ [Redundant System Settings]

Window



Displayed items

Item		Description	Setting range	Default
Watching Standby System Setting		Set this item not to detect continuation errors when: • A communication error with the other system has occurred or communications with the other system have been disabled. • The standby system CPU module is off or has been reset, or a stop error has occurred. This item is enabled only for the control system CPU module.	• Disable • Enable	Enable
Standby System Output Setting		Set this item to enable the output (Y) from the standby system for system inspection or adjustment in the standby system. • To diagnose devices that are connected to the standby system by using a both systems execution program, set "Enable". However, set the standby system output setting to "Disable" when common devices have been connected to both the control system and standby system. • When "Enable" is set, do not specify outputs (Y) to be used in the standby system as the data to be tracked. If such an operation is performed, the outputs (Y) are overwritten with the control system data and output from the standby system. (Even though the standby system CPU module is in the STOP state, the output (Y) received in tracking transfer is output.)	• Disable • Enable	Disable
Backup Mode Setting		☞ Page 745 Operating status, ☞ Page 746 Safety operation mode		
Control/Standby System Start-up Setting	Other system Start-up Timeout Setting	Sets a timeout time taken until communications with the other system are enabled after the own system is started up and the initial processing is completed. • When "Set" is selected, a stop error will occur in the own system unless communications are performed with the other system even though the timeout time has come. • When "Not Set" is set, the own system will wait until communications with the other system are established.	• Not Set • Set	Set
	Time-out Period		3 to 1800s (in increments of 1s)	60s
	Control System Start-up Setting (Switch Operation)	Set this item to enable the operation to be started in the control system with the switch operation (RUN → STOP → RUN) while the own system that has started is waiting for the start-up of the other system.	• Disable • Enable	Disable
	Control System Start-up Setting (Input (X))	Set this item to enable the operation to be started in the control system with the contact input (X) while the own system that has started is waiting for the start-up of the other system.	• Disable • Enable	Disable
	Input (X)		X0 to X2FFF	—

Item	Description	Setting range	Default
Setting to Wait Receiving Cyclic Data after Switching System*1	Set this item to wait for the execution of the sequence program until all CC-Link IE Field Network modules on the main base unit receive the latest cyclic data after system switching. ☞ Page 754 Setting to wait cyclic data receive after system switching	• Disable • Enable	Disable

*1 Before enabling the setting to wait cyclic data receive after system switching, check the versions of the CPU module, CC-Link IE Field Network module, and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

Standby system output setting

Output timing

When "Standby System Output Setting" has been set to "Enable", the output timing from the standby system is at the completion of the END processing or depends on the refresh group setting or refresh settings of each module. (☞ Page 72 Group setting for refresh)

Therefore, when a program set in the refresh group setting is a program executed in the control system, the I/O refresh is not performed while the module is running because the program is not executed in the standby system. (When the CPU module is in the STOP/PAUSE state, the I/O refresh is performed at the timing of the END processing.)

To execute a program set in the refresh group setting in the standby system, set "Both Systems Program Executions Setting". (☞ Page 747 Program Execution in Both Systems)

Operation at system switching

The operation performed at the system switching differs when "Standby System Output Setting" is set to "Enable". The following table shows the operation performed when "Standby System Output Setting" has been set to "Enable".

Item	New control system CPU module	New standby system CPU module
Output (Y)	The status of the old standby system is held and output refresh is performed.	The status of the old control system is held and output refresh is performed.
Direct access output (DY)	In the program execution after system switching, data is output when an instruction using the direct access output (DY) is executed.	■For program executed in both systems In the program execution after system switching, data is output when an instruction using the direct access output (DY) is executed. ■For program executed in the control system No operation is performed because the program does not operate.

For the program execution in both systems, refer to the following.

☞ Page 747 Program Execution in Both Systems

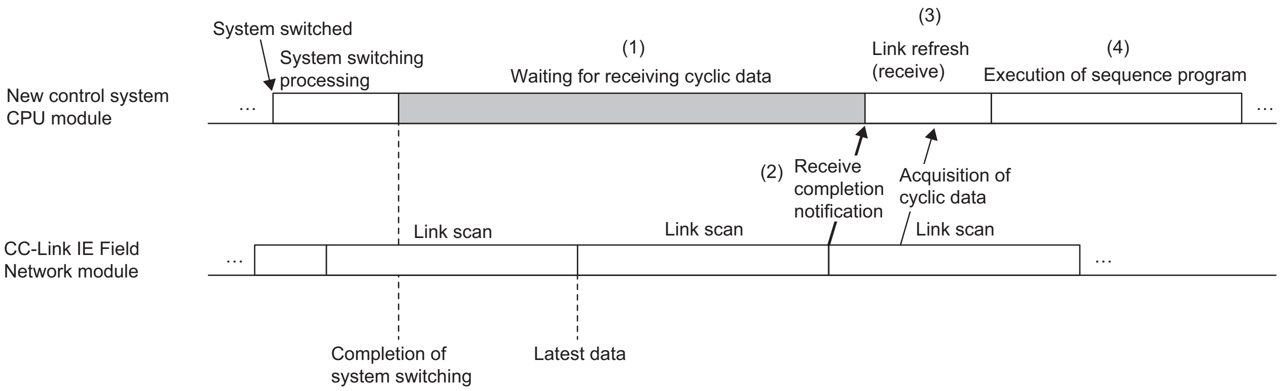
Setting to wait cyclic data receive after system switching

When system switching occurs in the redundant line structure of CC-Link IE Field Network, set this item to start the execution of the program in the new control system with new cyclic data obtained after system switching.

Restriction

Before enabling the setting to wait cyclic data receive after system switching, check the versions of the CPU module, CC-Link IE Field Network module, and engineering tool used. (☞ Page 1139 Added and Enhanced Functions)

When this setting is enabled, the execution of the program is suspended until new cyclic data is received in the new control system after system switching.



(1) After system switching has been completed, the CPU module stands by until it is notified of the completion of the cyclic data receipt after system switching by all CC-Link IE Field Network modules on the main base unit.
(2) When the CC-Link IE Field Network modules receive cyclic data from the remote I/O station after system switching, the CC-Link IE Field Network modules notify the CPU module of the completion of the cyclic data receipt.
(3) When notified of the completion of the cyclic data receipt by the CC-Link IE Field Network modules, the CPU module executes link refresh (receive).
(4) The CPU module executes the sequence program using the new cyclic data.

- When this setting is enabled, the extended time from the completion of system switching to the first output is added to the cyclic data receipt waiting time. (☞ Page 1086 Delay time until initial output after system switching (Tjo))
- This setting is executed in the new control system after system switching. This setting operates when the operating status of the CPU module in the new control system is RUN, STOP, or PAUSE. (This setting does not operate when a stop error has occurred in the CPU module in the new control system.)
- This setting operates when the link scan mode of the CC-Link IE Field Network is set to "Sequence Scan Asynchronous" or "Constant Link Scan". This setting does not operate when the link scan mode of the CC-Link IE Field Network is set to "Sequence Scan Synchronous Setting". (📖 MELSEC iQ-R CC-Link IE Field Network User's Manual (Application))
- When cyclic data cannot be received within the timeout time due to disconnection of the network cable or other causes while cyclic data receipt is waited for, waiting for cyclic data receipt is stopped, and the sequence program is executed. Occurrence of a timeout can be checked in SM1756 (Wait timeout for receiving cyclic data after system switching) and SD1756 (Module information on wait timeout for receiving cyclic data after system switching).

Precautions

- Scan time monitoring with the watchdog timer is interrupted while cyclic data receipt is waited for. Thus, no error is detected even if the scan time monitoring time has elapsed while cyclic data receipt is waited for.
- Constant scan is disabled while cyclic data receipt is waited for. Thus, no error is detected even if the constant scan setting time has elapsed. Constant scan is enabled after waiting for cyclic data receipt has been completed, and the sequence program is executed.
- Because device/label access service processing is not accepted while the cyclic data receipt is waited for, set the communication timeout time with external devices in consideration of the cyclic data receipt waiting time. (☞ Page 1087 Waiting time for cyclic data receive after system switching (Twcyc))

38.7 Redundant Function Module Communication Test

The hardware of the redundant function module is checked for an error when its communications are unstable.
The following table shows the test items included in the module communication test.

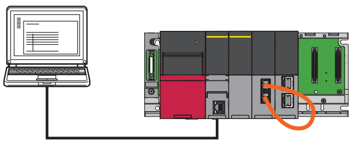
Test item	Description
Internal selfloopback test	Checks whether the communication function of the redundant function module normally operates.
External selfloopback test	Checks whether communications can be normally performed with a tracking cable that connects the IN connector and OUT connector of the redundant function modules.

Execution procedure of the module communication test

1. Connect the engineering tool directly to the standby system CPU module.

2. Set the CPU module operating status to the STOP state.

3. Connect the IN and OUT connectors of the standby system redundant function module with the tracking cable. (A cable disconnection error is detected in the control system.)



4. Open the "Redundant Operation" window of the engineering tool.

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

5. Click the [Execute Test] button of "Module Communication Test".

- LED status of the redundant function module during the module communication test

Status	RUN LED	ERR LED
Module communication test in execution	Flashing	Off
Completed successfully	On	Off
Completed with an error	On	On

6. When the test is completed with an error, take actions according to "Corrective Action" in the test result window.

7. When the test is completed successfully, connect the control system and standby system with the tracking cable. (☞ Page 696 Wiring the redundant function modules)

8. Click the [Close] button in the "Redundant Operation" window to exit the module communication test.

9. Set the CPU module operating status to the RUN state.

Point

A tracking communication error is detected when a tracking cable is disconnected.

Precautions

- Always connect the IN and OUT connectors of the redundant function module with the tracking cable before performing the module communication test.
- Perform the module communication test on the standby system CPU module. Performing the test on the control system CPU module may cause an unintended operation.

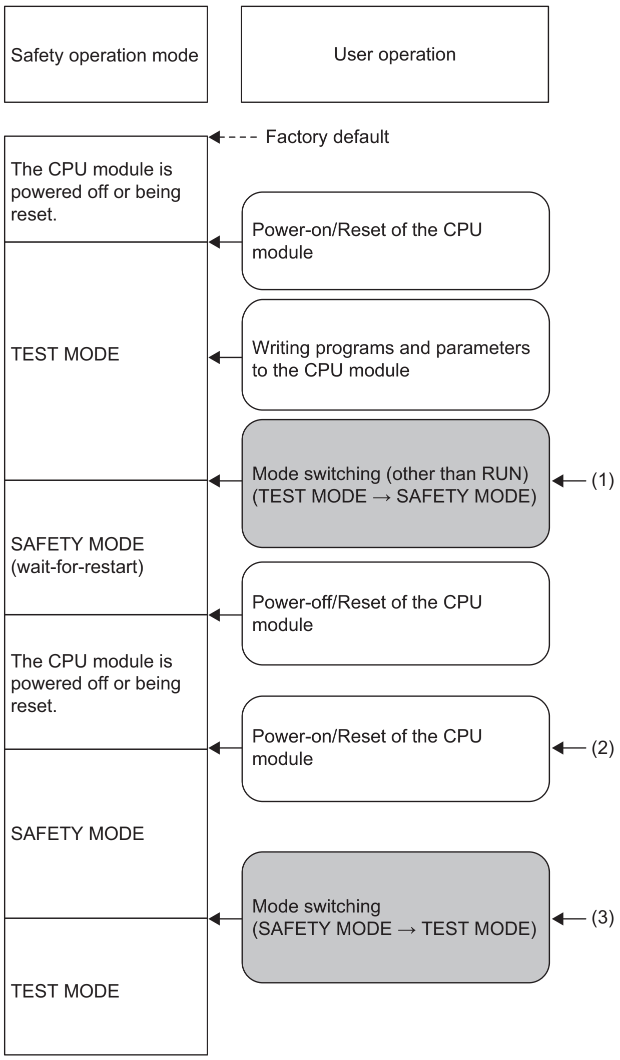
38.8 Switching Safety Operation Mode

This section describes how to switch the safety operation mode.

Safety operation mode transition timing

The following table shows the timing at which safety operation mode switches.

Safety operation mode switching direction	Safety operation mode switch timing
TEST MODE → SAFETY MODE	When the CPU module is powered off and on or is reset after the safety operation mode switching operation
SAFETY MODE → TEST MODE	At the safety operation mode switching operation



(1) After switching from TEST MODE to SAFETY MODE, the safety operation mode will be SAFETY MODE (wait-for-restart). (The status will be wait-for-restart until the CPU module is powered off or is reset.) An error occurs if the SIL2 Process CPU is switched to the RUN state when in SAFETY MODE (wait-for-restart).

(2) After switching the safety operation mode (TEST MODE → SAFETY MODE), the mode will switch to SAFETY MODE when the CPU module is powered off and on or is reset.

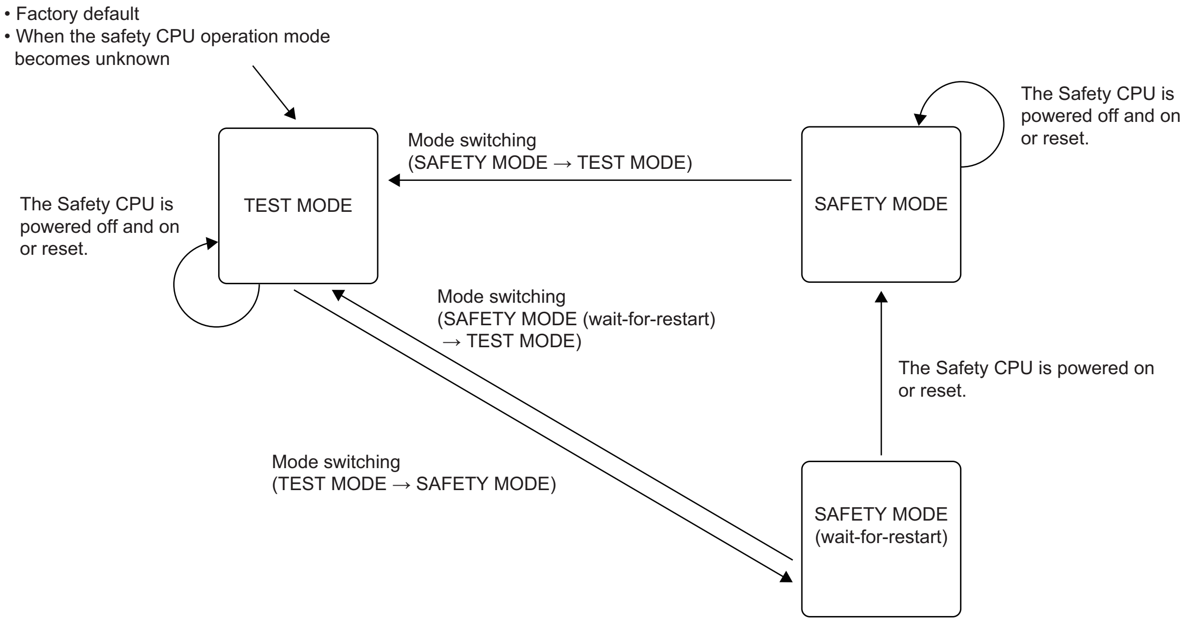
(3) After switching the safety operation mode (SAFETY MODE → TEST MODE), the mode will switch to TEST MODE.

Point

The SIL2 Process CPU retains the safety operation mode status even when the CPU module is powered off or is reset.

Safety operation mode state transition

The SIL2 Process CPU retains the safety operation mode even in the event of a power failure, and therefore the current safety operation mode state will remain unchanged even if the CPU module is powered off and on or is reset. The SIL2 function module does not retain the safety operation mode in the event of a power failure, and the safety operation mode will be the same as that of the SIL2 Process CPU when the CPU module is powered off and on or is reset.



Safety operation mode status	State transition by operation		
	Power off → on or reset	Safety operation mode switching operation	
		TEST MODE → SAFETY MODE	SAFETY MODE → TEST MODE
TEST MODE	→ TEST MODE	→ SAFETY MODE (wait-for-restart)	—
SAFETY MODE	→ SAFETY MODE	—	→ TEST MODE
SAFETY MODE (wait-for-restart)	→ SAFETY MODE	—	→ TEST MODE

Point

If [Online] ⇨ [User Authentication] ⇨ [Initialization of all PLC Data] is performed, the safety operation mode will revert to the default mode (TEST MODE). All programmable controller information can be initialized regardless of the safety operation mode.

Safety operation mode switching conditions

The following table lists the conditions under which the safety operation mode can be changed.

Safety operation mode switching direction	Condition
TEST MODE → SAFETY MODE	The user who operates the mode switching has "Developers" access level or higher, and is currently logged on.
	The safety operation mode is currently set to the TEST MODE.
	The SIL2 Process CPU operating status is other than RUN.
	The SIL2 Process CPU is not in the process of changing operating status.
	The following files held by the engineering tool, SIL2 Process CPU, and SIL2 function module match. • Safety CPU parameter • Safety module parameter • Safety program • Safety FB file • Safety global label setting file • Standard/safety shared label setting file
	The following files are not being written from another engineering tool. • Safety CPU parameter • Safety module parameter • Safety program • Safety FB file • Safety global label setting file • Standard/safety shared label setting file
SAFETY MODE → TEST MODE	The user who operates the mode switching has "Developers" access level or higher, and is currently logged on.
	The safety operation mode is currently set to SAFETY MODE or SAFETY MODE (wait-for-restart).
	The SIL2 Process CPU is not in the process of changing operating status.

How to switch the safety operation mode

The safety operation mode can be switched using the engineering tool.

[Online] ⇨ [Safety PLC Operation] ⇨ [Switch Safety Operation Mode]

The execution target of mode switching can be specified from the following options.

Execution target specification	Operation
Currently Specified Station	Switches the safety operation mode for the CPU module specified in the transfer setup.
Specify Both Systems	Switches the safety operation modes for the CPU modules of both systems.

For details, refer to the following.

GX Works3 Operating Manual

38.9 Safety Diagnostic Function

The following table lists the safety-specific diagnostic functions among the self-diagnostic functions for the SIL2 Process CPU.

Item[*1]		Description	Diagnostic timing	Error code
Memory diagnosis	RAM diagnosis	Detects errors occurring at program memory, device memory, and memory used by the system.	• At powering off and on • At reset • At END processing	3C20H, 3C21H, 3C22H, 3C2FH, 3C33H
	F/W diagnosis	Diagnoses whether the firmware stored in the ROM is corrupt.	• At powering off and on • At reset	3C10H
Program diagnosis	Operation circuit diagnosis	Detects errors at operation circuits used to operate safety programs.	• At powering off and on • At reset • At END processing	3C15H
	File verification	Detects errors in stored safety programs and safety parameters.	• At powering off and on • At reset • At END processing	2180H, 36E1H, 3C33H, 3C34H
		Diagnoses whether safety programs and safety parameters stored in both the SIL2 Process CPU and SIL2 function module are the same.	• At powering off and on • At reset	2180H, 3640H
Operation result verification	Output data verification	Verifies output results calculated in the respective SIL2 Process CPU and SIL2 function module.	At safety cycle processing	3E60H
System diagnosis	Time monitoring	Monitors the execution status of the respective SIL2 Process CPU and SIL2 function module processing.	Always	3E61H
	Microcomputer diagnosis	Detects microcomputer internal register errors.	• At powering off and on • At reset • At END processing	3C00H, 3C01H
Power supply voltage monitoring	Power supply voltage monitoring	Detects voltage errors to prevent operation with voltage outside the guaranteed operating range.	Always	— (Performs a shutdown.)
	Power supply voltage monitoring/shutdown circuit diagnosis	Monitors whether the circuit monitoring the power supply voltage and circuit used to perform shutdown are functioning properly.	• At powering off and on • At reset • At END processing	3E10H, 3E11H
Reset circuit monitoring	Reset circuit monitoring diagnosis	Diagnoses whether the reset has been performed correctly.	• At powering off and on • At reset	3E12H
Clock stop detection	Clock diagnosis	Detects clock errors (frequency errors).	• At powering off and on • At reset • At END processing	3C16H
CRC calculation circuit diagnosis	CRC calculation circuit diagnosis	Diagnoses whether the CRC calculation circuit is capable of correctly calculating CRC.	• At powering off and on • At reset • At END processing	3E01H
Module	Module OS diagnosis	Verifies whether the SIL2 Process CPU and SIL2 function module are functioning normally without running out of control.	• At powering off and on • At reset • At END processing	3C11H, 3C17H

*1 Self-diagnostics is performed in both the SIL2 Process CPU and SIL2 function module.

38.10 Identification Check for Safety Data

This function checks if the project data created using the engineering tool and the data in the SIL2 Process CPU are the same, and confirms that the program executed in SAFETY MODE is the one written by the user.

This function allows the user to inspect whether the files in the engineering tool and the SIL2 Process CPU match or not.

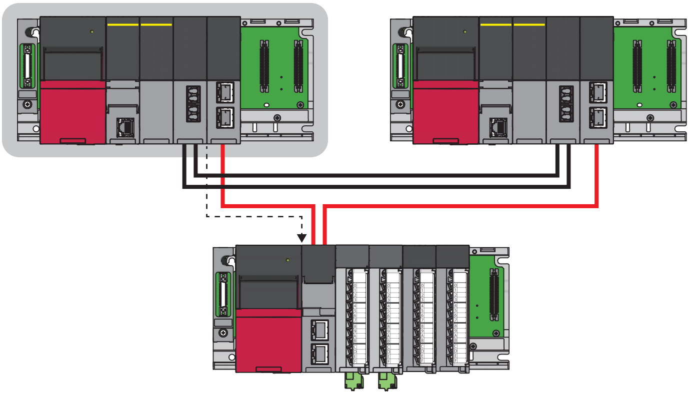
[Online] ⇨ [Safety PLC Operation] ⇨ [Check Safety Data Identity]

For details, refer to the following.

GX Works3 Operating Manual

38.11 Safety Communication Function

This function communicates data between the SIL2 Process CPU of the control system and modules supporting safety functions using safety protocols. Safety communication processing is not performed in the SIL2 Process CPU of the standby system.



Point

The SIL2 Process CPU performs safety communications between safety stations using the following network.

- CC-Link IE Field Network (safety communication function)

For details on the function and settings required for safety communications, refer to the following.

MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

Safety I/O hold time

The safety I/O hold time (Tioh) is the time until safety communications are disconnected (the safety output is turned off) after detection of a timeout for the safety refresh monitoring time of safety communications.

Safety I/O hold time setting

The safety I/O hold status of each safety connection can be checked with SA\SD1600 to SA\SD1663 (safety I/O hold status of each safety connection).

To prevent the safety communications from disconnecting at a system switching, configure the following setting.

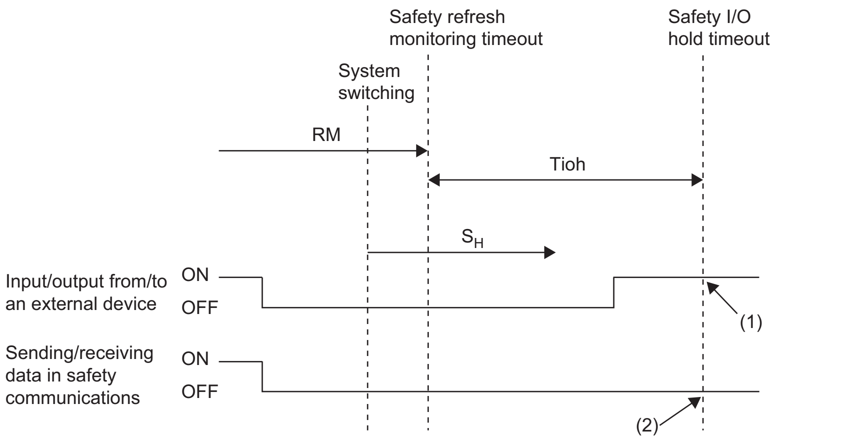
Safety I/O hold time (Tioh) > Safety data holding time at a system switching (S_H)

For the safety data holding time at a system switching, refer to the following.

MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

The following figures show the relationship between the input/output of the external device and sending/receiving data in safety communications.

- When safety communications are not recovered within the safety I/O hold time



RM: Safety refresh monitoring time

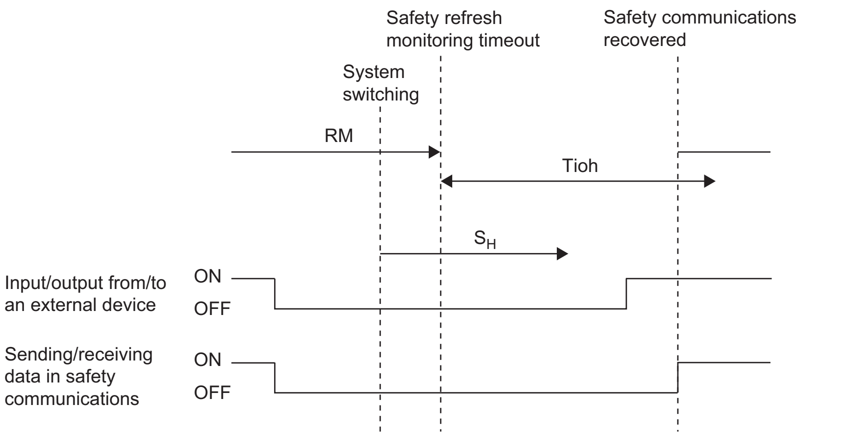
Tioh: Safety I/O hold time

S_H: Safety data holding time at a system switching

(1) The safety communications are disconnected. For the input values from the external device and output values to the external device, refer to the manual for each I/O module.

(2) A safety communication error occurs, and the safety station interlock is triggered.

- When the safety communications are recovered within the safety I/O hold time



RM: Safety refresh monitoring time

Tioh: Safety I/O hold time

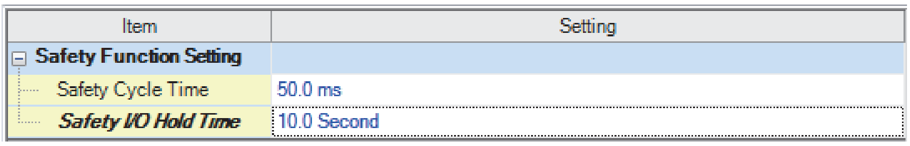
S_H: Safety data holding time at a system switching

Setting method

The following shows the setting method of the safety I/O hold time.

[CPU parameter] ⇨ [Safety Function Setting]

Window



Item	Setting
Safety Function Setting	
Safety Cycle Time	50.0 ms
Safety I/O Hold Time	10.0 Second

Displayed items

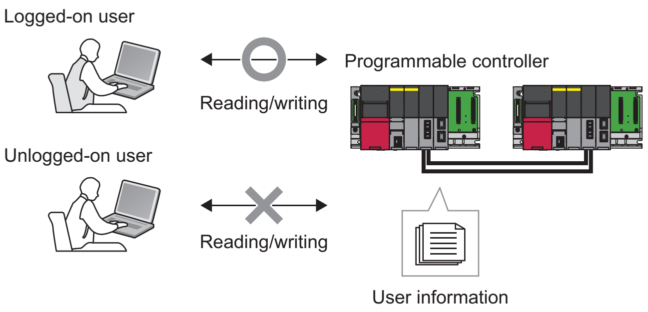
Item	Description	Setting range	Default
Safety I/O Hold Time	Sets the safety I/O hold time. If safety communications recovered successfully within the set safety I/O hold time, processing continues.	0.0 to 180.0s (in increments of 0.1s)	10.0s

Point

Note that the time until a timeout of safety communications increases in length by the amount of the setting value of the safety I/O hold time.

38.12 User Authentication Function of CPU Modules

This function prevents an unauthorized access to data (such as a program or parameters) written to a CPU module.
Operations are restricted by registering a user name/password for a CPU module.
To access a CPU module, logging on with the registered user information (user name/password) is required.



For details on the user authentication function of CPU modules, refer to the following.
GX Works3 Operating Manual

Point

- The user information registered in a CPU module needs to match with the user information of a project.
- For RnPSFCPUs (firmware version 12 or later), enhanced measures to reduce vulnerability are taken and partial changes are made to the following functions.
  · Functions and operations that require user authentication
  · Writing user information to a CPU module
  · Logging on to a CPU module

The enhanced vulnerability-measures are taken for GX Works3 (version 1.105K or later) as well.
Communications with the vulnerability-measures taken are performed by using an RnPSFCPU (firmware version 12 or later) and GX Works3 (version 1.105K or later) and enabling the setting to communicate only with GX Works3 with the vulnerability-measures enhanced version.
However, when that setting is enabled, some functions of the intelligent function module that read files, such as the user authentication and recording functions of the GOT sequence program monitor (circuit display), may become unavailable.
To use the GOT sequence program monitor (circuit display) or the intelligent function module that reads out files, disable the above-mentioned setting.
For details on the user authentication function, refer to the following.
GX Works3 Operating Manual

38.13 Online Change

When the online change (ladder block) is performed on the CPU module in one system, the change is also reflected on the CPU module in the other system.

Precautions

During an online change, avoid the following conditions.

- Changing the operating status of the CPU module from STOP (PAUSE) to RUN
- System switching
- Disconnection of the tracking cable
- Power-off or reset of the CPU module in the control system or standby system

When any of the above conditions is satisfied during an online change, a file mismatch is detected in the system consistency check or the online change fails.

■Action for when an online change has failed

If an online change has failed, take the following actions.

1. Get the CPU module ready for an online change and execute the online change with the same data as before the failure.

- If the online change succeeds, the action is completed.
- If the online change fails again, take the following actions.

2. Verify the data in the control system CPU module with the programmable controller from the engineering tool, and check whether the online change has been completed successfully.

- When the verification result is mismatched, the online change has not been completed successfully. Connect the engineering tool to the control system CPU module, execute an online change only to the control system, and proceed to step 3.
- When the verification result is matched but the program memory transfer window was not displayed at the online change, the possible cause is that data transfer to the program memory has not been completed successfully. Batch-write files only to the control system, and proceed to step 3.
- When the verification result is matched and the program memory transfer window is displayed (the online change is completed successfully) during online change, proceed to step 3.

3. Execute the memory copy from the control system to the standby system. When the memory copy has failed, power off and on or reset the standby system CPU module and execute the memory copy again. (☞ Page 736 Memory Copy from Control System to Standby System)

4. Power off and on or reset the standby system CPU module.

38.14 Constant Scan

This section describes the precautions for setting the constant scan in the system using the SIL2 Process CPU.

Increase in scan time

In the standby system, when the CPU module is powered off, a hardware failure has occurred, or a tracking cable has a failure, the scan time will increase in the control system.

When setting the constant scan, take one of the following measures.

- Set the setting time of the constant scan by adding the increase in the scan time when an error occurs. (☞ Page 1078 Increase in the scan time due to tracking transfer)
- When a failure has occurred and a continuation error occurs due to the excess of constant scan time, clear the error. (☞ Page 773 Error clear)

For program executed in both systems

The constant scan function is invalid for the standby system.

38.15 Remote Operation

In a system using the SIL2 Process CPU, the behavior of a remote operation (with the engineering tool) differs for each operation.

Operation	Behavior
Remote RUN*1/Remote STOP/Remote PAUSE	The CPU module operating status of a system specified in the transfer setup of the engineering tool or both systems can be changed.
Remote RESET	The CPU modules of both systems can be reset by performing the remote RESET operation on the control system CPU module. Only the standby system CPU module can be reset by performing the remote RESET operation on the standby system CPU module.

*1 When the safety operation mode is SAFETY MODE, the safety global devices, safety global labels, safety local labels, and standard/safety shared labels are not cleared.

Remote RUN/Remote STOP/Remote PAUSE

The CPU module operating status of a system specified in the transfer setup of the engineering tool or both systems can be changed.

Remote operation performed on the system specified in the engineering tool

When the following item is selected in "Execution Target", the remote operation is performed only on the CPU module of a system specified in the transfer setup of the engineering tool.

- Currently Specified Station

[Online] ⇨ [Remote Operation]

Remote operation performed on both systems

When the following item is selected in "Specify Execution Target", the remote operation is performed on the CPU modules of both systems.

- Specify Both Systems

[Online] ⇨ [Remote Operation]

Point

When "Specify Redundant CPU" is set to "Not Specified" in the transfer setup, select "All Stations Specified" or "Specified Group" to perform a remote operation on the CPU modules of both systems.

Remote RESET

The CPU modules of both systems can be reset by performing the remote RESET operation on the control system CPU module. Only the standby system CPU module can be reset by performing the remote RESET operation on the standby system CPU module.
Select "Control System" or "Standby System" in the transfer setup, and select "Currently Specified Station/Specify Both Systems" in "Specify Execution Target" of "Remote Operation".

Precautions

- When the control system CPU module is in the STOP state and the standby system CPU module is in the RUN state, performing the remote RESET operation on the control system CPU module causes system switching. To prevent system switching in the remote RESET operation, perform the remote RESET operation after setting both of the CPU modules to the STOP state.
- When a remote operation is performed on the CPU module of the control system or standby system through another path, the standby system CPU module cannot be reset by performing a remote RESET operation on the control system CPU module. To perform a remote RESET operation on the CPU module of the control system or standby system, use the path that was used to perform the remote operation on the standby system CPU module. Cancel the remote operation first and perform the remote RESET operation.
- When performing a remote RESET operation on both systems, select "Control System" in "Specify Redundant CPU". When "Not Specified" is selected, an error may occur depending on the timing of the operating status change of each CPU module.

[Online] ⇨ [Current Connection Destination] ⇨ [Specify Redundant CPU]

Precautions

The following describes the precautions on the remote operation in a system using the SIL2 Process CPU.

- When the control system or standby system is in the initial processing (the READY LED is flashing), do not perform the remote operation of "Specify Both Systems". If such an operation is performed, both systems may be recognized as being mismatched in the system consistency check because the operating statuses may be mismatched.

38.16 Device/Label Memory Area Setting

The capacity of each area on the device/label memory can be specified.

Default capacity

The following table lists the default capacity of each area.

Item	R08PSFCPU	R16PSFCPU	R32PSFCPU	R120PSFCPU
Standard device area	40K words	40K words	40K words	40K words
Safety device area	20K words	20K words	20K words	20K words
Safety label area	20K words	20K words	20K words	20K words
Safety local device area	0K word	0K word	0K word	0K word
Standard/safety shared label area	10K words	10K words	10K words	10K words
Standard label area	40K words	50K words	90K words	110K words
Standard latch label area	2K words	2K words	4K words	4K words
Standard local device area	0K word	0K word	0K word	0K word
File storage area	457K words	713K words	969K words	1481K words

Point

The standard local device area capacity obtained by subtracting the total capacity of the standard device area, standard label area, standard latch label area, file storage area, and safety device/safety label area capacity, and the total standard/safety shared label area from the total device/label memory is set. Note, however, that even if the total capacity of the standard device area, standard label area, and safety device/safety label area capacity is smaller than the following, the remaining capacity cannot be assigned to the standard local device area. (The remaining area will be an unused area.)

- R08PSFCPU: 50K words
- R16PSFCPU: 60K words
- R32PSFCPU: 70K words
- R120PSFCPU: 90K words

Point

The safety local device area capacity obtained by subtracting the total capacity of the safety device area and safety label area from the safety device/label memory capacity is set.

Setting range of capacity of each area

The following tables list the setting range of the capacity of each area on the device/label memory.*1

*1 The remaining capacity of other areas is automatically set as the capacity of the standard local device area and safety local device area. (☞ Page 767 Default capacity)

R08PSFCPU

Area	Setting range of capacity of each area		
	Without an extended SRAM cassette	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 588K words	2 to 1612K words	2 to 4684K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 586K words	0 to 1610K words	0 to 4682K words
Standard latch label area	0 to 544K words	0 to 1568K words	0 to 4640K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 544K words	0 to 1568K words	0 to 4640K words

R16PSFCPU

Area	Setting range of capacity of each area		
	Without an extended SRAM cassette	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 854K words	2 to 1878K words	2 to 4950K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 852K words	0 to 1876K words	0 to 4948K words
Standard latch label area	0 to 800K words	0 to 1824K words	0 to 4896K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 800K words	0 to 1824K words	0 to 4896K words

R32PSFCPU

Area	Setting range of capacity of each area		
	Without an extended SRAM cassette	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 1152K words	2 to 2176K words	2 to 5248K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 1150K words	0 to 2174K words	0 to 5246K words
Standard latch label area	0 to 1088K words	0 to 2112K words	0 to 5184K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 1088K words	0 to 2112K words	0 to 5184K words

R120PSFCPU

Area	Setting range of capacity of each area		
	Without an extended SRAM cassette	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 1684K words	2 to 2708K words	2 to 5780K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 1682K words	0 to 2706K words	0 to 5778K words
Standard latch label area	0 to 1600K words	0 to 2624K words	0 to 5696K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 1600K words	0 to 2624K words	0 to 5696K words

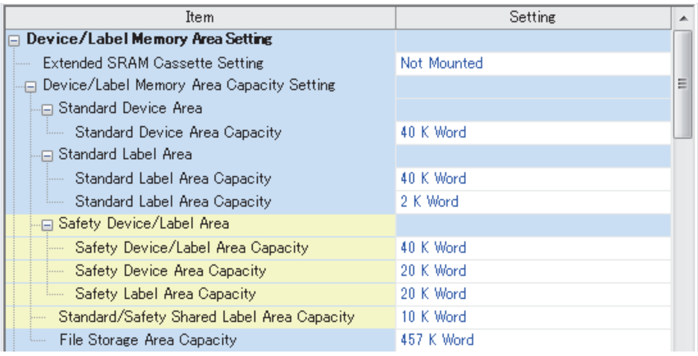
Setting method

The capacity in each data area of the device/label memory can be changed.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Setting]

Operating procedure

"Device/Label Memory Area Setting" window



1. Set whether to use an extended SRAM cassette or not in "Extended SRAM Cassette Setting".
2. Set the capacity of each area in "Device/Label Memory Area Capacity Setting".

Displayed items

Item			Description	Setting range	Default
Extended SRAM Cassette Setting			Sets the capacity of the extended SRAM cassette if used.	• Not Mounted • 2MB*1 • 8MB	Not Mounted
Device/Label Memory Area Capacity Setting	Standard Device Area	Standard Device Area Capacity	Sets the capacity of the device area used for standard global devices.	Page 767 Setting range of capacity of each area	Page 767 Default capacity
	Standard Label Area	Standard Label Area Capacity	Sets the capacity of the label area used for standard global labels and standard local labels.		
		Standard Latch Label Area Capacity	Sets the capacity of the standard latch label area used for latch-type labels.		
	Safety Device/ Label Area	Safety Device/ Label Area Capacity	Sets the total capacity for the safety device area and safety label area.		
		Safety Device Area Capacity	Sets the capacity of the device area used for safety global devices.		
		Safety Label Area Capacity	Sets the capacity of the label area used for safety global labels and safety local labels.		
	Standard/Safety Shared Label Area Capacity		Sets the capacity of the label area used for standard/safety shared labels.		
	File Storage Area Capacity		Sets the capacity of the file storage area used to store files, such as file register files.		

*1 Check the version of the engineering tool. (Page 1139 Added and Enhanced Functions)

Point

Note that the total of the capacity of each area (including the capacity of the local device area/safety local device area) and safety special relay/safety special register area capacity should not exceed the capacity of the device/label memory. (MELSEC iQ-R CPU Module User's Manual (Startup))

Standard device area setting range

The number of points of each device used in standard programs and capacity in which the total number of device points can be stored are set. Set the total number of device points within the device area range.

Type	Device name	Symbol	Range of use*1	Increment for setting
Bit	Input	X	X0 to X2FFF	—
	Output	Y	Y0 to Y2FFF	—
	Internal relay	M	M0 to M94674943	64 points
	Link relay	B	B0 to B5A49FFF	64 points
	Annunciator	F	F0 to F32767	64 points
	Link special relay	SB	SB0 to SB5A49FFF	64 points
	Edge relay	V	V0 to V32767	64 points
	Latch relay	L	L0 to L32767	64 points
Word	Timer	T	T0 to T5259711	32 points
	Retentive timer	ST	ST0 to ST5259711	32 points
	Long timer	LT	LT0 to LT1479295	1 point
	Long retentive timer	LST	LST0 to LST1479295	1 point
	Counter	C	C0 to C5259711	32 points
	Long counter	LC	LC0 to LC2784543	32 points
	Data register	D	D0 to D5917183	4 points
	Link register	W	W0 to W5A49FF	4 points
	Link special register	SW	SW0 to SW5A49FF	4 points

*1 This range is the maximum when the R120PSFCPU with an extended SRAM cassette (8MB) (NZ2MC-8MBSE) is used. The number of points varies depending on the model of the SIL2 Process CPU and whether to use an extended SRAM cassette.

Safety device area setting range

The number of points of each safety device used in safety programs and capacity in which the total number of safety device points can be stored are set. Set the total number of safety device points within the safety device area range.

Type	Device name	Symbol	Range of use	Increment for setting
Bit	Safety input	SA\X	SA\X0 to SA\X2FFF	—
	Safety output	SA\Y	SA\Y0 to SA\Y2FFF	—
	Safety internal relay	SA\M	SA\M0 to SA\M638975	64 points
	Safety link relay	SA\B	SA\B0 to SA\B9BFFF	64 points
Word	Safety timer	SA\T	SA\T0 to SA\T35487	32 points
	Safety retentive timer	SA\ST	SA\ST0 to SA\ST35487	32 points
	Safety counter	SA\C	SA\C0 to SA\C35487	32 points
	Safety data register	SA\D	SA\D0 to SA\D39935	4 points
	Safety link register	SA\W	SA\W0 to SA\W9BFF	4 points

38.17 RAS Functions

This section describes the RAS functions of the SIL2 Process CPU.

Self-diagnostic function

This section describes the self-diagnostic function of the SIL2 Process CPU module and SIL2 function module. As for the SIL2 Process CPU, only the parts that differ from the standard CPU are described.

How to check errors

Errors in the SIL2 Process CPU can be checked in the same way as for the standard CPU. (☞ Page 138 Self-Diagnostics Function)

Errors in the SIL2 function module can be checked in the following ways.

■Using the buffer memory

When the SIL2 function module detects an error, the corresponding error code is stored in Un\G0 (Latest self-diagnostics error code). If multiple errors are detected, the latest error code is stored in Un\G0. Up to 16 error codes can be stored in Un\G10 (Self-diagnostics error code 1) to Un\G25 (Self-diagnostics error code 16). (The error codes of the 17th error and later are not stored.)

■Using the LED of the safety function module

Like intelligent function modules, the error status can be checked with the ERROR LED. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

■Using the engineering tool

Like intelligent function modules, the error status of the entire system and the history of errors currently occurred or events can be checked on the "Module Diagnostics" window. (📖 GX Works3 Operating Manual)

Error detection setting

■Applicable errors to the error detection setting

The following table lists the errors to set whether or not to be detected in the SIL2 Process CPU.

Error name	Error code
Power shutoff (either of the redundant power supply modules)	1010H
Failure (either of the redundant power supply modules)	1020H
Battery error	1090H
Module verification error	2400H, 2401H
Fuse blown error	2420H

CPU module operation setting at error detection

The "Instruction Execution Error" setting under the "CPU Module Operation Setting at Error Detected" is applied only for standard programs. A stop error always occurs in safety programs. (Even if "Continue" is selected, a stop error occurs.)

[CPU Parameter] ⇨ [RAS Setting] ⇨ [CPU Module Operation Setting at Error Detection]

■Applicable errors to the CPU module operation upon error detection setting

The following table lists the applicable errors to the setting that specifies the CPU module operation of when the specific errors are detected in the SIL2 Process CPU.

Error name	Error code
Memory card error	2120H, 2121H
Module verification error	2400H, 2401H
Fuse blown error	2420H
I/O number or network number specification error	2800H, 2801H, 2802H, 2803H, 2804H, 2805H, 2806H, 2807H, 2810H
Device, label, or buffer memory specification error	2820H, 2821H, 2822H, 2823H, 2824H
File name specification error	2840H, 2841H, 2842H
Operation error	3400H, 3401H, 3402H, 3403H, 3404H, 3405H, 3406H, 3420H, 3421H, 3422H, 3423H, 3440H, 3441H, 34A0H

CPU module operation setting

■Applicable errors to the CPU module operation setting

The following table lists the applicable errors to the setting that specifies the CPU module operation of when the specific errors have occurred in each intelligent function module in the system using the SIL2 Process CPU.

Error name	Error code
Module moderate error	1200H
Module major error	2441H, 2442H, 2450H

Error clear

This function clears all the existing continuation errors occurring in the SIL2 Process CPU or SIL2 function module at once. However, for the SIL2 function module, errors in the other system cannot be cleared. Clear the errors separately in each system.

Errors that can be cleared

This function can be used to clear only the continuation errors listed in the following table.

■SIL2 Process CPU

Error name	Error code
Power shutoff	1000H
Power shutoff (either of the redundant power supply modules)	1010H
Failure (either of the redundant power supply modules)	1020H
Invalid power supply module	1030H
Power supply module configuration error	1031H
ROM write count error	1080H
Battery error	1090H
Memory card access error	1100H
SNTP clock setting error	1120H
Default gateway/gateway IP address error	1124H
Own node port number error	1128H
Open specification port number error	1129H
Specified IP address error	112DH
Connection establishment failed	112EH
Socket communications response send error	1133H
TCP connection timeout	1134H
IP address error	1152H
Connection number acquisition error	1155H
Receive buffer securement error	1157H
UDP/IP send failed	1165H
TCP/IP send failed	1166H
Unsend data send error	1167H
Redundant IP address error	1180H
Module moderate error	1200H
Another CPU module moderate error	1220H
Annunciator ON	1800H
Operation continuation error	1810H
Receive queue full	1830H
Receive processing error	1831H
Transient data error	1832H
Constant scan time error	1900H
Safety cycle time exceeded	1A00H
Safety cycle processing error	1A01H
Destination station error in safety communications	1A40H[*1]
Parameter mismatch in safety communication destination station	1A50H[*1]
Device mismatch in safety communication destination station	1A51H[*1]
Device version mismatch in safety communication destination station	1A52H[*1]
Timeout in safety communications	1A60H[*1], 1A61H[*1], 1A62H[*1], 1A63H[*1], 1A64H[*1], 1A65H[*1], 1A66H[*1], 1A80H
Safety communications receipt data error	1A70H[*1], 1A71H[*1], 1A72H[*1], 1A73H[*1]
Network configuration mismatch	1B00H
System consistency check error (operating status)	1B20H
System consistency check error (safety operation mode)	1B21H
Redundant system error	1B40H, 1B42H, 1B43H

38

Error name	Error code
Standby system CPU module error	1B60H, 1B61H
Tracking communications disabled	1B70H
Tracking communication error	1B71H, 1B78H
Tracking transfer error	1B80H, 1B81H, 1B82H
Redundant function module error	1BA0H
System switching error	1BD0H, 1BD1H
Memory card error	2120H, 2121H
Module verification error	2400H, 2401H
Fuse blown error	2420H
Module major error	2441H, 2442H, 2450H
Another CPU module major error	2461H, 2462H, 2470H
I/O number or network number specification error	2800H, 2801H, 2802H, 2803H, 2804H, 2805H, 2806H, 2807H, 2810H
Device, label, or buffer memory specification error	2820H, 2821H, 2822H, 2823H, 2824H
File name specification error	2840H, 2841H, 2842H
Operation error	3400H, 3401H, 3402H, 3403H, 3404H, 3405H, 3406H, 3420H, 3421H, 3422H, 3423H, 3424H, 3440H, 3441H, 34A0H

*1 If a safety communications error (minor error) occurs between safety stations, a safety station interlock will be triggered for safety communications between relevant stations, and safety communications will not be resumed until the safety station interlock is released. The safety station interlock is released with safety special register areas (safety station interlock release request for each safety connection). (Page 1026 List of Safety Special Register Areas)
For details on the safety station interlock status, refer to the following.
MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

■SIL2 function module

Error name	Error code
ROM write count error	1080H
Safety cycle time exceeded	1A00H
Destination station error in safety communications	1A40H[*1]
Parameter mismatch in safety communication destination station	1A50H[*1]
Device mismatch in safety communication destination station	1A51H[*1]
Device version mismatch in safety communication destination station	1A52H[*1]
Timeout in safety communications	1A60H[*1], 1A61H[*1], 1A62H[*1], 1A63H[*1], 1A64H[*1], 1A65H[*1], 1A66H[*1], 1A80H[*1]
Safety communications receipt data error	1A70H[*1], 1A71H[*1], 1A72H[*1], 1A73H[*1]

*1 If a safety communications error (minor error) occurs between safety stations, a safety station interlock will be triggered for safety communications between relevant stations, and safety communications will not be resumed until the safety station interlock is released. The safety station interlock is released with safety special register areas (safety station interlock release request for each safety connection). (Page 1026 List of Safety Special Register Areas)
For details on the safety station interlock status, refer to the following.
MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

How to clear errors

Errors in the SIL2 Process CPU can be cleared in the same way as for the standard CPU. (☞ Page 147 How to clear errors)
The following describes how to clear errors in the SIL2 function module. The errors in the other system cannot be cleared. Clear the errors separately in each system.

■Using the engineering tool

Clear errors with the module diagnostics function of GX Works3. (📖 GX Works3 Operating Manual)

■Using the buffer memory

Clear errors using operations of buffer memory.

1. Check the standard program and identify the detected continuation error using Un\G0 (Latest self-diagnostic error code).
2. Eliminate the cause of the detected continuation error.
3. Set 1 in Un\G50 (Error clear) in the standard program to clear the error.[*1]

*1 After the error is cleared, 0 is automatically set in Un\G50 by the system.

Clearing errors on the standby system CPU module from the control system CPU module

Errors on the standby system can be cleared from the control system CPU module by using SM1679 (Error reset (the other system)) in programs or external devices.

Error clearing procedure

Use SM1679 to clear errors.

1. Eliminate the causes of all the continuation errors that have occurred on the standby system.
2. Turn off and on SM1679 of the control system CPU module to clear the errors. When multiple continuation errors have occurred, all the errors are cleared at once.

Precautions

- Error clear operation with SM1679 can be performed by the control system CPU module only. A continuation error on the standby system CPU module cannot be cleared by turning off and on SM1679 of the standby system CPU module.
- The cause of an error which has occurred in a module other than the target CPU module for the error clear cannot be eliminated even though the error is cleared using SM1679.
- If the error cause is not eliminated after the error clear operation, the same error is detected again.
- The error clear processing is performed in the END processing. To clear an error, execute the END instruction while SM1679 is on.

38.18 SLMP Communications

This section describes the precautions on communications using the SLMP.

System switching

There are the following notes when the system IP address matching function is not used.

■Re-setting of the connection destination

When the relay CPU module is in the communication-disabled state (power-off, reset, or tracking cable disconnection) at system switching, the connection destination needs to be set again for communications using the SLMP.

■Re-execution of the write command

When "Control System" or "Standby System" has been selected in the transfer setup and the systems are switched, a target system mismatch is detected by a command issued during the system switching, and a communication error occurs. If a communication error occurs while a data write command is being issued, the data write command needs to be issued for the new control system.

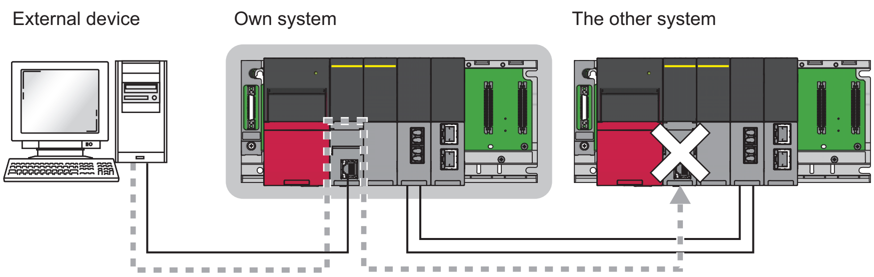
Remote operation

If a remote operation command is executed, the CPU modules enter different operating statuses and thus systems cannot be switched.

If the safety operation mode is SAFETY MODE, the safety global devices, safety global labels, safety local labels, and standard/safety shared labels are not cleared even when the clear mode is set to perform the clear processing using remote RUN.

Communicating with other systems

For SLMP communications via the built-in Ethernet port of the CPU module, when communications are performed to the other system that cannot respond (power-off, reset, or tracking cable disconnection), a timeout error may occur.



39 SAFETY DEVICES, SAFETY LABELS, AND CONSTANTS

This chapter describes the safety devices, safety labels, and constants.

39.1 Safety Devices

A safety device is a device used in safety programs. The safety devices can be used only in the safety programs.

Restriction

- The safety devices cannot be used in the standard programs.
- An index modification and indirect specification cannot be performed in safety programs.

List of safety devices

The following table lists the safety devices.

Classification	Type	Device name	Symbol	Number of points (default)	Changeability with parameter settings	Notation
Safety user devices	Bit	Safety input	SA\X	8K points	Selectable either 8K or 12K points	Hexadecimal
	Bit	Safety output	SA\Y	8K points		Hexadecimal
	Bit	Safety internal relay	SA\M	6K points	Changeable (☞ Page 770 Safety device area setting range)	Decimal
	Bit	Safety link relay	SA\B	4K points		Hexadecimal
	Bit/word	Safety timer	SA\T	512 points		Decimal
	Bit/word	Safety retentive timer	SA\ST	0 points		Decimal
	Bit/word	Safety counter	SA\C	512 points		Decimal
	Word	Safety data register	SA\D	12K points		Decimal
	Word	Safety link register	SA\W	4K points		Hexadecimal
Safety system devices	Bit	Safety special relay	SA\SM	4K points	Unchangeable	Decimal
	Word	Safety special register	SA\SD	4K points		Decimal

Point

For details on standard devices used for the SIL2 Process CPU, refer to the following.

☞ Page 374 DEVICES

Safety user devices

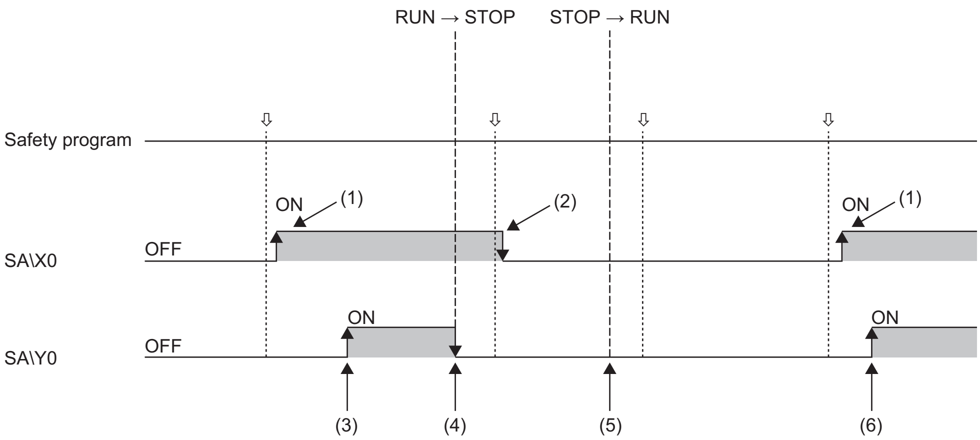
This section describes the safety user devices.

Safety input (SA\X)

Safety inputs are used to give instructions or data to the SIL2 Process CPU with external devices such as emergency stop buttons, safety plugs, door switches, and light curtains.

Safety output (SA\Y)

Safety outputs are used to output safety program control results to devices such as external relays and contactors.
When the status changes from STOP to RUN, the value immediately before the RUN state is retained, and the value is updated at the moment safety cycle processing is executed for the first time after the status is changed to the RUN state.



⇩: Safety cycle processing start
(1) Turns on when safety communications are received.
(2) Turns off when safety communications are received.
(3) SA\Y0 turns on when SA\X0 turns on in the safety program.
(4) SA\Y0 turns off when the status of the CPU module is switched to STOP.
(5) The value is maintained when the status changes from STOP → RUN.
(6) Turns on when a safety program is executed.

Safety internal relay (SA\M)

This device is used as an auxiliary relay within the SIL2 Process CPU. The following operations for the SIL2 Process CPU turn off all safety internal relays.

- Powering off and on
- Reset

Safety link relay (SA\B)

This device is used as a device when sending and receiving safety data between SIL2 Process CPUs over CC-Link IE Field Network.

■Refreshing network modules using safety link relay

Data is sent and received between SIL2 Process CPUs. The safety communications send/receive device range is set in the CC-Link IE Field Network safety communication settings. Locations that are not used for safety communications send/receive devices can be used for other applications.

Safety timer (SA\T)/safety retentive timer (SA\ST)

This device starts measurement when the safety timer coil is turned on. When the current value reaches a setting value, time is up and the contact is turned on. This safety timer is an up-timing type device and therefore the current value matches a setting value when the safety timer time is up. Operations other than the following are the same as those for the timer. (☞ Page 383 Timer)

■Safety timer types

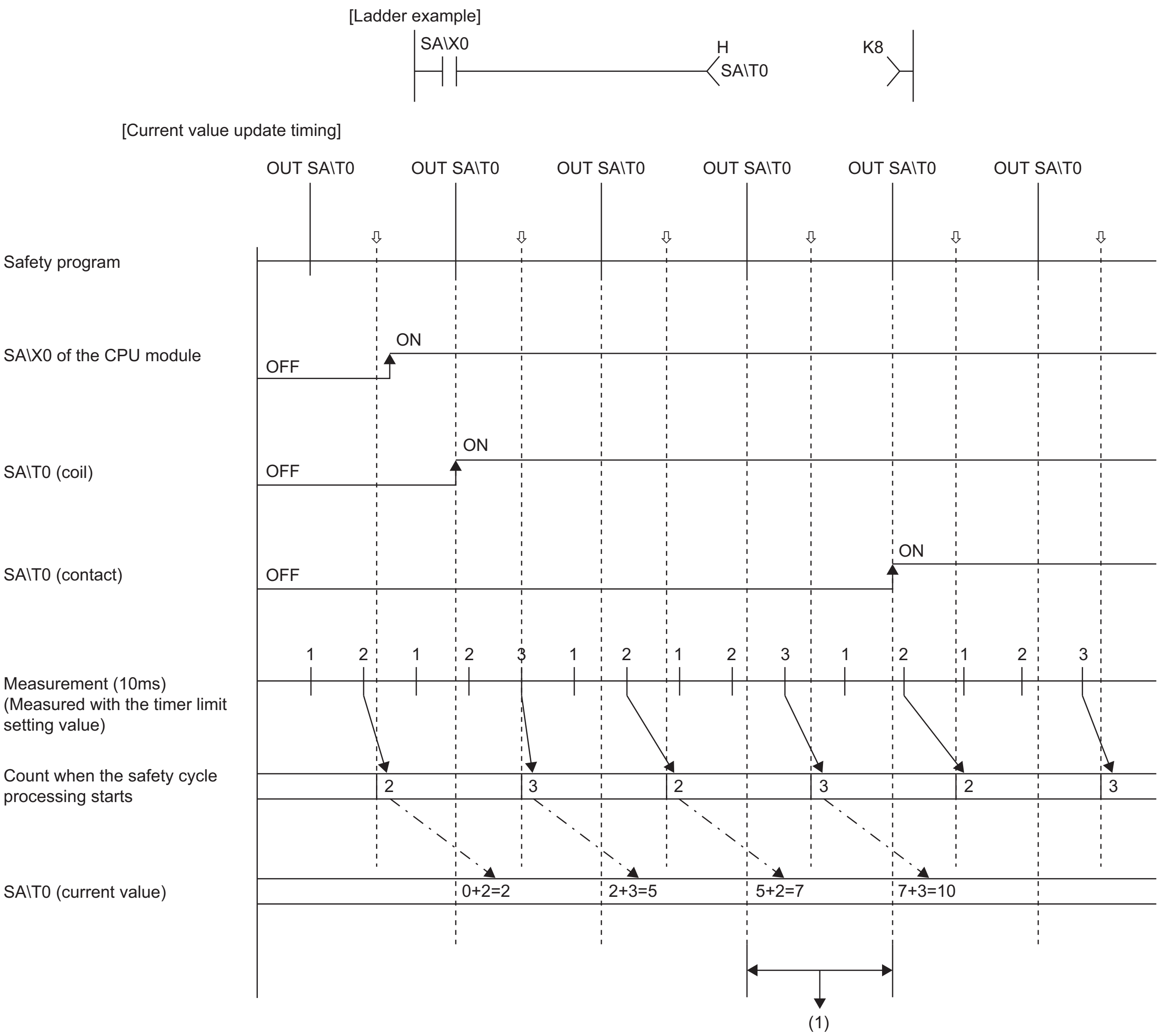
There is a safety timer (SA\T) that retains the current value in 16-bit units. In addition, there is a safety retentive timer (SA\ST) that retains the current value even if the coil turns off.

■Safety timer accuracy

The current value is measured when safety cycle processing starts. The value for the elapsed time since the previous safety cycle processing was started until the present is added to the current value when the OUT SA\T☐ instruction is executed. If the safety timer coil is off at the execution of the OUT SA\T☐ instruction, the current value is not updated. The maximum response accuracy of the timer is the "elapsed time since previous safety cycle processing was started until the present + timer limit setting".

Ex.

Timer limit setting = 10ms, SA\T0 setting value = 8



⇩: Safety cycle processing start

(1) Time accuracy - (Elapsed time since the previous safety cycle processing was started until the present + Timer limit setting) to (Elapsed time since the previous safety cycle processing was started until the present)

(2) Sets the coefficient when starting safety cycle processing.

Safety counter (SA\C)

This device counts the number of rising operation of the input condition in the program. The safety counter is an up-timing type device and therefore when the count value matches a setting value, the count reaches its upper limit and the contact is turned on. Operations other than the following are the same as those for the counter. (☞ Page 391 Counter)

■Safety counter types

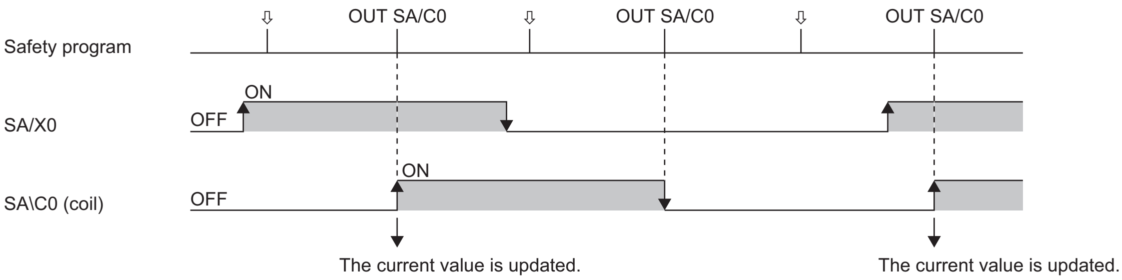
There is a safety counter (SA\C) that retains the counter value in 16-bit units.

■Counting process

When the safety counter coil is executed, the safety counter coil is turned on/off, the current value is updated (count value +1), and the contact is turned on/off. The current value is updated (count value +1) when the safety counter coil input is rising (off to on). The current value is not updated when the coil input is off, on to on, and on to off.

[Ladder example]

[Current value update timing]



⇩: Safety cycle processing start

■Resetting the safety counter

The safety counter current value is not cleared even when the counter coil input is turned off. To clear (reset) the safety counter current value and turn off the contact, issue the RST SA\C□ instruction. When the RST SA\C□ instruction is executed, the counter value is cleared and the contact is turned off.

Safety data register (SA\D)

This device can store numerical values.

Safety link register (SA\W)

This device is used as a device when sending and receiving safety data between SIL2 Process CPUs over CC-Link IE Field Network.

■Refreshing network modules using safety link register

Data is sent and received between SIL2 Process CPUs. The safety communications send/receive device range is set in the CC-Link IE Field Network safety communication settings. Locations that are not used for safety communications send/receive devices can be used for other applications.

Safety system devices

This section describes the safety system devices.

Safety special relay (SA\SM)

This relay stores the SIL2 Process CPU status relating to safety control. (☞ Page 1023 List of Safety Special Relay Areas)

Safety special register (SA\SD)

This register stores the SIL2 Process CPU status relating to safety control. (☞ Page 1026 List of Safety Special Register Areas)

39.2 Safety Global Devices

A safety global device is a device that can be shared by all the safety programs.

39.3 Safety Local Devices

A safety local device is a device used individually by each safety program.

Devices that can be used as a safety local device

The following devices can be used as a safety local device.

- Safety internal relay (SA\M)
- Safety timer (SA\T)
- Safety retentive timer (SA\ST)
- Safety counter (SA\C)
- Safety data register (SA\D)

How to set safety local devices

The range and usability of the safety local devices can be set in the same way as the standard local devices. (☞ Page 417 Setting method for the local devices)

Point

Set safety local devices within the number of device points set in the safety device point quantity setting. The number of safety local device points used is calculated by the following calculation formula. Set the number of safety local device points used so that the number is equal to or less than the capacity of the safety local device area.

Total number of safety local device points used = ((A ÷ 16) + B + (C × 2) ÷ 16)) × D

- A: Number of points of the safety local device (SA\M)
- B: Number of points of the safety local devices (SA\D, SA\T (current value), SA\ST (current value), and SA\C (current value))
- C: Number of points of the safety local devices (SA\T (contact/coil), SA\ST (contact/coil), and SA\C (contact/coil))
- D: Number of programs using the safety local device

How to specify safety local devices

To specify safety local devices in a safety program, add "#".

Ex.

SA\#D100, SA\K4#M0

Point

In the program, safety local devices are displayed with "#" in front of the device name. This helps users to distinguish local devices from global devices.

SM777 setting

Regardless of the SM777 (Local device setting in interrupt programs) setting, local devices/safety local devices of a program file in the storage location are always used in the standard program/safety program.

39.4 Safety Label

A label used in safety programs is called a safety label. Information not described in this section is same as that of standard labels. (☞ Page 420 LABELS)

Safety label types

There are three safety label types. Only the following labels can be used in safety programs.

- Safety global label*1
- Standard/safety shared label*2
- Safety local label

*1 Safety devices can be assigned.

*2 This label can be used in both standard programs and standard function blocks.

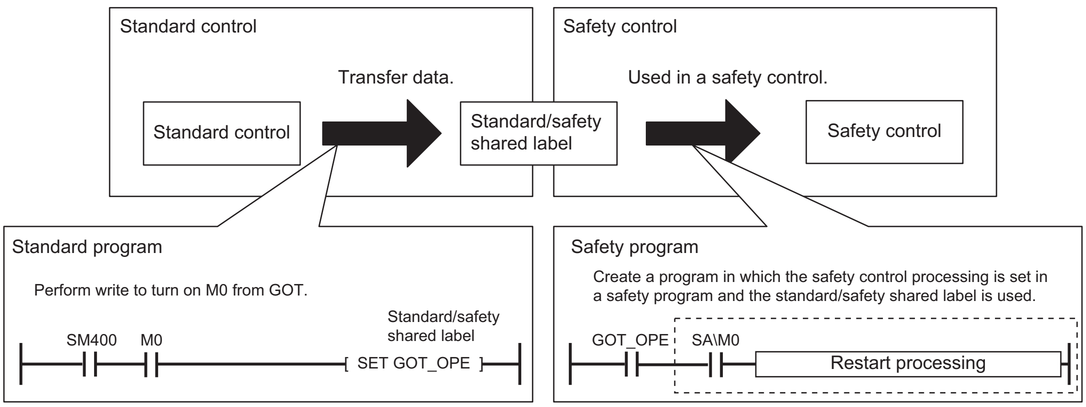
Restriction

- An initial value cannot be set to safety labels and standard/safety shared labels in the CPU parameter.
- To use aliases, a safety label can assign only safety labels as its original data location. Likewise, a standard/safety shared label can assign only standard/safety shared labels. A standard label cannot assign safety labels and standard/safety shared labels as its original data location. (GX Works3 Operating Manual)

How to use standard/safety shared labels

A standard/safety shared label is used to pass device data from a safety program to a standard program, and vice versa. When a standard/safety shared label is used in a safety program as shown in the examples below, the program needs to be created so that the safety state is secured.

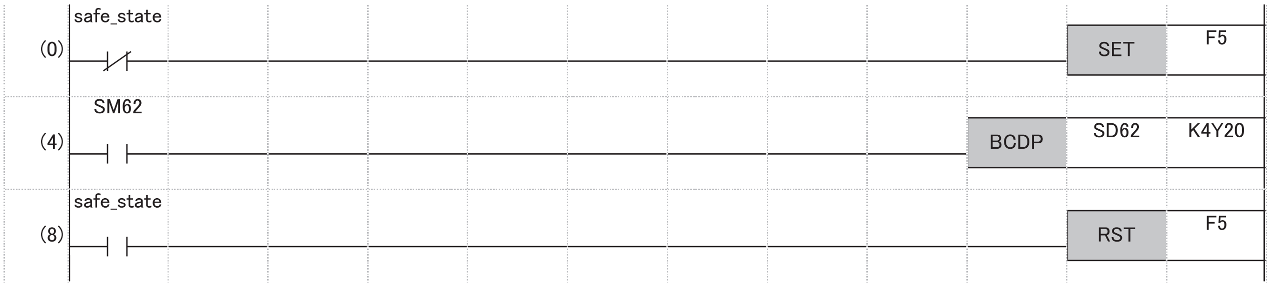
■To restart safety control by the command from the GOT



■To use the annunciator (F)

The safe state signal status can be controlled using the annunciator (F) in the standard program. The safe state signal status is passed from the safety program to the standard program via the standard/safety shared label (safe_state), and the status is controlled with the annunciator No.5. If an error is detected with the annunciator, the corresponding annunciator number is output to Y20.

- Standard program



(0) When the safe state signal turns off, the annunciator No.5 turns on.
(4) The annunciator number detected by SM62 (Annunciator) is output to Y20.
(8) When the safe state signal turns on, the annunciator No.5 turns off.

Classes

The following table lists the availability of the classes of safety global labels and standard/safety shared labels.

○: Applicable, ×: Not applicable

Class	Availability	
	Safety global label	Standard/safety shared label
VAR_GLOBAL	○	○
VAR_GLOBAL_CONSTANT	○	○
VAR_GLOBAL_RETAIN	×	×

The following table lists the availability of the classes of safety local labels.

○: Applicable, ×: Not applicable

Class	Availability		
	Safety program	Safety function	Safety function block
VAR	○	○	○
VAR_CONSTANT	○	○	○
VAR_RETAIN	×	×	×
VAR_INPUT	×	○	○
VAR_OUTPUT	×	○	○
VAR_OUTPUT_RETAIN	×	×	×
VAR_IN_OUT	×	×	○
VAR_PUBLIC	×	×	○
VAR_PUBLIC_RETAIN	×	×	×

39

Data types

Primitive data type

The following table lists the availability of primitive data types.

○: Applicable, ×: Not applicable

Data type		Availability
Bit	BOOL	○
Word [unsigned]/bit string [16 bits]	WORD	○
Double word [unsigned]/bit string [32 bits]	DWORD	○
Word [signed]	INT	○
Double word [signed]	DINT	○
Single-precision real number	REAL	×
Double-precision real number	LREAL	×
Time	TIME	○
String	STRING	×
String [Unicode]	WSTRING	×
Timer	TIMER	○
Retentive timer	RETENTIVETIMER	○
Long timer	LTIMER	×
Long retentive timer	LRETENTIVETIMER	×
Counter	COUNTER	○
Long counter	LCOUNTER	×
Pointer	POINTER	×

Structures

The structure definition is shared by standard programs and safety programs. However, it cannot be used in the following cases.

- A member of the primitive data type which cannot be used in safety programs exists.
- An initial value is set in the structure definition.

39.5 Constants

The decimal constant (K) and hexadecimal constant (H) can be used in safety programs. The specification method is the same as that for standard programs. (Example: K1234, H1FFF) (☞ Page 450 CONSTANTS)

40 PRECAUTIONS ON PROGRAMMING

This chapter describes the precautions on programming for a system using the SIL2 Process CPU.

40.1 Instructions Not Available in System Using SIL2 Process CPU

This section describes the instructions not available in a system using the SIL2 Process CPU.

Instructions that need to be executed again in a new control system

For an instruction that requires several scans for completing the processing, the instruction will be continuously executed when the system switching is performed during execution of the instruction. When a completion device has been used in an execution program of the control system, the completion device will not turn on even though the instruction is completed after the system switching from the control system to the standby system. However, the completion device will turn on after the system switching from the control system to the standby system and then to the control system again. Note that the completion status of the completion device is not reflected to the tracking device. When the system switching is performed during execution of an instruction, execute the instruction again as required.

Classification	Instruction symbol
Data processing instructions	SORTD(_U), DSORTD(_U)
Reading/writing data instructions	SP.DEVST, SP.FREAD, SP.FWRITE
Open/close processing instructions	SP.SOCOPEN, SP.SOCCLOSE
Socket communications instructions	SP.SOCRCV, S.SOCRCVS, SP.SOCSND, SP.SOCCINF, SP.SOCCSET, SP.SOCRMODE, S(P).SOCRDATA
Module dedicated instructions	Instructions that require several scans for completing processing

For some instructions, an error will occur if an instruction is executed during execution of the same instruction. For the operation for each instruction that is executed again during execution of the same instruction, refer to the following.

📖 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)
📖 MELSEC iQ-R Programming Manual (Module Dedicated Instructions)

Re-execution of instruction when systems are switched during instruction execution

When the system switching is performed while an instruction that requires several scans is being executed, the instruction can be executed again in the new control system after the system switching by using the programs such as following.

■REMFR instruction

When the system switching is performed while the instruction is being executed (M201 = ON), SM1643 (ON for only one scan after system switching (standby system to control system)) will turn on for one scan in the new control system and the REMFR instruction will be executed again on the station number 10 of the network number 1.

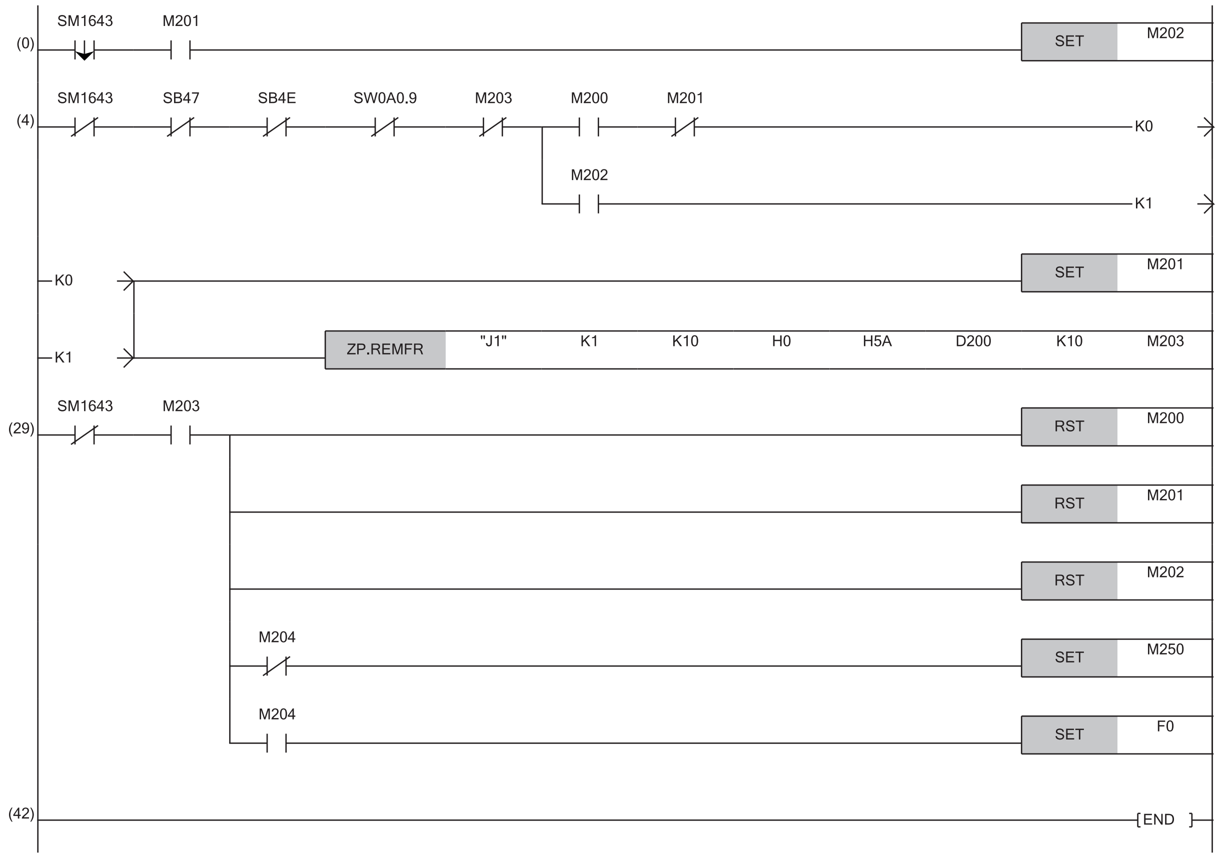
- Devices used

Device	Description
SM1643	ON for only one scan after system switching (standby system to control system)
SB47*1	Baton pass status of own station
SW0A0.9*1	Baton pass status of each station
M200*2	Reading request
M201*2	At instruction execution
M202*2	ON at instruction re-execution request due to system switching
M203*2	Completed without an error
M204*2	Completed with an error

*1 For details on the link special relay (SB) and link special register (SW), refer to the manuals for the network used.

*2 Change the device number according to the system.

- Program example



■REMTO instruction

When the system switching is performed while the instruction is being executed (M101 = ON), SM1643 (ON for only one scan after system switching (standby system to control system)) will turn on for one scan in the new control system and the REMTO instruction will be executed again on the station number 10 of the network number 1. When the system switching is performed while a write instruction such as the REMTO instruction is being executed, execution of the instruction may have been suspended before completion of writing data to the target module. Thus, insert an interlock in the new control system to read the X signals and buffer memory status of the target module and to determine whether or not to execute the instruction again.

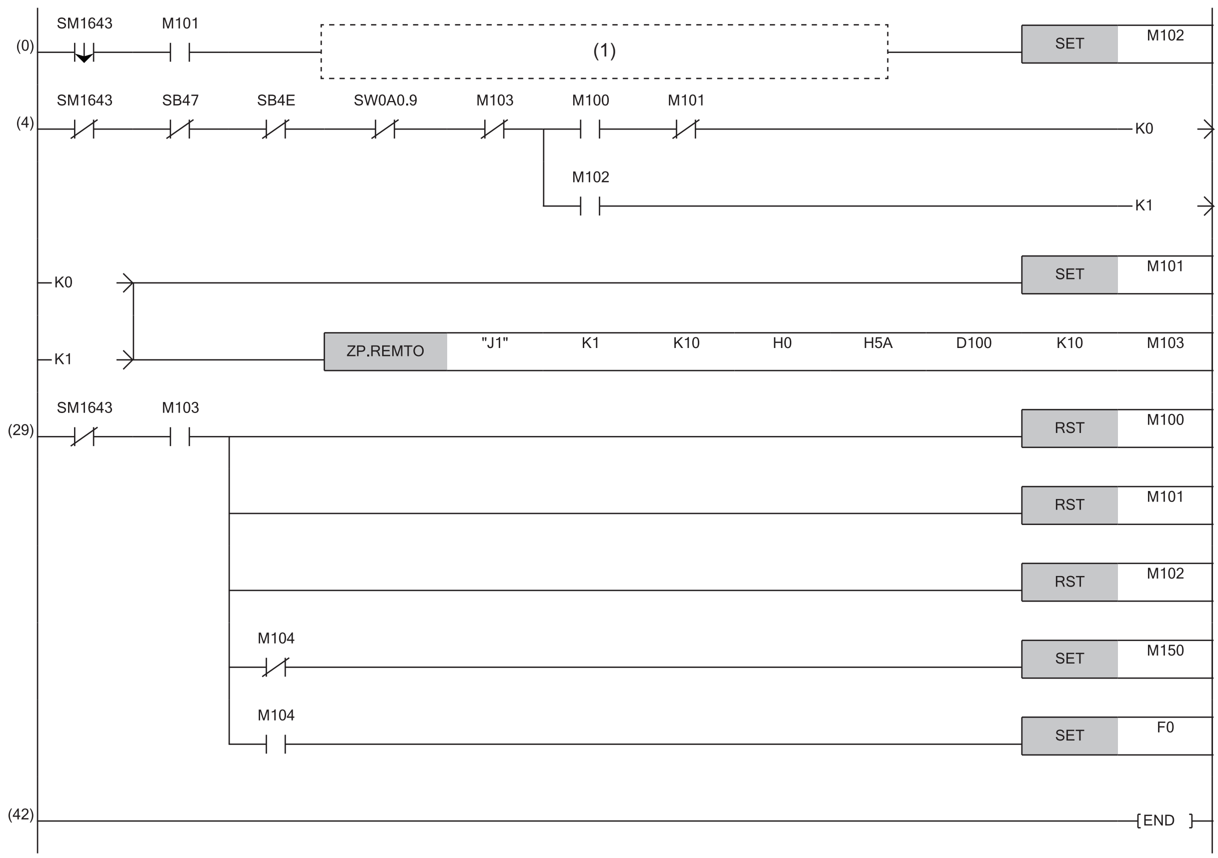
• Devices used

Device	Description
SM1643	ON for only one scan after system switching (standby system to control system)
SB47*1	Baton pass status of own station
SW0A0.9*1	Baton pass status of each station
M100*2	Write request
M101*2	At instruction execution
M102*2	ON at instruction re-execution request due to system switching
M103*2	Completed without an error
M104*2	Completed with an error

*1 For details on the link special relay (SB) and link special register (SW), refer to the manuals for the network used.

*2 Change the device number according to the system.

• Program example



(1) Add an interlock circuit for determining whether or not to execute the instruction again (according to the X signals and buffer memory status of the target module) as required.

Instructions whose operations vary depending on tracking of the signal flow memory

This section describes the instructions whose operations after the system switching vary depending on whether the signal flow memory is tracked or not. The operations vary when one of the following instructions is executed among program organization units that have the signal flow memory, to which tracking can be performed.

Classification/type of special relay	Instruction symbol
Rising instruction	LDP, ANDP, ORP, LDPI, ANDPI, ORPI, PLS, MEP, EGP, SET F, RST F, FF, LEDR, DUTY, LOGTRG, LOGTRGR, □P (including MOVP and INCP), SP.□, JP.□, GP.□, ZP.□
Falling instruction	LDF, ANDF, ORF, LDFI, ANDFI, ORFI, PLF, MEF, EGF
SCJ instruction	SCJ
Data processing instructions	SORTD(_U), DSORTD(_U)
TIMCHK instruction	TIMCHK
XCALL instruction	XCALL
Rising instruction using SM1643 as an execution condition	—

Rising instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, a rising instruction whose execution condition turned on during system switching will not be executed.

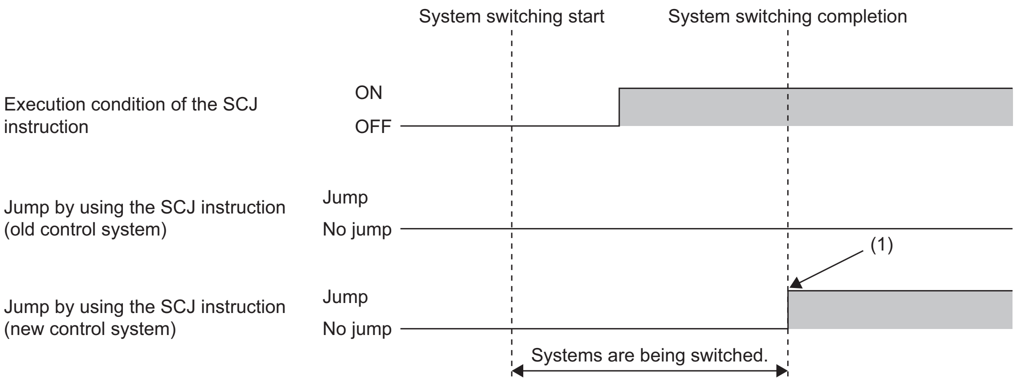
Falling instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, a falling instruction whose execution condition turned off before the system switching will be executed.

SCJ instruction

■When the signal flow memory is not tracked

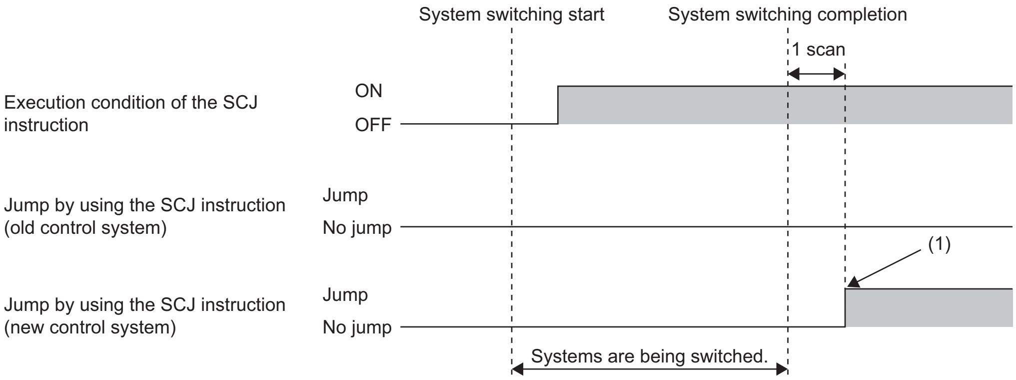
Once the system switching is performed, the execution condition of the SCJ instruction turns on after the signal flow memory of the new control system turns off. In the new control system, processing jumps to the pointer specified by the SCJ instruction in the first scan.



(1) After system switching, the processing jumps in the first and later scans.

■When the signal flow memory is tracked

When the system switching is performed, the execution condition of the SCJ instruction turns on while the signal flow memory remains off. In the new control system, the processing jumps to the pointer specified by the SCJ instruction in the second scan.



(1) After system switching, the processing jumps in the second and later scans.

Data processing instructions

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, the SORTD(_U)/DSORTD(_U) instruction in the first scan after the system switching will be executed not as the first execution but as continuous execution (continuous processing). When the instruction is executed for the first time, the instruction is executed without data to be stored in the devices used by the system, causing an unintended operation.

TIMCHK instruction

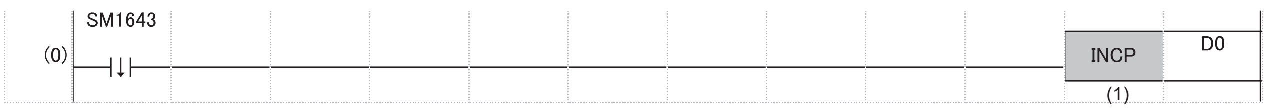
If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, the TIMCHK instruction in the first scan after the system switching will be executed not as the first execution but as continuous execution (continuous processing). At the first execution, the current value is not cleared and the device that turns on at timeout is not turned off. The instruction is executed with the status at the previous measurement.

XCALL instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, the subroutine program will not be executed when the execution condition of the XCALL instruction remains off in the first scan after system switching.

Rising instruction using SM1643 as an execution condition

SM1643 is the special relay that turns on for one scan in the new control system after system switching. If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on and a rising instruction cannot be executed. To execute a rising instruction where SM1643 has been set as an execution condition, use a falling edge pulse operation contact (LDF/AND/ORF instruction) and create a program in which the rising instruction is to be executed at the falling edge of SM1643, as shown below. However, when the falling edge of SM1643 has been set as the execution condition, the target instruction will be executed in the second scan after the system switching.



(1) The instruction will be executed in the second scan after the system switching.

Instructions that affect the status of another instruction when executed

When one of the following instructions is executed and the status of another instruction changes, the new status will not be tracked to the other system. When the system switching is performed during execution of an instruction, execute the instruction again as required.

Classification		Instruction symbol
Program execution control instructions	Disabling interrupt programs	DI
	Enabling interrupt programs	EI
	Disabling interrupt programs with specified priority or lower	DI
	Interrupt program mask	IMASK
	Disabling/enabling the specified interrupt pointer	SIMASK
File register operation instructions	Switching the file register block number	RSET(P)
	Changing the file register file name	QDRSET(P)
Timing check instruction	Generating timing pulses	DUTY

Instructions that cause different operation results between the control system and standby system

This section describes the instruction that causes different operation results in both systems after system switching.

PID control instructions

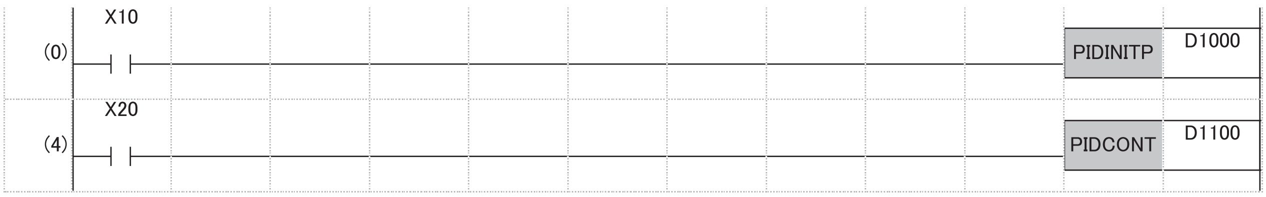
When using the following PID control instructions, include the number of device points used by the systems in tracking target data. Otherwise, the instructions cause different operation results between the control system and standby system after system switching.

n: Total number of loops

Classification		Instruction symbol	Number of device points used
PID control instructions (Inexact differential)	Registering the PID control data to the CPU module	S(P).PIDINIT	2 + n × 4
	PID operation	S(P).PIDCONT	10 + n × 23
PID control instructions (Exact differential)	Registering the PID control data to the CPU module	PIDINIT(P)	2 + n × 10
	PID operation	PIDCONT(P)	10 + n × 18

Ex.

When the number of loops is eight, the PIDINITP instruction needs 2 + 8 × 10 = 82 words and the PIDCONT instruction needs 10 + 8 × 18 = 154 words. Thus, include D1000 to D1081 and D1100 to D1253 in tracking target data.



Precautions for using the COM or ZCOM instruction

When the network module link refresh is performed by using the COM or ZCOM instruction, output from the remote I/O station may change after system switching. To prevent this, do not perform the network module link refresh with the COM or ZCOM instruction. For the COM instruction, whether or not to perform the network module link refresh can be set using SM775 (Selection of refresh processing during the COM instruction execution) and SD775 (Selection of refresh processing during the COM instruction execution). Set SM775 and SD775 and perform only the I/O refresh and device/label access service processing with the COM instruction.

○: Selectable, ×: Not selectable

Instruction	Refresh processing	In a system using the SIL2 Process CPU
COM instruction	I/O refresh	○
	Network module link refresh	○*1
	Intelligent function module refresh	×*2
	Refresh using the CPU buffer memory of the multiple CPU system (END)	×*3
	Device/label access service processing (communications with the engineering tool, GOT, or other external devices)	○
ZCOM instruction	Network module link refresh	○*1
	Intelligent function module refresh	×*2

*1 When the COM or ZCOM instruction is executed, tracking transfer is not performed. Thus, when the system switching is attempted after execution of the instruction and before completion of tracking transfer, the system switching is performed without tracking transfer. Thus, even though output to the network module is changed by the COM or ZCOM instruction with the CPU module in the control system, the change will not be reflected to the CPU module in the standby system. After the system switching, the CPU module in the new control system outputs old values to the network module, and output from the network module may vary.

*2 This processing is not selectable because no intelligent function modules can be mounted on main base units.

*3 This processing is not selectable because a multiple CPU system cannot be built.

Precautions for using the ADRSET instruction

Even though a file is written to both the control system and standby system, different addresses are assigned to the systems. To continue the processing in the new control system, use the ADRSET instruction to obtain indirect addresses again.

40.2 Interrupt from Modules

This section describes the precautions for interrupts from modules.

When the old control system is switched to the new standby system

The old control system retains the interrupt factors that have occurred even after the system is switched to the new standby system by system switching before execution of an interrupt program. After the systems are switched again, the interrupt program of an interrupt factor that the old control system has retained will be executed.
Since the interrupt factor that has occurred on the old control system is not inherited to the new control system, the interrupt program of an interrupt factor that has occurred on the old control system will not be executed on the new control system.

When the old standby system is switched to the new control system

The old standby system retains the interrupt factors that have occurred. After the old standby system is switched to the new control system by system switching, the interrupt program of an interrupt factor that the old standby system has retained will be executed on the new control system.
When the old standby system has retained multiple interrupt factors, the scan time may greatly increase.

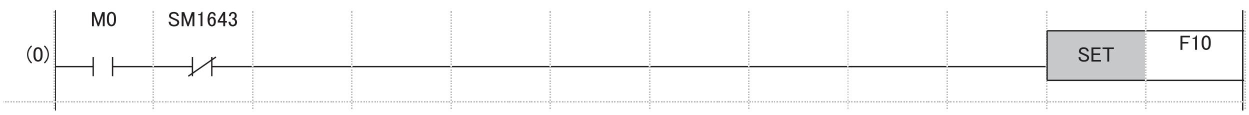
40.3 Precautions for Using the Annunciator (F)

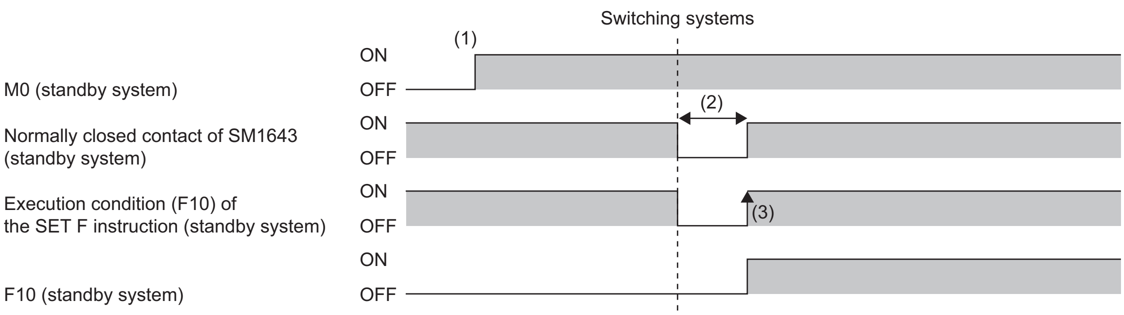
This section describes the precautions for using the annunciator (F).

When the SET F instruction is used to register the annunciator

When the execution condition of the SET F☐ instruction has been satisfied at system switching, the annunciator that is turned on by the SET F☐ instruction can be registered in the new control system. The SET F☐ instruction registers the annunciator at the rising edge of the execution condition. To register the annunciator in the new control system at system switching, add a normally closed contact of SM1643 (ON for only one scan after system switching (standby system to control system)) as the AND condition in the execution conditions of the SET F☐ instruction.

Ex.





(1) When M0 turns on in the control system before system switching, M0 of the standby system also turns on as a result of tracking transfer.
(2) The contact turns off for one scan after system switching.
(3) When the execution condition turns on, annunciator information is registered by the SET F☐ instruction.

When the OUT F instruction is used to register the annunciator

When the execution condition of the OUT F☐ instruction has been satisfied at system switching, annunciator information is registered in the new control system at system switching.

40.4 Precautions on Timers and Timer Function Blocks

This section describes the precautions on timers and timer function blocks at system switching.

Current values at system switching

For the timer (T), retentive timer (ST), and a timer function block TIMER_□_M, the current values of the timers in the first scan of the CPU module of the new control system will not be updated after system switching.

Timeout before system switching

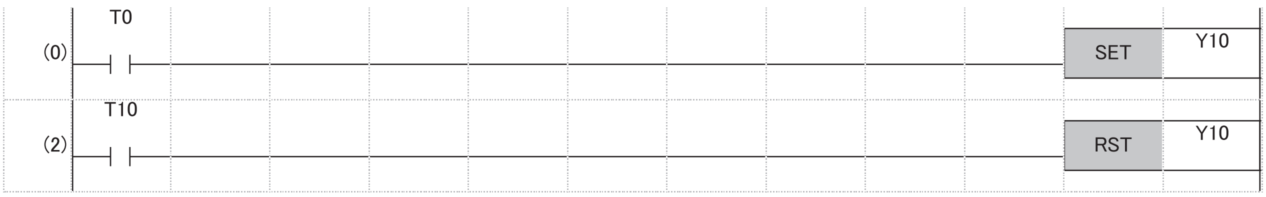
Depending on the timing to perform system switching such as power-off, tracking transfer processing is suspended and tracking data may not be reflected to the CPU module in the new control system. For the timer (T), retentive timer (ST), long timer (LT), long retentive timer (LST), and timer function blocks TIMER_□_M, TP(_E), TON(_E), and TOF(_E), the timer whose time has been up before system switching may go into the state in which no timeout has occurred in the first scan after system switching.

When values are output (writing values to the buffer memory and the output (Y)) with a timer contact or an output variable, the timer statues may go into the state in which no timeout has occurred as described above, causing chattering of the output. To transfer data with modules or external devices using the output (Y) or buffer memory, a program may not properly function due to chattering of output after the system switching. To output values (writing data to the buffer memory and the output (Y)) to modules or external devices with a timer contact or an output variable, consider the time taken for data to be transferred from the CPU module in the control system to the CPU module in the standby system after the time is up.

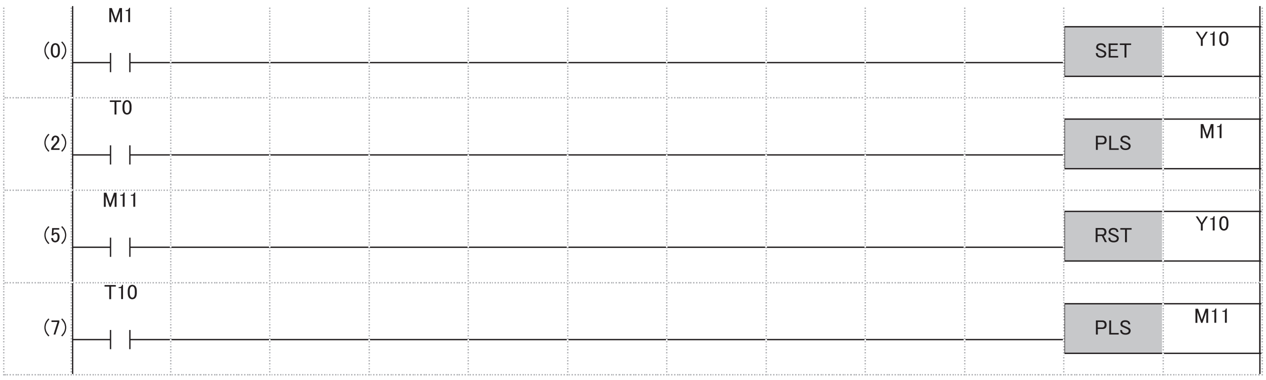
Ex.

Program that delays outputting values by one scan after the time of the timer (T) is up

[Without measures]



[With measures]



Point

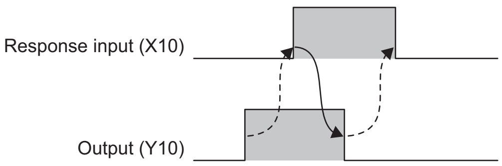
In the CPU parameter, select "Transfer" (default setting) in "Signal Flow Memory Tracking Setting" of "Redundant System Settings". (☞ Page 728 Tracking transfer setting for the signal flow memory)

40.5 Precautions on Access to Intelligent Function Module or External Devices

Depending on the timing of system switching cause to be caused, such as power-off, tracking processing is suspended and device/label data may not be applied to the CPU module in the new control system after the system switching.
Consequently, output data may differ from device/label data of the CPU module of the new control system. In communication with intelligent function modules or external devices using the output (Y) or buffer memory, programs may not properly function due to a mismatch in the device data after system switching.
For command output (such as output (Y), startup by writing data to the buffer memory, and clear) to the intelligent function module and external devices, consider the time taken for tracking the execution condition of command output from the CPU module in the control system to the CPU module in the standby system.
The following example shows a program that outputs data one scan later after the command output condition is satisfied.

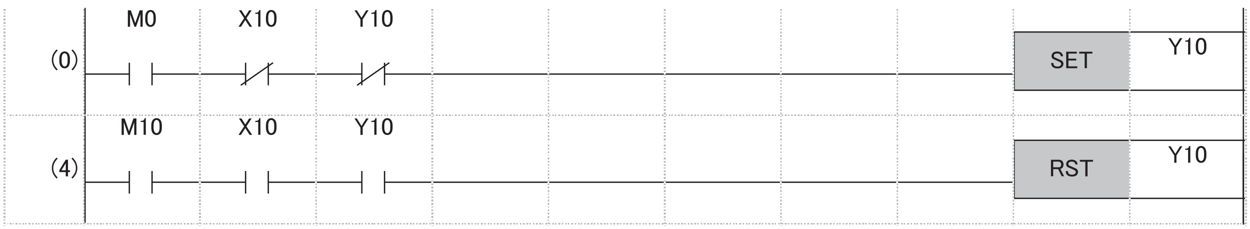
Ex.

When response input is returned to output

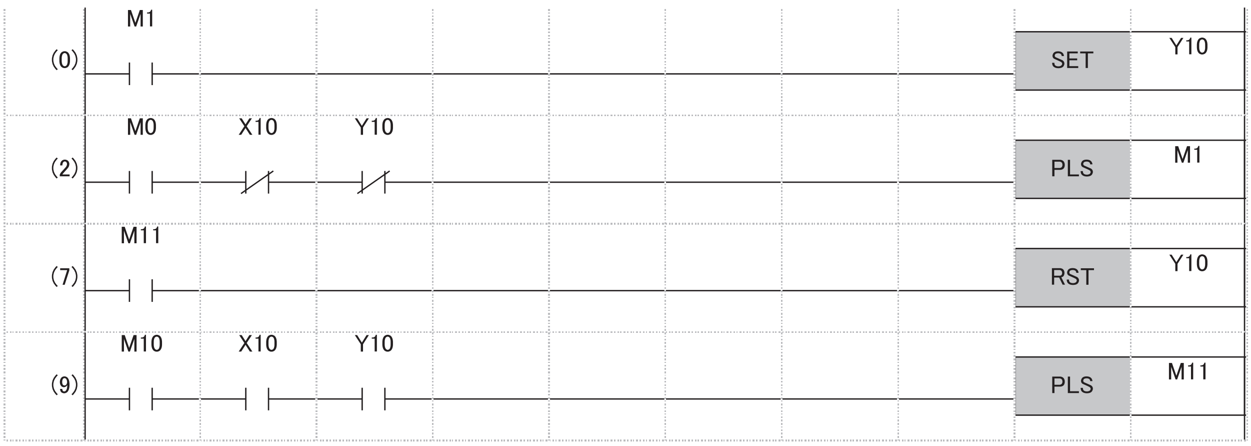


In the following program, turning on M0 turns on the output (Y10) and turning on M10 turns off the output (Y10).

[Without measures]



[With measures]



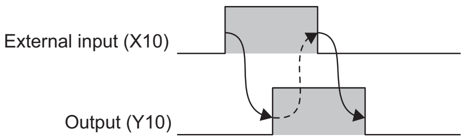
(2) PLS M1 delays SET Y0 by one scan.
(9) PLS M11 delays RST Y10 by one scan.

Point

In the CPU parameter, select "Transfer" (default setting) in "Signal Flow Memory Tracking Setting" of "Redundant System Settings". (☞ Page 728 Tracking transfer setting for the signal flow memory)

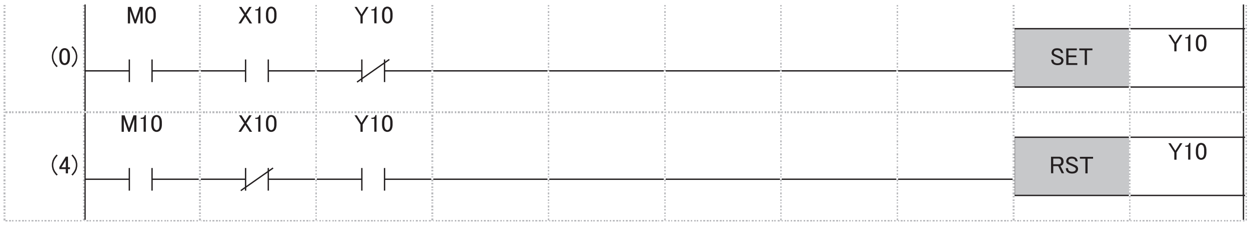
Ex.

When output is returned to external input

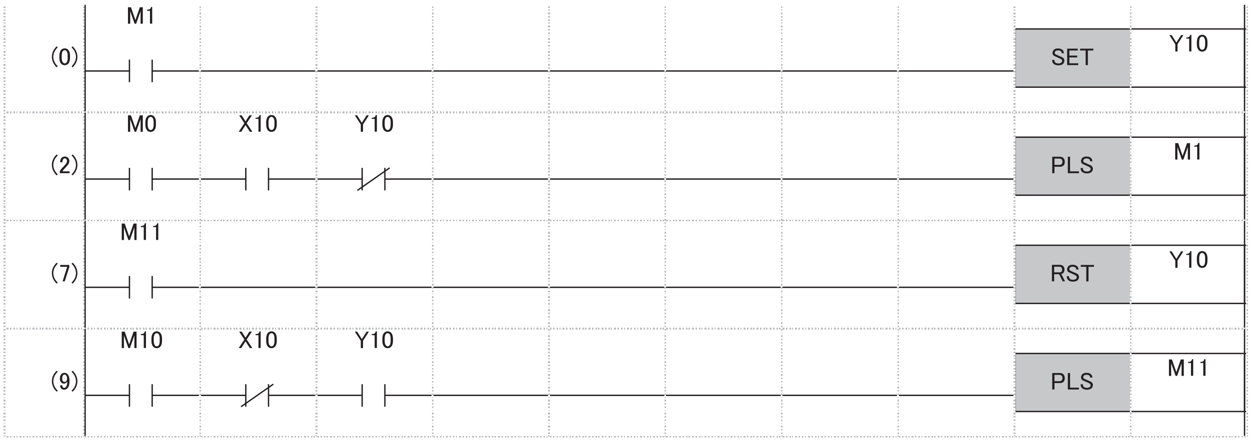


In the following program, turning on M0 turns on the output (Y10) and turning on M10 turns off the output (Y10).

[Without measures]



[With measures]



(2) PLS M1 delays SET Y10 by one scan.

(9) PLS M11 delays RST Y10 by one scan.

Point

In the CPU parameter, select "Transfer" (default setting) in "Signal Flow Memory Tracking Setting" of "Redundant System Settings". (☞ Page 728 Tracking transfer setting for the signal flow memory)

40

40.6 Precautions on Writing Data from GOT or External Devices

When data is written from the GOT or external devices, the tracking data may not be applied to the CPU module of the new control system depending on the timing of system switching cause to be caused, such as power-off.
Consequently, data that is written from the GOT or external devices before system switching may be lost. Write the data again after the system switching.

40.7 Precautions on Outputting in the Middle of the Scan

When values are output during execution of a program with the following devices or setting, the program will be executed again from the step 0 in the new control system after system switching. Therefore, values may be output twice, before and after system switching. Output results may differ between the two outputs.

- Link direct device (Jn\Y)
- Direct access output (DY)
- Refresh at execution of a specified program ("Refresh Group Setting" of "Program Setting")

Before outputting values with one of the above devices or setting, check that the above operation does not cause any problem in the system design phase. If any problem occurs, do not perform the external output until the status of the received output signal becomes stable, or take other measures with external circuits.

Point

In the program of a system using the SIL2 Process CPU, performing the output refresh by the END processing is recommended instead of outputting with the link direct device (Jn\Y), direct access output (DY), or refresh at execution of a specified program ("Refresh Group Setting") in the middle of the scan.

41 MAINTENANCE AND INSPECTION FOR A SYSTEM USING SIL2 PROCESS CPU

This chapter describes the maintenance and inspection for a system using the SIL2 Process CPU.

41.1 Module Replacement in a System Using the SIL2 Process CPU

This section describes the methods of replacing modules in a system using the SIL2 Process CPU while the system is operating.

- Replacing modules in a control system with the online module change function
- Replacing modules in a standby system (To replace modules in a standby system, power off the standby system or use the online module change function.)

○: Replaceable, ×: Not replaceable

Replacement target	Replacement		Reference
	Control system	Standby system	
CPU module and SIL2 function module*4	×	○	Page 800 Replacing a CPU module and SIL2 function module
Power supply module	×	○	Page 800 Replacing a power supply module
Redundant power supply module	○*1	○	Page 801 Replacing a redundant power supply module
Redundant function module	○*3	○*2	Page 801 Replacing a redundant function module
Network module	×	○	Page 801 Replacing a network module
Main base unit	×	○	Page 802 Replacing a main base unit

*1 This module in a redundant power supply system can be replaced.
*2 To replace this module without powering off the system, use the online module change function. (MELSEC iQ-R Online Module Change Manual)
*3 When an error has been detected on a redundant function module in the control system, a continuation error occurs on the CPU module, and the control system and the standby system continue operating without being switched. Perform online module change if the redundant function module has failed. (MELSEC iQ-R Online Module Change Manual)
*4 Since the CPU module and SIL2 function module are used as a pair, replace them at the same time.

Point

Since the standby system needs to be powered off for replacing their modules, a continuation error occurs on the CPU module of the control system. After the replacement, clear the continuation error that has occurred on the CPU module of the control system. (Page 145 Error Clear)

The following describes how to replace modules in the standby system (by powering off the standby system).

Replacing a CPU module and SIL2 function module

This section describes the procedure for replacing a CPU module and SIL2 function module.

1. Check the system of the replacement target CPU module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target CPU module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED is on on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes.

3. Replace the CPU module and SIL2 function module in the standby system.

Replace the CPU module and SIL2 function module in the standby system with a new CPU module and SIL2 function module with the same models as those of the CPU module and SIL2 function module in the control system. The pair versions of the CPU module and SIL2 function must be the same. If an SD memory card or an extended SRAM cassette has been inserted to the replacement target CPU module, insert it to the new CPU module.

4. Change the switch status of the CPU module in the standby system.

Set the RUN/STOP/RESET switch of the CPU module in the standby system to the RUN position.

5. Power on the standby system.

Power on the standby system.

6. Execute the memory copy from the control system to the standby system.

Execute the memory copy using the engineering tool. (☞ Page 736 Memory Copy from Control System to Standby System)

When the memory copy is completed, the MEMORY COPY LED of the redundant function module in the control system turns off and that in the standby system turns on.

7. Restart the standby system.

Restart the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on.

8. Check the safety operation mode.

Change the safety operation mode of the CPU module in the standby system to SAFETY MODE. (☞ Page 756 Switching Safety Operation Mode)

Replacing a power supply module

This section describes the procedure for replacing the power supply module.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED is on on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes.

3. Replace the power supply module in the standby system.

Replace the power supply module in the standby system.

4. Power on the standby system.

Check if wiring to the power supply module and the power supply voltage are correct, and power on the standby system.

When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on.

Replacing a redundant power supply module

Select the standby system or control system and power off and replace the redundant power supply module in the system. After that, power off and replace the module in the other system. Since the redundant power supply module that is not the replacement target supplies power to the modules on the base unit, controls can be continuously performed during replacement of the other redundant power supply module.
For the replacement procedure, refer to the following.
📖 MELSEC iQ-R Module Configuration Manual

Replacing a redundant function module

This section describes the procedure for replacing the redundant function module.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED is on on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes.

3. Replace the redundant function module in the standby system.

Replace the redundant function module in the standby system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on.

5. Check the modules for errors.

Check that no error has occurred on the CPU module or redundant function module in the standby system. If the ERROR LED of either of the modules is on or flashes, check the cause of an error and eliminate the error cause.

Replacing a network module

This section describes the procedure for replacing the network module.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED is on on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes.

3. Replace the network module in the standby system.

Replace the network module in the standby system with a new network module whose model is the same as that of the network module in the control system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on.

5. Perform network diagnostics on the standby system.

Check that no error has occurred on the network module.

Replacing a main base unit

This section describes the procedure for replacing the main base unit.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED is on on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes.

3. Replace the main base unit in the standby system.

Replace the main base unit in the standby system with a new main base unit whose model is the same as that of the main base unit in the control system. When mounting modules to the new base unit, mount each module to the slots of the unit in the same order as that of the base unit of the control system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on.

5. Check the modules for errors.

Check that no error has occurred on the CPU module or redundant function module in the standby system. If the ERROR LED of either of the modules is on or flashes, check the cause of an error and eliminate the error cause.

APPENDICES

Appendix 1 Error Codes

The CPU module stores the corresponding error code in the special register (SD) upon detection of an error by the self-diagnostic function. If an error occurs when the data communications are requested from the engineering tool, intelligent function module, or network system connected, the CPU module returns the corresponding error code to the request source. The error details and cause can be identified by checking the error code. The error code can be checked in either of the following ways:

- Module diagnostics by the engineering tool (MELSEC iQ-R CPU Module User's Manual (Startup))
- Special register (SD0 (Latest self-diagnostics error code), SD10 to SD25 (Self-diagnostic error number)) (Page 966 List of Special Register Areas)

This section describes errors that may occur in the CPU module and actions to be taken for the errors.

Error code system

All error codes are given in hexadecimal format (4 digits) (16-bit unsigned integer). The type of error includes the error, which is detected through the self-diagnostic function of each module, and the common error, which is detected during data communications between modules. The following table lists the error detection type and the error code ranges.

Error detection type		Range	Description
By the self-diagnostic function of each module	Minor error	1000H to 1FFFH	Error code specific to each module, such as self-diagnostic errors
	Moderate error	2000H to 3BFFH	
	Major error	3C00H to 3FFFH	
During data communications between modules		4000H to 4FFFH	Error in the CPU module
		6F00H to 6FFFH	Error in the redundant function module
		7000H to 7FFFH	Error in the serial communication module
		B000H to BFFFH	Error in the CC-Link module
		C000H to CFBFH	Error in the Ethernet-equipped module
		CFC0H to CFFFH	Error in CC-Link IE Field Network Basic
		D000H to DFFFH	Error in the CC-Link IE Field Network module
		E000H to EFFFH	Error in the CC-Link IE Controller Network module
		F000H to FFFFH	Error in the MELSECNET/H network module or MELSECNET/10 network module

A

Detailed information

Upon detection of an error by the self-diagnostic function, the detailed information of the error cause is stored together with an error code. The detailed information can be checked using the engineering tool. The following detailed information is added to each error code. (Up to two types of information are stored together with an each error code. The types differ depending on the error code.) Detailed information 1 and 2 for the latest error code can also be checked in the special register (SD). (☞ Page 966 List of Special Register Areas)

Detailed information	Item	Description
Detailed information 1	Error location information[*1]	Information on the location in a program, such as step numbers
	Drive/file information	Information on the corresponding drive name and file name
	Parameter information	Information on the parameter, such as parameter storage location and parameter type
	System configuration information	Information on the system configuration, such as I/O numbers and power supply numbers
	Safety station system configuration information	Information on the own station or other stations at a safety communication error
	Frequency information	Information on the frequency, such as the number of writes to memory
	Time information	Information on time
	TEST MODE continuous RUN prevention setting	Continuous RUN allowable time in TEST MODE
	Failure information	Information on failures
	System switching information	Information on the system switching cause or system switching failure cause
	Data type (tracking transfer) information	Information on tracking transfer setting data
	Tracking transfer trigger information	Information on tracking block numbers when data are transferred
	Extension cable information	Information on the base unit to which the extension cable in which the error occurs is connected
Detailed information 2	Drive/file information	Information on the corresponding drive name and file name
	Annunciator information	Information on the annunciator areas
	Parameter information	Information on the parameter, such as parameter storage location and parameter type
	System configuration information	Information on the system configuration, such as I/O numbers and power supply numbers
	Process control instruction processing information	Information on processing blocks of the process control instructions
	Error information of other stations (CC-Link IE TSN/CC-Link IE Field)	Information on errors occurred in other stations during safety communications[*2]
	Program error information	Program error code

*1 The step number which is displayed in the error location information, is the one that is counted from the start of the file. It may differ from the step number of the program which is displayed in error jump of the engineering tool.

*2 For details on error classifications and error codes, refer to the manual of each device connected. Note that if the connected device is a SIL2 Process CPU or a Safety CPU, and the error classification is "350", the error code that occurred at the connection destination is displayed for the error code in decimal notation. Furthermore, "0s" in the occurrence date and time information, and in high-order digits are omitted. For example, if the occurrence time (hhmmss) is 09:10:05, the displayed value will be 91005.

Operation when an error occurs

There are two types of errors: continuation errors and stop errors.

Stop error

If a stop error occurs, the CPU module stops its operation and the operating status changes to STOP. Modules can communicate with the CPU module even after the stop error occurs in the CPU module. The external output of each module is controlled in accordance with the output mode setting in error. (☞ Page 139 CPU module operation upon error detection setting) For the case where the stop error occurs in the multiple CPU system, the stop error target CPU module (all CPU modules or only the error CPU module) can be set in parameter. (☞ Page 329 Stop setting)

Continuation error

If a continuation error occurs, the CPU module continues its operation. (The operating status remains the same.)

Safety output operation when error occurs

If a stop error occurs, safety output turns off, and safety communications do not function. Other safety input device/safety label (including internal safety devices) data is retained.

How to clear errors

Continuation errors can be cleared. For the error clear method, refer to the following.

- CPU module: ☞ Page 145 Error Clear
- SIL2 function module: ☞ Page 773 Error clear
- Safety function module: ☞ Page 650 Error clear

When a redundant system or a system using the SIL2 Process CPU is configured, an error in the standby system can be cleared from the control system CPU module by using SM1679 (Error reset (the other system)) in programs or external devices. (☞ Page 561 Clearing errors on the standby system CPU module from the control system CPU module, ☞ Page 775 Clearing errors on the standby system CPU module from the control system CPU module)

A

List of error codes

Codes of errors detected by the self-diagnostic function (1000H to 3FFFH)

The following table lists error codes detected by the self-diagnostic function.

Error code	Error name	Error details and cause	Stop/continue	Action	Detailed information	Diagnostic timing
1000H	Power shutoff	• A momentary power failure has occurred. • The power supply has been shut off.	Continue	• Check the power supply status.	—	Always
1010H	Power shutoff (either of the redundant power supply modules)	• Power has been shut off or the power supply voltage has dropped in either of the redundant power supply modules on the redundant power supply base unit. Or, only one redundant power supply module is mounted.	Continue	• Check the power supplied to the power supply modules on the base unit. • Check that two power supply modules are mounted on the base unit. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative. • If error detection is not desirable, change the error detection setting in the CPU parameters.	System configuration information	Always
1020H	Failure (either of the redundant power supply modules)	• A failure has been detected in either of the redundant power supply modules on the redundant power supply base unit.	Continue	• The possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative. • If error detection is not desirable, change the error detection setting in the CPU parameters.	System configuration information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1030H	Invalid power supply module	• An invalid power supply module has been mounted on the redundant power supply base unit.	Continue	• Mount only applicable power supply modules. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.	System configuration information	Always
1031H	Power supply module configuration error	• A power supply module other than the redundant power supply module has been mounted on the redundant power supply base unit.	Continue	• Mount only applicable power supply modules. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.	System configuration information	Always
1080H	ROM write count error	■CPU module • The number of writes to the flash ROM (data memory, program memory, and system memory (memory used by the system when the CPU module executes its function)) exceeded 100000 times. • Files are frequently written by the event history function. ■SIL2 function module, safety function module • The number of writes to the flash ROM exceeded 100000 times. (Number of writes > 100000)	Continue	■CPU module • Replace the CPU module. ■SIL2 function module • Replace the SIL2 function module. ■Safety function module • Replace the safety function module.	Frequency information	At power-on, at RESET, at write
1090H	Battery error	• The voltage of the battery built in the CPU module has dropped below the specified value. • The connector of the battery built in the CPU module is disconnected. • The connector of the battery built in the CPU module is not securely connected.	Continue	• Replace the battery. • Connect the battery connector to use the file storage area in the device/label memory or the latch function. • Check the connection status of the battery connector. If it is loose, securely connect the connector. • If error detection is not desirable, change the error detection setting in the CPU parameters.	—	Always
1100H	Memory card access error	• Data cannot be written to the memory card because the write protect switch is locked.	Continue	• Unlock the write protect switch of the memory card.	—	Always
1120H	SNTP clock setting error	• Time setting has failed when the CPU module is powered on or reset.	Continue	• Check if the time is correctly set in parameter. • Check if the specified SNTP server is operating normally and there is no failure on the network accessing the SNTP server computer.	—	At power-on, at RESET
1124H	Default gateway/ gateway IP address error	• The default gateway is not set correctly. • The gateway IP address is not set correctly. • The default gateway/ gateway IP address (network address after the subnet mask) is different from that of the IP address of the own node.	Continue	• Correct the default gateway IP address. • Set the same network address as that of the IP address.	Parameter information	Always
1128H	Own node port number error	• The port number is incorrect.	Continue	• Correct the port number.	—	Always
1129H	Open specification port number error	• The port number setting of the external device is incorrect.	Continue	• Correct the port number of the external device.	—	Always
112CH	All-station specification error	• Request with all stations specification failed.	Continue	• Execute request with the currently specified station.	—	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
112DH	Specified IP address error	• The IP address setting of the external device for the open processing is incorrect.	Continue	• Correct the IP addresses. • Check if the class of the IP address is set to A/B/C.	—	Always
112EH	Connection establishment failed	• A connection could not be established in the open processing.	Continue	• Check the operation of the external device. • Check if the open processing has been performed in the external device. • Check and correct the port number of the module, IP address/port number of the external device, and opening method. • When the firewall is set in the external device, check if the access is permitted. • Check if the Ethernet cable is disconnected. • If error detection is not desirable, turn on the error detection invalidation setting (bit 0 of SD49).	—	Always
1133H	Socket communications response send error	• The response send failed during socket communications.	Continue	• Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the Ethernet communication load is high because of the Ethernet functions being executed. Use the Ethernet functions so that the communication load is reduced. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • If error detection is not desirable, turn on the error detection invalidation setting (bit 0 of SD49).	—	Always
1134H	TCP connection timeout	• A TCP ULP timeout error has occurred in the TCP/IP communication. (The external device does not send an ACK response.)	Continue	• Check the operation of the external device. • Check and correct the TCP ULP timeout value. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the Ethernet communication load is high because of the Ethernet functions being executed. Use the Ethernet functions so that the communication load is reduced. • Check if the connection cable is disconnected. • If error detection is not desirable, turn on the error detection invalidation setting (bit 0 of SD49).	—	Always
1152H	IP address error	• The IP address is not set correctly.	Continue	• Correct the IP addresses.	Parameter information	Always
1155H	Connection number acquisition error	• The specified connection was already closed in TCP/IP communications. • Open processing is not performed.	Continue	• Perform the open processing for the specified connection. • Check if the open processing has been performed in the external device. • If error detection is not desirable, turn on the error detection invalidation setting (bit 0 of SD49).	—	Always
1157H	Receive buffer securement error	• The specified connection was already closed in UDP/IP communications. • Open processing is not performed.	Continue	• Perform the open processing for the specified connection. • Check if the open processing has been performed in the external device. • If error detection is not desirable, turn on the error detection invalidation setting (bit 0 of SD49).	—	Always

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1165H	UDP/IP send failed	• Data was not sent correctly with UDP/IP.	Continue	• Check the settings for connection with the external device. • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the Ethernet communication load is high because of the Ethernet functions being executed. Use the Ethernet functions so that the communication load is reduced. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute a PING test. If the test completes with an error, take an action to correct the error. • If error detection is not desirable, turn on the error detection invalidation setting (bit 0 of SD49).	—	Always
1166H	TCP/IP send failed	• Data was not sent correctly with TCP/IP.	Continue	• Check the settings for connection with the external device. • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the Ethernet communication load is high because of the Ethernet functions being executed. Use the Ethernet functions so that the communication load is reduced. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute a PING test. If the test completes with an error, take an action to correct the error. • If error detection is not desirable, turn on the error detection invalidation setting (bit 0 of SD49).	—	Always
1167H	Unsend data send error	• Unsent data found, but could not be sent.	Continue	• Check the settings for connection with the external device. • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the Ethernet communication load is high because of the Ethernet functions being executed. Use the Ethernet functions so that the communication load is reduced. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute a PING test. If the test completes with an error, take an action to correct the error.	—	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1180H	Redundant IP address error	• The same IP address has been set as the system A IP address, system B IP address, and/or control system IP address • Network addresses of the system A IP address, system B IP address, and control system IP address are different.	Continue	• Set different IP addresses for the system A IP address, system B IP address, and control system IP address. • Set the same network address for the system A IP address, system B IP address, and control system IP address.	Parameter information	At power-on, at RESET
11A0H	PID operation error	• A value outside the range ($T_S \leq 0$) was specified for the sampling time (T_S).	Continue	• Check and correct the sampling time (T_S) value.	Error location information	At instruction execution
11A1H	PID operation error	• A value outside the range (α < 0 or 100 $\leq \alpha$) was specified for the input filter constant (α).	Continue	• Check and correct the input filter constant (α) value.	Error location information	At instruction execution
11A2H	PID operation error	• A value outside the range (K_P < 0) was specified for the proportional gain (K_P).	Continue	• Check and correct the proportional gain (K_P) value.	Error location information	At instruction execution
11A3H	PID operation error	• A value outside the range (T_I < 0) was specified for the integral time (T_I).	Continue	• Check and correct the integral time (T_I) value.	Error location information	At instruction execution
11A4H	PID operation error	• A value outside the range (K_D < 0 or 201 $\leq K_D$) was specified for the derivative gain (K_D).	Continue	• Check and correct the derivative gain (K_D) value.	Error location information	At instruction execution
11A5H	PID operation error	• A value outside the range (T_D < 0) was specified for the derivative time (T_D).	Continue	• Check and correct the derivative time (T_D) value.	Error location information	At instruction execution
11A6H	PID operation error	• A value less than or equal to the operation cycle was specified for the sampling time (T_S).	Continue	• Check and correct the sampling time (T_S) value.	Error location information	At instruction execution
11A7H	PID operation error	• A process value variation (ΔPV) overflow occurred.	Continue	• Check and correct the control data setting values.	Error location information	At instruction execution
11A8H		• A deviation (EV) overflow occurred.				
11A9H		• An integral calculation value overflow occurred.				
11AAH		• A derivative gain (K_D) value overflow occurred.				
11ABH	PID operation error	• A differential calculation value overflow occurred.	Continue	• Check and correct the control data setting values.	Error location information	At instruction execution
11ACH		• A PID operation result overflow occurred.				
11ADH	PID operation error	• A value smaller than the output lower limit setting value was specified for the output upper limit setting value.	Continue	• Check and correct the output upper limit setting value or output lower limit setting value.	Error location information	At instruction execution
11AEH	PID operation error	• A value smaller than 0 was specified for the input variation amount warning setting value or output variation amount warning setting value.	Continue	• Change the input variation amount warning setting value or output variation amount warning setting value to a value greater than 0.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
11AFH	PID operation error	• The deviation when starting auto tuning (step response method) became 150 or less. • The deviation when finishing auto tuning (step response method) was 1/3 or greater than the deviation when starting.	Continue	• Check and correct the control data. • If the deviation when starting auto tuning (step response method) is not 150 or more, specify a set value for auto tuning.	Error location information	At instruction execution
11B0H	PID operation error	• The movement direction assumed from the relationship between the set value and process value when starting auto tuning (step response method) and the output value movement direction do not match.	Continue	• Perform auto tuning again after correcting the relationship between the set value, auto tuning (step response method) output value, and process value. • Check and correct the control data.	Error location information	At instruction execution
11B1H	PID operation error	• The input value (PV) did not change correctly, and auto tuning (step response method) did not function normally.	Continue	• Set a long time for the sampling time (T_S). • Set a larger value for the input filter constant (α). • Check the input value (PV).	Error location information	At instruction execution
11B2H	PID operation error	• A value equal to or lower than the output lower limit (LLV) was specified for the auto tuning (limit cycle method) output upper limit (ULV).	Continue	• Check and correct the output upper limit (ULV) and output lower limit (LLV) values.	Error location information	At instruction execution
11B3H	PID operation error	• A value outside the range (SHPV < 0) was specified for the auto tuning (limit cycle method) process value threshold (hysteresis) width (SHPV).	Continue	• Check and correct the process value threshold (hysteresis) width (SHPV) setting value.	Error location information	At instruction execution
11B4H	PID operation error	• The system area used for auto tuning (limit cycle method) was rewritten.	Continue	• Check whether the occupied system area has been rewritten with the PID instruction.	Error location information	At instruction execution
11B5H	PID operation error	• As the auto tuning (limit cycle method) process time was exceeded, it was not possible to properly acquire the τ and τon time. (τon > τ, τon < 0, τ < 0)	Continue	• Increase the difference (ULV-LLV) between the auto tuning output value upper and lower limits, or reduce the input filter constant (α) and auto tuning process value threshold (hysteresis) width (SHPV) values.	Error location information	At instruction execution
11B6H	PID operation error	• An overflow occurred with the proportional gain (K_P) calculated when performing auto tuning (limit cycle method).	Continue	• Ensure that the process value (PV) change increases. • Reduce the difference (ULV-LLV) between the auto tuning output value upper and lower limits.	Error location information	At instruction execution
11B7H	PID operation error	• The integral time (T_I) calculated when performing auto tuning (limit cycle method) was outside the 0 to 32767 range.	Continue	• Increase the difference (ULV-LLV) between the auto tuning output value upper and lower limits, or reduce the input filter constant (α) and auto tuning process value threshold (hysteresis) width (SHPV) values.	Error location information	At instruction execution
11B8H		• The derivative time (T_D) calculated when performing auto tuning (limit cycle method) was outside the 0 to 32767 range.				
1200H	Module moderate error	• A moderate error has been notified from the intelligent function module connected.	Stop/ continue	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, identify the error module, and eliminate the error cause.	System configuration information	Always
1210H		• An inter-module synchronous signal error has been notified from the intelligent function module connected.	Continue			

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1220H	Another CPU module moderate error	• A moderate error has been notified from another CPU module.	Stop/ continue	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, identify the error module, and eliminate the error cause. • Check the mounting status and reset status of other CPU modules.	System configuration information	Always
1240H	Inter-module synchronization processing error	• The execution interval of a synchronous interrupt program has exceeded the set value. • The inter-module synchronous interrupt program (I44) did not complete within the inter-module synchronization cycle.	Continue	• Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take the following actions: (1) Check and correct the processing of the interrupt program (I44) so that it can be completed within the period specified in the fixed scan interval setting. (2) Check and correct the fixed scan interval setting value. • If error detection is not desirable, change the error detection setting in the CPU parameters.	Time information	At interrupt occurrence
1241H	Inter-module synchronization processing error	• The execution interval of a synchronous interrupt program has exceeded the set value. • A cycle where the inter-module synchronous interrupt program (I44) was not executed was detected.	Continue	• Check and correct the interrupt disable sections and the interrupt programs with a high priority so that the inter-module synchronous interrupt program can be executed. • If error detection is not desirable, change the error detection setting in the CPU parameters.	—	At interrupt occurrence
1260H	Multiple CPU synchronization processing error	• The execution interval of a synchronous interrupt program has exceeded the set value. • The multiple CPU synchronous interrupt program (I45) did not complete within the multiple CPU fixed scan communication cycle.	Continue	• Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take the following actions: (1) Check and correct the processing of the interrupt program (I45) so that it can be completed within the period specified in the fixed scan interval setting. (2) Check and correct the fixed scan interval setting value. • If error detection is not desirable, change the error detection setting in the CPU parameters.	Time information	At interrupt occurrence
1262H		• The program execution section of a synchronous interrupt program has been exceeded. • The multiple CPU synchronous interrupt program (I45) did not complete within the program execution section.				
1800H	Annunciator ON	• The annunciator (F) on status has been detected.	Continue	• Check the detailed error information (annunciator information) by executing module diagnostics using the engineering tool, and correct the program corresponding to the displayed annunciator number.	Error location information and annunciator information	At instruction execution
1810H	Operation continuation error	• The PALERT(P) instruction was executed.	Continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool and check the corresponding program - one that outputs the program error code.	Error location information and program error information	At instruction execution
1811H	Control CPU error	• An error has been detected in the CPU module.	Continue	• Check the error details by executing module diagnostics using the engineering tool and take an action.	—	Always

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1812H	Module error	• Tracking communications could not be performed successfully due to an error in the redundant function module.	Continue	• The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	—	Always
1830H	Receive queue full	• The number of transient receive request exceeded the upper limit of simultaneously processable requests.	Continue	• Reduce the frequency of transient transmission, and perform transmissions again. • Increase the frequency of transient transmission by using the COM instruction.	—	Always
1831H	Receive processing error	• Transient reception failed.	Continue	• Reduce the frequency of transient transmission, and perform transmissions again.	System configuration information	Always
1832H	Transient data error	• Too much transient transmission processing exists and transient transmission cannot be performed.	Continue	• Correct the number of transient transmission executions.	—	Always
1845H	Transient data buffer full	• Too much transient transmission processing exists and transient transmission cannot be performed.	Continue	• Correct the number of transient transmission executions.	—	Always
1860H	Network error	• Tracking communications stopped due to an error in the network or in the redundant function module.	Continue	• Check the network status by executing module diagnostics using the engineering tool and take an action. • If the error code is displayed again even after taking an action, please consult your local Mitsubishi representative.	—	Always
1861H	Tracking communication error	• Tracking communications could not be performed successfully due to an error in the network or in the redundant function module.				
1900H	Constant scan time error	• The scan time exceeded the constant scan time set in the CPU parameters.	Continue	• Check and correct the constant scan time setting.	Time information	At END instruction execution
1A00H	Safety cycle time exceeded	• Processing of the safety program, from safety input to safety output, did not complete within the safety cycle time.	Continue	• Check and correct the safety cycle time.	Time information	At interrupt occurrence
1A01H	Safety cycle processing error	• The safety cycle processing takes too much time, and the safety cycle time was exceeded. • Safety cycle processing was not executed because of execution of the following instruction or function: (1) Instruction with a long processing time (2) Device test with execution condition (3) Recording function			—	
1A20H	Continuous RUN time in TEST MODE exceeded	• The continuous RUN time in TEST MODE was exceeded.	Continue	• Change the mode to SAFETY MODE.	TEST MODE continuous RUN prevention setting	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1A40H	Destination station error in safety communications	• Error information has been received from a safety communication destination station. • The safety protocol version of the device actually connected to perform safety communications and the version set in the safety communication setting do not match.	Continue	■SIL2 Process CPU, SIL2 function module • Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check "Station No.", "Slot No. (another station)", and "Base No. (another station)". If the destination station is an intelligent device station, execute the system monitor for the remote head module in the station and check the error details. ■Safety CPU, safety function module • Check the numerical value (station number) in the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and take either of the following actions. Note that when the destination station does not detect an error, the safety protocol version that is not supported by the station is specified in the safety communication setting. Change the safety protocol version in the relevant safety communication setting to match the version that is supported by the safety communication destination station, and write parameters to the destination station and CPU module. (1) If the destination station is a remote device station or remote I/O station, read error history data of the destination station by executing a command from the CC-Link IE TSN Configuration window or the CC IE Field Configuration window, and check the details of the error. (2) If the destination station is a master/local station, execute the system monitor for the station, and check the details of the error.	System configuration information, safety station system configuration information, error information of other stations (CC-Link IE TSN/CC-Link IE Field)	At interrupt occurrence

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1A50H	Parameter mismatch in safety communication destination station	• The safety approvals codes written to the CPU module and to the safety communication destination station do not match. • The parameters written to the CPU module and to the safety communication destination station do not match.	Continue	■SIL2 Process CPU, SIL2 function module • Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check the displayed station number. Check the safety communication setting method and procedure referring to the manual for the safety communication destination station, and write parameters to the safety communication destination station and the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the module performing safety communications. Please consult your local Mitsubishi representative. ■Safety CPU, safety function module • Check the numerical value (station number) in the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and take either of the following actions. Note that if the same error code is displayed again after taking one of the following actions, the possible cause is a hardware failure of the CPU module or the module performing safety communications. Please consult your local Mitsubishi representative. (1) Check the safety approvals code set in the safety communication setting and the safety approvals code of the unit performing safety communication, and write the parameters to the safety communication destination station and CPU module. (2) Check the safety communication setting method and procedure referring to the manual for the safety communication destination station, and write parameters to the safety communication destination station and the CPU module.	System configuration information, safety station system configuration information	At interrupt occurrence

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1A51H	Device mismatch in safety communication destination station	• The model or production information on the safety communication destination station is incorrect. • The device actually connected to perform safety communications and the device set in the network configuration settings do not match.	Continue	■SIL2 Process CPU, SIL2 function module • Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check the displayed station number. Then, match the connected device and the device set in the network configuration settings. • Write parameters to the safety communication destination station and the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the module performing safety communications. Please consult your local Mitsubishi representative. ■Safety CPU, safety function module • Check the numerical value (station number) in the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and take the following action. Note that if the same error code is displayed again after taking one of the following actions, the possible cause is a hardware failure of the CPU module or the module performing safety communications. Please consult your local Mitsubishi representative. (1) Change the connected device and the device set in the network configuration settings to match, and write parameters to the safety communication destination station and the CPU module.	System configuration information, safety station system configuration information	At interrupt occurrence
1A52H	Device version mismatch in safety communication destination station	• The version of the device actually connected to perform safety communications and the version of the device set in the network configuration settings do not match.	Continue	■SIL2 Process CPU, SIL2 function module • Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check the displayed station number. Then, match the version of the connected device and the version set in the network configuration settings or the profile data. • Write parameters to the safety communication destination station and the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the module performing safety communications. Please consult your local Mitsubishi representative. ■Safety CPU, safety function module • Check the numerical value (station number) in the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and take the following action. Note that if the same error code is displayed again after taking one of the following actions, the possible cause is a hardware failure of the CPU module or the module performing safety communications. Please consult your local Mitsubishi representative. (1) Change the device version of the connected device and the device version or profile data set in the network configuration settings to match, and write parameters to the safety communication destination station and CPU module.	System configuration information, safety station system configuration information	At interrupt occurrence

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1A53H	Safety protocol version mismatch in safety communication destination station	The safety protocol version of the device actually connected to perform safety communications and the version set in the safety communication setting do not match.	Continue	• Check the numerical value (station number) in the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and take the following action. Note that if the same error code is displayed again after taking one of the following actions, the possible cause is a hardware failure of the CPU module or the module performing safety communications. Please consult your local Mitsubishi representative. (1) Change the safety protocol version in the relevant safety communication setting to match the safety communication setting of the safety communication destination station, and write parameters to the safety communication destination station and the CPU module.	System configuration information	At interrupt occurrence

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1A60H to 1A66H	Timeout in safety communications	• A timeout error occurred during safety communications.	Continue	■SIL2 Process CPU, SIL function module • Check and correct the safety refresh monitoring time and the transmission interval monitoring time setting values. • The possible cause is malfunction due to noise or cable failure. Check that there is no error in a transmission path by executing CC-Link IE Field diagnostics. Check the distance of wires and cables and the grounding status of each device as well. • Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check "Station No.", "Slot No. (another station)", and "Base No. (another station)". If the destination station is an intelligent device station, execute the system monitor for the remote head module in the station and check the error details. • Check that no online operation is being performed from a peripheral such as the engineering tool, or no program or parameter is being written to the safety communication destination station. • The possible cause is a hardware failure of the module performing safety communications. Please consult your local Mitsubishi representative. ■Safety CPU, safety function module • Check the numerical value (station number) in the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and take any of the following actions. Note that if the same error code is displayed again after taking one of the following actions, the possible cause is a hardware failure of the module performing safety communications. Please consult your local Mitsubishi representative. (1) Check and correct the safety refresh monitoring time and the transmission interval monitoring time setting values. (2) The possible cause is malfunction due to noise or cable failure. Check that there is no error in a transmission path by executing CC-Link IE TSN/CC-Link IE Field diagnostics. Check the distance of wires and cables and the grounding status of each device as well. (3) Check that no online operation is being performed from a peripheral such as the engineering tool, or no program or parameter is being written to the safety communication destination station. (4) Check the system configuration by referring to the manual for the network module performing safety communications. (5) If the destination station has detected an error and the destination station is a remote device station or remote I/O station, read error history data of the destination station by executing a command from the CC-Link IE TSN Configuration window or the CC IE Field Configuration window, and check the details of the error. If the destination station is a master/local station, execute the system monitor for the station, and check the details of the error.	System configuration information, safety station system configuration information	At interrupt occurrence

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1A70H to 1A73H	Safety communications receipt data error	• The received data is abnormal.	Continue	■SIL2 Process CPU, SIL2 function module • Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check "Station No.", "Slot No. (another station)", and "Base No. (another station)". If the destination station is an intelligent device station, execute the system monitor for the remote head module in the station and check the error details. • The possible cause is a hardware failure of the module performing safety communications. Please consult your local Mitsubishi representative. ■Safety CPU, safety function module • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and check the displayed station number. If the destination station is a remote device station or remote I/O station, read error history data of the destination station by executing a command from the CC-Link IE TSN Configuration window or the CC IE Field Configuration window, and check the details of the error. If the destination station is a master/local station, execute the system monitor for the station, and check the details of the error. • The possible cause is a hardware failure of the module performing safety communications. Please consult your local Mitsubishi representative.	System configuration information, safety station system configuration information	At interrupt occurrence
1A80H	Timeout in safety communications	• Safety communications with a destination station could not be resumed within a safety I/O hold time.	Continue	• Check and correct the safety I/O hold time. • The possible cause is malfunction due to noise or cable failure. Check that there is no error in a transmission path by executing CC-Link IE Field diagnostics. Check the distance of wires and cables and the grounding status of each device as well. • Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check "Station No.", "Slot No. (another station)", and "Base No. (another station)". If the destination station is an intelligent device station, execute the system monitor for the remote head module in the station and check the error details. • The possible cause is a hardware failure of the module performing safety communications. Please consult your local Mitsubishi representative.	Safety station system configuration information	At interrupt occurrence
1AA0H	Safety I/O refresh error	• The CPU module has detected safety I/O refresh error.	Continue	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the mounting status of the module corresponding to the displayed I/O number. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or another module where the error has been detected. Please consult your local Mitsubishi representative.	System configuration information	At interrupt occurrence

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1AB0H to 1AB3H	Safety I/O refresh error	• The CPU module has detected safety I/O refresh error.	Continue	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and check the module corresponding to the displayed I/O number. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or another module where the error has been detected. Please consult your local Mitsubishi representative.	System configuration information	At interrupt occurrence
1AC0H to 1AC6H	Safety I/O refresh timeout	• A timeout error has occurred during safety I/O refresh.	Continue	• Check the numerical value (I/O number) in the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and take any of the following actions. (1) Check and correct the safety I/O refresh timeout time and the safety output refresh monitoring time setting values. (2) Check the mounting status of the module where the error was detected. (3) Eliminate the error cause from the module where the error was detected. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or another module where the error has been detected. Please consult your local Mitsubishi representative.	System configuration information	At interrupt occurrence
1AD0H to 1AD3H	Safety input refresh data error	• The safety input refresh data is abnormal.	Continue	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and check the module corresponding to the displayed I/O number. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or another module where the error has been detected. Please consult your local Mitsubishi representative.	System configuration information	At interrupt occurrence
1B00H	Network configuration mismatch	• The CC-Link IE Field Network configuration differs between the systems A and B. (The error is detected in the standby system.)	Continue	• Check that all network cables are connected correctly. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the network cable. Please consult your local Mitsubishi representative.	System configuration information	Always
1B20H	System consistency check error (operating status)	• The operating status of the CPU module differs between the systems A and B. (The error is detected in the standby system.)	Continue	• Set the same operating status to the CPU modules in both systems.	—	Always
1B21H	System consistency check error (safety operation mode)	• The safety operation mode differs between the systems A and B. • Switching of the safety operation mode is executed for the systems A and B. (The error is detected in the standby system.)	Continue	Set the same safety operation mode to the CPU modules in both systems.	—	Always

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1B40H	Redundant system error	• Both systems were found to be set as system A or system B when tracking cables were connected. (The error is detected in the control system or the standby system.)	Continue	• Set one system as system A and the other system as system B by performing online operation. Then, restart the CPU modules in both systems.	—	Always
1B42H	Redundant system error	• The CPU module operated in redundant mode was connected to the CPU module set to process mode with tracking cables. (The error is detected in the control system or the standby system.)	Continue	• Read the project stored in the CPU module operated in redundant mode, and write it to the CPU module set to process mode. Then, reset the CPU module set to process mode, and run it again.	—	Always
1B43H	Redundant system error	• The firmware version of the CPU module connected is not compatible with the other one. • The Process CPU and the SIL2 Process CPU are connected. • In a redundant system with redundant extension base unit, the extension cable is not connected between the main base unit of one system and the extension base unit. Or, the extension base unit cannot be recognized by the other system. (The error is detected in the control system or the standby system.)	Continue	• Replace one CPU module with the one having a firmware version that can be used with the other CPU module together, referring to the MELSEC iQ-R Module Configuration Manual. Then, restart the system. • Replace one CPU module in either of two systems so that the same CPU module models are used in both systems. Then, write a project to the replaced CPU module and restart the system. • Check the system configurations of both systems and correct them. • Connect the extension cable securely and properly to the connector of the redundant extension base unit, and then restart the CPU module in which the error has been detected. If the same error code is displayed again, the possible cause is a failure of the extension cable. Replace the extension cable.	—	Always
1B48H	Extension cable failure	• The extension cables between the redundant extension base units were not duplicated at start-up. (The error is detected in the control system.)	Continue	• Check the detailed information (extension cable information) of the error by executing module diagnostics using the engineering tool, and connect additional extension cables to the extension cable connectors to duplicate the extension cables. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or extension cable. Please consult your local Mitsubishi representative.	Extension cable information	At power-on, at RESET
1B4AH	Extension cable failure	• An error of the extension cable (inactive side) between the redundant extension base units has been detected. (The error is detected in the control system.)	Continue	• Check the detailed information (extension cable information) of the error by executing module diagnostics using the engineering tool, and connect the extension cable securely and properly to the connectors of the base units. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or extension cable. Please consult your local Mitsubishi representative.	Extension cable information	Always
1B50H	Parameter error (redundant function)	• In a redundant system with redundant extension base unit, the module extension parameter used by the modules mounted on the extension base unit is written in the CPU module. (The error is detected in the control system or the standby system.)	Continue	• Delete the module extension parameter from the CPU module and write it to modules on the extension base unit.	—	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1B60H	Standby system CPU module error	• In backup mode, the standby system CPU module has not started up. (The error is detected in the control system.)	Continue	• If the standby system is powered off, power on the system. • If the standby system CPU module is in the RESET state, clear the RESET state. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	—	Always
1B61H	Standby system CPU module error	• In backup mode, a stop error of the standby system CPU module was detected. (The error is detected in the control system.)	Continue	• Eliminate the error cause, and restart the system. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	—	Always
1B70H	Tracking communications disabled	• Data communications with the other system cannot be performed even in backup mode. (The error is detected in the control system or the standby system.)	Continue	• If the standby system is powered off, power on the system. • If a WDT error has occurred in the other system CPU module, eliminate the error cause, and restart the system. • Connect tracking cables securely and properly to the connectors of the redundant function modules in both systems. (One cable shall be connected between the IN connector of the module in system A and the OUT connector of the module in system B. The other cable shall be connected between the OUT connector of the module in system A and the IN connector of the module in system B.) • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.	—	Always
1B71H	Tracking communication error	• Data communications with the other system cannot be performed in one of the two tracking cables even in backup mode. (The error is detected in the control system or the standby system.)	Continue	• Connect tracking cables securely and properly to the connectors of the redundant function modules in both systems. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module or tracking cable. Please consult your local Mitsubishi representative.	—	Always
1B78H	Tracking communication error	• During tracking transfer, a tracking communication error occurred. (The error is detected in the control system or the standby system. Note that if the error is detected in the standby system, the detailed information is not stored.)	Continue	• If the other system is powered off, power on the system. • If a WDT error has occurred in the other system CPU module, eliminate the error cause, and restart the system. • Connect tracking cables securely and properly to the connectors of the redundant function modules in both systems. • Set the program execution time of the standby system shorter than that of the control system. If the shorter value cannot be set, correct the SD1662 value. • If the load on the communications with the external devices via the other system is high, perform the communications without going through the other system or lighten the load on the communications. • Check the power supply status. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.	Data type (tracking transfer) information	At END instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1B80H	Tracking transfer error	• The tracking data set in the CPU parameters exceeds the allowable tracking transfer range. (The error is detected in the control system.)	Continue	• Set the device/label data within the allowable range in the CPU parameters ("Tracking Device/Label Setting").	Tracking transfer trigger information	At END instruction execution
1B81H	Tracking transfer error	• The file register capacity has been set in the CPU parameters of the control system CPU module is less than the file register capacity set for tracking transfer. (The error is detected in the control system.)	Continue	• Check and correct the tracking transfer settings in the CPU parameters so that they will be within the device range or the file register capacity. • Check and correct the device settings or the file register capacity setting in the CPU parameters.	—	At END instruction execution
1B82H		• The file register data is transferred from the control system CPU module to the standby system CPU module exceeding the capacity of the receive side. (The error is detected in the standby system.)				
1BA0H	Redundant function module error	• An error has been detected in the redundant function module.	Continue	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the status of the redundant function module. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	Always
1BB0H	File name specification error	• In redundant mode, the POFF(P) instruction (Changing the program execution type to standby type (output off)) has been executed for an SFC program in which the program execution type is set to scan execution type. • In redundant mode, the PSCAN(P) instruction (Changing the program execution type to scan execution type) has been executed for an SFC program in which the program execution type is set to standby type.	Continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and check the specified file.	Error location information, drive/file information	At instruction execution
1BC0H	Program execution time error (redundant function)	• In a redundant system with redundant extension base unit, the program execution time of the standby system exceeded 200ms. (The error is detected in the standby system.)	Continue	• Set a program execution time of the standby system to be within 200ms.	Time information	At END instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1BD0H	System switching error	• The systems were not switched. There was a cause of system switching failure. (The error is detected in the control system.)	Continue	• Check the cause of system switching failure in the detailed information (system switching information) of the error by executing module diagnostics using the engineering tool, eliminate the error cause, and switch the systems from the engineering tool again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or tracking cable. Please consult your local Mitsubishi representative.	System switching information	At system switching execution
1BD1H	System switching error	• The systems were not switched by using the SP.CONTSW instruction because SM1646 (System switching by a user) was off. (The error is detected in the control system.)	Continue	• To switch the systems by using the SP.CONTSW instruction, turn on SM1646 and then execute the instruction.	Error location information	At instruction execution
2000H	Module configuration error	• The module type set in the system parameters ("I/O Assignment Setting") differs from that of the module actually mounted.	Stop	• Re-set the module type in the system parameters in accordance with the CPU module or intelligent function module actually mounted.	System configuration information	At power-on, at RESET
2001H	Module configuration error	• The I/O numbers set in the system parameters ("I/O Assignment Setting") are overlapping between modules.	Stop	• Re-set the I/O numbers in the system parameters in accordance with the intelligent function module or I/O module actually mounted.	System configuration information	At power-on, at RESET
2002H	Module configuration error	• The number of points assigned to the intelligent function module in the system parameters ("I/O Assignment Setting") is smaller than that of the module actually mounted.	Stop	• Re-set the number of points in the system parameters in accordance with the intelligent function module actually mounted.	System configuration information	At power-on, at RESET
2004H	Module configuration error	• Nine or more CC-Link IE Controller Network modules and/or MELSECNET/H network modules are mounted in the entire system. (The CC-Link IE built-in Ethernet interface module is included if the module is used as a CC-Link IE Controller Network module.) • Five or more MELSECNET/H network modules are mounted in the entire system.	Stop	• Reduce the number of CC-Link IE Controller Network modules and/or MELSECNET/H network modules to eight or less in the entire system. (The CC-Link IE built-in Ethernet interface module is included if the module is used as a CC-Link IE Controller Network module.) • Reduce the number of MELSECNET/H network modules to four or less in the entire system.	System configuration information	At power-on, at RESET
2005H	Module configuration error	• Two or more interrupt modules (QI60) with no interrupt pointer setting are mounted. • The interrupt pointer numbers are overlapping in the interrupt module (QI60) with no interrupt pointer setting and a module with an interrupt pointer setting.	Stop	• Mount only one QI60 in the entire system. • Configure the interrupt pointer setting for the QI60. • Correct the interrupt pointer setting.	System configuration information	At power-on, at RESET, at STOP → RUN state
2006H	Module configuration error	• A module is mounted on the 65th slot or later.	Stop	• Remove the module mounted on the 65th slot or later.	System configuration information	At power-on, at RESET

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2007H	Module configuration error	• A module is mounted on the slot whose number is later than that specified in the system parameters ("I/O Assignment Setting").	Stop	• Remove the module mounted on the slot whose number is later than that specified in the system parameters.	System configuration information	At power-on, at RESET
2008H	Module configuration error	• A module is mounted over or across the maximum number of I/O points (4096).	Stop	• Remove the module mounted over or across the maximum number of I/O points (4096). • Replace the module mounted on the last slot with the one that does not exceed the maximum number of I/O points (4096).	System configuration information	At power-on, at RESET
2009H	Module configuration error	• There is no response from the I/O module or intelligent function module accessed.	Stop	• Check and correct the I/O assignment setting in the system parameters. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the I/O module or intelligent function module. Please consult your local Mitsubishi representative.	System configuration information	Always
200AH	Module configuration error	• Several different series of MELSECNET/H network modules are managed by one CPU module.	Stop	• The series of MELSECNET/H network modules that are managed by one CPU module should be the same.	System configuration information	At power-on, at RESET
200BH	Module configuration error	• Recorder module or camera recorder module for which the module parameters have not been set is mounted. • There are two or more recorder modules or camera recorder modules that are set to "main". • The recorder module or camera recorder module that is set to "main" does not exist. • The CPU module or the recorder module that does not support the "sub" of the camera recorder module is mounted.	Stop	• Remove the recorder module or the camera recorder module for which the module parameters have not been set. • Use only one recorder module or one camera recorder module set to "main". • Mount the recorder module or the camera recorder module set to "main". • Check the firmware version of the module and use the appropriate version.	System configuration information	At power-on, at RESET
2020H	Module configuration error	• There is a mounted module that is not supported, or there is a mounted module that is not supported by the network type (module model name) set in system parameters ("I/O Assignment Setting").	Stop	• Remove the unsupported module if any. • Check whether modules are supported by the network type (module model name) set in the system parameters. • If all the modules are supported, the possible cause is a hardware failure of the CPU module, base unit, I/O module, or intelligent function module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2021H	Module configuration error	• In a multiple CPU system, the control CPU of the Q series intelligent function module incompatible with the multiple CPU system is set to other than CPU No.1.	Stop	• Replace the Q series intelligent function module with the one (function version B) compatible with the multiple CPU system. • Set the control CPU of the Q series intelligent function module incompatible with the multiple CPU system to CPU No.1.	System configuration information	At power-on, at RESET
2022H	Module configuration error	• A power supply module other than the redundant power supply module has been mounted on the redundant power supply base unit.	Stop	• Mount only the redundant power supply module. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module, CPU module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2040H	CPU module configuration error	• The number of CPU modules set in the system parameters ("I/O Assignment Setting") differs from the number of CPU modules actually mounted. • The CPU module is mounted on the slot different from the one specified in the system parameters (I/O assignment setting). • Two or more Safety CPUs are mounted.	Stop	• Correctly set the number of CPU modules (including the empty setting) in the system parameters in accordance with the number of CPU modules actually mounted. • Correctly set the system parameters so that the setting and actual CPU module mounting status will be the same. • Mount only one Safety CPU.	System configuration information	At power-on, at RESET
2041H	CPU module configuration error	• The CPU module is not mounted on the slot that is set for the CPU module in the system parameters ("I/O Assignment Setting"). • The CPU module is mounted on the slot that is set for empty in the system parameters ("I/O Assignment Setting"). • An I/O module or intelligent function module is mounted between the CPU modules. • Arrangement of CPU modules mounted is not appropriate to configure a multiple CPU system.	Stop	• Check and correct the I/O assignment setting in the system parameters. • Correctly set the number of CPU modules (including the empty setting) in the system parameters in accordance with the number of CPU modules actually mounted. • Remove the I/O module or intelligent function module mounted between the CPU modules.	System configuration information	At power-on, at RESET
2043H	CPU module configuration error	• The CPU module is mounted on the inapplicable slot.	Stop	• Mount the CPU module on the applicable slot (CPU slot or I/O slot 0 to 6). • Remove the CPU module from the inapplicable slot.	System configuration information	At power-on, at RESET
2044H	CPU module configuration error	• The host CPU No. set in the system parameters ("I/O Assignment Setting") differs from the one determined by the mounting position of the CPU module.	Stop	• Re-set the host CPU No. in the system parameters in accordance with the mounting position of the CPU module.	System configuration information	At power-on, at RESET, at STOP → RUN state
2046H	CPU module configuration error	• Another CPU module was accessed, but there was no response.	Stop	• Check and correct the I/O assignment setting in the system parameters. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of another CPU module. Please consult your local Mitsubishi representative.	System configuration information	Always
2050H	CPU module configuration error	• An unsupported CPU module is mounted.	Stop	• Remove the unsupported CPU module. If all the CPU modules are supported, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2051H	CPU module configuration error	• A CPU module that does not support the online module change (direct change) function is used in a multiple CPU system.	Stop	• Refer to the MELSEC iQ-R Online Module Change Manual, and correct the system configuration. • To disable the online module change (direct change) function, set "Direct change setting" to "Disable" in the CPU parameters. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2052H	CPU module configuration error	• A multiple CPU system was configured with CPU modules incompatible with multiple CPU systems.	Stop	• Check and correct the system configuration.	System configuration information	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2060H	Base unit configuration error	• Eight or more extension base units are connected.	Stop	• Reduce the number of extension base units to seven or less.	System configuration information	At power-on, at RESET
2061H	Base unit configuration error	• Any of the following base units is connected: QA1S3□B, QA1S5□B/ QA1S6□B, QA6□B, QA6ADP+A5□B/A6□B, or QA1S6ADP-S1+A1S5□B/ A1S6□B.	Stop	• Remove the inapplicable base unit: QA1S3□B, QA1S5□B/QA1S6□B, QA6□B, QA6ADP+A5□B/A6□B, and QA1S6ADP-S1+A1S5□B/A1S6□B.	System configuration information	At power-on, at RESET
2063H	Base unit configuration error	• Extension base unit levels are overlapping.	Stop	• Check and correct the level setting of the extension base units.	System configuration information	At power-on, at RESET
2070H	Base unit configuration error	• An unsupported base unit is connected. • A GOT is bus-connected to the Q series extension base unit.	Stop	• Check the firmware version of the CPU module, and replace the CPU module with the one supporting the use of redundant extension base units. • Disconnect the unsupported base unit. If all the base units are supported, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative. • Disconnect the GOT bus-connected to the Q series extension base unit.	System configuration information	At power-on, at RESET
2080H	Inter-module synchronization configuration error	• An inter-module synchronization signal error has been detected.	Stop	• The possible cause is a hardware failure of the CPU module, base unit, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
20E0H	Module unrecognized	■CPU module • A module that the CPU module cannot recognize is mounted. • In a multiple CPU system, the module cannot be recognized because the control CPU setting in the system parameters is different from that of other CPU modules. ■Redundant function module • Data communications with the CPU module cannot be performed.	Stop	■CPU module • Mount only applicable modules. • Correct the system parameter settings for the CPU No.2 and later in accordance with those of the CPU No.1. • The possible cause is a hardware failure of the I/O module or intelligent function module accessed. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	System configuration information	Always
2100H	Memory error	• No extended SRAM cassette is inserted. • The capacity of extended SRAM cassette set in does not match "Extended SRAM Cassette Setting" in the CPU parameter with the actual capacity of the cassette inserted.	Stop	• Check that an extended SRAM cassette is inserted. Or, correct the capacity set in "Extended SRAM Cassette Setting" in the CPU parameter so that it matches with the actual capacity of the cassette inserted. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or extended SRAM cassette. Please consult your local Mitsubishi representative.	Drive/file information, parameter information	At power-on, at RESET
2101H	Memory error	• An extended SRAM cassette or a battery-less option cassette is inserted or removed while the programmable controller is powered on.	Stop	• Do not insert or remove an extended SRAM cassette or a battery-less option cassette during operation. • Check if an extended SRAM cassette or a battery-less option cassette is securely inserted to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the extended SRAM cassette or battery-less option cassette. Please consult your local Mitsubishi representative.	Drive/file information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2102H	Memory error	• An error has been detected in the extended SRAM cassette or the battery-less option cassette.	Stop	• Check if an extended SRAM cassette or a battery-less option cassette is securely inserted to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the extended SRAM cassette, battery-less option cassette, or CPU module. Please consult your local Mitsubishi representative.	Drive/file information	Always
2103H	Memory error	• An unsupported extended SRAM cassette or battery-less option cassette is inserted.	Stop	• Replace the extended SRAM cassette or battery-less option cassette with the one supported by the CPU module.	Drive/file information	Always
2104H	Memory error	• A battery-less option cassette with the factory default settings or one that was used in another CPU module is inserted.	Stop	• Initialize the battery-less option cassette. (To initialize the battery, use the engineering tool and select [Online] - [CPU Memory Operation] - [Device/Label Memory].) If the same error code is displayed again, the possible cause is a hardware failure of the battery-less option cassette or CPU module. Please consult your local Mitsubishi representative.	—	At power-on, at RESET
2105H	Memory error	• "Battery-less Option Cassette Setting" in the CPU parameter is set as "Mounted", but no battery-less option cassette is inserted.	Stop	• Check that a battery-less option cassette is inserted. Or, correct "Battery-less Option Cassette Setting" in the CPU parameter. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or battery-less option cassette. Please consult your local Mitsubishi representative.	Drive/file information, parameter information	At power-on, at RESET
2106H	Memory error	• "Battery-less Option Cassette Setting" in the CPU parameter is set as "Not Mounted", but a battery-less option cassette is inserted.	Stop	• Remove the battery-less option cassette. Or, correct "Battery-less Option Cassette Setting" in the CPU parameter. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information, parameter information	At power-on, at RESET
2120H	Memory card error	• The memory card was removed without the card being disabled.	Stop/ continue	• Disable the memory card, and then remove it.	Drive/file information	Always
2121H	Memory card error	• An error has been detected in the memory card.	Stop/ continue	• Format the memory card, re-insert the memory card, or replace the memory card. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information	Always
2122H	Memory card error	• The CPU module did not start because the memory card was not restored during the startup processing.	Stop	• Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the memory card. Replace the memory card.	Drive/file information	At power-on, at RESET
2150H	Initial processing time error	• Since the initial processing takes time, the host CPU in a multiple CPU system cannot respond to communication requests from other CPU modules in initial communications. (Other CPU modules cannot start up.)	Stop	Take the following action to shorten the initial processing time. • Check and correct the restoration setting and the number of files to be restored if the automatic data restoration function is being executed.	—	At power-on, at RESET
2180H	Invalid file	• An invalid file has been detected.	Stop	• Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, select the correct file name, and write the specified file to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Drive/file information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2181H	Invalid file	• Due to firmware update or restoration (after the firmware update) of the CPU module, a program file of which file structure is not supported by the new version of the firmware or the operation mode of the CPU module has been written.	Stop	• After reading program files from the programmable controller, format the memory and write the program files. Then, reset the CPU module and run it again.	Drive/file information	At power-on, at RESET
2182H	Invalid file	• The program file is incorrect. Or, the program file is not correctly written.	Stop	• Write the program file to the CPU built-in memory again.	Drive/file information	At power-on, at RESET
21A0H	File specification error	• The file specified in the CPU parameters does not exist. • The memory card is disabled by tuning on SM606 (SD memory card forced disable instruction). • The file register file does not exist in the specified memory when the file register setting is set to "Use Common File Register in All Programs" and the file capacity is not set in the CPU parameters ("File Setting"). • The file specified in the memory card parameters ("Boot Setting") does not exist in the memory card.	Stop	• Turn off SM606. (Cancel the disabled state.) • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, select the correct file name, and write the specified file to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, memory card, or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Drive/file information, parameter information	At instruction execution, at interrupt occurrence, at power-on, at RESET, at STOP → RUN state, at END instruction execution
21A1H	File specification error	• The file specified in parameter cannot be created.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the name and size of the file corresponding to the displayed parameter number. • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and take the following actions: (1) Format the corresponding drive. (2) Delete unnecessary files on the corresponding drive to increase free space. (3) Unlock the corresponding drive if it is locked.	Drive/file information, parameter information	At write, at power-on, at RESET, at STOP → RUN state
21A2H	File specification error	• The CPU module model set to the file using the engineering tool differs from that of the CPU module actually mounted.	Stop	• Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and correct the CPU module model set to the file in accordance with that of the CPU module actually mounted.	Drive/file information	At write, at power-on, at RESET, at STOP → RUN state
21A3H	File specification error	The files names are overlapping.	Stop	• Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and take the following actions: (1) Format the corresponding drive. Write the program(s) and FB program(s) to the CPU module again. Then, reset the CPU module and run it again. (2) Correct the file names so that they do not overlap.	Drive/file information	At write, at power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2200H	Parameter error	• The system parameter file and CPU parameter file do not exist. • The memory card parameter file or module extension parameter file stored in the memory card cannot be accessed because the memory card is disabled by turning on SM606 (SD memory card forced disable instruction).	Stop	• Write the system parameter file and CPU parameter file to the CPU module. • Turn off SM606. (Cancel the disabled state.)	Parameter information	At power-on, at RESET, at STOP → RUN state
2220H	Parameter error	• The parameter setting is corrupted. • Parameters that the firmware version of the corresponding CPU does not support are written.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and write the displayed parameter setting to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, memory card, or module (SIL2 function module, safety function module, I/O module, or intelligent function module) connected. Please consult your local Mitsubishi representative. • Check the firmware version of the CPU module and use a supported product. Then, write parameters again.	Parameter information	At power-on, at RESET, at STOP → RUN state, at write
2221H	Parameter error	• The set value is out of range.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, memory card, or module (SIL2 function module, safety function module, I/O module, or intelligent function module) connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state, at END instruction execution, at instruction execution, at module access
		An unsupported parameter exists. • Other than 1 is set to the group number in the CC-Link IEF Basic setting. • For the simple CPU communication function, "Initial Communication Setting" is enabled.	Stop	Check the firmware version of the CPU module and use a supported product. Use the product in a supported category. • Set the number of occupied stations so that the total number is 16 stations or less, and set 1 to the group number. • Disable "Initial Communication Setting".	Parameter information	At power-on, at RESET, at STOP → RUN state, at END instruction execution, at instruction execution, at module access
2222H	Parameter error	• Use of the function that is not supported by the module is enabled in parameter. • The module is non-operational. • Parameters that the firmware version of the module does not support were written.	Stop	• Remove the unsupported module if any. • Use functions supported by the module. • Check the status of the module. • Check the firmware version of the module and use a supported product. • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, memory card, or module (SIL2 function module, safety function module, I/O module, or intelligent function module) connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2223H	Parameter error	• The parameter that needs a reset of the CPU module was modified and overwritten.	Stop	• Reset the CPU module and run it again.	Parameter information	Always
2224H	Parameter error	• A memory area cannot be ensured. • A global label setting file exists when the label assignment area set in the CPU parameters is 0K word.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, display the error-detected area by clicking the [Error Jump] button, and increase the capacity of the area. (If the capacity of the area cannot be increased, decrease the capacity of other areas.) • Reduce the number of labels or local devices used. • If no global label is used, delete the global label setting file.	Parameter information	At write, at power-on, at RESET, at STOP → RUN state
2225H	Parameter error	• The model set in parameter differs from the model actually mounted.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and write the displayed parameter setting to the CPU module.	Parameter information	At write, at power-on, at RESET, at STOP → RUN state
2226H	Parameter error	• The SFC setting in the CPU parameters is incorrect. (Block 0 was set to start automatically, however, block 0 does not exist.)	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number.	Parameter information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
2227H	Parameter error	• The execution type of the SFC program set in the CPU parameter program settings is other than the scan execution type.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number.	Parameter information	At power-on, at RESET, at STOP → RUN state
2228H	Parameter error	• The memory area set by the parameter cannot be secured.	Stop	• Check the firmware version of the CPU module and use a supported product. Then, write parameters again.	Parameter information	At power-on, at RESET, at STOP → RUN state
222AH	Parameter error	• The parameter that needs a reset of the CPU module was modified and overwritten. • The parameter containing information that needs a reset of the CPU module was modified and overwritten. (1) The safety module parameter was overwritten after the I/O assignment settings for the system parameters were changed.	Stop	• Reset the CPU module and run it again. • If the system parameter settings have not been overwritten, write the system parameter settings. Then, reset the CPU module and run it again.	Parameter information	At power-on, at RESET, at STOP → RUN state
2240H	Parameter error (module)	• In a multiple CPU system, the I/O module or intelligent function module controlled by another CPU module is specified in the module parameters.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, safety function module, I/O module, or intelligent function module connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2241H	Parameter error (module)	• The I/O numbers set in the system parameters differ from those of the module actually mounted. • The target module is not mounted on the slot where the system parameters and module parameters are set. • The module type set in parameter differs from that of the module actually mounted.	Stop	• Check if the system configuration displayed on the system monitor window of the engineering tool match the actual system configuration. • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, safety function module, I/O module, or intelligent function module connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state, at END instruction execution, at instruction execution, at module access
2242H	Parameter error (module)	• The intelligent function module has detected a module parameter error.	Stop	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed I/O number. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or the intelligent function module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET, at STOP → RUN state
2260H	Parameter error (network)	• Network numbers are overlapping.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or the intelligent function module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET
2262H	Parameter error (network)	■Programmable controller CPU, Safety CPU • When the station number of the MELSECNET/H network module is 0, parameters of PLC to PLC network are set. • The station type set in the module parameters differs from that of the module actually mounted. • Parameters that the firmware version of the CC-Link IE TSN master/local module does not support are written. ■Process CPU, SIL2 Process CPU • When the station number of the MELSECNET/H network module is 0, parameters of PLC to PLC network are set. • The station type set in the module parameters differs from that of the module actually mounted. • In the CPU module in redundant mode, "RJ71GF11-T2" (the model name cannot be set for the redundant system) is selected to the model name in the I/O assignment setting of the system parameter.	Stop	■Programmable controller CPU, Safety CPU • Correct the station number of the MELSECNET/H network module. • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or the intelligent function module. Please consult your local Mitsubishi representative. • Check the firmware version of the CC-Link IE TSN master/local module and use a supported product. Then, write parameters again. ■Process CPU, SIL2 Process CPU • Correct the station number of the MELSECNET/H network module. • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. • In the CPU module in redundant mode, select "RJ71GF11-T2(MR)", "RJ71GF11-T2(SR)", or "RJ71GF11-T2(LR)" (model names can be set for the redundant system) to the model name in the I/O assignment setting of the system parameter. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the intelligent function module connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2263H	Parameter error (network)	• Even though the CC-Link IE module or MELSECNET/H network module is mounted, a different CC-Link IE module or MELSECNET/H network module is set in the system parameters ("I/O Assignment Setting"), or CC-Link IE module or MELSECNET/H network module parameters have not been set.	Stop	• Set the system parameters and module parameters. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or the intelligent function module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state
2280H	Parameter error (refresh)	• The refresh setting is not set correctly. (Data were refreshed exceeding the file register capacity.) • The refresh settings (number of points) are different from those of other numbered CPU modules.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number so that the data are refreshed within the specified device range. (Take the following actions: increase the number of file register points, create a file register file having a capacity for all of the target data to be refreshed, or reduce the refresh device range.) • Rewrite the refresh settings (number of points) in the CPU parameters for all the CPU modules. (Use the same number of points in the refresh settings for all the CPU modules.)	Parameter information	At power-on, at RESET, at STOP → RUN state, at END instruction execution, at instruction execution, at module access
2281H	Parameter error (refresh)	• A device that cannot be used as a refresh device is specified.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number.	Parameter information	At power-on, at RESET, at STOP → RUN state
2282H		• The number of specified refresh points is invalid.				
2283H		• The total number of refresh points exceeded the maximum limit.				
22E0H	Parameter verification error	• In a multiple CPU system, the system parameter settings of the host CPU module differ from those of other CPU modules. • In a multiple CPU system, the system parameter settings are overwritten only to the host CPU module, and the settings differ from those of other CPU modules.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the system parameter settings corresponding to the displayed number for the CPU No.2 and later. The settings need to be the same among all the CPU modules. (The module synchronization setting and fixed scan communication setting need to be the same between the CPU modules that use these functions.) • When the system parameter settings are changed, update the settings of all the CPU modules connected. (The system parameter settings must be same in all the CPU modules.)	Parameter information, system configuration information	At write, at power-on, at RESET, at STOP → RUN state
2300H	Security key authentication error	• The security key set to the program does not match the one registered to the CPU module (or cassette).	Stop	• Check and correct the security key setting.	Drive/file information	At power-on, at RESET, at STOP → RUN state
2301H		• The security key is set to the program, but it is not registered to the CPU module (or cassette).				
2302H	Security key authentication error	• The security key set to the file is corrupted and does not match the one registered to the CPU module. • The security key registered to the CPU module is corrupted and does not match the one set to the file.	Stop	• Write the file to the CPU module again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	—	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2303H	Security key authentication error	• The security key is registered to the CPU module and cassette.	Stop	• Check and correct the security key setting.	—	At power-on, at RESET, at STOP → RUN state
2320H	Remote password setting error	• The start I/O number of the remote password target module is set to other than 0H to 0FF0H. • There is a problem on the slot specified by the start I/O number of the remote password setting. (1) No module is mounted. (2) The mounted intelligent function module does not support the remote password setting.	Stop	• Set the start I/O number of the remote password target module within the range 0H to 0FF0H. • On the specified slot, mount an intelligent function module that supports the remote password setting.	System configuration information	At power-on, at RESET, at STOP → RUN state
2321H	Remote password setting error	• In a multiple CPU system, the module controlled by another CPU module is specified by the start I/O number of the remote password setting.	Stop	• Check and correct the remote password setting.	System configuration information	At power-on, at RESET, at STOP → RUN state
2400H	Module verification error	• The module information at power-on differs from the information of modules actually mounted. • The I/O module or intelligent function module is not mounted properly or was removed during operation.	Stop/ continue	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and check the module corresponding to the displayed slot number. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative. • If error detection is not desirable, change the error detection setting in the CPU parameters.	System configuration information	Always
2401H	Module verification error	• A CPU module, I/O module, or intelligent function module was mounted on the base unit during operation.	Stop/ continue	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and check the module corresponding to the displayed slot number. • Do not mount a CPU module, I/O module, nor intelligent function module during operation. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative. • If error detection is not desirable, change the error detection setting in the CPU parameters.	System configuration information	Always
2420H	Fuse blown error	• The output module with a blown fuse has been detected.	Stop/ continue	• Check the FUSE LED of each output module, and replace the one with the FUSE LED on. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and replace the module corresponding to the displayed slot number. • If error detection is not desirable, change the error detection setting in the CPU parameters.	System configuration information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2440H	Module major error	• In a multiple CPU system, the control CPU setting in the system parameters is different from that of other CPU modules. • In a multiple CPU system, other CPU modules (SIL2 Process CPU or Safety CPU) have detected a parameter verification error. • An error has been detected in the I/O module or intelligent function module during the initial processing.	Stop	• Correct the system parameter settings for the CPU No.2 and later in accordance with those of the CPU No.1. • Eliminate the error cause of another CPU module (Safety CPU or SIL2 Process CPU). • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2441H	Module major error	• An error has been detected in the I/O module or intelligent function module when the instruction was executed.	Stop/ continue	• Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	Error location information, system configuration information	At instruction execution
2442H		• An error has been detected in the I/O module or intelligent function module during the END processing.	Stop/ continue		System configuration information	At module access
2443H		• An error has been detected in the I/O module or intelligent function module.	Stop			
2450H	Module major error	• A major error has been notified from the intelligent function module connected. • The I/O module or intelligent function module is not mounted properly or was removed during operation.	Stop/ continue	• Take measures to reduce noise. • Check the connection status of the extension cable. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and check the module corresponding to the displayed slot number. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	System configuration information	Always
2460H	Another CPU module major error	• An error has been detected in another CPU module during the initial processing.	Stop	• Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the host CPU module or another CPU module where the error has been detected. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2461H		• An error has been detected in another CPU module when the instruction was executed.	Stop/ continue		Error location information, system configuration information	At instruction execution
2462H		• An error has been detected in another CPU module during the END processing.	Stop/ continue		System configuration information	At END instruction execution
2463H		• An error has been detected in another CPU module.	Stop		System configuration information	At power-on, at RESET
2470H		• A major error has been notified from another CPU module.	Stop/ continue		System configuration information	Always
2480H	Multiple CPU error	• In a multiple CPU system, an error has been detected in the CPU module where "Stop" is set in the operation mode setting parameter. • Any CPU module other than CPU No.1 is mounted in the inapplicable slot. (An error occurs in the CPU module mounted in the inapplicable slot.)	Stop	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, identify the error module, and eliminate the error cause. • Remove the CPU module from the inapplicable slot.	System configuration information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2481H	Multiple CPU error	• In a multiple CPU system, any CPU module other than CPU No.1 was disconnected from the base unit during operation. Or, any CPU module other than CPU No.1 was reset.	Stop	• Check the mounting status and reset status of the CPU modules other than CPU No.1.	System configuration information	Always
24C0H 24C1H	System bus error	• An error was detected on the system bus.	Stop	■CPU module, safety function module • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, extension cable, or module (safety function module, I/O module, or intelligent function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C2H	System bus error	• The I/O module or intelligent function module is not mounted properly or was removed during operation. • An error was detected on the system bus.	Stop	■CPU module, safety function module • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and check the module corresponding to the displayed slot number. • Check the connection status of the extension cable. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, extension cable, or module (safety function module, I/O module, or intelligent function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	System configuration information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
24C3H	System bus error	• An error was detected on the system bus.	Stop	■CPU module, safety function module • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, extension cable, or module (safety function module, I/O module, or intelligent function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C4H	System bus error	• An error was detected on the system bus.	Stop	• Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C5H					—	
24C6H	System bus error	• An error was detected on the system bus.	Stop	■CPU module • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or extension cable. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	—	At module access
24C8H	System bus error	• An error was detected on the system bus.	Stop	• Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	—	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
24D0H	System bus error	• The extension level setting of the Q series extension base unit is overlapping with that of any other extension base units. • An unsupported base unit is connected. • Incorrect connections of the extension cables have been detected in the redundant extension base unit. • An error was detected on the system bus.	Stop	• Check and correct the level setting of the Q series extension base unit. • Check the connection status of the extension cable. • Check that the 10m mark is printed on the base unit connected when using the ten-meter extended cables (RC100B). (MELSEC iQ-R Module Configuration Manual) • Disconnect the unsupported base unit. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and identify the extension cables incorrectly connected. Connect the OUT1 connector to the IN1 connector of the next level, and connect the OUT2 connector to the IN2 connector of the next level. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or extension cable. Please consult your local Mitsubishi representative.	System configuration information	Always
24E0H	System bus error	• An error was detected on the system bus.	Stop	• Take measures to reduce noise. • Check the mounting status and reset status of the CPU modules other than CPU No.1. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	System configuration information	Always
2500H	WDT error	■CPU module • The set value of the scan time monitoring time setting has been exceeded. (1) The initial (1st) scan time exceeded the execution monitoring time set in the CPU parameters. (2) The execution time of the fixed scan interrupt program exceeded the interrupt execution interval. • A safety function module that supports extension of the number of points for safety input device and safety output device is not used. ■SIL2 function module, safety function module • The set value of the scan time monitoring time setting has been exceeded. (1) The initial (1st) scan time exceeded the execution monitoring time set in the CPU parameters.	Stop	■CPU module • Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take the following actions: (1) Check and correct the program so that it can be executed within the set time. (2) Check and correct the scan time monitoring time setting in the CPU parameters. • Check and correct the fixed scan interrupt program so that the processing completes within the interrupt execution interval. • Check the firmware version of the safety function module and use a supported safety function module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■SIL2 function module, safety function module • Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take the following actions: (1) Check and correct the program so that it can be executed within the set time. If the same error code is displayed again, the possible cause is a hardware failure of the SIL2 function module or safety function module. Please consult your local Mitsubishi representative.	Time information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2501H	WDT error	■CPU module • The set value of the scan time monitoring time setting has been exceeded. (1) The 2nd or later scan time exceeded the execution monitoring time set in the CPU parameters. (2) The execution time of the fixed scan interrupt program exceeded the interrupt execution interval. ■SIL2 function module, safety function module • The set value of the scan time monitoring time setting has been exceeded. (1) The 2nd or later scan time exceeded the execution monitoring time set in the CPU parameters.	Stop	■CPU module • Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take the following actions: (1) Check and correct the program so that it can be executed within the set time. (2) Check and correct the scan time monitoring time setting in the CPU parameters. • Check and correct the fixed scan interrupt program so that the processing completes within the interrupt execution interval. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■SIL2 function module, safety function module • Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take the following actions: (1) Check and correct the program so that it can be executed within the set time. If the same error code is displayed again, the possible cause is a hardware failure of the SIL2 function module or safety function module. Please consult your local Mitsubishi representative.	Time information	Always
2520H	Invalid interrupt	• Even though an interrupt was requested, there is no interrupt factor.	Stop	• Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, safety function module, I/O module, intelligent function module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	At interrupt occurrence
2521H					—	
2522H	Invalid interrupt	• An interrupt was requested from the module with no interrupt pointer setting.	Stop	• Check and correct the interrupt pointer setting in the module parameters. • Take measures so that no interrupt is requested from the module with no interrupt pointer setting. • Check and correct the interrupt setting in the buffer memory of the intelligent function module. • Correct the BASIC program executed in the QD51.	System configuration information	At interrupt occurrence
2610H	Inter-module synchronization signal error	• An execution interval error of the synchronization interrupt program has been detected. • An inter-module synchronization error has been detected.	Stop/ continue	• Check the module set as the inter-module synchronous master. • Correct the inter-module synchronous master settings. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	—	Always
2611H		• An inter-module synchronization error has been detected. • An error of the module set as the inter-module synchronous master has been detected.	Stop		System configuration information	At power-on, at RESET, at END instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2630H	Multiple CPU synchronization signal error	• An execution interval error of the synchronization interrupt program has been detected. • A multiple CPU synchronization error has been detected.	Stop/ continue	• Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	—	Always
2631H		• A multiple CPU synchronization error has been detected.	Stop		System configuration information	At power-on, at RESET, at END instruction execution
2800H	I/O number or network number specification error	• The specified I/O number is out of range (other than 0 to FFH, 3E0 to 3E3H).	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
2801H		• The I/O number of the module that does not exist was specified.	Stop/ continue		Error location information, system configuration information	At instruction execution
2802H	I/O number or network number specification error	• The I/O number of the module that does not support the instruction was specified. • The dedicated instruction specified in the program cannot be executed in the specified module or mode.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the execution propriety (including support status and execution mode) of the dedicated instruction, referring to the manual for the target module.	Error location information, system configuration information	At instruction execution
2803H	I/O number or network number specification error	• The I/O number of the module that cannot be specified in the instruction was specified.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information, system configuration information	At instruction execution
2804H		• The specified network number is out of range (other than 1 to 239).				
2805H		• The network number that does not exist was specified.				
2806H	I/O number or network number specification error	• An I/O module or intelligent function module controlled by another CPU module was specified.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Delete the link direct device that specifies a network module controlled by another CPU module from the program. • Specify a network module controlled by host CPU module by using the link direct device.	Error location information, system configuration information	At instruction execution
2807H	I/O number or network number specification error	• The module cannot be identified in the instruction that requires a specification of the I/O module or intelligent function module. (There is a mistake in the string specifying the module.)	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
2810H	I/O number or network number specification error	• The I/O module or intelligent function module specified in the instruction cannot execute the instruction.	Stop/ continue	• The possible cause is a hardware failure of the I/O module or intelligent function module specified in the instruction. Please consult your local Mitsubishi representative.	Error location information, system configuration information	At instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2820H	Device, label, or buffer memory specification error	■CPU module • The device or label area used in the instruction exceeded the specified range. • The file register file is not set or was accessed without setting it in the CPU parameters (file setting). ■Safety function module • The device or label area used in the instruction exceeded the specified range.	Stop/ continue	■CPU module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Set the file register file in parameter, and access the file. ■Safety function module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information, process control instruction processing information	At instruction execution, at END instruction execution
2821H	Device, label, or buffer memory specification error	• The device or label areas used in the instruction to store data are overlapping.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information, process control instruction processing information	At instruction execution
2822H		• The device or label that cannot be used in the instruction was specified.			Error location information	
2823H		• The buffer memory area of the module specified in the instruction exceeded the specified range. • The module specified in the instruction does not have buffer memory.				
2824H		• The access prohibited area in the buffer memory was accessed.				
2840H	File name specification error	• The file specified in the instruction does not exist.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, create the specified file, and write it to the CPU module. Or, set all the required files in the CPU parameters (file setting).	Error location information, drive/file information	At instruction execution
2841H	File name specification error	• The program file specified in the instruction is not set in the CPU parameters (program setting).	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and set the specified program file in the CPU parameter.	Error location information, drive/file information	At instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2842H	File name specification error	• A file that cannot be specified with the instruction was specified.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and check the specified file.	Error location information, drive/file information	At instruction execution
3000H	Boot function execution error	• The boot setting in the memory card parameters is incorrect.	Stop	• Check and correct the boot setting in the memory card parameters.	Drive/file information	At power-on, at RESET
3001H	Boot function execution error	• When the boot function was executed, the file format processing failed.	Stop	• Reset the CPU module and execute the boot function again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information	At power-on, at RESET
3003H	Boot function execution error	• When the boot function was executed, the file passwords did not match.	Stop	• Check and correct the file password settings of the transfer source and transfer destination files. • Delete the boot setting.	Drive/file information	At power-on, at RESET
3004H	Boot function execution error	• When the boot function was executed, the CPU built-in memory capacity was exceeded.	Stop	• Check and correct the boot setting. • Delete unnecessary files in the CPU built-in memory.	Drive/file information	At power-on, at RESET
3005H	Boot function execution error	• When the boot function is executed, the security key registered in the CPU module (or cassette) does not match the one that locks the boot source program. • When the boot function is executed, the security key is not registered in the CPU module (or cassette) even though the boot source program is locked with the security key. • The program files and FB files that are written to an SD memory card using the memory card operation of the engineering tool is set to boot targets.	Stop	• Check and correct the security key setting. • Delete the boot settings from the memory card parameter. • Write the boot target program files and FB files to the SD memory card on the CPU module using the online data operation of the engineering tool.	Drive/file information	At power-on, at RESET
300CH	Execution error of functions using memory card	• The firmware update cannot be executed because the firmware update file is not stored in the memory card. • The automatic restoration function with the SD CARD OFF button cannot be executed because the system file for the function is not stored in the memory card.	Stop	• If the automatic restoration function with the SD CARD OFF button cannot be executed, take either of the following actions. (1) Check that the system file for automatic restoration with the SD CARD OFF button is stored. (2) Turn on bit 2 of SD955 (Automatic restoration with the SD CARD OFF button) before backing up the data.	—	At power-on, at RESET
3010H	Data restoration function execution error	• The model of the restoration target CPU module differs from the model of the backup source CPU module.	Stop	• Execute the data restoration function to the CPU module whose model is the same as that of the backup source CPU module. • Turn off bit 0 of SD955 to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3011H	Data restoration function execution error	• Reading of backup data from an SD memory card completed with an error.	Stop	• Replace an SD memory card, and execute the function again. • The backup data may have been corrupted. Execute the data restoration function using another backup data. • Turn off bit 0 of SD955 to disable the automatic data restoration function.	CPU module data backup/ restoration folder information, drive/file information	At power-on, at RESET
3012H	Data restoration function execution error	• Writing of backup data to the CPU built-in memory completed with an error.	Stop	• The possible cause is a hardware failure of the restoration target CPU module. Execute the data restoration function to another CPU module.	CPU module data backup/ restoration folder information, drive/file information	At power-on, at RESET
3013H	Data restoration function execution error	• The system file does not exist in the backup data to be restored. • File(s) in the system file information does not exist in the folder of the backed up data.	Stop	• The backup data may have been corrupted. Execute the data restoration function using another backup data. • Turn off bit 0 of SD955 to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET
3014H	Data restoration function execution error	• Data was restored to the CPU module where the same data with a file password has already been stored.	Stop	• Delete file passwords, and execute the CPU module data backup/restoration function. • Turn off bit 0 of SD955 to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET
3015H	Data restoration function execution error	• A folder with a value that matches the restoration target date folder setting value or number folder setting value does not exist in the SD memory card. • The restoration target data setting value is out of range. • The restoration target date folder setting value or number folder setting value is out of range.	Stop	• Check and correct the restoration target date folder setting value or number folder setting value, and execute the function again. • Check and correct the restoration target data setting value, and execute the function again. • Turn off bit 0 of SD955 to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET
3016H	Data restoration function execution error	• The automatic data restoration function was executed with the CPU module where an SD memory card was not inserted.	Stop	• Insert or re-insert an SD memory card, and execute the function again. • Turn off bit 0 of SD955 to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET
3017H	Data restoration function execution error	• The automatic data restoration function was executed exceeding the maximum memory capacity of the CPU module. • The automatic data restoration function was executed exceeding the maximum number of files that can be stored in the CPU module.	Stop	• Execute the function so that the maximum memory capacity will not be exceeded. • Execute the function so that the maximum number of storable files will not be exceeded. • Turn off bit 0 of SD955 to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET
3018H	Data restoration function execution error	• The status (such as programs, parameters, and file structure) of the CPU module differs from that of when the data backup function was executed.	Stop	• Match the CPU module status to the one at the time of backup, and execute the function again. • Set all data as the backup/restoration target data, and execute the data restoration function. • Turn off bit 0 of SD955 to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
301AH	Data restoration function execution error	• During restoration, an error has been detected in the safety function module.	Stop	• Eliminate the error cause from the safety function module and execute the function again. • Disable the automatic restoration setting from the engineering tool.	CPU module data backup/ restoration folder information	At power-on, at RESET
301BH	Data restoration function execution error	• The parameter restored in the safety function module is different from the reflected parameter.	Stop	• Reset the CPU module and run it again.	CPU module data backup/ restoration folder information	At power-on, at RESET
301CH	Data restoration function execution error	• The automatic restoration function with the SD CARD OFF button cannot be executed because the button has been pressed for more than 10 seconds after the READY LED had begun to flash.	Stop	• Release the SD CARD OFF button within 10 seconds after the READY LED begins to flash. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	CPU module data backup/ restoration folder information	At power-on, at RESET
3070H	Operation stop error	• The PABORT instruction was executed.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool and check the corresponding program - one that outputs the program error code.	Error location information and program error information	At instruction execution
3100H	Program error	• The program includes any instruction that cannot be used or decoded in the CPU module.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the program to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Error location information	At power-on, at RESET, at STOP → RUN state, at instruction execution
3101H	Program error	• The program contains a dedicated SFC program instruction even though it is not an SFC program.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Error location information	At write, at power-on, at RESET, at STOP → RUN state
3120H	Program error	• The CPU module does not support the dedicated instruction executed.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the firmware version of the CPU module and use a supported product.	Error location information	At power-on, at RESET, at STOP → RUN state, at instruction execution
3121H	Program error	• The number of devices used in the dedicated instruction specified in the program is incorrect.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3122H	Program error	• The function block or function specified in the program does not exist.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Error location information	At instruction execution
3130H	Program error	• The link direct device specified in the instruction cannot be executed.	Stop	• Check and correct the link direct device setting of the CPU parameter.	Error location information	At instruction execution
3140H	END instruction error	• The END (FEND) instruction does not exist in the program.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Error location information	At power-on, at RESET, at STOP → RUN state
3141H	FB/FUN program error	• The structure of FB/FUN program is incorrect.	Stop	• Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Error location information	At instruction execution
3142H	Temporary area error	• The temporary area was used incorrectly.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and check the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3143H	FB/FUN program error	• The FB/FUN program file is incorrect.	Stop	• Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Error location information	At instruction execution
3160H to 3163H	SFC program block, step error	• The SFC program configuration is illegal.	Stop	• Take measures to reduce noise. • Write the SFC program to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Error location information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
3170H	SFC program block, step error	• The number of SFC program steps exceeds the total number of step relays (S).	Stop	• Correct the program so that the number of SFC program steps does not exceed the total number of step relays (S). • Check and correct the number of device setting step relays (S) in the CPU parameters.	Error location information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
3171H	SFC program block, step error	• The total number of SFC program blocks (max. step No. + 1) exceeds the total number of step relays (S).	Stop	• Correct the program so that the total number of SFC program blocks (max. step No. + 1) does not exceed the total number of step relays (S). • Check and correct the number of device setting step relays (S) in the CPU parameters.	Error location information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
3180H	SFC program configuration error	• The SFC program configuration is illegal.	Stop	• Take measures to reduce noise. • Write the SFC program to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Error location information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
3190H						At SFC program execution
3191H						
3192H	SFC program configuration error	• A self step number was specified for the specification destination step number for the jump transition.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the SFC program to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Error location information	At SFC program execution
3193H		• A self step number was specified for the specification destination step number for the reset step.				
31A0H	SFC program block, step specification error	• An attempt was made to start an SFC program block that was already running.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Turn on Start/stop SFC program (SM321) if it is off.	Error location information	At SFC program execution

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
31A1H	SFC program block, step specification error	• A non-existent SFC program block was specified.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Turn on SM321 (Start/stop SFC program) if it is off. • Check the SFC program has existed. • Check the execution status of the SFC program.	Error location information	At instruction execution, at SFC program execution
31A2H	SFC program block, step specification error	• The specified block exceeds the range that can be used in the SFC program.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Turn on SM321 (Start/stop SFC program) if it is off.	Error location information	At instruction execution
31B1H	SFC program block, step specification error	• A non-existent SFC program step was specified.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Turn on SM321 (Start/stop SFC program) if it is off. • Check the SFC program has existed. • Check the execution status of the SFC program.	Error location information	At instruction execution, at SFC program execution
31B2H	SFC program block, step specification error	• The specified step exceeds the range that can be used in the SFC program.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Turn on SM321 (Start/stop SFC program) if it is off.	Error location information	At instruction execution
31B3H	SFC program block, step specification error	• The number of simultaneous active block steps that can be specified in the SFC program exceeds the permissible value.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution, at SFC program execution
31B4H		• The total number of simultaneous active steps that can be specified in the SFC program exceeds the permissible value.				At instruction execution, at SFC program execution
31B5H		• A SET Sn/BLm/Sn and RST Sn/BLm/Sn instruction were specified for the self step in the step operation output.				At instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3200H	Program execution error	■Programmable controller CPU, Process CPU, SIL2 Process CPU • Memory/Device Setting set in the CPU parameters differ from those assigned to the sequence programs, FB programs, and global label setting file. (After Memory/ Device Setting were modified, only the CPU parameters were written to the CPU module.) ■Safety CPU • Memory/Device Setting set in the CPU parameters differ from those assigned to the sequence programs, FB programs, and global label setting file. (After Memory/ Device Setting were modified, only the CPU parameters were written to the CPU module.) • The device/label assignment for the safety program, safety FB file, safety global label setting file, and standard/safety shared label setting file does not match the memory/device setting in the CPU parameter of the Safety CPU. ■SIL2 function module, safety function module • The device/label assignment for the safety program, safety FB file, safety global label setting file, and standard/safety shared label setting file does not match the memory/device setting in the CPU parameter of the CPU module. • Parameters that require resetting after being rewritten at the CPU module were rewritten.	Stop	■Programmable controller CPU, Process CPU, SIL2 Process CPU • After Memory/Device Setting are modified, write the sequence program file(s), FB file(s), and global label setting file together with the CPU parameter file to the CPU module. • If no global label is used, delete the global label setting file. ■Safety CPU • After Memory/Device Setting are modified, write the sequence program file(s), FB file(s), and global label setting file together with the CPU parameter file to the CPU module. • If no global label is used, delete the global label setting file. • Batch write the CPU parameters, safety CPU parameters, safety program, safety FB file, safety global label setting file, and standard/ safety shared label setting file. • If not using safety global labels or standard/ safety shared labels, delete the safety global label setting file and standard/safety shared label setting file. ■SIL2 function module, safety function module • Batch write the CPU parameters, safety CPU parameters, safety program, safety FB file, safety global label setting file, and standard/ safety shared label setting file. • If not using safety global labels or standard/ safety shared labels, delete the safety global label setting file and standard/safety shared label setting file. • Reset the CPU module and run it again.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3201H	Program execution error	• Even though no program is set in the CPU parameters, multiple program files exist.	Stop	• Set a program in the CPU parameter. • Delete unnecessary program files.	Drive/file information	At power-on, at RESET, at STOP → RUN state

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3202H	Program execution error	■CPU module • The program file is incorrect. Or, the program file is not written properly. • For the subroutine type FB, "Use MC/MCR to Control EN" of "Inherent Property" is set to "Yes". • In the option settings of the engineering tool, "Use Phase Processing Instructions" is set to "Yes". ■Safety function module • The safety program file is incorrect. Or, the safety program file is not written properly.	Stop	■CPU module • Write the program file to the CPU built-in memory again. • Set "Use MC/MCR in EN Control" in "Inherent Property" of subroutine-type FB to "No" and write the program file to the CPU built-in memory again. • Replace the CPU module with the one that supports the use of the subroutine-type FB for which "Use MC/MCR in EN Control" of "Inherent Property" is set to "Yes". • Set "Use Phase Processing Instructions" to "No" in the option settings of the engineering tool and write the program file to the CPU built-in memory again. • Replace the CPU module with the one that supports the use of the phase processing instructions. ■Safety function module • Write the program again.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3203H	Program execution error	• No program file exists.	Stop	• Check if the system parameter file, CPU parameter file, and program file exist. • Write the system parameter file, CPU parameter file, and program file to the CPU built-in memory.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3204H	Program execution error	• Two or more SFC programs were executed.	Stop	• Ensure that only one SFC program is executed.	Drive/file information	At SFC program execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3205H	Program execution error	■ Programmable controller CPU, Safety CPU • After the global label setting file was modified, only the modified file was written to the programmable controller. Or, without writing the modified global label setting file, only the sequence program file(s) and FB file(s) were written to the programmable controller. (The global label setting file was not written to the programmable controller.) • After an FB file was modified, only the modified FB file was written to the programmable controller. Or, without writing the modified FB file, only the program file(s) and global label setting file were written. (The modified FB file was not written to the programmable controller.) • After the "Access from External Device" setting of the global label setting was changed, only the global label assignment information was written to the programmable controller. Or, after the "Access from External Device" setting was changed, only the sequence program file(s) and FB file(s) were written to the programmable controller. (The global label setting file was not written.) • When the "Access from External Device" setting is not selected, the sequence program file(s), FB file(s), and global label setting file are written to the programmable controller without initializing the memory where the global label assignment information file is stored.	Stop	• Write all the sequence program file(s), FB file(s), global label setting file, and global label assignment information file to the programmable controller. • If no global label is used, delete the global label setting file. • If the "Access from External Device" setting is not selected, initialize the memory where the global label assignment information file is stored, and write the sequence program file(s), FB file(s), and global label setting file to the programmable controller.	Drive/file information	At power-on, at RESET, at STOP → RUN state

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3205H	Program execution error	■ Process CPU • After the global label setting file was modified, only the modified file was written. Or, without writing the global label setting file, only the program file(s) and FB file(s) were written. • After an FB file was modified, only the modified file was written. Or, without writing the modified FB file, only the program file(s) and global label setting file were written. • After the "Access from External Device" setting of the global label setting was changed, only the global label assignment information was written. Or, after the "Access from External Device" setting was changed, only the sequence program file(s) and FB file(s) were written. • When the "Access from External Device" setting is not selected, the sequence program file(s), FB file(s), and global label setting file are written without initializing the memory where the global label assignment information file is stored. • The global label assignment information that does not correspond to the firmware version of the CPU module was written due to memory copy or restoration.	Stop	• Write all the sequence program file(s), FB file(s), global label setting file, and global label assignment information file to the programmable controller. • If no global label is used, delete the global label setting file. • If the "Access from External Device" setting is not selected, initialize the memory where the global label assignment information file is stored, and write the sequence program file(s), FB file(s), and global label setting file to the programmable controller.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3206H	Program execution error	• After the sequence program was modified, only the modified sequence program file was written to the programmable controller. Or, without writing the modified sequence program file, only the initial local label value file was written to the programmable controller. • After the global label setting file was modified, only the modified file was written to the programmable controller. Or, without writing the modified global label setting file, only the initial global label value file was written.	Stop	• Write both the sequence program file(s) and initial local label value file to the programmable controller. • Write both the global label setting file and initial global label value file to the programmable controller. • If no initial global label value is used, delete the initial global label value file. • If no initial local label value is used, delete the initial local label value file.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3207H	Program execution error	• The signal flow area to be used in the FB program cannot be secured.	Stop	• Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and identify the error program file. Then, reduce the number of instructions that use the signal flow area of the FB program used in the sequence program file.	Drive/file information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3208H	Program execution error	• The label file is incorrect.	Stop	• Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, select the correct file name, and write the specified file to the CPU module.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3220H	SFC program execution error	• Unable to execute the SFC program.	Stop	• Take measures to reduce noise. • Write the SFC program and CPU parameters to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
3221H 3222H	SFC program execution error	• Unable to execute the SFC program.	Stop	• Take measures to reduce noise. • Write the SFC program to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
32FEH	License authentication error	• A device or a label that is required for the license authentication does not exist.	Stop	• Please consult your local Mitsubishi representative.	Error location information	At instruction execution
32FFH	License authentication error	• The specified license key is incorrect.	Stop	• Check the license key and specify correct one.	Error location information	At instruction execution
3300H	Pointer setting error	• The total number of points of local or global pointers used in the program exceeded the points set in the CPU parameters (pointer device area).	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check and correct the pointer device area point setting in the CPU parameters.	Error location information	At power-on, at RESET, at STOP → RUN state
3301H	Pointer setting error	• The total number of points of pointer-type labels used in the program exceeded the points set in the CPU parameters (pointer-type label assignment area).	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check and correct the pointer-type label assignment area point setting in the CPU parameters.	Error location information	At power-on, at RESET, at STOP → RUN state

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3302H	Pointer setting error	• Multiple global pointers with the same number or pointer-type global labels are used in the program. (The pointer numbers or the labels are overlapping.)	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At power-on, at RESET, at STOP → RUN state
3303H	Pointer setting error	• Multiple local pointers with the same number or pointer-type local labels are used in the program. (The pointer numbers or the labels are overlapping.)				
3320H	Interrupt pointer setting error	• The interrupt pointer numbers used in the files are overlapping.				
3340H	FOR-NEXT instruction error	• The NEXT instruction was not executed even though the FOR instruction was executed. Or, there are more FOR instructions than NEXT instructions.				At END instruction execution
3341H	FOR-NEXT instruction error	• The NEXT instruction was executed even though the FOR instruction was not executed. Or, there are more NEXT instructions than FOR instructions.				At instruction execution
3342H	FOR-NEXT instruction error	• The BREAK instruction was executed even though the FOR instruction was not executed.				
3360H	Nesting depth error	• The number of nesting levels in the subroutine program exceeded its limit (16).	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the number of nesting levels (16 or less).	Error location information	At instruction execution
3361H		• The number of nesting levels in the FOR instruction exceeded its limit (16).				
3362H		• The number of nesting levels in the DI instruction exceeded its limit (16).				
3363H	Nesting depth error	• The number of nesting levels in the function block or function exceeded its limit (32).	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the number of nesting levels (32 or less).	Error location information	At instruction execution
3380H	Pointer execution error	• The pointer specified in the instruction does not exist.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
3381H		• The RET instruction does not exist in the executed subroutine program.				At END instruction execution
3382H		• The RET instruction exists before the FEND instruction in the main program.				At instruction execution
33A0H	Interrupt pointer execution error	• The interrupt pointer corresponding to the interrupt input does not exist.	Stop	• Check if the program corresponding to the interrupt pointer number set in the module parameters exists.	—	At instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
33A1H	Interrupt pointer execution error	• The IRET instruction does not exist in the executed interrupt program. • The STOP instruction has been executed in an interrupt program.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
33A2H		• The IRET instruction exists before the FEND instruction in the main program.				
33A3H		• The IRET instruction or STOP instruction was executed in the fixed scan execution type program.				
33A4H		• The IRET instruction or STOP instruction has been executed in an event execution type program.				
33C0H	FB/FUN execution error	• Before the FB/FUN program ends, the call source program ended.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the program to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Error location information	At instruction execution
33D0H	Temporary area exceeded	• The secured temporary area size exceeded its limit.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the number of nesting levels in the function.	Error location information	At instruction execution

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3400H	Operation error	• Division where the divisor is zero was performed.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information, process control instruction processing information	At instruction execution
3401H	Operation error	• Data that cannot be converted by using the data conversion instruction was input.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information, process control instruction processing information	At instruction execution
3402H	Operation error	• The operation was performed with the invalid data (-0, denormalized number, NaN (not a number), or ±∞).	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information, process control instruction processing information	At instruction execution
3403H	Operation error	• An overflow has occurred during the operation.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information, process control instruction processing information	At instruction execution
3404H	Operation error	• A string that is not supported in the instruction was specified.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information, process control instruction processing information	At instruction execution
3405H	Operation error	• The input data was out of range.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information, process control instruction processing information	At instruction execution
3406H	Operation error	• The operation result is out of the output range. (The operation result of the instruction that concatenate character strings exceeded the allowable number of characters.)	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information, process control instruction processing information	At instruction execution
3420H	Operation error	• The link direct device, module access device, or CPU buffer memory access device is specified for both (s) and (d) used in the BMOV instruction.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
3421H	Operation error	• When writing data to the data memory by using the SP.DEVST instruction, the number of writes per day exceeded the number set in SD771. • The value set in SD771 is out of range.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and check if the SP.DEVST instruction is used correctly in the program. • Execute the SP.DEVST instruction again on another day, or change the value in SD771. • Set the value in SD771 within the settable range.	Error location information	At instruction execution
3422H	Operation error	• The structure of the PID control instruction is incorrect.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the PID control instruction structure.	Error location information	At instruction execution
3423H	Operation error	• The size of data to be sent/ received by the socket communications instruction exceeds the allowable range.	Stop/ continue	• Check and change the send data size of the CPU module or the external device. • If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Error location information	At instruction execution
3424H	Operation error	• A second SFC program was started with an instruction while an SFC program was running.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and check the SFC program execution status.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3425H	Operation error	• The clock data specified with the DATEWR(P) instruction is less than 1 hour from the daylight saving time starting time.	Stop/ continue	• Check the detailed error information (error location information) by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the clock data specified with the instruction or the daylight saving time setting starting time.	Error location information	At instruction execution
3426H	Operation error	• The specified file name (before a period) or extension includes two or more "*". • The specified file name (before a period) or extension includes "*" and "?". • A wild card ("*", "?") is used in a location where it cannot not be used. • A file that cannot be transferred has been specified. • A file name is not specified. • A delimiter for a drive number has been specified with characters other than ":\" or ":/".	Stop/ continue	• Check the usage of wild card characters. • Use a file that can be transferred. • Specify a file name. • Specify a delimiter for a drive number with ":\" or ":/".	Error location information	At instruction execution
3427H	Operation error	The control data (d1) of the SP.FREAD and SP.FWRITE instruction are specified with an invalid combination between the execution/ completion type and the data type specification.	Stop/ continue	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and change the combination to a valid one.	Error location information	At instruction execution
3430H	Operation error	• An instruction has been executed without setting parameters which are required when the instruction is executed.	Stop/ continue	• Set parameters required to execute the instruction.	Error location information	At instruction execution
3440H	Operation error	• In a multiple CPU system, the multiple CPU dedicated instruction (the one whose symbol starts with D(P)) was executed when "Do Not Use" was set to "Fixed scan communication function" in the system parameters ("Multiple CPU Setting").	Stop/ continue	• Change "Fixed scan communication function" to "Use". • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and change the multiple CPU dedicated instruction to the one whose symbol starts with M(P).	Error location information	At instruction execution
3441H	Operation error	• In a multiple CPU system, the number of data points was specified exceeding the multiple CPU dedicated instruction areas applicable for each CPU module.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and change the number of data points for the multiple CPU dedicated instruction.	Error location information	At instruction execution
3460H	Operation error (redundant function)	• In a redundant system with redundant extension base unit, a dedicated instruction has been executed to the module mounted on the extension base unit. (The error is detected in the control system or the standby system.)	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and delete the dedicated instruction for the module on the extension base unit from the program.	Error location information	At instruction execution

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3461H	Operation error (redundant function)	• In a redundant system with redundant extension base unit, an instruction for accessing the module mounted on the extension base unit from the standby system has been executed. (The error is detected in the standby system.)	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and delete the instruction that causes the error from the program. • Turn on SM1762 (Behavior setting for access from standby system to extension base unit) and then execute the instruction.	Error location information	At instruction execution
34A0H	Operation error	• Response data of the socket communications instruction cannot be created.	Stop/ continue	• Increase the request interval. • Decrease the number of request nodes. • Wait for a response to the previous request before sending the next request. • Correct the timeout value.	Error location information	At END instruction execution
3600H	Safety system configuration error	■SIL2 Process CPU, SIL2 function module • The SIL2 function module controlled by the host CPU module is not mounted to the right of the CPU module. • The slot right of the CPU module is set as "Empty" in the system parameters (I/O Assignment Setting). ■Safety CPU, safety function module • The safety function module controlled by the host CPU module is not mounted to the right of the Safety CPU. • The slot right of the Safety CPU is set as "Empty" in the system parameters (I/O Assignment Setting).	Stop	■SIL2 Process CPU, SIL2 function module • Mount the SIL2 function module controlled by the host CPU module to the right of the CPU module. • Re-set the system parameters (I/O Assignment Setting) so that the SIL2 function module is right of the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, the SIL2 function module, or the base unit. Please consult your local Mitsubishi representative. ■Safety CPU, safety function module • Mount the safety function module controlled by the host CPU module to the right of the Safety CPU. • Re-set the system parameter I/O assignment settings so that there is a safety function module to the right of the Safety CPU. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU, safety function module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
3601H	Safety system configuration error	• The safety function module controlled by the host CPU module is not mounted on the main base unit. • The safety function module is mounted on the slot that is set to "Empty" in the system parameters ("I/O Assignment Setting").	Stop	• Mount the safety function module controlled by the host CPU module on the main base unit. • Re-set the system parameters ("I/O Assignment Setting") in accordance with the safety function module actually mounted. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, safety function module, or base unit. Please consult your local Mitsubishi representative.	—	At power-on, at RESET
3602H	Safety system configuration error	• Two or more Safety CPUs are mounted.	Stop	• Mount only one Safety CPU.	System configuration information	At power-on, at RESET
3608H	Safety system configuration error	• Two or more SIL2 function modules are mounted.	Stop	• Mount only one SIL2 function module.	System configuration information	At power-on, at RESET
3610H	Safety system configuration error	■SIL2 Process CPU • The SIL2 function module with a pair version that differs from the one of the SIL2 Process CPU is mounted. ■Safety CPU • The safety function module with a pair version that differs from the one of the Safety CPU is mounted.	Stop	■SIL2 Process CPU • Mount the safety function module with the same pair version as the one of the SIL2 function module. ■Safety CPU • Mount the safety function module with the same pair version as the one of the Safety CPU.	System configuration information, pair version information[*1]	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3620H	Control CPU module error	• An error has been detected in the control CPU module.	Stop	• Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the control CPU module or this module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
3621H	SIL2 system configuration error	• A module with a firmware version that does not support the safety functions that are available to the system using the SIL2 Process CPU is mounted. • A module that does not support the system using the SIL2 Process CPU is mounted.	Stop	• Check the firmware version of the corresponding module. If the module that does not support the safety functions that are available to the system using the SIL2 Process CPU is mounted, remove the module from the base unit. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and check the module corresponding to the displayed slot number. If the module does not support the system using the SIL2 Process CPU, remove the module from the base unit.	System configuration information	At power-on, at RESET
3622H	SIL2 system configuration error	• The redundant function module is not mounted to the right of the SIL2 function module. • The slot right of the SIL2 function module is set as "Empty" in the system parameters (I/O Assignment Setting).	Stop	• Mount the redundant function module to the right of the SIL2 function module. • Re-set the system parameters (I/O Assignment Setting) so that the redundant function module is right of the SIL2 function module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, the SIL2 function module, the redundant function module, or the base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
3640H	Safety mutual monitoring error	■SIL2 Process CPU, SIL2 function module • A file mismatch has been detected in the mutual monitoring between the SIL2 Process CPU and SIL2 function module. ■Safety CPU, safety function module • A file mismatch has been detected in the mutual monitoring between the Safety CPU and safety function module.	Stop	• Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again	—	At power-on, at RESET
3641H	Safety mutual monitoring error	■SIL2 Process CPU, SIL2 function module • An error has been detected in the SIL2 Process CPU or SIL2 function module in the mutual monitoring between the SIL2 Process CPU and SIL2 function module. • The SIL2 Process CPU or the SIL2 function module is not mounted properly or was removed during operation. ■Safety CPU, safety function module • An error has been detected in the Safety CPU or safety function module in the mutual monitoring between the Safety CPU and safety function module. • The Safety CPU or the safety function module is not mounted properly or was removed during operation.	Stop	• Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	System configuration information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3680H	Parameter error (safety function)	■SIL2 Process CPU, SIL2 function module • Because a SIL2 function module controlled by the host CPU module is not mounted, the host CPU module cannot communicate parameters. ■Safety CPU • Because a safety function module controlled by the host CPU module is not mounted, the host CPU module cannot communicate parameters.	Stop	■SIL2 Process CPU, SIL2 function module • Mount a SIL2 function module or a safety function module controlled by the host CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative. ■Safety CPU • Mount the safety function module controlled by the host CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or safety function module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET
3681H	Parameter error (safety function)	• Connection between modules controlled by the host CPU module is set in the safety communication settings. • The contents in the safety module parameter are broken.	Stop	• Check the numerical value (parameter number) in the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and take either of the following actions. Note that if the same error code is displayed again after taking one of the following actions, the possible cause is a hardware failure of the CPU module or module (SIL2 function module, safety function module, or intelligent function module) connected. Please consult your local Mitsubishi representative. (1) For the safety communication setting, check and correct the parameter settings. (2) For the safety I/O refresh setting, write the safety module parameters again.	Parameter information	At power-on, at RESET
3682H	Parameter error (safety function)	• The transmission interval monitoring time in the safety communication setting is shorter than the safety cycle time, or is set to the same value. • The safety output refresh interval monitoring time in the safety I/O refresh monitoring time setting is shorter than the safety cycle time, or is set to the same value.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter settings corresponding to the displayed number so that the transmission interval monitoring time in the safety communication settings will not be shorter than or equal to the safety cycle time. • Correct the parameter setting so that the safety output refresh interval monitoring time in the safety I/O refresh monitoring time setting is not shorter than or equal to the safety cycle time.	Parameter information	At power-on, at RESET
3683H	Parameter error (safety function)	• The safety refresh monitoring time in the safety communication setting is shorter than the transmission interval monitoring time, or is set to the same value. • The safety I/O refresh timeout time is shorter than the safety output refresh interval monitoring time in the safety I/O refresh monitoring time setting, or is set to the same value.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter settings corresponding to the displayed number so that the safety refresh monitoring time in the safety communication settings will not be shorter than or equal to the transmission interval monitoring time. • Correct the parameter setting so that the safety I/O refresh timeout time is not shorter than or equal to the safety output refresh interval monitoring time in the safety I/O refresh monitoring time setting.	Parameter information	At power-on, at RESET
3684H	Parameter error (safety function)	• The safety CPU parameters do not exist.	Stop	• Write the safety CPU parameters.	Parameter information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3685H	Parameter error (safety function)	• Passive is selected for the open system in the safety communication setting when the operation mode of the CPU module is redundant mode.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter settings corresponding to the displayed parameter number. Then, select Active for the open system of the safety communication setting.	Parameter information	At power-on, at RESET
3686H	Parameter error (safety function)	• Safety communication setting parameters that the firmware version of the network module does not support are written.	Stop	• Check the numerical value (parameter number) in the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and take either of the following actions. (1) For the safety communication setting, match the safety protocol version that is set in the safety communication setting and the safety protocol version that is supported by the network module. Or, check the firmware version of the network module and use the supported product. Then, write parameters again. (2) For the safety I/O refresh setting, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET
3687H	Parameter error (safety function)	• The start address in the safety I/O refresh device setting is not set.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or I/O module with safety functions. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET
3690H	Parameter error (safety function)	• The corresponding Safety CPU does not exist. • Multiple control CPUs were detected.	Stop	• Remove the safety function module. • Use a control CPU for a single Safety CPU. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU, safety function module, or base unit. Please consult your local Mitsubishi representative.	—	At power-on, at RESET
36A0H	Parameter error (safety function)	• A value exceeding the maximum number of safety programs to be executed is set in the program settings.	Stop	• Correct the number of safety programs so that it is within the maximum number to be executed.	Parameter information	At power-on, at RESET
36A1H	Parameter error (safety function)	• A value exceeding the maximum number of safety FB files to be executed is set in the FB/FUN file settings.	Stop	• Correct the number of safety FB files so that it is within the maximum number to be executed.	Parameter information	At power-on, at RESET
36C0H	Safety operation mode switching error	• An attempt was made to change the CPU operating status to RUN when the safety operation mode is SAFETY MODE (wait-for-restart).	Stop	• Reset the CPU module and run it again.	—	Always
36E0H	File specification error	• The program settings set in the CPU parameters do not match the program file category (standard/safety) written in the CPU module.	Stop	• Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again	Drive/file information	At power-on, at RESET, at STOP → RUN state

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
36E1H	File error	• Safety programs and safety FB files, that the data writing to the programmable controller did not complete successfully, exist. • Some safety programs and safety FBs are not in the normal state.	Stop	• Write safety programs and safety FBs to the programmable controller again and check that the data writing has completed successfully. Then, reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or safety function module. Please consult your local Mitsubishi representative.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3700H	System consistency check error (system configuration)	• The module mounting status of each slot or the module model names differ between the systems A and B. • The module model names set in the system parameters ("I/O Assignment Setting") differ between the systems A and B. (The error is detected in the control system or the standby system.)	Stop	• Match the module mounting status of each slot and the model of the modules used between the systems, and restart the CPU module. • Set the same module model name in the system parameters ("I/O Assignment Setting") for both systems.	System configuration information	At power-on, at RESET, at tracking cable connection, at operation mode change
3701H	System consistency check error (CPU module)	• The CPU module model names differ between the systems A and B. • The link direct device setting of the CPU parameters differs between the systems A and B, or a CPU module that does not support the extended mode of the link direct device setting is mounted. (The error is detected in the control system or the standby system.)	Stop	• Mount the same CPU module, and restart the CPU module. • Match the link direct device setting of the CPU parameters in both systems, or replace the CPU module with the one with a firmware version supporting the extended mode of the link direct device setting. Then, restart the system.	—	At power-on, at RESET, at tracking cable connection, at operation mode change
3710H	System consistency check error (SD memory card mounting status)	• The memory card mounting status differs between the systems A and B. (The error is detected in the control system or the standby system.)	Stop	• Match the memory card mounting status between the systems, and restart the CPU module.	—	At power-on, at RESET
3711H	System consistency check error (memory card write protect switch status)	• The write protect switch status of the memory card differs between the systems A and B. (The error is detected in the control system or the standby system.)	Stop	• Match the write protect switch status between the systems, and restart the CPU module.	—	At power-on, at RESET
3714H	System consistency check error (file)	• A file mismatch was detected in the system consistency check. (The error is detected in the standby system.)	Stop	• Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, format the corresponding drive of the standby system CPU module, write all files to the CPU module, and restart the standby system CPU module. (Set the same number of steps as that of the control system CPU module in "Allocate Memory for Online Program Change" on the "Program Detail Setting" window.) • Execute the memory copy to match files in the control system and the standby system. Then, reset the CPU module of the standby system and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information	At write, at power-on, at RESET, at STOP → RUN, at system switching execution, at tracking cable connection

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3730H	Redundant function module communication error	• Data communications with the redundant function module has failed.	Stop	• Check that there is no error in the CPU module, base unit, or redundant function module. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	Always
3740H	Redundant system error	• The operation mode differs between the systems A and B. (The error is detected in the standby system.)	Stop	• Restart the standby system CPU module.	—	At power-on, at reset, at tracking cable connection
3741H	Redundant system error	• Both systems were set as system A or system B at start-up. • The CPU module waiting for a start-up of the other system detected that the system A/B setting is not set to the CPU module of the other system.	Stop	• Set one system as system A and the other system as system B by performing online operation. Then, restart the CPU modules in both systems. • Set system A or B to the CPU module with no system setting, and restart the CPU modules in both systems. • Restore the backup data including the system A/B settings to set one system as system A and the other system as system B.	—	At power-on, at reset, at tracking cable connection
3742H	Redundant system error	• The systems were started without system A/B setting.	Stop	• Set one system as system A and the other system as system B by performing online operation. Then, restart the CPU modules in both systems. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, or tracking cable. Please consult your local Mitsubishi representative.	—	At power-on, at RESET
3743H	Redundant system error	• The system setting (system A or B) was not determined.	Stop	• Set one system as system A and the other system as system B by performing online operation. Then, restart the CPU modules in both systems.	—	At power-on, at RESET
3744H	Redundant system error	• Both systems are set as control systems. (The error is detected in the system B (control system).)	Stop	• Restart the CPU module in the system B.	—	At tracking cable connection
3745H	Redundant system error	• The systems were started with one CPU module set to redundant mode and the other CPU module set to process mode. • The systems were started in a state where the CPU modules with firmware versions that cannot be used together are mounted. • The systems were started with the Process CPU and the SIL2 Process CPU.	Stop	• When using the CPU modules with firmware versions that cannot be used together, replace either CPU module with the one that can be used together referring to the MELSEC iQ-R Module Configuration Manual. When using the Process CPU and the SIL2 Process CPU together, replace either CPU module so that the same CPU module models are used in both systems. Then, write a project to the CPU module in process mode or to the one after replacement and restart the CPU modules in both systems.	—	At power-on, at RESET

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3746H	Tracking communications disabled	• Data communications with the other system could not be performed within the period of time set in the CPU parameters ("Other System Start-up Timeout Setting").	Stop	• If the other system CPU module is powered off, restart the CPU modules in both systems. • If a WDT error has occurred in the other system CPU module, eliminate the error cause, and restart the system. • Connect tracking cables securely and properly to the connectors of the redundant function modules in both systems, and restart the CPU module(s) with the error. (One cable shall be connected between the IN connector of the module in system A and the OUT connector of the module in system B. The other cable shall be connected between the OUT connector of the module in system A and the IN connector of the module in system B.) • Set a longer time in the CPU parameters ("Other System Start-up Timeout Setting"). • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.	—	At power-on, at RESET
3747H	Redundant system error	• The system cannot be started as a control system because the CPU module of the other system has been stopped due to a stop error.	Stop	• Eliminate the error cause, and restart the CPU modules in both systems.	—	At power-on, at RESET
3748H	Redundant system error	• The systems were started in a state where the CPU modules with firmware versions that cannot be used together are mounted. • The systems were started with the Process CPU and the SIL2 Process CPU. • In a redundant system with redundant extension base unit, the extension cable is not connected between the main base unit of one system and the extension base unit. Or, the extension base unit cannot be recognized by the other system. (The error is detected in the control system or the standby system.)	Stop	• Replace one CPU module with the one having a firmware version that can be used with the other CPU module together, referring to the MELSEC iQ-R Module Configuration Manual. Then, restart the system. • Mount the same CPU module, and restart the CPU module. • Check the system configurations of both systems and correct them. • Connect the extension cable securely and properly to the connector of the redundant extension base unit, and then restart the CPU module in which the error has been detected. If the same error code is displayed again, the possible cause is a failure of the extension cable. Replace the extension cable.	—	At power-on, at RESET
374BH	Redundant system error	• In a redundant system with redundant extension base unit, data communications with the other system could not be performed at start-up. • In a redundant system with redundant extension base unit, an error has been detected in the extension base unit. (The error is detected in the standby system.)	Stop	• Connect tracking cables securely and properly to the connectors of the redundant function modules in both systems, and restart the CPU module(s) with the error. (One cable shall be connected between the IN connector of the module in system A and the OUT connector of the module in system B. The other cable shall be connected between the OUT connector of the module in system A and the IN connector of the module in system B.) • If the other system CPU module is powered off, restart the CPU modules in both systems. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.	—	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
374CH	Redundant system error	• In a redundant system with redundant extension base unit, the other system did not respond during initial communication at start-up. • An error has occurred at start-up since the startup procedure of the redundant system with redundant extension base unit was not followed. (The error is detected in the standby system.)	Stop	• Restart the system in which the error has been detected. • Check the startup procedure of the redundant system with redundant extension base unit, and start up the system following the procedure.	—	At power-on, at RESET
374DH	Redundant system error	• In a redundant system with redundant extension base unit, the other system has started while only the standby system is operating. • In a redundant system with redundant extension base unit, an error has been detected in the extension base unit. (The error is detected in the control system.)	Stop	• Restart the CPU modules in both systems. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or extension cable. Please consult your local Mitsubishi representative.	—	At power-on, at RESET
374EH	Redundant system error	• The systems have been started with a configuration where CPU modules which are set to process mode are used with redundant extension base units.	Stop	• Set the CPU module to redundant mode, and restart the system. Or, remove the redundant extension base units, and restart the system. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
3750H	Extension cable failure	• Incorrect connections of the extension cables have been detected in the redundant extension base unit. (The error is detected in the control system.)	Stop	• Check the detailed information (extension cable information) of the error by executing module diagnostics using the engineering tool and identify the extension cables incorrectly connected. Connect the OUT1 connector to the IN1 connector of the next level, and connect the OUT2 connector to the IN2 connector of the next level.	Extension cable information	Always
3752H	System bus error	• In a redundant configuration of extension cables, a bus access error has been detected in an extension cable (active side) between redundant extension base units. • In a single configuration of extension cables, a bus access error has been detected in an extension cable between extension base units. (The error is detected in the control system.)	Stop	• Check the detailed information (extension cable information) of the error by executing module diagnostics using the engineering tool, and connect the extension cable, in which the error has been occurred, securely and properly to the connectors of the base units. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or extension cable. Please consult your local Mitsubishi representative.	Extension cable information	Always
3755H	System bus error	• In a redundant system with redundant extension base unit, a bus access error has been detected in an extension cable between a main base unit and the redundant extension base unit. (The error is detected in the control system or the standby system.)	Stop	• Check if the extension cable is properly connected. If not, power off the main base unit and reconnect the cable. • If the extension cable is connected properly, the possible cause is a hardware failure of the CPU module, base unit, or extension cable. Please consult your local Mitsubishi representative.	Extension cable information	Always

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3749H	Redundant system error	• Both systems are set as control systems. (The error is detected in both systems.)	Stop	• Take measures to reduce noise. • Reset the CPU modules in both systems and run them again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module of the own or other system. Please consult your local Mitsubishi representative.	—	At END instruction execution
3760H	Control system CPU module error	• In backup mode, a stop error of the control system CPU module was detected. (The error is detected in the standby system.)	Stop	• Eliminate the error cause, and restart the CPU modules in both systems.	—	Always
3780H	Redundant system configuration error	• A network module that does not support the redundant system function is mounted on the main base unit. • A network module with a firmware version that does not support the redundant system function is mounted on the main base unit. • An Ethernet interface module whose network type of the port 1 and/or port 2 is set to an option other than "Ethernet" is mounted on the main base unit.	Stop	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed slot number. If the module does not support the redundant system function, remove the module from the base unit. • Check the firmware version of the corresponding module. If the module does not support the redundant system function, remove the module from the base unit. • Set the network type of both port 1 and port 2 to "Ethernet" in the system parameters ("I/O Assignment Setting"). If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or network module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
3781H	Redundant system configuration error	• Two or more redundant function modules are mounted.	Stop	• Mount only one redundant function module.	System configuration information	At power-on, at RESET
3782H	Redundant system configuration error	• An extension base unit is connected.	Stop	• Remove all extension base units.	System configuration information	At power-on, at RESET
3783H	Redundant system configuration error	• In redundant mode, nine or more CC-Link IE Field Network modules or CC-Link modules are mounted.	Stop	• Reduce the number of CC-Link IE Field Network modules and CC-Link modules to eight or less for each.	System configuration information	At power-on, at RESET
3784H	Redundant system configuration error	• Although the control CPU is operating in redundant mode, the redundant function module is not mounted on the main base unit.	Stop	• Mount the redundant function module on the main base unit, and restart the system. Or, set the CPU module to process mode, and restart the system. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	—	At power-on, at RESET, at STOP → RUN state
3785H	Redundant system configuration error	• In a redundant system with redundant extension base unit, a base unit other than the redundant extension base unit is connected to the first extension level. (The error is detected in the control system or the standby system.)	Stop	• Connect the redundant extension base unit to the first extension level. If the same error code is displayed again, the possible cause is a hardware failure of the base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3786H	Redundant system configuration error	• In a redundant system with redundant extension base unit, redundant extension base units and other types of extension base units are mixed in the second and later extension levels. (The error is detected in the control system.)	Stop	• In a redundant configuration of extension cables, only the redundant extension base units should be used in the second and later extension levels. After replacing the extension base units, restart the system. • When extension cables are not redundant, no redundant extension base units should be used in the second and later extension levels. After replacing the redundant extension base units, restart the system.	System configuration information	At power-on, at RESET
3787H	Redundant system configuration error	• In a redundant system with redundant extension base unit, an unsupported module is mounted on an extension base unit. • In a redundant system with redundant extension base unit, a module with unsupported firmware version is mounted on an extension base unit. • In a redundant system with redundant extension base unit, an Ethernet interface module whose network type of the port 1 and/or port 2 is set to an option other than "Ethernet" is mounted on an extension base unit. (The error is detected in the control system.)	Stop	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed slot number. If the corresponding module does not support the redundant system with redundant extension base unit, remove the module from the base unit. • Check the firmware version of the corresponding module. If the module does not support the redundant system with redundant extension base unit, remove the module from the base unit. • Set the network type of both port 1 and port 2 to "Ethernet" in the system parameters ("I/O Assignment Setting"). If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or network module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
3788H	Redundant system configuration error	• In a redundant system with redundant extension base unit, the number of slots on the main base unit differs between system A and system B. • In a redundant system with redundant extension base unit, the module mounting status of each slot or the models of the modules used differ between system A and system B. • In a redundant system with redundant extension base unit, the module model name set in the system parameters ("I/O Assignment Setting") differs between system A and system B. (The error is detected in the control system or the standby system.)	Stop	• Use the main base units with the same number of slots in both systems. • Match the module mounting status of each slot and the models of the modules used on the main base unit between the systems, and restart the CPU module. • Set the system parameters ("I/O Assignment Setting") so that the parameters in both systems are the same.	System configuration information	At power-on, at RESET
3789H	Redundant system configuration error	• In a redundant system with redundant extension base unit, the main base units in both systems are not connected to the same redundant extension base unit. (The error is detected in the control system or the standby system.)	Stop	• Connect both main base units to the same redundant extension base unit.	—	At power-on, at reset, at tracking cable connection

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
378AH	Redundant system configuration error	• In a redundant system with redundant extension base unit, an unsupported extension base unit is connected to the second or later extension levels. (The error is detected in the control system.)	Stop	• Remove the unsupported extension base unit. If all the extension base units are supported, the possible cause is a hardware failure of the CPU module or extension base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
3790H	Parameter error (redundant function)	• The redundant function module was mounted next to the control CPU set to process mode.	Stop	• Set the CPU module to redundant mode, and restart the system. Or, remove the redundant function module, and restart the system. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET, at STOP → RUN state
37A0H	Parameter error (redundant function)	• The range set for tracking transfer in the CPU parameters exceeds the device range or the file register capacity set in the CPU parameters.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, correct the parameter setting corresponding to the displayed number, and take the following actions: (1) Check and correct the tracking transfer settings in the CPU parameters so that they will be within the device range or the file register capacity. (2) Check and correct the device settings or the file register capacity setting in the CPU parameters.	Parameter information	At power-on, at RESET
37A1H 37A2H	Parameter error (redundant function)	• A device that cannot be used is specified for tracking transfer in the CPU parameters.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, correct the parameter setting corresponding to the displayed number, and write parameters to the CPU module again. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or redundant function module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET
37A3H	Parameter error (redundant function)	• Total amount of data set for tracking transfer in the CPU parameters exceeds the maximum limit.	Stop	• Set the tracking capacity of each block within the allowable range in the CPU parameters.	Parameter information	At power-on, at RESET
37A4H	Parameter error (redundant function)	• In the CPU module which operates in redundant mode, parameters for the unsupported function have been set or parameters related to a redundant system have not been set.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, correct the parameter setting corresponding to the displayed number, and write parameters to the CPU module again. • Correct the operation mode set in the engineering tool. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at write
37A5H	Parameter error (redundant function)	• In the CPU module, which does not support SFC programs when operating in redundant mode, the number of points of the step relay (S) is set to other than 0 in the CPU parameters.	Stop	• Replace the CPU module with the one supporting SFC programs in redundant mode. • Set the number of points of the step relay (S) to 0 in the CPU parameters.	Parameter information	At power-on, at RESET, at write
37A6H	Parameter error (redundant function)	• "Both Systems Program Executions Setting" of "Program Setting" in the CPU parameters is set to "Both systems executions" for SFC programs.	Stop	• Check "Program Setting". Set "Both Systems Program Executions Setting" to "Control system execution" for all the SFC programs and write the CPU parameters to the CPU module again.	Parameter information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
37A7H	Parameter error (redundant function)	• Use of the redundant functions that are not supported by the modules is enabled in the parameters.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed slot number. If the module does not support the redundant function, remove the module from the base unit.	Parameter information, system configuration information	At power-on, at RESET
37A8H	Parameter error (redundant function)	• In a redundant system with redundant extension base unit, an interrupt pointer has been set to modules mounted on an extension base unit. (The error is detected in the control system or the standby system.)	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, correct the parameter setting corresponding to the displayed number, and delete the setting of the interrupt pointer for modules mounted on an extension base unit.	Parameter information	At power-on, at RESET
37A9H	Parameter error (redundant function)	• In a redundant system with redundant extension base unit, the number of slots on the redundant extension base unit is set to nine slots or more in the system parameter (I/O assignment setting). (The error is detected in the control system or the standby system.)	Stop	• Check the I/O assignment setting in the system parameter, correct the number of slots on the redundant extension base unit to eight slots or less.	Parameter information, system configuration information	At power-on, at RESET
37C0H	Program error	• The program includes an instruction that cannot be used in the operation mode set in the engineering tool.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and delete the program. Then, write the program to the CPU module again.	Error location information	At write, at power-on, at RESET, at STOP → RUN state
37C1H	Program execution error	• In the CPU module, which does not support SFC programs when operating in redundant mode, SFC programs are set in the CPU parameters ("Program Setting").	Stop	• Replace the CPU module with the one supporting SFC programs in redundant mode. • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and delete all SFC programs registered. Then, write CPU parameters to the CPU module again.	Drive/file information	At write, at power-on, at RESET, at STOP → RUN state
37D0H	Online module change error (redundant function)	• In a redundant system with redundant extension base unit, the system switching was performed when the online module change is in progress on the main base unit of the control system. (The error is detected in the standby system.)	Stop	• Power off the standby system and change the modules. Then, power on the standby system.	System configuration information	At system switching execution
37E0H	Stopped operation of the standby system due to the memory copy function	• The operation of the standby system has been suspended because the memory copy function has been executed.	Stop	• Power off and on or reset the CPU module after the memory data has been copied from the control system to the standby system.	—	At memory copy

A

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3C00H	Hardware failure	• A hardware failure has been detected.	Stop	■CPU module, SIL2 function module, safety function module • Take measures to reduce noise. • Check the mounting status of the CPU module, SIL2 function module, and safety function module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	Failure information	Always
3C01H						
3C02H						At power-on, at RESET, at END instruction execution, at interrupt occurrence
3C03H						Always
3C0FH				■CPU module, SIL2 function module, safety function module • Take measures to reduce noise. • Check the mounting status of the CPU module, SIL2 function module, and safety function module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, extended SRAM cassette, battery-less option cassette, or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.		

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3C10H	Hardware failure	• A hardware failure has been detected.	Stop	■CPU module, SIL2 function module, safety function module • Take measures to reduce noise. • Check the mounting status of the CPU module, SIL2 function module, and safety function module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET
3C11H						At END instruction execution, at instruction execution
3C12H	Hardware failure	• The waveform of the voltage out of the specified range has been detected in the power supply module. • A hardware failure has been detected in the power supply module, CPU module, base unit, or extension cable. (In a redundant power supply system, the error is detected when two power supply modules failed.)	Stop	• Check the waveform of the voltage applied to the power supply module. • Check the mounting status of the CPU module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module, CPU module, base unit, or extension cable. Please consult your local Mitsubishi representative.	Failure information	Always
3C13H	Hardware failure	• A hardware failure has been detected.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3C14H	Hardware failure	• A hardware failure has been detected.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3C15H 3C16H	Hardware failure	• A hardware failure has been detected.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module, SIL2 function module, and safety function module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution, at power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3C17H	Hardware failure	• A hardware failure has been detected.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module, SIL2 function module, and safety function module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Failure information	Always
3C20H	Memory error	• An error has been detected in the memory.				At power-on, at RESET
3C21H						At END instruction execution, at power-on, at RESET
3C22H						
3C2FH				■CPU module, SIL2 function module, safety function module • Take measures to reduce noise. • Check the mounting status of the CPU module, SIL2 function module, and safety function module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.		Always
3C30H	Memory error	• An error has been detected in the memory.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module, SIL2 function module, and safety function module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Failure information	At instruction execution
3C31H						Always
3C32H						
3C33H						At power-on, at RESET
3C34H	Memory error	• An error has been detected in the memory.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3E00H	Operation circuit error	■CPU module, SIL2 function module, safety function module • An error has been detected in the CPU module, SIL2 function module, or safety function module. ■Redundant function module • An error has been detected in the redundant function module.	Stop	■CPU module, SIL2 function module, safety function module • Take measures to reduce noise. • Check the mounting status of the CPU module, SIL2 function module, and safety function module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • Reset the CPU module. • If the error code is displayed again even after taking an action, please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET
3E01H	Operation circuit error	• An error has been detected in the CPU module, SIL2 function module, or safety function module.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module, SIL2 function module, and safety function module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution, at power-on, at RESET
3E10H	Circuit error	• An error has been detected in the CPU module, SIL2 function module, or safety function module.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module, SIL2 function module, and safety function module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution, at power-on, at RESET
3E11H						
3E12H						At power-on, at RESET
3E20H	Program execution error	• The entire program was executed without executing the END instruction.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module, SIL2 function module, and safety function module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution
3E22H	Program execution error	• The FB/FUN program did not complete successfully.				At instruction execution
3E60H	Safety mutual monitoring error	■SIL2 Process CPU, SIL2 function module • A hardware failure has been detected in the mutual monitoring between the SIL2 Process CPU and SIL2 function module. ■Safety CPU, safety function module • A hardware failure has been detected in the mutual monitoring between the Safety CPU and safety function module.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module, SIL2 function module, and safety function module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or module (SIL2 function module or safety function module) connected. Please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET, at interrupt occurrence
3E61H						Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3E68H	Hardware failure	• A hardware failure has been detected.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always

*1 The pair version information is shown only for the Safety CPU with firmware version "19" or later.

Codes of errors detected by other than the self-diagnostic function (4000H to 4FFFH)

The following table lists the codes of errors, other than those detected by the self-diagnostic function of the CPU module.

■Error codes returned to request source during communications with the CPU module

- Errors generated when the data communications are requested from the engineering tool, intelligent function module, or network system connected
- Errors generated with the data logging function

These error codes are not stored in SD0 because they are not detected by the self-diagnostic function of the CPU module.

Error code	Error name	Error details and cause	Action
4000H	Common error	• Serial communication sum check error	• Connect the serial communication cable correctly. • Take measures to reduce noise.
4001H	Common error	• An unsupported request was executed. (The request was executed to the CPU module that does not support the request.)	• Check the command data of the SLMP/MC protocol. • Check the CPU module model name selected in the engineering tool. • Check the target CPU module model name. • Check that the target network number is not duplicated.
4002H	Common error	• An unsupported request was executed.	• Check the command data of the SLMP/MC protocol. • Check the CPU module model name selected in the engineering tool. • Execute the request again. • If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.
4003H	Common error	• Command for which a global request cannot be performed was executed.	• Check the command data of the SLMP/MC protocol.
4004H	Common error	All the operations to the CPU module are disabled because of the following reason: • The CPU module is starting up.	• Perform operations to the CPU module again after the start-up processing ends.
4005H	Common error	• The amount of data handled according to a specified request is out of range.	• Check the command data of the SLMP/MC protocol.
4006H	Common error	• Initial communication has failed.	• When using serial communication, inquire of the external device manufacturer for support conditions. • When using serial communication, check the CPU module model name selected in the engineering tool. • When using Ethernet communication, shift the communication start timing.
4008H	Common error	• The CPU module is BUSY. (The buffer is not vacant.)	• Execute the request again after the specified period of time has elapsed.
400BH	Common error	• A set value of data handled according to a specified request is invalid.	• Check the request data of a dedicated instruction. • Check the command data of the SLMP/MC protocol.
4010H	CPU module operation error	• The request cannot be executed because the CPU module is running.	• Set the operating status of the CPU module to STOP, and execute the request again.
4013H	CPU module operation error	• The request cannot be executed because the CPU module is not in the STOP state.	• Set the operating status of the CPU module to STOP, and execute the request again.
4021H	File related error	• The specified drive (memory) does not exist or there is an error.	• Check the specified drive (memory) status. • Back up data in the CPU module, and then initialize the memory.
4022H	File related error	• The file with the specified file name or file No. does not exist. • The specified program block does not exist. ■When CPU Module Logging Configuration Tool is used The data logging was started in the following state. • A program name (program No.) that does not exist was specified.	• Check the specified file name and file No. • Check the specified program block name. ■When CPU Module Logging Configuration Tool is used • Check the specified program name.
4023H	File related error	• The file name and file No. of the specified file do not match.	• Delete the file and then recreate the file.
4024H	File related error	• The specified file cannot be handled by a user.	• Do not access the specified file.
4025H	File related error	• The specified file is processing the request from another engineering tool.	• Forcibly execute the request. Or, execute the request again after the processing being performed ends.
4026H	File related error	• The file password set in advance to the target drive (memory) must be specified.	• Specify the file password set in advance, and then access to the drive (memory).
4027H	File related error	• The specified range is larger than the file size.	• Check the specified range and access within that range.

Error code	Error name	Error details and cause	Action
4028H	File related error	• The same file already exists.	• Forcibly execute the request. Or, change the file name and execute the request again.
4029H	File related error	• The specified file capacity cannot be obtained.	• Review the specified file capacity, and execute the request again.
402AH	File related error	• The specified file is abnormal.	• Back up data in the CPU module, and then initialize the memory.
402BH	File related error	• The request cannot be executed in the specified drive (memory).	• Set the operating status of the CPU module to STOP, and execute the request again.
402CH	File related error	• The request cannot be executed currently.	• Execute the request again after a while.
402FH	File related error	• Writing of a file did not complete.	• Back up data internally in the CPU module, and delete the corresponding file or initialize the drive 4. Then, write the file to the programmable controller again. • Write the program restoration information. Then, read the program files from the programmable controller again.
4030H	Device specification error	• The specified device is not supported. ■When CPU Module Logging Configuration Tool is used • The data logging specifying a device that is not supported was started.	• Check the specified device.
4031H	Device specification error	• The specified device number is out of range. • The CPU module does not support the specified device. ■When CPU Module Logging Configuration Tool is used • The data logging specifying a device number that does not exist was started.	• Check the specified device number. • Check the device assignment of the CPU module. • Check the specified device.
4032H	Device specification error	• The device modification was incorrectly specified. Or, the unusable device (TS, TC, SS, SC, CS, or CC) was specified in any of the following SLMP/MC protocol commands; Read random, Write random (in units of words), Entry monitor device, or Execute monitor. ■When CPU Module Logging Configuration Tool is used • The data logging specifying a device modification that is not supported was started.	• Check the device modification method. • Check the specified device.
4033H	Device specification error	• Writing cannot be done because the specified device is for system use.	• Do not write the data in the specified device, and do not turn on or off.
4034H	Device specification error	• The dedicated instruction cannot be executed since the completion device for the dedicated instruction does not turn on.	• Since the completion device for the SREAD or SWRITE instruction does not turn on in the CPU module on the target station, execute the instruction again after setting the operating status of the CPU module on the target station to the RUN status.
4035H	Device specification error	• The request cannot be executed because the specified device is in the write-protect range.	• Do not write data to the specified device. • Check the setting of the write-protect function for device data (from outside the CPU module) for the request destination CPU module, and execute the request to the available device range again. • Clear the setting of the write-protect function for device data (from outside the CPU module), and execute the request again.
4036H	Device specification error	• The request cannot be executed because the write-protect function for device data (from outside the CPU module) is enabled.	• Check and correct the specified device, and execute the request again. ·Do not use indirect specification and index modification. ·Do not use the R device. (Use the ZR device.) • Disable the setting of the write-protect function for device data (from outside the CPU module), and execute the request again.
4038H	Device specification error	• The request cannot be executed because the write-protect function for device data (from outside the CPU module) is enabled.	• Disable the setting of the write-protect function for device data (from outside the CPU module), and execute the request again.
403FH	Device specification error	• The link direct device specified cannot be executed.	• Set the link direct device setting of the CPU parameter to the "Extended Mode (iQ-R Series Mode)".
4040H	Intelligent function module specification error	• The request cannot be executed to the specified intelligent function module.	• Check whether the specified module is the intelligent function module having the buffer memory.

Error code	Error name	Error details and cause	Action
4041H	Intelligent function module specification error	• The access range exceeds the buffer memory range of the specified intelligent function module.	• Check the start address and access number of points and access using a range that exists in the intelligent function module.
4042H	Intelligent function module specification error	• The specified intelligent function module cannot be accessed.	• Check that the specified intelligent function module is operating normally. • Check the specified module for a hardware fault.
4043H	Intelligent function module specification error	• The intelligent function module does not exist in the specified position. ■When CPU Module Logging Configuration Tool is used • The data logging specifying a device that does not exist or cannot be accessed was started.	• Check the I/O number of the specified intelligent function module.
4044H	Intelligent function module specification error	• A control bus error occurred during access to the intelligent function module.	• Check the specified intelligent function module and other modules and base units for a hardware fault.
4049H	Intelligent function module specification error	• A request cannot be processed because the module extension parameter of the positioning module is being used for the positioning control.	Turn off module ready (Yn+0) of the positioning module or execute the processing again after removing the extension parameter of the positioning module from the target data.
4050H	Protect error	• The request cannot be executed because the write protect switch of the SD memory card is on.	• Turn off the write protect switch.
4051H	Protect error	• The specified drive (memory) cannot be accessed.	Check the following and take action. • Is the drive (memory) the one that can be used? • Is the specified drive (memory) correctly installed?
4052H	Protect error	• The specified file attribute is read only, so the data cannot be written.	• Do not write data in the specified file. Or, change the file attribute.
4053H	Protect error	• An error occurred when writing data to the specified drive (memory).	• Check the specified drive (memory). Or, write data again after changing the corresponding drive (memory).
4054H	Protect error	• An error occurred when deleting the data in the specified drive (memory).	• Check the specified drive (memory). Or, delete data again after replacing the specified drive (memory).
4060H	Online registration error	• The online debug function (such as online change) and the data logging function are being executed with another engineering tool. • Data is being written to the flash ROM (data memory, program memory, and system memory) and the SD memory card. • The global label assignment information is being written to the programmable controller (data memory). ■When CPU Module Logging Configuration Tool is used • An attempt was made to write or delete data logging settings or to execute data logging to the setting registered by another request source.	• Execute the function again after the processing of the function executed from another engineering tool ends. • If the operation of another engineering tool is on hold, resume and finish the operation of another engineering tool, and then execute the function again. • Execute the function again after the completion of writing to the flash ROM and the SD memory card. If the same error code is displayed again, reset the CPU module.
4061H	Online registration error	• Settings for the online debug function (such as online change) are incorrect. • The CPU module is powered off or reset during the monitoring.	• Register an online debug function (such as online change and external input/output forced on/off), and then execute the function. • Execute again after checking the communication route such as the communication cable. • Power on or reset the CPU module and execute the monitoring again.
4063H	Online registration error	• The registered number of locked files exceeded the maximum value.	• Execute the request again after the file access from another engineering tool ends.
4064H	Online registration error	• The specified contents of the online debug function (such as the online program change), data logging function, memory dump function, or real-time monitor function are incorrect. ■When CPU Module Logging Configuration Tool is used • The trigger logging was started in a state that the trigger condition has already been satisfied.	• Check the set data of the online debug function (such as the online program change), data logging function, memory dump function, and real-time monitor function. • Execute again after checking the communication route such as the communication cable. ■When CPU Module Logging Configuration Tool is used • Clear the satisfied trigger condition, and execute the trigger logging again.

Error code	Error name	Error details and cause	Action
4065H	Online registration error	• The device assignment information differs from the parameter. • The cassette set in the cassette setting in the CPU parameter differs from the one actually inserted. • An unsupported extended SRAM cassette or battery-less option cassette is inserted. • In the CPU module which operates in redundant mode, the number of device points of the step relay (S) is set to other than 0 in the device setting of the CPU parameters.	• Check the device assignment of the CPU module or the device assignment of the request data. • Correct the cassette setting in the CPU parameters so that it matches with the cassette actually inserted. • Replace the extended SRAM cassette or the battery-less option cassette with the one supported by the CPU module. • Set the number of device points of the step relay (S) to 0 in the device setting of the CPU parameters.
4066H	Online registration error	• The specified file password is incorrect.	• Check and specify the correct file password.
4067H	Online registration error	• Monitor communication has failed.	• Execute again after checking the communication route such as the communication cable.
4068H	Online registration error	• Operation is disabled because it is being performed with another engineering tool.	• Execute the request again after the processing of the function executed from another engineering tool ends.
406AH	Online registration error	• The drive (memory) number other than 0 to 4 was specified.	• Check the drive and specify the drive number correctly.
406BH	Online registration error	• Online operation was interrupted due to a CPU module error.	• Check the status of the CPU module by executing the module diagnostics. Identify the error, and take a corrective action referring to the troubleshooting section.
406CH	Online registration error	• The number of functions that can be executed simultaneously exceeds the limit.	• Execute again after stopping the functions that are executed in another engineering tool.
406DH	Online registration error	• The operation cannot be performed because the operation is performed from the same activation source.	• Execute again after the operation from the same activation source has completed.
406EH	Online registration error	• The specified operation cannot be maintained.	• Check and correct the operation.
4070H	Verification error	• The program not yet corrected and the one corrected by online program change are different. • The execution program that was written to the programmable controller (including online change) or the execution program that was written by using online change operation differs from the program restoration information to be written.	• Read the program from the CPU module to match it with that of the engineering tool, and then execute the online change again. • Write the program including the program restoration information to the programmable controller (including online change), or execute online change.
4071H	Verification error	• After the realtime monitor started, the CPU parameters in the CPU module has been changed or deleted.	• Start the realtime monitor again. • Write the project of GX Works3 at the time of the start of the realtime monitor to the CPU module.
4072H	Verification error	• After the realtime monitor started, the global label setting file in the CPU module or the sequence program file having a monitoring target program name has been changed or deleted.	• Read the project of GX Works3 from the CPU module, save it, and load it into GX LogViewer. Then, start the realtime monitor again. • Write the project of GX Works3 at the time of the start of the realtime monitor to the CPU module. • Check and correct the SD940 setting.
4080H	Any other error	• Request data error ■When CPU Module Logging Configuration Tool is used • Request or setting data error	• Check the request data that has been specified. ■When CPU Module Logging Configuration Tool is used • Check the specified data, and write it to the CPU module again.
4081H	Any other error	• The search target data cannot be detected.	• Check the data to be searched.
4082H	Any other error	• The specified command cannot be executed because it is being executed.	• Execute the command again after the processing of the command from another engineering tool ends.
4083H	Any other error	• An attempt was made to perform operation to the program not registered in parameter. ■When CPU Module Logging Configuration Tool is used • The data logging specifying the program that is stored in the CPU module but not registered in the CPU parameters was started.	• Register the program in parameter.
4084H	Any other error	• The specified pointer (P or I) does not exist.	• Check if the pointer (P or I) exist in the data.
4085H	Any other error	• The pointer (P or I) cannot be specified because the program is not specified in parameter.	• Register the program to be executed in parameter first, and specify the pointer (P or I).
4086H	Any other error	• The specified pointer (P or I) has already been added.	• Check and correct the pointer number to be added.
4087H	Any other error	• The number of pointers (P or I) exceeds its limit.	• Check and correct the specified pointer (P or I).

Error code	Error name	Error details and cause	Action
4088H	Any other error	• The specified step number is not at the head of the instruction. • The program differs from that stored in the CPU module. ■When CPU Module Logging Configuration Tool is used • The data logging specifying the step number that does not exist or is not specified as the start number of the instruction was started.	• Check and correct the specified step number. • Read the program from the CPU module to match it with that of the engineering tool, and then execute the online change again.
4089H	Any other error	• An attempt was made to insert/delete the END instruction by online program change.	• Check the specified program file contents. • Set the operating status of the CPU module to STOP, and write the program.
408AH	Any other error	• The file capacity exceeded after the online change was executed.	• Check the capacity of the specified program file. • Set the operating status of the CPU module to STOP, and write the program.
408BH	Any other error	• The remote request cannot be executed.	• Change the status of the CPU module so that the remote request can be executed, and execute the request again. • For remote operation, set the parameter to "Enable remote reset".
408DH	Any other error	• The instruction code that cannot be handled exists.	• Check whether the model of the used CPU module is correct or not. • The program where online change was attempted includes the instruction that cannot be handled by the CPU module specified for the project. Check the program and delete the instruction.
408EH	Any other error	• The write step is illegal. • The program differs from that stored in the CPU module.	• Set the operating status of the CPU module to STOP, and write the program. • The starting position of online program change is not specified with the correct program step number. Check whether the engineering tool supports the model and version of the CPU module that is specified for the project. • Read the program from the CPU module to match it with that of the engineering tool, and then execute the online program change again.
40A0H	SFC device specification error	• A block No. outside the range was specified.	• Check the setting and correct it.
40A1H	SFC device specification error	• The number of blocks exceeds the range.	• Check the set quantity and correct it.
40A2H	SFC device specification error	• A step No. outside the range was specified.	• Check the setting and correct it.
40A3H	SFC device specification error	• The number of steps exceeds the range.	• Check the set quantity and correct it.
40A4H	SFC device specification error	• A sequence step No. outside the range was specified.	• Check the setting and correct it.
40A5H	SFC device specification error	• The specified device is outside the range.	• Check the set quantity and correct it.
40A6H	SFC device specification error	• The block specification pattern and step specification pattern are incorrect.	• Check the setting and correct it.
40A7H	SFC device specification error	■R00CPU, R01CPU, R02CPU: • A block No. that does not exist in the 0 to 127 range was specified. ■CPU module other than above models: • A block No. that does not exist in the 0 to 319 range was specified.	• Check the setting and correct it.
40A8H	SFC device specification error	■R00CPU, R01CPU, R02CPU: • A step No. that does not exist in the 0 to 127 range was specified. ■CPU module other than above models: • A step No. that does not exist in the 0 to 511 range was specified.	• Check the setting and correct it.

Error code	Error name	Error details and cause	Action
40B0H	SFC file related error	• The drive (memory) specified with the SFC program file operation is incorrect.	• Check the setting and correct it.
40B1H	SFC file related error	• The SFC program specified with the SFC program file operation does not exist.	• Check the specified file name and correct it.
40B2H	SFC file related error	• The program specified with the SFC program file operation is not an SFC program.	• Check the specified file name and correct it.
40B3H	SFC file related error	• An attempt was made to rewrite a dedicated SFC instruction by changing an SFC program online.	• Set the operating status of the CPU module to STOP, and write the SFC program.
40B4H	SFC file related error	• An attempt was made to change or delete the active block.	• Inactivate the target block.
40B5H	SFC file related error	• The number of SFC steps after the program change exceeded the maximum.	• Reduce the number of SFC steps to be added and execute the online change again.
40B6H	SFC file related error	• The specified block does not exist.	• Read the program from the CPU module to match it with that of the engineering tool, and then execute the online change again.
40B7H	SFC file related error	• The online change cannot be executed to the standby type SFC program.	• After setting the operating status of the CPU module to STOP, write the SFC program to the programmable controller.
40B8H	SFC file related error	• The device number of SFC information device is outside the range.	• Check and correct the setting of the block information.
40B9H	SFC file related error	• The changed SFC program is incorrect.	• Execute again after checking the communication route such as the communication cable.
40BAH	SFC file related error	• The online change for each block cannot be executed to the SFC block whose number of sequential steps exceeds 32K steps.	• To execute the online change for each block, the number of sequential steps of the target SFC block must be within 32K steps. • After setting the operating status of the CPU module to STOP, write the SFC program to the programmable controller.
40BBH	SFC file related error	• The online change cannot be executed because writing to the programmable controller in the STOP state has just completed or a program execution error has occurred.	• After the operating status of the CPU module is changed from STOP to RUN, execute the online SFC block change or the online program change. • After preventing the program execution error from occurring, execute the online SFC block change or the online program change.
40BDH	SFC file related error	• Online change (SFC block) execution error	• Read the program from the CPU module to match it with that of the engineering tool, and then execute the online change again. • After setting the operating status of the CPU module to STOP, write the SFC program to the programmable controller.
40BEH	SFC file related error	• The program cannot be changed online because the part to be changed has an active step (a step holding operations).	• Omit the step from the part. • Deactivate the step.
40C0H	Label communication error	• The specified label name does not exist.	• Check the label setting. If the "Access from External Device" checkbox is not selected, check the checkbox.
40C1H	Label communication error	• Label access with a label name has failed because the specified array element number is outside the range.	• Specify the array element number within the set range.
40C2H	Label communication error	• Label access with a label name has failed because the bit array type label is not specified by bit specification.	• Specify the label by bit specification, and access again.
40C3H	Label communication error	• Label access with a label name has failed because the word array type label is not specified by word specification.	• Specify the label by word specification, and access again.
40C4H	Label communication error	• Label access with a label name has failed because the number of labels used in the program exceeded its limit.	• Reduce the number of labels in the program, and access the labels multiple times.
40C5H	Label communication error	• Label access with a label name has failed because the global label setting file and global label assignment information do not match. • The request execution has failed because the global label is being modified by the CPU module specified.	• Check the access from an external device of the specified global label setting, and write the global label setting file and the global label assignment information to the programmable controller together. • Execute the label access again after a while.

Error code	Error name	Error details and cause	Action
40C6H	Label communication error	• The request execution has failed because the global label is being modified by the CPU module specified.	• After a while, write the data to the programmable controller/ execute online change or execute the label access again.
40C7H	Label communication error	• Although changes of the global label were written to the programmable controller, the reflection operation (STOP → RUN or Power-on and reset) of the global label setting file is not performed. • The request cannot be executed because the data in the global label setting file being processed and the specified consistency check data are not the same.	• Perform the reflection operation (STOP → RUN or Power-on and reset) of the global label setting file. • Correct the details of the global label setting, and write the file to the programmable controller again.
40C8H	Label communication error	• Registration of a label definition has failed because the number of registered labels reached its limit.	• Reduce the number of registered labels with the "Access from External Device" checkbox selected.
40C9H	Label communication error	• A label definition could not be registered because the registration target memory capacity has been exceeded its limit.	• Reduce the number of registered labels with the "Access from External Device" checkbox selected. • If the data memory is specified as the registration target memory, change the memory to the SD memory card. • Change the settings of the functions that use the registration target memory.
40CAH	Label communication error	• A label definition could not be changed, added, or deleted because the label communication data does not exist in the CPU module. ■When CPU Module Logging Configuration Tool is used • An attempt was made to change, add, or delete labels without creating the label communication data.	• Write the label communication data to the programmable controller.
40CBH	Label communication error	• Data are not written to the CPU module because the data type of the specified label does not match the size of the write data.	• Change the size of data written from the external device (SLMP/MC protocol device) so that it matches the data type of the label in the specified CPU module.
40CCH	Label communication error	• Online program change has failed because the global label setting file before modification and the global label assignment information do not match.	• Write the global label setting file and the global label assignment information to the programmable controller together after modification.
40CEH	Label communication error	• An attempt was made to access a label that cannot be accessed with a label name.	• Change the data type of the specified label to the one other than "Function Block" or "Pointer". • Change the class of the specified label to the one other than "VAR_GLOBAL_CONSTANT". • Specify a device that is not being modified. (Bit-specified word devices and digit-specified bit devices can be specified.) • Change the data type of the specified label to the one other than "Bit and two-dimensional array" or "Bit and three-dimensional array".
40D0H	Label communication error	• The target CPU module does not support "Access from External Device" of "Global Label Setting".	• Disable "Access from External Device", and then write the data to the programmable controller again.
40D1H	Label communication error	• When the byte is specified for the unit for reading/writing data, an odd-numbered value is specified for the read/write array data length.	• Specify an even-numbered value for the read/write array data length.
4100H	Any other error	• Hardware failure of the CPU module.	• Replace the CPU module.
4101H	Any other error	• Serial communication connection was executed for a different CPU module series.	• Check the CPU module series.
4103H	Any other error	• The instruction written by online program change is incorrect or invalid.	• Execute online program change again. Or, set the operating status of the CPU module to STOP, and write the program.
4104H	Any other error	• An instruction(s) that cannot be executed by the CPU module set to process mode or redundant mode is included in the instructions targeted for online program change.	• Delete the instruction(s) that cannot be executed, and execute the online program change function again. Or, set the operating status of the CPU module to STOP, and write the program.
4105H	Any other error	• Hardware failure of the CPU module internal memory	• The possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.
4108H	Any other error	• The device monitor/test cannot be performed normally.	• Execute the function again. Check that the access prohibited area is not accessed, and execute the function again.

Error code	Error name	Error details and cause	Action
410AH	Any other error	• The specified command cannot be executed because the online program change is being executed. ■When CPU Module Logging Configuration Tool is used • The data logging where a step No. is specified as the collection start condition or trigger condition was started during the online program change. • The data logging where a label (global label or local label) is specified as the collection start condition, data collection target, or trigger condition was started during the online program change. • The data logging setting file where a label (global label or local label) is specified as the collection start condition, data collection target, or trigger condition was written during the online program change.	• Execute the request again after the online program change.
410BH	Any other error	• The monitor condition registration was cleared after the online program change was executed.	• Execute the registration of monitoring condition again after the online program change.
410CH	Any other error	• Writing to the specified data is not supported.	• Check that the version of the engineering tool used is correct. • Check the settings and make a correction.
410EH	Any other error	• When the execution status of the online program change is in error, the online program change command was issued.	• Issue the online program change cancel command, and execute the function again.
410FH	Any other error	• During the execution of the online program change function, the cancel command was issued by the same request source.	• Issue the command again after the currently-performed processing ends.
4110H	CPU module error	• The request cannot be executed because the CPU module is in a stop error state.	• Reset the CPU module and execute the request again.
4111H	CPU module error	• The request cannot be executed because the other CPU modules, except the host CPU module, have not yet started in a multiple CPU system.	• Execute the request again after all the other CPU modules have started.
4121H	File related error	• The specified drive (memory) or file does not exist.	• Execute again after checking the specified drive (memory) or file.
4122H	File related error	• The specified drive (memory) or file does not exist.	• Execute again after checking the specified drive (memory) or file.
4123H	File related error	• The specified drive (memory) is abnormal. ■When CPU Module Logging Configuration Tool is used • The data logging was started to the memory having an error.	• Initialize the memory, and restore the drive (memory) to its normal state.
4124H	File related error	• The specified drive (memory) is abnormal.	• Initialize the memory, and restore the drive (memory) to its normal state.
4125H	File related error	• The specified drive (memory) or file is performing processing.	• Execute again after a while.
4126H	File related error	• The specified drive (memory) or file is performing processing.	• Execute again after a while.
4127H	File related error	• File password mismatch	• Execute again after checking the file password.
4128H	File related error	• File password mismatch with copy destination	• Execute again after checking the file password.
4129H	File related error	• The request cannot be executed since the specified drive (memory) is ROM.	• Execute again after changing the target drive (memory).
412AH	File related error	• The request cannot be executed since the specified drive (memory) is ROM.	• Execute again after changing the target drive (memory).
412BH	File related error	• The specified drive (memory) is write-inhibited.	• Execute again after changing the write inhibit condition or drive (memory).
412CH	File related error	• The specified drive (memory) is write-inhibited.	• Execute again after changing the write inhibit condition or drive (memory).
412DH	File related error	• The specified drive (memory) does not have enough free space.	• Execute again after increasing the free space of the drive (memory).
412EH	File related error	• The specified drive (memory) does not have enough free space.	• Execute again after increasing the free space of the drive (memory).
412FH	File related error	• The drive (memory) capacity differs between the drive (memory) copy destination and copy source.	• Execute again after checking the drive (memory) copy destination and copy source.
4130H	File related error	• The drive (memory) type differs between the drive (memory) copy destination and copy source.	• Execute again after checking the drive (memory) copy destination and copy source.
4131H	File related error	• The file name of the file copy destination is the same as that of the copy source.	• Execute again after checking the file names.

Error code	Error name	Error details and cause	Action
4132H	File related error	• The specified number of files does not exist.	• Execute again after checking the specified data.
4133H	File related error	• The specified drive (memory) has no free space.	• Execute again after increasing the free space of the drive (memory).
4134H	File related error	• The attribute specification data of the file is wrong.	• Execute again after checking the specified data.
4135H	File related error	• The date/time data of the engineering tool (personal computer) is out of range.	• Execute again after checking the clock setting of the engineering tool (personal computer).
4136H	File related error	• The specified file already exists.	• Execute again after checking the specified file name.
4137H	File related error	• The specified file is read-only.	• Execute again after changing the condition of the specified file.
4138H	File related error	• Simultaneously accessible files exceeded the maximum.	• Execute again after decreasing file operations.
4139H	File related error	• The size of the specified file has exceeded that of the existing file.	• Execute again after checking the size of the specified file.
413AH	File related error	• The specified file has exceeded the already existing file size.	• Execute again after checking the size of the specified file.
413BH	File related error	• The same file was simultaneously accessed from different engineering tools. ■When CPU Module Logging Configuration Tool is used • An operation was performed to a file being accessed.	• Execute again after a while.
413CH	File related error	• The specified file is write-inhibited.	• Execute again after changing the file condition.
413DH	File related error	• The specified file capacity cannot be secured.	• Execute again after increasing the capacity of the specified drive (memory).
413EH	File related error	• Operation is disabled for the specified drive (memory).	• Execute again after changing the target drive (memory).
413FH	File related error	• The file is inhibited to be written to the file storage area.	• Execute again after changing the specified drive (memory).
414AH	Intelligent function module specification error	• In a multiple CPU system, operation was performed to a non-controlled intelligent function module or network module.	• Execute the operation again from the control CPU of the target module.
414CH	Intelligent function module specification error	• The specified buffer memory address cannot be accessed.	• Execute again after checking the buffer address.
4150H	File related error	• An attempt was made to initialize the drive (memory) protected by the system.	• Do not initialize the target drive (memory) as it cannot be initialized.
4151H	File related error	• An attempt was made to delete the file/folder protected by the system.	• Do not delete the target file as it cannot be deleted.
4160H	Online registration error	• The registered number of I/O devices of the forced on/off target exceeded the maximum.	• Cancel the registration of I/O devices of the forced on/off target that is not used.
4168H	Online registration error	• The number of registered settings of the device test with execution conditions has exceeded 32.	• Disable the settings of the device test with execution conditions registered in the CPU module. Alternatively, reduce the number of executional conditioned device tests to be registered at a time.
4169H	Online registration error	• No settings of the device test with execution conditions are registered.	• Check the number of registered settings of the device test with execution conditions in the CPU module, and disable the settings.
416AH	Online registration error	• The specified execution conditions do not exist. (Device test with execution conditions)	• Check whether the execution conditions (program block, step No., and execution timing) specified for disabling settings are registered in the CPU module.
416BH	Online registration error	• Other than the ladder program was specified for the registration of the device test with execution conditions.	• Check and correct the program block specified when the settings of the device test with execution conditions are registered or disabled.
41C1H	File related error	• The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41C2H	File related error	• File open specification data for file access is wrong.	• Execute again after checking the specification data.
41C3H	File related error	• Simultaneously accessible files exceeded the maximum.	• Execute again after decreasing file operations.
41C4H	File related error	• Simultaneously accessible files exceeded the maximum.	• Execute again after decreasing file operations.
41C5H	File related error	• The specified file does not exist. ■When CPU Module Logging Configuration Tool is used When an attempt was made to re-register the data logging with the previous settings, the corresponding file did not exist.	• Execute again after checking the file.

Error code	Error name	Error details and cause	Action
41C7H	File related error	• The specified file/folder or drive (memory) does not exist.	• Execute again after checking the file/folder or drive (memory).
41C8H	File related error	• The size of the specified file has exceeded that of the existing file.	• Execute again after checking the size of the specified file. • If the error recurs after re-execution, the file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41C9H	File related error	• Access to the file sector has failed. • The format information data of the target drive (memory) is abnormal.	• Back up data in the CPU module, and then initialize the memory.
41CAH	File related error	• Access to the file sector has failed. • The format information data of the target drive (memory) is abnormal.	• Back up data in the CPU module, and then initialize the memory.
41CBH	File related error	• The file name is specified in a wrong method.	• Execute again after checking the file name.
41CCH	File related error	• The specified file does not exist. Or, the specified subdirectory does not exist. ■When CPU Module Logging Configuration Tool is used • The data logging was started in a state that sub-folders for storing data logging files (or folders) cannot be created or accessed. Or, sub-folders cannot be created or accessed while the data logging is being performed or the logged data is being saved.	• Execute again after checking the name of the file and subdirectory.
41CDH	File related error	• An access to the file is prohibited in the system. ■When CPU Module Logging Configuration Tool is used • The data logging was started in a state that files (or folders) cannot be created or accessed because a file (or folder) with the same name exists. Or, files (folders) cannot be created or accessed while the data logging is being performed or the logged data is being saved.	• Do not access the specified file or subdirectory. • Execute again after checking the file and subdirectory. • Execute again after checking the open mode of the file.
41CEH	File related error	• The file cannot be written because the specified file is read-only.	• Execute again after checking the attribute of the specified file.
41CFH	File related error	• The specified drive (memory) has been used exceeding the capacity.	• Execute again after checking the drive (memory) capacity.
41D0H	File related error	• The specified drive (memory) has no free space. Or, the number of files in the directory of the specified drive (memory) has exceeded the maximum.	• Execute again after increasing the free space of the drive (memory). • Delete files in the drive (memory), and execute the function again.
41D1H	File related error	• The file name is specified in a wrong method. • The SD memory card is disabled by SM606 (SD memory card forced disable instruction).	• Execute again after checking the file name. • If the error recurs after re-execution, the file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory. • Cancel the SD memory card forced disable instruction.
41D5H	File related error	• The file of the same name exists.	• Forcibly execute the request, or execute after changing the file name.
41D6H	File related error	• The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41D7H	File related error	• The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41D8H	File related error	• The specified file is being accessed.	• Execute again after a while.
41DFH	File related error	• The specified drive (memory) is write-protected.	• Execute again after canceling the write protect of the specified drive (memory).
41E0H	File related error	• The specified drive (memory) is abnormal or does not exist.	• Check that an SD memory card is inserted, and execute the function again. • Back up data, and then initialize the memory.
41E1H	File related error	• Access to the SD memory card has failed.	• Back up data, and then write the data to the data memory.
41E4H	File related error	• Access to the SD memory card has failed.	• Execute the operation again after checking that the SD memory card has been inserted. • Execute the operation again after replacing the SD memory card. • Back up data, and then initialize the memory.

Error code	Error name	Error details and cause	Action
41E7H	File related error	• The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41E8H	File related error	• The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41E9H	File related error	• The specified file is being accessed.	• Execute again after a while.
41EBH	File related error	• The file name is specified in a wrong method.	• Execute again after checking the file name.
41ECH	File related error	• The file system of the specified drive (memory) is logically corrupted.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41EDH	File related error	• The specified drive (memory) does not have continuous free space. (The free space for file is sufficient but the continuous free space is insufficient.)	• Execute again after deleting unnecessary files.
41EFH	File related error	• Creation of the power failure backup data in the specified drive (memory) has failed.	• Execute the operation again after checking that the SD memory card has been inserted.
41F0H	File related error	• The power failure backup data of the specified drive (memory) are corrupted.	• Execute the operation again after checking that the SD memory card has been inserted.
41F3H	File related error	• The file size is larger than the following: • the value to be acquired when 2 bytes are subtracted from 4G bytes.	• Specify a smaller value for the file size when creating a file or changing the file size. Alternatively, divide the file so that each file size is smaller.
41F4H	File related error	• The request cannot be executed because the operation is prohibited by the system.	• Do not request the file operation because it is prohibited by the system.
41F5H	File related error	• The command for the (split storage) program was executed to the file other than the (split storage) program file. Or, the command not for the (split storage) program was executed to the (split storage) program file.	• Check the command, and request the command applicable to the target file.
41F6H	File related error	• The following files cannot be written because the write-protect function for device data (from outside the CPU module) is enabled. ·Initial device value file ·File register file	• Do not write the specified file. • Disable the setting of the write-protect function for device data (from outside the CPU module), and execute the request again.
41F8H	File related error	• The data is being accessed from another engineering tool.	• Data are being written to the program memory or being transferred to the backup memory. Access the file after the currently-performed processing ends.
41FAH	File related error	• Program was written beyond the area where the program can be executed.	• Execute again after reducing either the already written program or newly written program.
41FBH	File related error	• The specified file is already being manipulated from the engineering tool.	• Execute again after the currently performed operation is completed.
41FCH	File related error	• An attempt was made to initialize the drive (memory) being used.	• Stop all accesses to the specified drive (memory), and execute the request again.
41FDH	File related error	• There are no data written to the data memory.	• Write all the required files to the programmable controller.
41FEH	File related error	• An SD memory card is not inserted. • The SD memory card is being disabled. • The SD memory card is disabled by SM606 (SD memory card forced disable instruction). ■When CPU Module Logging Configuration Tool is used • The data logging was started when the CPU module is in the following state: no SD memory card is inserted; the CARD READY.LED is not on (green); or the SD memory card is forcibly disabled.	• Insert the SD memory card. • Remove the SD memory card, and insert it again. • Cancel the SD memory card forced disable instruction.
41FFH	File related error	• The type of the SD memory card differs.	• Check the type of the SD memory card.
4200H	Online module change function error	• In a redundant system with redundant extension base unit, the system switching occurred when the online module change is in progress.	• Check SD1617 (Online module change progress status) with the CPU module of the control system and resume the online module change operation.
4201H	Online module change function error	• In a redundant system with redundant extension base unit, the request cannot be executed because a module on the main base unit of the control system is being changed online.	• Issue the request again after the online module change processing ends.

Error code	Error name	Error details and cause	Action
4202H	Online module change function error	• The request cannot be executed because the specified module is being changed online. • The request cannot be executed because the redundant function module is restarting.	• Issue the request again after the online module change processing ends. • If the online module change operation cannot be completed, power off the system, and change the module. • Issue the request again after the restart of the redundant function module has completed.
4203H	Online module change function error	• In a redundant system with redundant extension base unit, modules on the main base unit of the standby system cannot be changed online.	• Power off the standby system before changing modules on the main base unit of the standby system online.
4204H	Online module change function error	• In a redundant system with redundant extension base unit, the online module change cannot be executed to the modules on an extension base unit from the CPU module of the standby system.	• Execute the online module change from the CPU module of the control system.
421DH	Online module change function error	• Redundant operations cannot be performed because the redundant function module is being changed online or is restarting.	• Perform redundant operations after the online module change processing ends or the restart has completed.
4240H	Redundant system related error	Any of the following operations is requested to the CPU module in the standby system, but cannot be performed because they are not supported. • Operation mode change • System switching • Memory copy from control system to standby system • Control system forced start-up while waiting for a start-up of the other system • External input/output forced on/off function	• Perform these operations to the CPU module in the control system.
4241H	Redundant system related error	Data communications cannot be performed with the other system because of any of the following reasons: • The CPU module of the other system is powered off or in a reset state. • The tracking communication stops due to an error in the CPU module of the own or other system. • Tracking cables are disconnected or incorrectly connected, or failed. • The redundant function module of the own or other system is being changed online, a module communication test is being performed for the module, a hardware failure has occurred in the module, or the module is restarting. • System A/B has not been determined in the CPU modules.	• If a WDT error has occurred in the CPU module of the own or other system, eliminate the error cause and then execute the operation again. • Set the own system as system A or B and the other system as the other. • Check that there is no error or failure in the CPU module, tracking cables, redundant function modules, and perform a retry. Or, perform a retry after the online module change processing or the module communication test ends. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.
4243H	Redundant system related error	• The request cannot be executed because a stop error has occurred in the CPU module of the standby system.	• Check the stop error, eliminate the error cause, and then execute the request again.
4244H	Redundant system related error	• The request cannot be executed because the operating status of the CPU module in the control system differs from that of the CPU module in the standby system.	• Match the operating status of the CPU module between the systems, and execute the request again.
4246H	Redundant system related error	• The request cannot be executed because the operation mode is being changed or the systems are being switched.	• Execute the request again after the operation mode change processing or the system switching processing ends.
4247H	Redundant system related error	• The request cannot be executed because the memory copy function to the other system is being executed.	Execute the request again after the memory copy processing ends. The completion status can be checked in the following special relay areas. • SM1654 (Memory copy being executed): Off • SM1655 (Memory copy completion): On
4248H	Redundant system related error	Data communications cannot be performed with the system specified as a connection destination (request destination module I/O number) because of either of the following reasons: • The request was issued during the system switching processing. • The system specified as a connection destination (request destination module I/O number) does not exist.	Check the following, and execute the request again. • The system switching processing has completed. • The redundant system has started up successfully. • When the memory copy with the special relay and special register is used, 03D1H (Standby system CPU module) is stored in SD1653.
4249H	Redundant system related error	• The request cannot be executed because the system setting (system A or B, control system or standby system) has not been determined.	• Set one system as system A or control system and the other system as system B or standby system, and execute the request again. Or, do not specify any connection destination system (request destination module I/O number), and execute the request again.

Error code	Error name	Error details and cause	Action
424AH	Redundant system related error	• The request cannot be executed because the system A, system B, control system, or standby system is set as a connection destination (request destination module I/O number).	• Do not specify any connection destination system (request destination module I/O number), and execute the request again.
424BH	Redundant system related error	The request cannot be executed because system switching is disabled because of either of the following reasons: • SM1646 (System switching by a user) is off. • The DCONTSW instruction is being executed.	Perform either of the following operations to enable system switching, and execute the request again. • Turn on SM1646 (System switching by a user). • Execute the ECONTSW instruction.
424CH	Redundant system related error	• The request cannot be executed because the online program change function is being executed.	• Execute the request again after the online program change processing ends.
424EH	Redundant system related error	• The request cannot be executed because an unsupported system switching is specified.	• Take measures to reduce noise, and execute the request again. If the same error code is displayed again, the possible cause is a hardware failure of the target module. Please consult your local Mitsubishi representative.
424FH	Redundant system related error	• The request cannot be executed because system switching due to a different cause was executed during execution of system switching by the engineering tool.	• Check that the systems are switched successfully. If not, monitor SD1644 (Cause of system switching failure), eliminate the error cause, and then execute the function again.
4251H	Redundant system related error	• The request cannot be executed because the redundant system is in separate mode.	• Change the operation mode to backup mode, and execute the request again.
4252H	Redundant system related error	• The systems cannot be switched because an error has occurred in an intelligent function module of the standby system.	• Identify an error module by monitoring SD1646 (System switching request status from a network module of the other system), eliminate the error cause, and then switch the systems again.
4256H	Redundant system related error	• The request cannot be executed because a timeout error has occurred in tracking communications.	• Check if the tracking cables have been properly connected. If the same error code is displayed again even after the tracking cables are connected properly, the possible cause is a hardware failure of the CPU module, redundant function module, or tracking cable. Please consult your local Mitsubishi representative.
4258H	Redundant system related error	• The operation mode cannot be changed because the control system in separate mode is in a state waiting for transition to RUN.	• Change the operating status of the CPU module whose PROGRAM RUN LED is flashing to RUN by using the RUN/STOP/RESET switch or remote operation, and change the operation mode again.
4259H	Redundant system related error	• The operation mode cannot be changed from separate mode to backup mode because the communication route differs from that of when the mode was changed from backup mode to separate mode.	• Change the operation mode using the same communication route of when the mode was changed from backup mode to separate mode.
425AH	Redundant system related error	The system cannot be started as the control system because of either of the following reasons: • The system A/B setting has not been determined. • The system is being changed to the control system. • A start-up operation has been performed in the standby system.	Take the following actions: • Set one system as system A and the other system as system B, restart the CPU module. • Check that the system has been changed to the control system. • Check the system status (control system/standby system).
425BH	Redundant system related error	• In a redundant system with redundant extension base unit, a function that is not supported by an intelligent function module on an extension base unit has been executed via the intelligent function module.	• Execute the function via a module on the main base unit. • Check the command data of the SLMP/MC protocol.
425EH	Redundant system related error	• Data communications cannot be performed with the other system because a module communication test is being executed on a redundant function module.	• Retry data communications after the module communication test ends.
425FH	Redundant system related error	• The memory copy function cannot be executed because the CPU module models differ between the control system and the standby system.	• Match the CPU module model between the systems, and execute the function again.
4269H	Any other error	• The remote RUN (function) cannot be executed.	• Execute the function again after a while.
426AH	Any other error	• The date and time data specified by an engineering tool (personal computer) are less than one hour from the start time of daylight saving time.	• Specify the time to other than one less than one hour from the start time of daylight saving time using an engineering tool (personal computer). Then, execute the function again. • Check and Correct the start time of the daylight saving time setting.

Error code	Error name	Error details and cause	Action
426BH	Any other error	• Initialization of the battery-less option cassette cannot be executed because, in the CPU parameters, the battery-less option cassette setting is set to "Not Mounted".	• Set the battery-less option cassette setting to "Mounted", write the CPU parameters, execute the initialization again.
426CH	Any other error	• The specified operation cannot be performed.	• Check the settings or the operating status of the recording function.
426DH	Any other error	• The operation cannot be performed because the recording function is running.	• Perform the operation after the recording function stops.
4270H	Debug related function error	• Data logging is being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data) to another memory.	• Register data logging to the memory where the data logging is being performed. Or, stop the data logging being performed and register again. ■When CPU Module Logging Configuration Tool is used • Start the data logging to the memory where the data logging is being performed. Or, stop the data logging being performed, and start the data logging.
4271H	Debug related function error	• The specified data logging is already being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).	• Stop the data logging. Or, write, delete, or register data logging to the setting number where no data logging is being performed.
4272H	Debug related function error	• The trigger logging specifying the device as a trigger condition is being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).	• Change the trigger condition. Or, stop the trigger logging being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data), and register another trigger logging. ■When CPU Module Logging Configuration Tool is used • Change the trigger condition. Or, stop the trigger logging being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data), and start another trigger logging.
4275H	Debug related function error	• Auto logging is being performed.	• After the auto logging, replace the SD memory card and execute again.
4276H	Debug related function error	• The function that cannot be executed during the data logging (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data) was executed.	• Stop the data logging, and then execute the function.
4277H	Debug related function error	• The number of saved files exceeded the specified number. ■When CPU Module Logging Configuration Tool is used • The data logging was started in a state where the number of saved files has exceeded the specified number. (The operation when the number of saved files exceeded is set to "Stop".) Or, the data logging was started in a state where the number of saved files has exceeded the specified number. (The operation when the number of saved files exceeded is set to "Overwrite".)	• The number of files saved in the storage destination memory has exceeded the setting value. Delete files, or change the storage destination and then register. ■When CPU Module Logging Configuration Tool is used • The number of files saved in the storage destination memory has exceeded the setting value. Delete files or change the storage destination, and then start the data logging.
4278H	Debug related function error	• An attempt was made to register data logging in a state where the saved file number has reached its maximum, FFFFFFFF. Or, the number reached to the maximum during the execution. ■When CPU Module Logging Configuration Tool is used • The data logging was started in a state where the saved file number has reached its maximum, FFFFFFFF. Or, the number reached to the maximum during the execution.	• The saved file number in the storage target memory has reached its maximum, FFFFFFFF. Delete files, or change the storage destination and then register. ■When CPU Module Logging Configuration Tool is used • The saved file number in the storage target memory has reached its maximum, FFFFFFFF. Delete files or change the storage target memory, and then perform the data logging.
427AH	Debug related function error	• The common setting file is corrupted. ■When CPU Module Logging Configuration Tool is used • The data logging was started to the memory where the corrupted common setting file exists.	• Write the common settings to the target memory again. ■When CPU Module Logging Configuration Tool is used • Write the common settings to the target memory again.

Error code	Error name	Error details and cause	Action
427BH	Debug related function error	• The data logging with the same file storage destination is being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data). ■When CPU Module Logging Configuration Tool is used • The data logging with the same file storage destination is being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).	• Stop the data logging destined for the same storage, and then register. Or, change the storage destination of the file, and then register. ■When CPU Module Logging Configuration Tool is used • Stop the data logging destined for the same storage, and then perform another data logging. Or, change the storage destination of the file, and then register.
4280H	File transfer function error	• A file transfer test was executed from another CPU Module Logging Configuration Tool during execution of a file transfer test.	• Execute the file transfer test again after the ongoing test is completed.
4282H	Debug related function error	• The registration was performed with the internal buffer capacity set to 0.	• Check and correct the internal buffer capacity setting.
4283H	Debug related function error	• An attempt was made to register trigger logging in a state that the specified number of records before trigger has exceeded the number of records that can be collected within the internal buffer capacity.	• Check and correct the internal buffer capacity setting. • Reduce the number of records before trigger.
4284H	Debug related function error	• The error codes registered for memory dump are incorrect.	• Check and correct the error codes.
4285H	Debug related function error	• A non-executable function has been executed during collection or save in memory dump.	• Execute the function again after the completion of save in memory dump.
4286H	Debug related function error	• An item that cannot be set in the Safety CPU is specified in the data logging settings.	• Correct the data logging settings.
4287H	Debug related function error	• A safety program is specified in the step No. specification.	• Ensure that no safety program is specified in the step No. specification.
4288H	Debug related function error	• The request cannot be executed because the number of characters used in the name of the specified file exceeded its limit.	• Rename the file within 61 one-byte characters (including underscore, serial number (8 digits), period, and extension), and execute the request again.
4289H	Debug related function error	• Items that cannot be set as data collection conditions are set.	• Check and correct the data collection condition setting.
428AH	Data logging function error	• The data logging has started while · Data is being written to the programmable controller online. · The global label assignment information is being written to the programmable controller (data memory).	• Start the data logging after the online data write to the programmable controller has completed. • Start the data logging after the writing of the global label assignment information to the programmable controller has completed.
428CH	Data logging function error	Execution fails because any of the following files does not exist. • CPU parameter file • Global label setting file • Program file with the target program name	Write the following files to the CPU module. • CPU parameter file • Global label setting file • Program file with the target program name
4290H	Data logging function error	• The program file or global label setting file was changed during the realtime monitor where a label (global label or local label) is specified as a monitor target, timing condition, or trigger condition.	• Do not change the program file or global label setting file during the realtime monitor where a label (global label or local label) is specified. • Stop the realtime monitor where a label (global label or local label) is specified. • Check and correct the setting of the special register (SD940), and execute the function again.
4291H	Debug related function error	• Online program change has been executed during the execution of the real-time monitor where a step number is specified as "Timing" or "Trigger condition".	• After the completion of the online program change, check the step number specified and execute real-time monitor again.
4292H	Debug related function error	• Real-time monitor does not begin during the execution of real-time monitor.	• Start real-time monitor after the stop of the real-time monitor being executed.
4293H	Debug related function error	• Execution fails because the internal buffer exceeds its maximum capacity.	• Check and correct the settings of internal buffer capacity, and then try again.
4294H	Debug related function error	• Execution fails because the settings of internal buffer capacity has been changed during the execution of functions that consume the internal buffer.	• Deactivate the functions that consume the internal buffer, and then try again. Or, restore the internal buffer capacity to the previous state, and then try again.
42A3H	Safety related error	• The safety operation mode specification details lie outside the range.	• Change the safety operation mode again.

A

Error code	Error name	Error details and cause	Action
42A4H	Safety related error	• An attempt was made to change the safety operation mode to TEST MODE while in TEST MODE, or to change the safety operation mode to SAFETY MODE while in SAFETY MODE.	• Check whether an attempt was made to change the current safety operation mode to the same mode.
42A5H	Safety related error	• An operation that cannot be executed in SAFETY MODE was performed.	• Change the safety operation mode to TEST MODE and execute again.
42A7H	Safety related error	• The safety operation mode cannot be changed to SAFETY MODE.	• Change the CPU module operating status to other than RUN and execute again. • Execute again after completing writing of the safety program, safety FB file, safety CPU parameters, safety module parameters, safety global label setting file, and standard/safety shared label setting file.
42A8H	Safety related error	• The safety operation mode cannot be changed to TEST MODE.	• Change the safety operation mode again.
42A9H	Safety related error	• A CRC error occurred during communication with the Safety CPU.	• Take measures to reduce noise, and retry data communications. • If the same error code is displayed again, the possible cause is a hardware failure of the applicable module. Please consult your local Mitsubishi representative.
42B0H		• Communications between the CPU module and the safety function module/SIL2 function module failed.	
42B4H		• An error was detected in the safety mutual monitoring between the CPU module and the safety function module/ SIL2 function module.	
42B8H	Safety related error	• The size of the program stored in the program memory and the size of the program being executed differ.	• Restart the Safety CPU. • If the same error code is displayed again, the possible cause is a hardware failure of the applicable module. Please consult your local Mitsubishi representative.
42B9H	Safety related error	• The Safety CPU received a request from an unsupported device.	• Send the request to the Safety CPU again from a supported device.
42BAH	Safety related error	• An SD memory card was specified for the write destination for the safety sequence program, safety FB file, safety CPU parameters, safety module parameters, safety global label setting file, and standard/safety shared label setting file.	• Try the write operation again.
42BBH	Safety related error	• The maximum capacity for safety programs and safety FB files that can be stored in the specified drive (memory) has been exceeding.	• Delete the safety programs or safety FB files stored in the Safety CPU drive (memory), and execute again. • Execute again after formatting Safety CPU drive (memory). • Reduce the safety program and safety FB file to be written to less than 40K steps and execute again.
42BCH	Safety related error	• A file mismatch has been detected between the CPU module and the safety function module/SIL2 function module.	• Check that the safety function module or SIL2 function module is mounted and format the memory. Write all files to the CPU module. Then, reset and run the CPU module and perform the processing again.
42BEH	Safety related error	• The number of safety sequence programs, safety FB files, safety CPU parameters, safety module parameters, safety global label setting files, and standard/safety shared label setting files exceeded the maximum number of files that can be stored on the specified drive (memory).	• Execute again after deleting the safety sequence programs, safety FB files, safety CPU parameters, safety module parameters, safety global label setting files, and standard/ safety shared label setting files on the drive (memory).
42BFH	Safety related error	• Initialization of the CPU module has been completed, but the safety function module/SIL2 function module has not been able to be initialized.	• To initialize the safety function module/SIL2 function module, mount the each module to the right of the CPU module. Power on and reset the CPU module. Then, initialize the each module again.
42C0H	Redundant system related error	• The SFC program was written by the online change to the CPU module that does not support the SFC program in the redundant mode.	• Check that the SFC program is not written to the CPU module that does not support the SFC program in the redundant mode.
42C1H	Redundant system related error	• Since the operation mode is changed or the systems were switched during the initial processing or the initial processing (when switched to RUN), it takes time to complete the processing.	• Check whether the processing ends or not after a while. • Take measures to reduce noise. • If the same error code is displayed again even after the redundant system is restarted, the possible cause is a hardware failure of the CPU module, redundant function module, or tracking cable. Please consult your local Mitsubishi representative.

Error code	Error name	Error details and cause	Action
42C2H	Redundant system related error	• The memory copy function is executed to the CPU module that does not support the selection writing of program restoration information while no program restoration information is in the CPU module of the control system.	• Execute the memory copy after the program restoration information is written to the CPU module of the control system. • Replace the CPU module of the control system with the one that supports the selection writing of program restoration information.
42C3H	Redundant system related error	• The request cannot be executed because the other system is not a standby system.	• Change the other system to a standby system and try again.
42C4H	Redundant system related error	• The request cannot be executed because the operation is not supported by the other system.	• Replace the CPU module with the one that supports the requested operation and try again.
42C5H	Redundant system related error	• In a redundant system with redundant extension base unit, access to a module on an extension base unit is attempted when the standby system is specified in "Specify Redundant CPU" in the "Specify Connection Destination" window. • In a redundant system with redundant extension base unit, access to a module on an extension base unit, such as a file operation, is attempted either connecting the device directly to the CPU module of the standby system or via a module on the main base unit of the standby system. • In a redundant system with redundant extension base unit, access to other stations via a module on the main base unit of the standby system and a module on an extension base unit is attempted.	• Specify the control system in "Specify Redundant CPU" in the "Specify Connection Destination" window and access a module on an extension base unit. • Access a module on an extension base unit either connecting the device to the CPU module of the control system directly or via a module on the main base unit of the control system. • Access other stations via the control system in a redundant system with redundant extension base unit.
42DEH	Redundant system related error	• The request cannot be executed because the safety operation modes of two CPU modules differ between the control system and standby system.	• Match the safety operation modes of two CPU modules between the control system and standby system, and execute the request again.
42DFH	Redundant system related error	• The request cannot be executed because an error has been detected in the communications between the CPU modules of the control system and the standby system.	• Check if the tracking cables have been properly connected. If the same error code is displayed again even after the tracking cables are connected properly, the possible cause is a hardware failure of the CPU module, SIL2 function module, redundant function module, or tracking cable. Please consult your local Mitsubishi representative.
42F0H	Safety related error	• The request cannot be executed because the safety operation mode change is in progress.	• Execute the request again after completion of the safety operation mode change that is currently in progress.
42F1H	Safety related error	• The operation executed is failed in the safety function module.	• Eliminate the error cause of the safety function module and execute the operation again. • Mount the safety function module with the same pair version as the one of the Safety CPU.
42F2H	Safety related error	• The parameter restored in the safety function module is different from the reflected parameter.	• Execute the request again after completion of the safety operation mode change that is currently in progress.
42F3H	Safety related error	• The request cannot be executed because the safety function module or SIL2 function module cannot be accessed.	• Check that the safety function module or SIL2 function module is mounted and execute the request again. • Check that the safety function module or SIL2 function module operates normally and execute the request again.
42F4H	Safety related error	• Initialization of the CPU module has been completed, but the safety function module cannot be initialized.	• To initialize the safety function module, mount it on the main base unit. After power on the system and reset the CPU module, initialize the safety function module again,
42F5H	Safety related error	• A file mismatch has been detected between the CPU module of the request side and the specified CPU module or between the safety function module and SIL2 function module.	• Check that the safety function module or SIL2 function module is mounted and format the memory. Write all files to the CPU module. Then, reset and run the CPU module and perform the processing again.
4300H	Security function error (user authentication)	• An operation requiring authentication was executed when no authentication was received from the CPU module. • An operation requiring authentication was executed from a startup source or function that does not support user authentication.	• Log in to the CPU module. • Perform operation from a device which supports the user authentication function.
4309H	Security function error (user authentication)	• The specified user name has not been registered in the CPU module.	• Register user management information in the CPU module. • Log in with a registered user name.

A

Error code	Error name	Error details and cause	Action
430CH	Security function error (user authentication)	• An unauthorized file access has been attempted by a user with an insufficient access level. • A file was accessed by an unauthorized user (access level) with the user authentication function enabled (file access control enabled). • The CPU module does not support the access level of the user.	• Access files within the authorized scope. • Files should be accessed by authorized users. • Files should be accessed by users with an access level supported by the CPU module.
430DH	Security function error (user authentication)	• An operation requiring authentication was executed from a startup source that does not support the user authentication function with the user authentication function enabled.	• Perform the operation from a device that supports the user authentication function.
4310H	Security function error (user authentication)	• Logon failed.	• Log on with a registered user name/password. • If the password authentication is disabled (lockout), wait for the specified time and try again.
4311H	Security function error (user authentication)	• Logon failed.	• Wait for a while and try again.
4314H	Security function error (user authentication)	• A user authentication operation was performed with the user authentication function disabled.	• Perform the operation after enabling the user authentication function.
4318H	Security function error (user authentication)	• An attempt was made to change the user management information while the CPU module was logged on by a user.	• The operation should be performed after the other user logs off.
4319H	Security function error (user authentication)	• An attempt was made to initialize all programmable controller information (operation that disables user authentication function) while logged on.	• The operation should be performed after logging off.
431BH	Security function error (user authentication)	• An unauthorized operation has been attempted by a user with an insufficient access level. • The CPU module does not support the access level of the user.	• Check the access level of the user. • Files should be accessed by users with an access level supported by the CPU module.
431CH	Security function error (user authentication)	• The number of simultaneously logged on users exceeds the maximum value, and therefore the system is unable to authenticate the applicable user.	• Unnecessary logged on users should log off. • Adjust the number of log ons to within the maximum value.
431DH	Security function error (user authentication)	• An operation that requires the authentication was performed with no user management information registered in the CPU module.	• Register user management information in the CPU module.
431EH	Security function error (user authentication)	• The access level of the operator does not match with the level registered in the CPU module.	• Log on with a user whose access level is registered in the CPU module.
431FH	Security function error (user authentication)	• The operation has been performed from the engineering tool that does not support the user authentication mode.	• Use the engineering tool that supports the user authentication mode.
4320H	Security function error (user authentication)	• The log-off operation has been performed when the operation is specified for the both systems but the own system is in the log-off state.	• In the redundant mode, perform the log-off operation while it is specified for the other system.
4321H	Security function error (user authentication)	• The user authentication function has not been executed properly. • The user authentication function has been executed on the CPU module whose operation mode is the redundant mode while a different operation mode is set with the engineering tool. • The request cannot be executed because a password change or a user information change took a long time and a timeout error has occurred.	• Execute the request again after a while. • Set the operation mode to the redundant mode with the engineering tool and execute the function again. • Change the password or the user information again. • If the same error code is displayed again, the possible cause is a hardware failure of the target module. Please consult your local Mitsubishi representative.
4322H	Security function error (user authentication)	• The request cannot be executed because another user authentication function is being executed. • The request cannot be executed because the memory copy function to the other system is being executed. • The request cannot be executed because an error has occurred while another user authentication function was being executed. • The request cannot be executed because the user authentication synchronization function for both systems is being executed.	• Execute the request again after another user authentication function has been completed. • Execute the request again after the memory copy function to the other system has been completed. • Execute the request again after a while.

Error code	Error name	Error details and cause	Action
4323H	Security function error (user authentication)	• The log-off operation has been performed when the operation is specified for the both systems but the other system is in the log-off state.	• Perform the log-off operation while it is specified for the own system.
4324H	Security function error (user authentication)	• Logon failed. • The user information of the programmable controller does not match between the connected system and the other system.	• Log on with a registered user name/password. • Set the user information of the both systems again. • To match the user information with those of the control system, execute memory copy.
4325H	Security function error (user authentication)	• Log-on to the both systems cannot be performed because the number of users who have simultaneously logged on to the both systems exceeds the maximum number. • The user information of the programmable controller does not match between the connected system and the other system.	• Log on with a registered user name/password. • Set the user information of the both systems again. • To match the user information with those of the control system, execute memory copy.
4326H	Security function error (user authentication)	• An operation was performed from the engineering tool with the version where enhanced vulnerability-measures were not taken.	• Use the engineering tool with the version where enhanced vulnerability-measures are taken.
432CH	Security function error (user authentication)	• An operation was performed with the combination of the following systems: a system with a CPU module where the enhanced vulnerability-measures are not taken; a system with a CPU module where the "setting to communicate only with GX Works3 with the vulnerability-measures enhanced version" is enabled.	• In the connected system or in another system, disable the "setting to communicate only with GX Works3 with the vulnerability-measures enhanced version".
433CH	Maintenance and inspection error	• The error was not cleared. (Error clear was performed during execution of error clear.)	• Execute again after a while. If the same error code is displayed again, the possible cause is a hardware failure of the target module. Please consult your local Mitsubishi representative.
433DH	Maintenance and inspection error	• The target module cannot handle the error clear.	• Check the target module. (Check the module in which the error occurred.)
4400H	Security function error	• A file protected by a password has been opened without unlocking the password.	• Enter a correct password and perform password authentication.
4401H	Security function error	• Read password authentication has failed when required. • The file password format is incorrect.	• Set a correct read password and perform password authentication. • Access the file with the correct method.
4402H	Security function error	• Write password authentication has failed when required. • The file password format is incorrect.	• Set a correct write password and perform password authentication. • Access the file with the correct method.
4403H	Security function error	• Both passwords for reading and for writing that are set upon Create, Change, Delete, or Disable do not match the previous ones.	• Set correct passwords for both reading and writing, and perform password authentication.
4404H	Security function error	• A file error was detected before or after performing Create, Change, or Delete.	• Initialize the drive including the target file by initializing the memory. • Write the target file to the programmable controller again, and then register or cancel the file password.
4408H	Security function error	• Password authentication failed when access was required.	• Set a correct password and perform password authentication again.
4409H	Security function error	• Password authentication failed when access was required.	• Set a correct password and perform password authentication again 1 minute later.
440AH	Security function error	• Password authentication failed when access was required.	• Set a correct password and perform password authentication again 5 minutes later.
440BH	Security function error	• Password authentication failed when access was required.	• Set a correct password and perform password authentication again 15 minutes later.
440CH 440DH	Security function error	• Password authentication failed when access was required.	• Set a correct password and perform password authentication again 60 minutes later.
440EH	Security function error	• The security function was activated and password authentication cannot be performed.	• Set a correct password and perform password authentication again after a certain period of time.
440FH	Security function error	• An operation was performed to the firmware update prohibited file with a file password set.	• Disable the file password setting.

Error code	Error name	Error details and cause	Action
4410H	Security function error	• The file of the locked CPU module is accessed without the security key authentication.	• Register the security key which locks the CPU module to the engineering tool. • When the project is opened, lock the project with the security key which locks the CPU module. • When the CPU module is locked, the access control target file cannot be accessed using the following functions or external devices. · FTP server function · SLMP/MC protocol · GOT · EZSocket
4412H	Security function error	• The security key cannot be registered to the CPU module due to the failure of the internal memory where the security key is registered. Or, the security key of the CPU module cannot be deleted.	• Hardware failure of the CPU module. Replace the CPU module.
4413H	Security function error	• Since the CPU module is locked and 32 engineering tools are reading and writing programs simultaneously, another engineering tool cannot read or write programs.	• Wait until the number of engineering tools which are reading and writing programs decreases to 31 or less, and read or write programs.
4414H	Security function error	• The request cannot be executed because the CPU module is locked.	• Do not execute the request because it cannot be executed by the CPU module which is locked.
4415H	Security function error	• The request cannot be executed because the CPU module is not locked.	• Do not execute the request because it cannot be executed by the CPU module which is not locked.
4416H	Security function error	• The request cannot be executed because the CPU module lock or unlock processing is being performed.	• Execute the request again after the lock or unlock processing ends.
4417H	Security function error	• An attempt was made to write or delete the security key with no extended SRAM cassette or battery-less option cassette inserted in the CPU module.	• Check that an extended SRAM cassette or battery-less option cassette is correctly inserted, and then perform the operation again.
4418H	Security function error	• The security key cannot be changed or deleted because any locked program exists in the CPU module.	• Unlock all the programs in the CPU module.
4422H	Security function error	• The access target CPU module does not support the security key information stored in the engineering tool.	• Change the security key information version of the engineering tool in accordance with the version supported by the target CPU module.
4424H	Security function error	• The security key set to the file written by executing the file batch online change function does not match the one registered to the CPU module (or extended SRAM cassette/ battery-less option cassette).	• Set the same security key registered to the CPU module to the file, and then execute the file batch online change function.
4425H	Security function error	• The file with a security key was written by executing the batch file online change function, but no security key is registered to the CPU module (or extended SRAM cassette/ battery-less option cassette).	• Clear the security key set to the file, and then execute the online change function.
4800H	iQ Sensor Solution related error	• The specified command cannot be executed because the iQ Sensor Solution data backup/restoration function is being executed.	• Execute the command again after the processing of the data backup/restoration function ends.
4801H	iQ Sensor Solution related error	• When the iQ Sensor Solution data backup/restoration is requested, the target module does not exist. • When the iQ Sensor Solution data restoration is requested, the specified backup folder does not exist. • The value in the special register (SD) related to the iQ Sensor Solution data backup/restoration is out of range.	• Check and correct the value in the special register, and execute the function again. • Check that the backup data created by the MELSEC iQ-R series module is specified.
4802H	iQ Sensor Solution related error	• The number of backup folders used by the iQ Sensor Solution data backup/restoration function has reached to its limit.	• Delete an existing backup folder(s), and execute the function again. • Set "Automatic specification (Folder deletion supported)" in the special register for setting a backup target folder number.
4803H	iQ Sensor Solution related error	• The system file does not exist in the specified backup data, or the system file is corrupted.	• Specify another backup data.
4804H	iQ Sensor Solution related error	• The backup file does not exist in the specified backup data, or the backup file is corrupted.	• Specify another backup data.
4805H	iQ Sensor Solution related error	• When the iQ Sensor Solution data backup function is executed, no backup target device exists. • The data backup function was executed to a device which does not support the function.	• Check and correct the value in the special register, and execute the function again. • Execute the function to a device which supports the function.

Error code	Error name	Error details and cause	Action
4806H	iQ Sensor Solution related error	An SD memory card is in the following state while the function is being executed. • An SD memory card is not inserted. • An SD memory card is disabled by SM606 (SD memory card forced disable instruction).	• Insert or re-insert an SD memory card, and execute the function again. • Enable the SD memory card operation, and execute the function again.
4807H	iQ Sensor Solution related error	• Data communications cannot be performed with a target device of the iQ Sensor Solution data backup/restoration function.	• Check the operation of the target device. • Check the network status between the target device, such as cables, hubs, and routers. • The line may be busy. Perform a retry after a while. • Check and correct the communication timeout time of data backup/restoration specified in the special register.
4808H	iQ Sensor Solution related error	• When the backup data is restored to the device supporting iQ Sensor Solution, data communications cannot be performed with the restoration target device. • When backup data is restored to the device supporting iQ Sensor Solution, the device of the specified backup data and the restoration target device do not match.	• Check the operation of the target device. • Check the connections and network status between the target device, such as cables, hubs, and routers. • The line may be busy. Perform a retry after a while. • Check and correct the specified backup data and the device supporting iQ Sensor Solution (manufacturer, model name, version) specified as the restoration target device.
4809H	iQ Sensor Solution related error	• The data backup/restoration function of iQ Sensor Solution was executed to the module which does not support the function.	• Replace the module with the one which supports the function. • Execute the function to the module which supports the function.
480AH	iQ Sensor Solution related error	The function was executed to the CC-Link module whose operation had been switched from the standby master station to master station.	After powering off and on the master station and standby master station, execute the function again.
480BH	iQ Sensor Solution related error	The function was executed for the CC-Link module to which "Read Model Name of Device Station" is not set in the setting of the automatic detection function of connected devices.	Select "Read Model Name of Device Station" in the setting of the automatic detection function of connected devices, and execute the function again.
480CH	iQ Sensor Solution related error	• The specified command cannot be executed because the automatic detection of connected device function of iQ Sensor Solution is being executed.	• Execute the command again after the automatic detection processing ends.
480DH	iQ Sensor Solution related error	• The specified command cannot be executed because the communication setting reflection function of iQ Sensor Solution is being executed. • A communication timeout occurred in a command of a device supporting iQ Sensor Solution.	• Execute the command again after the communication setting reflection processing ends. • Check and correct the communication time check setting value using the engineering tool.
480EH	iQ Sensor Solution related error	• The specified command cannot be executed because the monitor function of iQ Sensor Solution is being executed. • The specified command cannot be executed because the sensor parameter read/write function of iQ Sensor Solution is being executed. • A communication timeout occurred in a command of a device supporting iQ Sensor Solution.	• Execute the function again after a while. • Execute the command again after the sensor parameter read/write processing ends. • Check and correct the communication time check setting value using the engineering tool.
4810H	iQ Sensor Solution related error	• The series of the CPU module where the iQ Sensor Solution data restoration function is to be executed differs from the series of the CPU module where the iQ Sensor Solution data backup function was executed. Or the restoration target module differs from the backup target module.	• Check the series of the CPU module where the data backup function was executed and the backup target module.
4811H	iQ Sensor Solution related error	• The series of the CPU module where the iQ Sensor Solution data backup function is to be executed differs from the series of the CPU module where last time the function was executed. Or the last backup target module differs from the current backup target module.	• Execute the data backup function to the module where the data backup function was executed last time. • Set a different folder number.
4812H	iQ Sensor Solution related error	• The functions that cannot be executed simultaneously with the data backup/restoration function of iQ Sensor Solution, such as the file transfer function (FTP server/client), are being executed.	• Execute the function again after a while.
4902H	Any other error	• The communications have stopped because an error occurred in another simple CPU communication function setting No. for which the same communication destination is specified in the simple CPU communication function.	• Eliminate the error cause of the simple CPU communication setting No. with an error.

Error code	Error name	Error details and cause	Action
4905H	Any other error	• The capacity of label area used exceeded its limit.	• Check the program, delete the unused local and global label definitions, compile the program, and then write the program to the programmable controller. • Change the label area size in parameter, compile the program, and then write the program to the programmable controller.
4906H	Any other error	The file register data cannot be cleared by either of the following reasons: • The QDRSET instruction was executed more than once after the CPU module was powered on or reset. • The file register setting parameter is set to the one other than "Use Common File Register in All Programs".	• If the QDRSET instruction is being executed, power off and on or reset the CPU module. • The operation cannot be performed if the file register setting parameter is not set to "Use Common File Register in All Programs".
4907H	Any other error	• The target CPU module does not support "Access from External Device" of "Global Label Setting".	• Disable "Access from External Device", and then write the data to the programmable controller again.
4908H	Any other error	• The specified program execution type is not supported.	• Check the program execution type specified.
4909H	Any other error	• A second SFC program has been started while an SFC program was running.	• Check the execution status of the SFC program.
490AH	Any other error	• An SFC program is stopped while "Stop Mode" is set to "Hold output after stop".	• Set "Stop Mode" to "Stop output after stop".
490BH	Any other error	• A program has started or stopped while the CPU module is in the STOP or PAUSE state.	• Set the CPU module to the RUN state and execute the program again.
490CH	Any other error	• A control system execution program has started or stopped in CPU module of the standby system when the redundant system is in backup mode.	• Execute the program in the CPU module of the control system. • Check the name of the program specified.
490DH	Any other error	• An SFC program has started or stopped when the operation mode of the CPU module is redundant mode.	• Check the name of the program specified.
4A00H	Network error	• Access to the specified station cannot be made since the routing parameters, network number/station, or network station <-> IP-related information setting are not set to the start source CPU module and/or relay CPU module. • For routing via a multiple CPU system, the control CPU module of the network module for data routing or the CPU module for data routing has not started. • The third byte of the IP address (network number) specified for the IP communication test is the same as that of the CPU module where the test is executed. • The CPU module that performs IP packet transfer is not the control CPU of the CC-Link IE module, which is on the path that IP packets travel.	• Set the routing parameters, network number/station, or network station <-> IP-related information setting for accessing the specified station to the related stations. • Retry after a while. Or, start communication after checking that the system for data routing has started. • Do not use the same third byte of the IP address (network number) specified for the IP communication test with that of the CPU module where the test is executed. • Set the CPU module that performs IP packet transfer as the control CPU of the CC-Link IE module, which is on the path that IP packets travel.
4A01H	Network error	• The network of the number set to the routing parameters does not exist. • The specified CPU module cannot be communicated through the network that is not supported by the CPU module.	• Check and correct the routing parameters set to the related stations. • Set communication through the network that is supported by the specified CPU module.
4A02H	Network error	• Access to the specified station cannot be made.	• Check the network module for error, or check that the modules are not in offline. • Check if the network numbers/PC numbers are correctly set.
4A03H	Network error	• A request for network test was issued.	• Check the request of the SLMP/MC protocol.
4A05H	Link related error (file related error)	• 121 stations or more are specified to the station number.	• Check the station number.
4A10H	Link related error (file related error)	• The number of files in the specified folder exceeded the limit.	• Reduce the number of files in the specified folder.

Error code	Error name	Error details and cause	Action
4A20H	IP communication test error	• The upper 2 bytes of the IP addresses do not match between the CPU module and the request destination device on the same path that IP packets travel. • The upper 2 bytes of the IP addresses do not match between the CPU module and the CC-Link IE module on the same path that IP packets travel. • The upper 2 bytes of the IP addresses do not match between the CC-Link IE modules on the same path that IP packets travel. • The upper 2 bytes of the IP addresses do not match between the request source device and the CPU module connected to the request destination device by Ethernet.	• Check and correct the IP address settings of the CPU module. • Check and correct the IP address of the request destination device. • Check and correct the IP address of the CC-Link IE module. • Check and correct the IP address of the request source device.
4A21H	IP communication test error	• The 3rd byte (Network No.) or 4th byte (Station No.) of the IP address of the CPU module is the number that is not available for CC-Link IE. • The 3rd byte (Network No.) or 4th byte (Station No.) of the IP address of the request destination device is the number that is not available for CC-Link IE.	• Check and correct the IP address settings of the CPU module. • Check and correct the IP address of the request destination device.
4A22H	IP communication test error	• The IP address is not set to the CC-Link IE module on the path that IP packets travel.	• Set the IP address to the CC-Link IE module used as a master station. • Check the communication status with the master station when the CC-Link IE module is used as a local station. • Replace the CC-Link IE module (master station) with the one that supports the IP packet transfer function. • Conduct the IP communication test again after the CC-Link IE module is started up. • Check and correct the IP address of the request destination device.
4A23H	IP communication test error	• The CPU module on the path that IP packets travel does not support the IP packet transfer function. • Routing parameters are set so that IP packets are routed to CPU modules that are incompatible with IP packet transfer.	• Replace the CPU module with the one supporting the IP packet transfer function. • Correct routing parameter so that IP packets are routed to the CPU module that supports the IP packet transfer function. • Check and correct the IP address of the request destination device.
4A24H	IP communication test error	• The CC-Link IE module on the path that IP packets travel does not support the IP packet transfer function. • Routing parameters are set so that IP packets are routed to the network module that does not support the IP packet transfer function. • The 3rd byte (Network No.) of the IP address of the device on the path that IP packets travel is overlapping with the network No. of the module connected to the CPU module and does not support the IP packet transfer function. • The 3rd byte (Network No.) of the IP address of the request destination device is overlapping with the network No. of the module connected to the CPU module and does not support the IP packet transfer function.	• Replace the CC-Link IE module with the one supporting the IP packet transfer function. • Correct routing parameter so that IP packets are routed to the CC-Link IE module that supports the IP packet transfer function. • Check and correct the setting so that the 3rd byte (Network No.) of the IP address of the device on the path that IP packets travel does not overlap with the network No. of the module connected to the CPU module and does not support the IP packet transfer function. • Check and correct the setting so that the 3rd byte (Network No.) of the IP address of the request destination device does not overlap with the network No. of the module connected to the CPU module and does not support the IP packet transfer function. • Check and correct the IP address of the request destination device.
4A25H	IP communication test error	• The IP packet transfer setting is not set. • Routing parameters are set so that IP packets are routed to the CPU module where IP packet transfer setting is not set.	• Select "Use" for the IP packet transfer function setting in the CPU parameters. • Correct routing parameters so that IP packets are routed to the CPU module where IP packet transfer setting is set. • Check and correct the IP address of the request destination device.
4A27H	IP communication test error	• The CPU module that performs IP packet transfer is not the control CPU of the CC-Link IE module, which is on the path that IP packets travel.	• Set the CPU module that performs IP packet transfer as the control CPU of the CC-Link IE module, which is on the path that IP packets travel.

Error code	Error name	Error details and cause	Action
4A28H	IP communication test error	• In the system where the CPU module is connected to the request destination device over Ethernet, the request path and the response path of IP packets differ. • When multiple CC-Link IE modules with the same network number are connected in a multiple CPU system, the CC-Link IE module mounted on the lowest number slot is not set as a controlled module of the CPU module that transfers IP packets. • When multiple CC-Link IE modules with the same network number are connected in a single or multiple CPU system, the station number of the CC-Link IE module mounted on the lowest number slot is not set as a relay station number in routing parameter.	• Correct the routing parameter setting so that IP packets travel the same path for both request and response transmission. • When multiple CC-Link IE modules with the same network number are connected in a multiple CPU system, set the module mounted on the lowest number slot as a controlled module that transfers IP packets. • When multiple CC-Link IE modules with the same network number are connected in a single or multiple CPU system, set the station number of the module mounted on the lowest number slot as a relay station number in routing parameter.
4A29H	IP communication test error	• The 3rd byte (network No.) of the IP address of the request destination device is overlapping with the 3rd byte of the IP address of the CPU module connected to the request source device over Ethernet.	• Check and correct the IP address settings of the CPU module. • Check and correct the IP address of the request destination device.
4A2AH	IP communication test error	• An IP address of a device on CC-Link IE network or the CPU module is not specified.	• Specify an IP address of a device on CC-Link IE network or the CPU module.
4B00H	Target module error	• An error occurred in the access destination or the relay station. • The specified connection destination (request destination module I/O number) is incorrect. • The access destination CPU module has not started up. • The target CPU No. specified does not exist.	• Check the error occurred in the specified access destination or the relay station, and take an action. • Check the connection destination (request destination module I/O number or PC number) in the request data of SLMP/MC protocol. • Check the stop error, and take an action. • Check and correct the target CPU No.
4B02H	Target module error	• The request is not addressed to the CPU module.	• Perform operation for the module that can execute the specified function.
4B03H	Target module error	• The specified route is not supported by the specified CPU module version. • The communication target CPU module is not mounted. • A device mounted on the specified route does not support communications.	• Check whether the specified route is supported or not. • Check the mounting status of the CPU module. • Check the stop error, and take an action.
4B04H	Target module error	• The specified connection destination (request destination module I/O number) is not supported.	• An invalid value is set as the start I/O number of the connection target module. Set the start I/O number of the target module correctly, and retry data communications.
4C00H	Data logging function error	• There is not enough free space for storing the result file in the target memory.	• Increase the free space, and create the result file again.
4C01H	Data logging function error	• Writing of the result file to the target memory has not completed successfully because the SD memory card is write-protected. • Writing of the result file to the target memory has not completed successfully because the folder/file structure is incorrect.	• Unlock the write protect switch of the SD memory card, and write the result file again. • Check that the target memory is broken. • Check that the folder or file to be used in the target memory is deleted.
4C02H	Data logging function error	• The SD memory card was removed while the data logging function was being executed (data logging status: Waiting RUN Not collected, Waiting to establish collection conditions Not collected, Waiting start Not collected, Pause, Collecting, Waiting trigger Collecting before trigger, Collecting after trigger, or Saving in progress). Or, writing to the SD memory card has not completed successfully.	• Insert the SD memory card and execute the function again. • Replace the SD memory card and execute the function again.
4C03H	Data logging function error	• The number of files in the root directory and subdirectory in the target memory exceeded the limit.	• Increase the free space of the drive (memory), and execute the function again. • Delete files in the drive (memory), and execute the function again.
4C04H	Data logging function error	• During auto logging, a data logging was not registered due to a registration failure of the data logging with another setting number.	• Clear the error, and start auto logging.

Error code	Error name	Error details and cause	Action
4C05H	Data logging function error	• The online change function was executed while the data logging function specifying the step number as a sampling or trigger condition was being executed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).	• Do not execute the online change function while the data logging function specifying the step number is being executed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data). • Stop the data logging function specifying the step number.
4C06H	Data logging function error	• System error	• Check the specified data, and write it to the CPU module again.
4C07H	Data logging function error	• A data logging is attempted to start with a data storage memory that is different from the one for the data logging already registered.	• Check and correct the setting for the data logging to be started.
4C08H	Data logging function error	• Three or more of the data logging that the CPU built-in memory (function memory) is specified as the data storage memory are attempted to start.	• Check and correct the setting for the data logging to be started. • Check that the version of the CPU module is designed to make it possible to start three or more settings of logging operation with "CPU built-in memory (function memory)" specified for the data storage destination memory.
4C09H	Data logging function error	• Free space in the target memory (data memory) is not enough.	• Increase the free space in the target memory (data memory) and execute the function again.
4C0AH	Data logging function error	• The target drive (data memory) is abnormal	• Initialize the memory to bring the drive (data memory) back to normal state.
4C0BH	Data logging function error	• A data logging file that is being transferred is deleted. • Reading from (an access to) a data logging file is failed. • The specified file does not exist. Or, the specified subdirectory does not exist.	• Check and correct the number of files to be saved in the file switching setting. • Check that the data logging file is not deleted. • Check that an SD memory card is inserted. • Check the file name and subdirectory name. Then, execute the function again.
4C0CH	Data logging function error	• The data logging file transfer function is executed while the access to the SD memory card is forcibly disabled. • The SD memory card is removed during the data logging file transfer function is executed.	• Clear the disabled state of the SD memory card and execute the function again. • Insert the SD memory card and execute the function again.
4C0DH	Data logging function error	• A data logging is attempted to start during a transfer of a result file.	• Restart the data logging after the completion of the data logging file transfer function.
4C0EH	Data logging function error	• A file transfer is stopped by the data logging file transfer stop request.	• Do not send the data logging file transfer stop request.
4C0FH	Data logging function error	• The program file or global label setting file was changed while the data logging function specifying the label (global label or local label) as the collection start condition, data collection target, or trigger condition was being executed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).	• Do not change the program file or global label setting file while the data logging function specifying the label (global label or local label) is being executed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data). • Stop the data logging specifying the label (global label or local label). • Check and correct the setting of SD940 (Stop direction at file change on label specification), and execute the function again.
4C10H	CPU module data backup/ restoration function error	• The maximum allowable capacity is exceeded. • The maximum allowable number of files is exceeded. • The upper limit of the backup folder number is exceeded. • The maximum length (255 characters) of the file path is exceeded.	• Increase the free space of the SD memory card and CPU module, and execute the function again. • Delete files in the SD memory card and CPU module, and execute the function again. • Delete backup data in the SD memory card and CPU module, and execute the function again. • Check and correct the folder structure or folder/file names of backup target data, and execute the data backup function again.
4C11H	CPU module data backup/ restoration function error	• An SD memory card is not inserted. • An SD memory card is disabled by SM606 (SD memory card forced disable instruction).	• Insert or re-insert an SD memory card, and execute the function again. • Enable the SD memory card operation, and execute the function again.

A

Error code	Error name	Error details and cause	Action
4C12H	CPU module data backup/ restoration function error	• Reading/writing of data from/to an SD memory card completed with an error.	• Check that an SD memory card is inserted, and execute the function again. • Replace an SD memory card, and execute the function again. • The backup data may have been corrupted. Execute the data restoration function using another backup data.
4C13H	CPU module data backup/ restoration function error	• Reading/writing of data from/to the CPU built-in memory completed with an error.	• Back up data in the CPU built-in memory, initialize the memory, and write the data back to the original memory. Then, execute the data backup/restoration function. • The possible cause is a hardware failure of the restoration target CPU module. Execute the data restoration function to another CPU module.
4C14H	CPU module data backup/ restoration function error	• The CPU module data backup/restoration function cannot be executed because a file password is set to the data. • Data was restored to the CPU module where the same data with a file password has already been stored.	• Delete file passwords, and execute the CPU module data backup/restoration function.
4C15H	CPU module data backup/ restoration function error	• Any of the following functions that cannot be executed simultaneously with the CPU module data backup/ restoration function is being executed: the file transfer function (FTP), data backup/restoration (iQ Sensor Solution function), IP address change function, firmware update function (via the engineering tool), memory copy function, or online module change function. • The CPU module data backup/restoration function is executed when the following functions are being executed. · Online program change · File transfer function (FTP server)	• Execute the function again after a while.
4C16H	CPU module data backup/ restoration function error	• The automatic backup setting by specification of day and time is turned on while the setting value (automatic backup date and time) are out of range. • The automatic backup setting by specification of time and day of the week is turned on while the setting value (automatic backup time and day of the week) are out of range.	• Review the setting value (automatic backup date and time) and turn on the automatic backup setting by specification of day and time. • Review the setting value (automatic backup time and day of the week) and turn on the automatic backup setting by specification of time and day of the week.
4C17H	CPU module data backup/ restoration function error	• The model of the restoration target CPU module differs from the model of the backup source CPU module.	• Execute the data restoration function to the CPU module whose model is the same as that of the backup source CPU module.
4C18H	CPU module data backup/ restoration function error	• Data was restored while the operating status of the CPU module is in RUN or PAUSE.	• Change the operating status of the CPU module to STOP, and execute the function again.
4C19H	CPU module data backup/ restoration function error	• The data restoration function was executed with backup files ($BKUP_CPU_INF.BSC and BKUP_CPU.BKD) not structured properly. • Data (file(s)) is missing in the backup file ($BKUP_CPU_INF.BSC) in the backup data folder. • The data restoration function was executed with a folder where no backup files ($BKUP_CPU_INF.BSC, BKUP_CPU.BKD, and BKUP_CPU_DEVLAB.BKD) are stored.	• The backup data may have been corrupted. Execute the data restoration function using another backup data.
4C1AH	CPU module data backup/ restoration function error	• A folder with a value that matches the restoration target date folder setting value or number folder setting value does not exist in the SD memory card. • The restoration target data setting value is out of range. • The restoration target date folder setting value or number folder setting value is out of range.	• Check and correct the restoration target date folder setting value or number folder setting value, and execute the function again. • Check and correct the restoration target data setting value, and execute the function again. • Disable the automatic restoration function with the SD CARD OFF button when not using it. Then, execute the function again.
4C1BH	CPU module data backup/ restoration function error	• The data restoration function was executed to the CPU module whose status (such as programs, parameters, and file structure) differs from that of when the data backup function was executed.	• Match the CPU module status to the one at the time of backup, and execute the function again. • Set all data as the restoration target data, and execute the automatic data restoration function.

Error code	Error name	Error details and cause	Action
4C1CH	CPU module data backup/ restoration function error	• An SD memory card is not inserted. • An SD memory card is disabled by SM606 (SD memory card forced disable instruction). • An SD memory card is write-protected.	• Insert or re-insert an SD memory card, and execute the function again. • Enable the SD memory card operation, and execute the function again. • Cancel the write protection, and execute the function again.
4C1EH	CPU module data backup/ restoration function error	• The status of the SFC program (such as step status and transition conditions) was changed during execution of the data backup function.	• Take measures so that the status of the SFC program does not change during execution of the data backup function, and execute the function again.
4C1FH	CPU module data backup/ restoration function error	• The specified command cannot be executed because the CPU module data backup/restoration function is being executed.	• Execute the command again after the data backup/ restoration processing ends.
4C20H	CPU module data backup/ restoration function error	• The data backup/restoration function was executed while the CPU module was in a state where this function could not be executed.	• Deselect the "Access from External Device" checkbox in the label setting window or delete labels with this item selected, and initialize the memory which is written the global label assignment information. Then, execute the data backup function again. • Initialize the memory where the global label assignment information is written and execute the restoration function again. • Check the firmware version of the CPU module and execute the automatic data restoration function again.
4C21H	CPU module data backup/ restoration function error	• The number of backup data stored in an SD memory card exceeds the upper limit value.	• Delete backup data in the SD memory card, and execute the function again. • Check and correct the settings of the upper limit value of the number of backup data, and execute the function again.
4C22H	CPU module data backup/ restoration function error	• Bit 5 of SD944 has been turned on while the set value of SD1353 is out of the allowable range.	• Check and correct the set value of SD1353, and turn on bit 5 of SD944.
4C23H	CPU module data backup/ restoration function error	• Upper limit value of the number of the backup data cannot be changed because a CPU data folder has already been in an SD memory card.	• After deleting the CPU data folder in the SD memory card and turning off bit 5 of SD944, turn on bit 5 of SD944 again.
4C24H	CPU module data backup/ restoration function error	• The data cannot be backed up because more backup data than the upper limit value of the number of backup data exists in the SD memory card.	• Delete the backup data exceeding the upper limit value, and execute the function again. • Check and correct the settings of the upper limit value of the number of backup data, and execute the function again.
4C26H	CPU module data backup/ restoration function error	• The automatic restoration function with the SD CARD OFF button cannot be executed because the button has been pressed for more than 10 seconds after the READY LED had begun to flash.	• Release the SD CARD OFF button within 10 seconds after the READY LED begins to flash. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.
4C40H	File transfer function (FTP client) error	• When files are specified by using wild card characters for the file transfer function instruction, the number of files matched exceeds the upper limit of the transferable number of files. • When files are specified by using wild card characters for the file transfer function instruction, no files are matched.	• Check and correct the wild card specification. • Check if the specified folder path exists.
4C43H	File transfer function (FTP client) error	• The number of processing completed files for sending or acquiring FTP client file is mismatched with the total number of processing files.	• Execute the function again.
4C44H	File transfer function (FTP client) error	The file transfer function (FTP client) is executed while the following functions are being executed. • CPU module data backup/restoration function • iQ Sensor Solution data backup/restoration function	• Execute the File transfer function (FTP client) again after the mentioned function is completed.
4C50H	Special relay and special register function error	• The free space in the data memory is not enough.	• Increase the free space in the target memory (data memory) and execute the function again.
4C51H	Special relay and special register function error	• Writing to the data memory is not completed successfully.	• Initialize the memory to bring the drive (data memory) back to normal state.

Error code	Error name	Error details and cause	Action
4C52H	Special relay and special register function error	• The function memory clear request or the data memory file transfer request is sent during a data logging.	• Do not send the function memory clear request or the data memory file transfer request during a data logging.
4C53H	Special relay and special register function error	The function memory clear request or the data memory file transfer request is sent during execution of the following functions. • File batch online change • Data transfer to the data memory • Writing the global label assignment information to the programmable controller (data memory)	• Send the function memory clear request or the data memory file transfer request after the completion of the functions.
4C54H	Special relay and special register function error	• The request cannot be executed because the CPU module is not in the STOP state.	• Set the operating status of the CPU module to STOP, and execute the request again.
4C55H	Special relay and special register function error	• The number of files in the root directory and subdirectory in the data memory exceeded the limit.	• Increase the free space in the data memory and execute the request again. • Delete files in the data memory, and execute the request again.
4C56H	Special relay and special register function error	• Reading from (an access to) a file is failed. • The specified file does not exist. Or, the specified subdirectory does not exist.	• Check that the file is not deleted. • Check the file name and subdirectory name. Then, execute the function again.
4D40H	Firmware update function error (Via engineering tool)	• Access to the flash ROM of the module to be updated has failed.	• Perform the firmware update to the module again.
4D41H	Firmware update function error (Via engineering tool)	• Access to the module to be updated has failed. • The module is unable to perform the firmware update. • An incorrect firmware update file (a firmware update file not for the module to be updated) has been used. • An invalid firmware update file has been used.	Check the following and perform the firmware update again. • No hardware failure occurs in the base unit or the module. • The module started up normally. • The module is able to perform the firmware update. • The correct firmware update file for the module is set in the engineering tool. • The name or contents of the firmware update file are not changed from their original state.
4D44H	Firmware update function error (Via engineering tool)	• A firmware update file of the version that cannot be installed on the module used has been used. • The module does not support the firmware update. • The module information cannot be read from the module to be updated.	• Use the module with a firmware version that supports the firmware update using the engineering tool. • Check if the module supports the firmware update. • Check that the module is mounted properly, reset the CPU module, and perform the firmware update again.
4D45H	Firmware update function error (Via engineering tool)	• The firmware update is disabled.	• Enable the firmware update and perform the operation again.
4D46H	Firmware update function error (Via engineering tool)	• The engineering tool and the CPU module are connected incorrectly. (The cable connection and/or connection settings in the engineering tool are not correct.)	• Check the cable connection and/or the settings in the "Specify Connection Destination" window that the CPU No.1 is connected via USB or an Ethernet port ("Ethernet Port Direct Connection"/"Connection via HUB"). • Check that the "Target PLC" is set to "Not Specified" in the "Specify Connection Destination" window of the engineering tool.
4D47H	Firmware update function error (Via engineering tool)	• The operation cannot be performed because the firmware update is being performed from another engineering tool. • The operation cannot be performed because the CPU No.1 was not reset after the last firmware update. • A communication error occurred in the last firmware update.	• Perform the firmware update to the module again after the completion of the update from the other engineering tool. • Manually reset the CPU module and perform the firmware update again.
4D48H	Firmware update function error (Via engineering tool)	• The firmware update cannot be performed due to a CPU module stop error. • The module may be faulty.	• Check the parameters. • Check whether the module is mounted correctly. • If the same error code is displayed again, please consult your local Mitsubishi representative.

Error code	Error name	Error details and cause	Action
4D49H	Firmware update function error (Via engineering tool)	• The CPU No.1 has been powered off or reset during the firmware update processing. • The engineering tool has been exited or a communication error has occurred during the firmware update processing.	• Perform the firmware update again.
4D4AH	Firmware update function error (Via engineering tool)	• The firmware update has been performed to the module that the CPU No.1 does not support. • The firmware update has been performed using the firmware update file that the CPU No.1 does not support. • An invalid firmware update file has been used.	• Update the CPU No.1 to the latest firmware version and perform the firmware update again. • Ensure that the name or contents of the firmware update file is not changed from its original state.
4D4BH	Firmware update function error (Via engineering tool)	• The CPU modules of CPU No.2 and later do not support the firmware update using the engineering tool.	• Update the CPU modules of CPU No.2 and later to the latest version by performing the firmware update using an SD memory card, and perform the operation again.
4D4CH	Firmware update function error (Via engineering tool)	• The module to be updated cannot be updated on the extension base unit. • The firmware update file set in the engineering tool cannot be used for the module on the engineering tool.	• Mount the module to be updated to the main base unit and perform the firmware update again. • Set the correct firmware update file for the module to be updated in the engineering tool, and perform the firmware update again.
4D4DH	Firmware update function error (Via engineering tool)	• A firmware data error has been detected during the firmware update processing.	• Perform the firmware update again.
4D4EH	Firmware update function error (Via engineering tool)	• The specified operation cannot be performed because the firmware update is being performed. • The specified operation cannot be performed because the CPU No.1 was not reset after the firmware update.	• Reset the CPU module after the completion of the firmware update and perform the specified operation again.
4D4FH	Firmware update function error (Via engineering tool)	• The firmware update has been performed to the module controlled by the CPU module of CPU No.2 or later.	• Set its control CPU to the CPU No.1 and perform the firmware update again. • Update the firmware of CPU No.1 to the latest version and perform the firmware update again.
4D50H	Firmware update function error (Via engineering tool)	• A remote operation has been performed to the CPU module of CPU No.2 or later during the firmware update processing. • The firmware update has been performed while the CPU module of CPU No.2 or later is the RUN state.	• Reset the CPU module and perform the firmware update again. • Change the operating status of the CPU module of CPU No.2 or later to STOP. Reset the CPU module and perform the firmware update again.
4D52H	Firmware update function error (Via engineering tool)	• The control CPU does not support firmware updates using the engineering tool.	• Update the control CPU to a version supporting the firmware update function, and perform the firmware update again.
4D53H	Firmware update function error (Via engineering tool)	• Performed an update on a module controlled by a CPU module executing another function.	• Check that the control CPU is not executing another function, and perform the firmware update again.
4D5AH	Firmware update function error (Via engineering tool)	• The firmware update cannot be performed because a tracking cable is connected.	• Disconnect the tracking cable and try again.
4D5BH	Firmware update function error (Via engineering tool)	• In a redundant system with redundant extension base unit, the firmware update has been executed while both systems are powered on.	• Power off the standby system. Reset the CPU module of the control system and execute the firmware update again.
4D70H	Web server function error	• The event history cannot be displayed because it is being updated in the target module.	• Check that events are not repeatedly detected in the target module.

A

■Error codes related to the online module change function

The following table lists the codes of errors related to the online module change function.
The code of an error detected during online module change is stored in SD1618 (Online module change error code).
The code of an error when a disable request is executed during online module change is stored in SD1619 (Disable request error code during online module change).

Error code	Error name	Error details and cause	Action
4110H	Online module change error	The online module change function was executed while the CPU module was in an error state (stop error).	The module cannot be changed online. Power off the programmable controller, and replace the module.
4111H		In a multiple CPU system, the online module change function was executed even though other CPU modules have not started up.	Execute the function again after all the CPU modules in the system start up.
4202H		• An attempt was made to change two modules directly at the same time. • A module that is not ready to be removed was removed.	• Continue processing for the module being changed online. Two modules cannot be changed online at the same time. For the other module, power off the programmable controller, and replace the module. • Turn on SM1602 (Module removal request flag) and check that the value, 5, is stored in SD1617 (Online module change progress status). Then, continue processing from the step for mounting a module.
4205H		The online module change function was executed in a multiple CPU system where a CPU module that does not support this function is included.	The module cannot be changed online. Power off the programmable controller, and replace the module.
4206H		An attempt was made to change the inter-module synchronization target module online.	The module cannot be changed online. Power off the programmable controller, and replace the module.
4210H		The selected module is out of the valid range.	Check and correct the target base unit number and/or slot number, and turn on SM1600 (Module selection request flag) again.
4214H		The model of the newly-mounted module differs from that of the module before online change. Or, the newly-mounted module is not upward compatible (module mismatch).	Mount the same model or the upward compatible module, and execute the request again. • Mount a module of the same model, and perform the module recognition processing again.
4215H		• The module recognition processing is performed without the module being mounted. • The module selection processing is performed to the slot that is set as "Empty". • The module newly-mounted to the system is not mounted correctly.	• Mount a module, and perform the module recognition processing again. • Check and correct the target base unit number and/or slot number, and turn on SM1600 (Module selection request flag) again. • Mount the module correctly.
4216H		The module newly-mounted to the system online has failed. (The CPU module cannot access to the module.) • The module has failed. • The module is not mounted correctly.	Use another module, and execute the request again. Or, power off the system, and replace the module with the one that supports the online module change function. • Replace the module with the normal one. • Mount the module correctly.
4218H		• The online change target module (before change) does not support the online module change function. • The module newly-mounted to the system online (after change) does not support the online module change function. • The online module change function was executed to change the MELSEC-Q series module.	• If an error occurs at the stage of selecting a module, the module cannot be changed online. Power off the programmable controller, and replace the module. • If an error occurs at the stage of recognizing a module, replace the module with the one that supports the online module change function, and continue the processing. • The module cannot be changed online. Power off the programmable controller, and replace the module.
421AH		The online module change function was executed to change a module controlled by another CPU module.	Execute the function from the control CPU module.
4222H		The request cannot be executed because the redundant function module is restarting.	Issue the request again after the restart of the redundant function module has completed.

Codes of errors detected by other than the self-diagnostic function (6F00H to 6FFFH)

The following table lists the codes of errors, other than those detected by the self-diagnostic function of the CPU module.

Error code	Error name	Error details and cause	Action
6F00H	Transient execution error	• The internal buffer area for transient processing is used to the maximum.	• Temporarily stop transient transmissions, or reduce the frequency. Then, perform transmissions again. Or, add the COM instruction to the programmable controller CPU and increase the frequency. If the same error occurs again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.
6F01H	Transient transmission timeout error	• In transient transmission, timeout has occurred without transmission completion.	• Review the network status. If the same error occurs again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.
6F02H 6F23H 6F24H	Module error	• An error has been detected in the redundant function module.	• Please consult your local Mitsubishi representative.
6F25H	Module communication test error	• The operating status of the CPU module is in a state other than STOP. (A test cannot be executed.)	• Change the operating status of the CPU module to STOP, and execute a module communication test.
6F27H	Module communication test being executed	• A module communication test execution request was received during the test. • A module communication test termination request was received during the test.	• Since the module communication test is being executed, execute the request again after the completion of the test.
6F28H	Module being changed online	• A module communication test execution request was received during online module change. • A module communication test termination request was received during online module change.	• Since a module is being changed online, execute the request again after the completion of the change.
6F29H	Module communication test error	• A module communication test execution request was received in the state where tracking cables are connected incorrectly.	• Connect one end of a tracking cable to the IN connector and the other end to the OUT connector of the test target redundant function module, and execute the request again.
6F30H	Transient execution error (redundant function module communication being stopped)	Data communications cannot be performed with the other system because of any of the following reasons: • The CPU module of the other system is powered off or in a reset state. • The tracking communication stops due to an error in the CPU module of the own or other system. • Tracking cables are disconnected or incorrectly connected, or failed. • The redundant function module of the own or other system is being changed online, a module communication test is being performed for the module, or a hardware failure has occurred in the module.	• If a WDT error has occurred in the CPU module of the own or other system, eliminate the error cause and then execute the operation again. • Check that there is no error or failure in the CPU module, tracking cables, redundant function modules, and perform a retry. Or, perform a retry after the online module change processing or the module communication test ends. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.
6F40H	Transient execution error	The number of transient transmission requests exceeded the upper limit of simultaneously processable requests.	• Temporarily stop transient transmissions, or reduce the frequency. Then, perform transmissions again.
6F44H 6F4AH 6F4BH 6F4CH 6F4DH 6F4EH	Transient frame error	• An invalid transient frame was received.	• Check and correct the request data (send data) on the request source side, and perform transmissions again.
6F60H	Module error	• An error has been detected in the redundant function module.	• Please consult your local Mitsubishi representative.

Codes of errors detected by other than the self-diagnostic function (C000H to CFFFH)

The following table lists the codes of errors related to the Ethernet-equipped module and CC-Link IE Field Network Basic. These error codes are not stored in SD0 because they are not detected by the self-diagnostic function of the CPU module.

Error code	Error name	Error details and cause	Action
C000H to CFBFH	📖 MELSEC iQ-R Ethernet User's Manual (Application)		
CFC0H	Cyclic transmission error (master station)	• Unable to execute cyclic transmission because multiple master stations exist in the same network address.	• Check the existence status of master station in network.
CFC1H	Cyclic transmission error (master station)	• Unable to execute cyclic transmission because the error occurred in cyclic transmission.	• Take measures to reduce noise. • If the same error is displayed again, please contact your local Mitsubishi representative.
CFC8H	Cyclic transmission error (master station)	• Unable to execute cyclic transmission because the device station controlled by other master station exists.	• Check the existence status of master station in network. • Check the device station where the error occurred.
CFC9H	Cyclic transmission error (master station)	• Unable to execute cyclic transmission because the device station of the same IP address exists in the same network address.	• Check the existence status of device station in network. • Check the device station where the error occurred.
CFD0H	Master station error	• The port No. (61450) used in CC-Link IE Field Network Basic has already been used.	• Check the port No. used in Ethernet function.
CFD1H	Master station error	• Invalid value has been set in subnet mask.	• Check the parameter setting.
CFE0H	Cyclic transmission error (device station)	• The cyclic transmission was executed for the device station controlled by other master station.	• Check the existence status of master station in network. • Check the device station where the error occurred.
CFE1H	Cyclic transmission error (device station)	• The unusable number of occupied stations has been specified from master station.	• Check the number of occupied stations setting in master station parameter (Network Configuration Settings). • Check the device station where the error occurred.
CFE8H	Cyclic transmission error (device station)	• There is no response from device station.	• Check the device station disconnection detection setting in master station parameter (Network Configuration Settings). • Check the existence status of device station in network. • Check the device station which is disconnected. • Take measures to reduce noise.
CFE9H	Cyclic transmission error (device station)	• The device station of the same IP address has existed in the same network address.	• Check the device station where the error occurred.
CFF0H	Device station error	• The error occurred in device station.	• Check the device station where the error occurred.

Appendix 2 Event List

The CPU module collects information, such as errors detected by the module; operations performed for the module; and network errors, from modules, and stores the collected data into the data memory or an SD memory card. (☞ Page 148 Event History Function)
When an event occurs, its event code and details can be read by using an engineering tool.

Point

For details on events occurred in each module, refer to the manual for the module used.

How to read the event list

The event list contains the following information.

Item	Description
Event code	ID number assigned to an event
Event type	Type of an event
Event category	Category of an event
Detected event	Description of a detected event
Detailed information 1 to 5	Details of a detected event

Detailed information

The following table lists the details of information displayed in the detailed information 1 to 5.

Detailed information	Item	Description
Detailed information 1	Operation source information	Information on the operation source • Connection port (such as Ethernet and USB) • I/O number • CPU number (a number assigned to CPU modules in a multiple CPU system) • Network number • Station number • IP address • User name (when the user authentication function is enabled)
	Event history file information	Information on the event history file
	Module information	Information on the target module (I/O number)
	Firmware update information (RnCPU)	Information (firmware versions before and after update) on the firmware update using an SD memory card for the RnCPU
	Firmware update information (RnPCPU)	Information (firmware versions before and after update) on the firmware update using an SD memory card for the RnPCPU
	Firmware update information (RnENCPU (CPU part))	Information (firmware versions before and after update) on the firmware update using an SD memory card for the RnENCPU (CPU part)
	Daylight saving time status	Information on the daylight saving time status (start/end)
	CPU module data backup/restoration information	Information on the CPU module data backup/restoration function (operation, result, error, data specification, initialization, latest data, special relay, special register, operation mode)
	CPU module data backup setting information	Information on the CPU module backup setting (backup function setting, error codes)
	Data backup/restoration information of iQ Sensor Solution	Information on the data backup/restoration function of iQ Sensor Solution (operation, target device, execution unit, target module, folder number setting method, total number of executions, number of successful executions, number of executions completed with an error, folder number)
	Information on the execution status of the data backup/restoration function of iQ Sensor Solution	Information on the execution status of the data backup/restoration function of iQ Sensor Solution (result, error category, error code)
	Information on the right to use the data backup/ restoration function of iQ Sensor Solution	Information on the right to use the data backup/restoration function of iQ Sensor Solution (right-to-use number, operation)
	System switching information	Information on the system switching cause, the cause of system switching failure, and the transition state of the systems (control system and standby system)
	Start-up information	Information on the start-up • Normal start-up • Start-up with SD memory card diagnostics
	Information on the initial processing cancellation	Information on the initial processing cancellation • SD memory card diagnostics
	Extension cable information	Information on the base unit to which the extension cable in which the error occurs is connected
	Automatic standby system recovery information	Information on the automatic standby system recovery

Detailed information	Item	Description
Detailed information 2	Communication speed and communication mode	Information on the communication speed and the communication mode
	Communication status	Information on the communication status
	Security key operation information	Information on the corresponding security key
	Remote password information	Information on the corresponding remote password
	File password information	Information on the corresponding file password
	Blocked IP address information	Information on the blocked IP address
	Drive/file information	Information on the corresponding drive name and file name
	Drive number and file name	Information on the corresponding drive number and file name
	Copy source drive/file information	Information on the corresponding drive name and file name
	Operation target information	Information on the operation target: I/O number
	Clock information (before change)	Clock information before change
	Remote operation type information	Information on the remote operation type
	User authentication information	Information on the user authentication function status (enabled/disabled)
	User management information copy	Information on user management information copy (succeeded/failed)
	Login/logout information	Login/logout information of the user authentication function
	File access control information	Information on file access control: access level
	Device/label information	Information on device/label clearing
	Device/label clearing information	
	Device name information	Information on the device name
	Device name information (user-specified)	Information on the device name (at index modification/indirect specification)
	Label name information	Information on the label name
	Label name information (user-specified)	Information on the label name (at index modification/indirect specification)
	Program start information	Information on the start of a specified program
	Program stop information	Information on the stop of a specified program
	System configuration information	Information on the system configuration
	Target folder information of the CPU module data backup/restoration function	Information on the data backup/restoration target folder (folder specification, date, number)
	Target device information of the data backup/ restoration function of iQ Sensor Solution	Information on the data backup/restoration target device (station number, station sub-ID number, ID number, IP address)
	System A/B setting information	Information on the system A/B setting
	System switching information	Information on the system switching cause, the cause of system switching failure, and the transition state of the systems (control system and standby system)
	Control system start-up cause information	Information on the start-up cause of the control system
	Error description (the other system)	Description of the error when an error was detected in the other system
	Cause of tracking communication stop	Cause of tracking communication stop
	Target station information	Error description (error codes) of target station
	Firmware update information (RnENCPU (network part))	Information (firmware versions before and after update) on the firmware update using an SD memory card for the RnENCPU (Network part)
	Firmware update information	Information (firmware versions before and after update, modules whose firmware versions are updated) on the firmware update that is performed using the engineering tool
	Restricted event category	Category of event subject to the event history logging restriction
Detailed information 3	Clock information (after change)	Clock information after change
	Copy destination drive/file information	Information on the corresponding drive name and file name
	Number of points	Information on the number of points
	Written value	Information on the written value
Detailed information 4	Device name information (target device)	Information (device name, address) on the device actually accessed
	Written value	Information on the written value
Detailed information 5	Target bit No.	Information on the target bit number
	Device name information (target device)	Information (device name, address) on the device actually accessed

A

Event list

The following table lists events related to the CPU module.

Event code	Event type	Event category	Detected event	Description	Detailed information 1	Detailed information 2	Detailed information 3	Detailed information 4	Detailed information 5
00100	System	Info	Link-up	■CPU module The CPU module has entered into the link-up state as a result of an operation such as connecting a network cable between the CPU module and an external device. ■Redundant function module The redundant function module has entered into the communication enable state. *This event code does not indicate the start of tracking communication.	Operation source information	Communication speed and communication mode	—	—	—
00110			Communication start/end (TCP)	Data communications with an external device through TCP connection started. Or, data communications with an external device through TCP connection ended.		Communication status			
00120			FTP connection start/end	FTP connection with an external device started. Or, FTP connection with an external device ended.					
00130			Receive frame error	A receive frame error was detected.		—			
00140			SNTP server time synchronization failure	Time setting by the time synchronization function failed because there was no response from the SNTP server.					
00400			Power-on/reset	The CPU module has been power-on or reset.	Start-up information*2				

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
00401	System	Info	Initial processing cancelled	The CPU module has been powered off or reset during initial processing.	Information on the initial processing cancellation	—	—	—	—
00410			Boot operation	Boot operation was performed.	—				
00411			SD memory card diagnostics completed	SD memory card diagnostics have completed.	—				
00420			Event history file generation	An event history file was generated.	Event history file information				
00421			Event history logging restricted	Event history logging from the module was restricted.	Module information	Restricted event category*3			
00430			SFC program continue start not possible	An SFC program could not be resumed, and an initial start was performed.	—	—			
00440			Safety operation mode undefined	The safety operation mode was undefined, and the system started in TEST MODE.	—				
00450			Daylight saving time start/end	Daylight saving time started. Or, daylight saving time ended.	Daylight saving time status				
00460			Label initialization	After the data is rebuild (reassigned), labels were initialized when the system was powered off and on or the operating status of the CPU module was changed from STOP to RUN. (Initial values were set or values were cleared to zero.)	—				
00700			Tracking communication start	Tracking communication started.					
			Auto system A/B setting	The system A/B setting was automatically set by the redundant system. (The setting was overwritten.)	System A/B setting information				

A

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
00800	System	Warning	Link-down	■CPU module The CPU module has entered into the link-down state as a result of an operation such as disconnecting a network cable between the CPU module and an external device. ■Redundant function module The redundant function module has entered into the communication disable state as a result of any of the following: • Any tracking cable was disconnected. • The other system was powered off. • An error occurred in the cable, connector, or module.	Operation source information	Communication speed and communication mode	—	—	—
00904	System	Warning	Socket communication send error	Sending a message over socket communication failed.	Operation source information	—	—	—	—
00906	System	Warning	Alive check error	The alive status of an external device could not be checked within the period specified by the response monitoring timer.	Operation source information	—	—	—	—

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
00907	System	Warning	Divided message receive timeout error	• All the data could not be received within the period specified by the response monitoring timer. • Data of the total data length could not be received. • The remaining part of the message divided into the TCP/IP level could not be received within the period specified by the response monitoring timer.	Operation source information	—	—	—	—
00908	System	Warning	IP composition timeout error	An IP composition timeout error occurred. (The specified period of time expired without receiving all of the divided data.)	Operation source information	—	—	—	—
00909	System	Warning	TCP-specified port number error	A port number being used for opened connection was set (for TCP/IP).	Operation source information	—	—	—	—
0090A	System	Warning	UDP-specified port number error	A port number being used for opened connection was set (for UDP/IP).	Operation source information	—	—	—	—
00A00	System	Warning	Error detection in the other system	An error was detected in the other system.	—	Error description (the other system)	—	—	—
00C02	System	Warning	Abnormal response from/to the other system	• An abnormal response was sent to the other system. • An abnormal response was sent from the other system.	—	Target station information	—	—	—

A

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
00C27	System	Warning	Tracking communication stop	Tracking communication has stopped as a result of any of the following: • The standby system CPU module has been powered off or reset. • Hardware failure of the CPU module has occurred. • An error has occurred in the redundant function module. • A WDT error has occurred. • Tracking cables has been pulled out or disconnected.	—	Cause of tracking communication stop	—	—	—
00C28			Retry	• A tracking communication retry was performed due to the change in communication route caused by loopback. • A tracking communication retry was performed due to a line status error, such as faulty cable and incorrect connector connection.		—			
00C29			Module restarted	The redundant function module has restarted because the module stopped due to the following causes: • Noise • Failure of the redundant function module					
00F00			System switching (by the system)	The systems were switched due to the cause on the redundant system side.	System switching information				
00F01			Auto memory copy (control system)	The control system automatically copied its memory to the standby system.	—				
00F02			Auto memory copy (standby system)	The memory of the control system was automatically copied to the standby system.					

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
00F03	System	Warning	Automatic standby system recovery	The standby system has recovered automatically.	Automatic standby system recovery information	—	—	—	—
00F04			Automatic standby system recovery failed	The automatic standby system recovery has failed.					
00F05			Latch data clear	Due to no battery and a memory error, the latch data was cleared to zero.	—				
1000 and after		Error	When a self-diagnostic error occurs, the error is stored as an event.						
10100	Security	Info	Security key registration/ deletion	A security key was registered or deleted.	Operation source information	Security key operation information	—	—	—
10200			Remote password lock	The remote password was set.		Remote password information			
10201			Remote password unlock	The remote password unlock processing was successfully completed.					
10202			Remote password unlock failed	The remote password unlock processing failed.					
10300			Access from an IP address blocked by the IP filter setting	An access from an IP address blocked by the IP filer setting was accepted.		Blocked IP address information			
10400			File password registration/ change/deletion	A file password was successfully registered, changed, or deleted.	Operation source information	File password information			
10401			File password registration/ change/deletion failed	Registration, change, or deletion of a file password failed.					
10402			File password unlock	A file password was successfully unlocked.					
10403			File password unlock failed	Unlock of a file password failed.					
10500			Forced invalidation setting	Forced invalidation was set.	—	—			
10501			Forced invalidation cancel	Forced invalidation was canceled.					

A

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
10600	Security	Info	User authentication function enabled/ disabled	The user authentication function was enabled or disabled.	Operation source information	User authentication information	—	—	—
10601			Login succeeded	The login processing of the user authentication function completed successfully.		Login/logout information			
10602			Login failed	The login processing of the user authentication function failed.					
10603			Log off successful	The log off processing of the user authentication function completed successfully.					
10604			Log off failed	The log off processing of the user authentication function failed.					
10605			Auto logout	The system automatically logged out the user by executing the user authentication function.					
10606			User management information update	The user management information was updated. (The information was added, changed, or deleted.)		—			
10607			User management information registration (Administrators)	An operator (Administrators) was registered to the user management information.					
10608			All-users log off successful	The all-users log off processing of the user authentication function completed successfully.		Login/logout information			
10609			All-users log off failed	The all-users log off processing of the user authentication function failed.					

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
10700	Security	Info	Access from unauthorized access level	A file was accessed from an access level not allowed by the user authentication function (file access control).	Operation source information	File access control information	—	—	—
14600			User management information copy successful (copy source)	Copy of the user management information to the other system completed successfully.		User management information copy			
14601			User management information copy failed (copy source)	Copy of the user management information to the other system failed.					
14602			User management information copy successful (copy destination)	Copy of the user management information completed successfully.					
14603			User management information copy failed (copy destination)	Copy of the user management information failed.					
18200		Warning	Initialization of all programmable controller information	All information in the programmable controller was initialized.		—			

A

Event code	Event type	Event category	Detected event	Description	Detailed information 1	Detailed information 2	Detailed information 3	Detailed information 4	Detailed information 5
20100	Operation	Info	Error clear	The error was cleared.	Operation source information	Operation target information	—	—	—
20200			Event history clear	The event history was cleared.		—			
20210			Scan time clear	The scan time was cleared.					
20300			SD memory card enabled	The SD memory card was enabled.	—				
20301			SD memory card forcibly disabled	The SD memory card forced disable function was executed and the SD memory card became ready to be removed.					
20400			■RnCPU Firmware update successful via SD memory card (RnCPU) ■Process CPU Firmware update successful via SD memory card (RnPCPU)	■RnCPU RnCPU firmware update using the SD memory card was performed and completed successfully. ■Process CPU RnPCPU firmware update using the SD memory card was performed and completed successfully.	■RnCPU Firmware update information (RnCPU) ■Process CPU Firmware update information (RnPCPU)				
20401			■RnCPU Firmware update failed via SD memory card (RnCPU) ■Process CPU Firmware update failed via SD memory card (RnPCPU)	■RnCPU RnCPU firmware update using the SD memory card was performed and was not completed successfully. ■Process CPU RnPCPU firmware update using the SD memory card was performed and was not completed successfully.					
20414			Firmware update successful via engineering tool	Firmware update using the engineering tool was performed and completed successfully.	Operation source information	Firmware update information			
20415			Firmware update failed via engineering tool	Firmware update using the engineering tool was performed and was not completed successfully.					

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
20420	Operation	Info	Firmware update successful via SD memory card (RnENCPU)	RnENCPU firmware update using the SD memory card was performed and completed successfully.	Firmware update information (RnENCPU (CPU part))	Firmware update information (RnENCPU (network part))	—	—	—
20421			Firmware update failed via SD memory card (RnENCPU)	RnENCPU firmware update using the SD memory card was performed and was not completed successfully.					
20500			CPU module data backup succeeded	Data in the CPU module were successfully backed up.	CPU module data backup/ restoration information	Target folder information of the CPU module data backup/ restoration function			
20501			CPU module data backup failed	Backup of data in the CPU module failed.					
20502			CPU module data restoration succeeded	Data were successfully restored to the CPU module.					
20503			CPU module data restoration failed	Restoration of data to the CPU module failed.					
20510			CPU module data backup setting disabled	Backup function cannot be set	CPU module data backup setting information	—			

A

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
20600	Operation	Info	iQ Sensor Solution data backup succeeded	Data in the target device supporting iQ Sensor Solution were successfully backed up.	Information on the execution status of the data backup/ restoration function of iQ Sensor Solution	Target device information of the data backup/ restoration function of iQ Sensor Solution	—	—	—
20601			iQ Sensor Solution data backup failed	Backup of data in the target device supporting iQ Sensor Solution failed.					
20602			iQ Sensor Solution data restoration succeeded	Data were successfully restored to the target device supporting iQ Sensor Solution.					
20603			iQ Sensor Solution data restoration failed	Restoration of data to the target device supporting iQ Sensor Solution failed.					
20610			iQ Sensor Solution data backup start/ cancel/end	The data backup processing of the target device supporting iQ Sensor Solution started, canceled, or ended.	Data backup/ restoration information of iQ Sensor Solution	—			
20611			iQ Sensor Solution data restoration start/cancel/end	The data restoration processing of the target device supporting iQ Sensor Solution started, canceled, or ended.					
20620			Right-to-use acquisition/ release	The right to use the data backup/ restoration function of iQ Sensor Solution was acquired or released.	Information on the right to use the data backup/ restoration function of iQ Sensor Solution				

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
24000	Operation	Info	Clock setting	The clock data was set.	Operation source information	Clock information (before change)	Clock information (after change)	—	—
24001			Remote operation request accepted	A remote request (RUN, STOP, or PAUSE) was accepted.		Remote operation type information	—		
24100			Operating status change (RUN)	The operating status of the CPU module was changed to RUN.	—	—			
24101			Operating status change (STOP)	The operating status of the CPU module was changed to STOP.					
24102			Operating status change (PAUSE)	The operating status of the CPU module was changed to PAUSE.					
24120			Starting the program	The program was started.	Operation source information	Program start information			
24121			Stopping the program	The program was stopped.		Program stop information			
24200			Creation of new folders, writes to files/folders*1	• A new folder was created. • A new file was created or data was written to a file.		Drive/file information			
24201			File copy*1	A file was copied.		Copy source drive/file information	Copy destination drive/file information		
24202			Folder/file rename*1	A folder name or file name was changed.					
24300			Safety operation mode change (SAFETY MODE (wait-for-restart))	The safety operation mode was changed to SAFETY MODE (wait-for-restart).		—	—		
			Module communication test	A module communication test was executed.					
24301			Safety operation mode change (TEST MODE)	The safety operation mode was changed to TEST MODE.					

A

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
24A00	Operation	Info	Write Device in word units (1 point)	Data was written to the device in word units.	Operation source information	Device name information	Written value	—	—
24A01						Device name information (user-specified)		Device name information (target device)[*4]	
24A10			Write Device in bit units (1 point)	Data was written to the device in bit units.		Device name information		—	
24A11						Device name information (user-specified)		Device name information (target device)[*4]	Target bit No.[*4]
24A20			Write Device in double-word units (1 point)	Data was written to the device in double-word units.		Device name information		—	—
24A21						Device name information (user-specified)		Device name information (target device)[*4]	
24A40			Write Device in word units (n points)	Data was written to the device in word units.		Device name information	Number of points	Written value	
24A50			Write Device in bit units (n points)	Data was written to the device in bit units.					
24B00			Write Device in word units (1 point)	Data was written to the device in word units.			Written value	—	
24B01						Device name information (user-specified)		Device name information (target device)[*4]	
24B10			Write Device in bit units (1 point)	Data was written to the device in bit units.		Device name information		—	
24B11						Device name information (user-specified)		Device name information (target device)[*4]	Target bit No.[*4]
24B20			Write Device in double-word units (1 point)	Data was written to the device in double-word units.		Device name information		—	—
24B21						Device name information (user-specified)		Device name information (target device)[*4]	
24B30			Write Device in quad-word units (1 point)	Data was written to the device in quad-word units.		Device name information		—	
24B31						Device name information (user-specified)		Device name information (target device)[*4]	

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
24B40	Operation	Info	Write Device in word units (n points)	Data was written to the device in word units.	Operation source information	Device name information	Number of points	Written value	—
24B50			Write Device in bit units (n points)	Data was written to the device in bit units.					
24B60			Write Device in double-word units (n points)	Data was written to the device in double-word units.					
24B70			Write Device in quad-word units (n points)	Data was written to the device in quad-word units.					
24C00			Write Label in word units (1 point)	Data was written to the label in word units.		Label name information	Written value	—	
24C01						Label name information (user-specified)		Device name information (target device)[*4]	
24C10			Write Label in bit units (1 point)	Data was written to the label in bit units.		Label name information		—	
24C11						Label name information (user-specified)		Device name information (target device)[*4]	Target bit No.[*4]
24C20			Write Label in double-word units (1 point)	Data was written to the label in double-word units.		Label name information		—	—
24C21						Label name information (user-specified)		Device name information (target device)[*4]	
24C30			Write Label in quad-word units (1 point)	Data was written to the label in quad-word units.		Label name information		—	
24C31						Label name information (user-specified)		Device name information (target device)[*4]	
24C40			Write Label in word units (n points)	Data was written to the label in word units.		Label name information		—	
24C41						Label name information (user-specified)		Written value[*4]	Device name information (target device)[*4]

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
25000	Operation	Info	Online module change	The online module change processing completed.	—	System configuration information	—	—	—
25010			Online extension cable change/addition	Processing of the online extension cable change/addition has completed.	Extension cable information	—			
25200			System A/B setting write	The system A/B setting was written to the CPU modules.	Operation source information	System A/B setting information			
26000			Redundant operation mode change (backup mode)	The operation mode in a redundant system was changed to backup mode.		—			
26001			Redundant operation mode change (separate mode)	The operation mode in a redundant system was changed to separate mode.					
2A200		Warning	Memory initialization*1	The memory was initialized.	Operation source information	Drive/file information			
2A201			Device/label zero clear	Values in a device or label were cleared to zero.		Device/label information/ Device/label clearing information			
2A202			Folder/file deletion*1	A folder or file was deleted.		Drive/file information			
2A205			Battery-less option cassette initialization	The battery-less option cassette was initialized.		Drive number and file name			
2B000			System switching (by a user)	The systems were switched due to the cause on the user side.		System switching information			
2B001			Memory copy execution by the engineering tool (control system)	The memory copy function was executed by the engineering tool.		—			
2B002			Memory copy execution by the special relay and special register (control system)	The memory copy function was executed by the special relay and special register.	—				
2B003			Memory copy execution by a user (standby system)	The memory copy function was executed by a user.					
2B004			Control system forced start-up	One system was forcibly started as a control system while waiting for a start-up of the other system.	Operation source information	Control system start-up cause information			

*1 As for file-related events such as writing and deleting a file, the following files are targeted:
· Program file
· FB program file
· Parameter file
· Data logging setting file (common setting file, individual setting file)
· Memory dump setting file
When the SIL2 Process CPU or Safety CPU is used, the following files are targeted as well:
· Safety program file
· Safety FB program file
· Safety parameter file

*2 The engineering tool displays "Start-up information" in detailed information 1 only when the Process CPU with firmware version "06" later or the SIL2 Process CPU is used.

*3 Detailed information 2 is stored only when detailed information 1 is CPU module (3E00H).

*4 The information may not be displayed depending on the target device or label.

Appendix 3 Troubleshooting by Symptom

If any function of the CPU module does not operate as designed, perform troubleshooting by checking the following items. If the ERROR LED or USER LED is on or flashing, eliminate the error cause using the engineering tool.

When the POWER LED of the power supply module turns off

For troubleshooting, refer to the following:

MELSEC iQ-R Module Configuration Manual

When the READY LED of the CPU module turns off

Check the following:

Check item	Action
Check if the CPU module is mounted on the main base unit properly.	Remove the CPU module from the main base unit, and mount it back on the main base unit.
Check if the READY LED of another module is on.	If the READY LED of another module is on, the CPU module detected a major error. Replace the CPU module.
Check if the READY LED turns on when the power supply module is replaced and the power is restored to the system. (Check the LED status after the power supply module on the extension base unit is also replaced.)	If the READY LED turns on, an error has occurred in the power supply module(s) before the replacement. Replace the power supply module.
Check if the READY LED does not turn on even after the power supply module is replaced and the power is restored to the system. (Check the LED status after the power supply module on the extension base unit is also replaced.)	If the READY LED does not turn on, an error has occurred in a module other than the power supply module. Repeatedly supply power to the system, returning the modules back to the system one by one. The last module mounted immediately before the READY LED turned off has been failed. Replace the module.
Check if the system was powered on immediately after power-off.	Wait for five seconds or longer after power-off and power on the system again.

If the READY LED of the CPU module does not turn on even after the items above are checked and the actions are taken, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.

When an error has occurred in a redundant function module

When the RUN LED turns off

When the RUN LED turns off after the redundant function module is powered on, check the following.

Check item	Action
Check if the redundant function module has been properly mounted.	If not, properly mount the module on the base unit.

If the RUN LED of the redundant function module still does not turn on even after the above action is taken, the possible cause is a hardware failure of the module. Please consult your local Mitsubishi representative.

When the ERR LED turns on or flashes

Check the following:

Check item	Action
Check if an error has occurred by performing the module diagnostics.	Take actions proposed by the module diagnostics.

If the ERR LED still turns on or flashes even after the above action is taken, perform a module communication test and check the module for a hardware failure. (☞ Page 555 Redundant Function Module Communication Test)

When the L ERR LED turns on

Check the following:

Check item	Action
Check if the tracking cables used are proper.	• Check if the tracking cables satisfy the standards. (📖 MELSEC iQ-R CPU Module User's Manual (Startup)) • Check that the length of each tracking cable is within the range of the specifications. (📖 MELSEC iQ-R CPU Module User's Manual (Startup)) • Check if no disconnection is found on each tracking cable.
Check if each tracking cable has been properly connected.	Check the connection status of each tracking cable by performing the module diagnostics. If a tracking cable has not been properly connected, connect it again.

If the L ERR LED still turns on even after the above actions are taken, perform a module communication test and check the redundant function module and tracking cables for failures. (☞ Page 555 Redundant Function Module Communication Test)

When an error has occurred in a SIL2 function module

When the READY LED turns off

When the READY LED of the SIL2 function module turns off, check the following.

Check item	Action
Check if the SIL2 function module has been properly mounted.	If not, properly mount the module on the base unit.
Check if the system was powered on immediately after power-off.	Wait for five seconds or longer after power-off and power on the system again.

If the READY LED of the SIL2 function module still does not turn on even after the above actions are taken, the possible cause is a hardware failure of the module. Please consult your local Mitsubishi representative.

When an error has occurred in a safety function module

When the READY LED turns off

When the READY LED of the safety function module turns off, check the following.

Check item	Action
Check if the safety function module has been properly mounted.	If not, properly mount the module on the base unit.
Check if the system was powered on immediately after power-off.	Wait for five seconds or longer after power-off and power on the system again.

If the READY LED of the safety function module still does not turn on even after the above actions are taken, the possible cause is a hardware failure of the module. Please consult your local Mitsubishi representative.

When the specific extension base unit cannot be recognized

For troubleshooting, refer to the following:

MELSEC iQ-R Module Configuration Manual

When the specific Q series extension base unit cannot be recognized

For troubleshooting, refer to the following:

MELSEC iQ-R Module Configuration Manual

When an error has occurred in an extension base unit for a redundant system

For troubleshooting, refer to the following:

MELSEC iQ-R Module Configuration Manual

When data cannot be written to the programmable controller

Check the following:

Check item	Action
Check if password is not registered.	Authenticate a password using the engineering tool.
Check if the SD memory card is write-protected while the SD memory card is being targeted for writing.	Clear the write-protect for the SD memory card.
When the target of the write is an SD memory card, check that the SD memory card has been formatted.	Carry out formatting for the SD memory card.
Check if the data to be written exceeds the capacity of CPU built-in memory or SD memory card.	Check the CPU built-in memory or SD memory card for their free space.
Check if a password has been set for the programmable controller.	Write user information to the programmable controller from the engineering tool.
Check if the user has logged on to the programmable controller.	Log on to the programmable controller from the engineering tool.

If data cannot be written to the programmable controller even after the items above are checked and the actions are taken, the possible cause is a hardware failure. Please consult your local Mitsubishi representative.

When data cannot be read from the programmable controller

Check the following:

Check item	Action
Check if password is not registered.	Authenticate a password using the engineering tool.
Check if the target memory to be read is correctly specified.	Check the read target memory (CPU built-in memory, SD memory card, or intelligent function module).
Check if a password has been set for the programmable controller.	Write user information to the programmable controller from the engineering tool.
Check if the user has logged on to the programmable controller.	Log on to the programmable controller from the engineering tool.
Check if program restoration information has not been written.	When program restoration information has not been written, data cannot be read from the programmable controller. When writing data to the programmable controller, write program restoration information.

A

If data cannot be read from the programmable controller even after the items above are checked and the actions are taken, the possible cause is a hardware failure. Please consult your local Mitsubishi representative.

When the operating status of the CPU module cannot be changed

Check the following:

Check item	Action
Check if a stop error has been occurred.	Check the error cause using an engineering tool and eliminate the error cause.
Check if the online change processing is left suspended.	Execute the online change again.

If the CPU module operating status cannot be changed even after the items above are checked and the actions are taken, the possible cause is a hardware failure. Please consult your local Mitsubishi representative.

When the Ethernet function cannot be used

For troubleshooting, refer to the following:

MELSEC iQ-R Ethernet User's Manual (Application)

When the CC-Link IE Field Network Basic function cannot be used

For troubleshooting, refer to the following:

CC-Link IE Field Network Basic Reference Manual

When cyclic data is turned off at the time of system switching

When cyclic data is turned off or temporarily turned off at the time of system switching, check the following.

Check item	Action
Check if the own station send range is the tracking range.	Check that the cyclic data in the send range is set to be transferred. (☞ Page 520 Tracking transfer setting)
Check if the interlock used in the program has no errors.	Take one of the following measures. ■Modify the program so that the interlock is triggered using the following labels. (📖 MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)) • 'Own station data link error status' (SB0049) • 'Own station master/sub master function operating status' (SB004E) • 'Each station redundant system data link error status' (SB01B0) • 'Redundant system master station data link error status' (SB01B1) • 'Redundant system station number 0 data link error status' (SB01BF) • 'Each station redundant system data link status' (SW01B0 to SW01B7) ■Check that "Setting to Wait Receiving Cyclic Data after Switching System" of the CPU parameter is enabled. If the setting is disabled, enable it. (☞ Page 554 Setting to wait cyclic data receive after system switching, ☞ Page 754 Setting to wait cyclic data receive after system switching).

When the CPU module database access function cannot be used

When the data source name cannot be set

Check the following:

Check item	Action
Check if use prohibited characters are used in "Data Source Name".	Set "Data Source Name" without using prohibited characters.
Check if the number of use prohibited characters in "Data Source Name" exceeds the range of use.	Set "Data Source Name" with the number of characters within the range of use.

When the connection test fails

When the connection test fails at the data source setting, check the following:

Check item	Action
Check if the CPU module supports the CPU module database access function.	Check that the CPU module supports the CPU module database access function. (☞ Page 1139 Added and Enhanced Functions)
Check if the IP address of the CPU module set in "Server" is correct.	Set the correct IP address of the CPU module in "Server". (☞ Page 281 Data Source Name Setup)
Check if 61461 is set in "Port".	Set 61461 in "Port". (☞ Page 281 Data Source Name Setup)
Check if the absolute path of the database of the CPU module set in "Database" is correct.	Set the correct absolute path of the database in "Database". (☞ Page 281 Data Source Name Setup)

When the ODBC server of the CPU module cannot be connected

When the ODBC server of the CPU module cannot be connected from the personal computer, check the following:

Check item	Action
Check if "invalid login/password" is displayed in the error message on the personal computer.	Check that the login name and password that are set in the module parameter of the engineering tool match those that are input to connect to the server. (☞ Page 278 Built-in database access setting)
Check if "cannot open connection to server; open init timeout" is displayed in the error message on the personal computer.	Check that the correct data source is selected to connect to the database of the CPU module.
Check if "Communication link failure" is displayed in the error message on the personal computer.	The maximum number of concurrently connectable databases (number of connections) is used (☞ Page 267 Database specifications) End unused applications connected with the database of the CPU module and connect with the database again.

When the connection is disconnected

When the connection of the personal computer and the CPU module to the ODBC server is disconnected, check the following:

Check item	Action
Check if the built-in database access timer setting in the module parameter of the engineering tool is too short.	Set the longer time for the built-in database access timer setting. (☞ Page 278 Built-in database access setting)

A

Appendix 4 List of Special Relay Areas

The following table lists items in the list.

Item	Description
No.	Special relay number
Name	Special relay name
Data stored	Data stored in the special relay and its meaning
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (program, engineering tool, GOT, or other testing operations from external device) • U/S: User and system <Set timing> • Every END: Data is set every time END processing is performed. • Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN). • Status change: Data is set when the status is changed. • Error: Data is set when an error occurs. • Instruction execution: Data is set when an instruction is executed. • Request: Data is set when requested by a user (using the special relay). • Writing: Data is set when a user performs a writing operation. • During END: Data is set when END processing is performed. • Power-on to RUN or STOP to RUN: Data is set when the operating status changes from power-on to RUN or from STOP to RUN. • System switching: Data is set when two systems are switched (between the control system and the standby system)
CPU	The following shows the supported CPU modules. Each of the CPU module is represented by the following symbols. • Rn: Programmable controller CPU • RnP: Process CPU • RnPSF: SIL2 Process CPU • RnSF: Safety CPU • ALL: All the above CPU modules

Point

Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

Diagnostic information

The following is the list of special rely areas relating to the diagnostic information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM0	Latest self-diagnostic error (including annunciator ON)	Off: No error On: Error	• This relay turns on when the self-diagnostics returns an error (including the case when an error is detected by turning the annunciator ON). • The ON state is maintained even after the error has been later cleared.	S (Error)	ALL
SM1	Latest self-diagnostic error (not including annunciator ON)	Off: No error On: Error	• This relay turns on when the self-diagnostics returns an error (not including the case when an error is detected by turning the annunciator ON). • The ON state is maintained even after the error has been later cleared. • This relay does not turn on when errors that are notified by the PALERT instruction or the PABORT instruction.	S (Error)	ALL
SM50	Error reset	Off→On:Error reset request On→Off:Error reset complete	• This relay clears the error state when the mode transfers from off to on. • This relay switches from on to off when the error reset has been completed.	U/S (Status change)	ALL
SM51	Battery low latch	Off: Normal On: Battery low	• This relay switches to on when the battery voltage of the CPU module drops below the specified value. • The ON state is maintained even after the battery voltage has been later recovered to the normal value. • This relay synchronizes with BAT LED.	S (Error)	ALL
SM52	Battery low	Off: Normal On: Battery low	This relay has the same function as SM51 except for switching to off after the battery voltage has been recovered to a normal value.	S (Error)	ALL
SM53	AC/DC DOWN	Off: No AC/DC DOWN detection On: AC/DC DOWN is detected	• This relay switches to on when a momentary power failure within 20ms is detected while the AC power supply module is in use. This relay can be reset when power is turned off and on. (In a redundant system with redundant extension base unit, if a momentary power failure occurs in a module on an extension base unit, the power failure will be detected in both systems.) • This relay switches to on when a momentary power failure within 10ms is detected while the DC power supply module is in use. This relay can be reset when power is turned off and on. (In a redundant system with redundant extension base unit, if a momentary power failure occurs in a module on an extension base unit, the power failure will be detected in both systems.)	S (Error)	ALL
SM56	Instruction execution fault	Off: Normal On: Instruction execution fault state	• This relay switches to on when an error which can be classified as instruction execution fault is returned. • The ON state is maintained even after the error has been later cleared.	S (Error)	ALL

A

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM60	Fuse Blown	Off: Normal On: Fuse blown is detected	• This relay is on when at least one output module is in fuse blown state and the ON state is maintained even after later recovering to the normal state. • The fuse blown state check is also done for output modules on the remote I/O station. • In a redundant system with redundant extension base unit, this relay is set as follows: · If a fuse blown occurs in an output module on an extension base unit, the on/off state is stored only in the CPU module of the control system, but not stored in the CPU module of the standby system. · When the systems are switched, the state before system switching is held. · When the error is cleared, the CPU module of the system where the error is cleared clears the value.	S (Error)	ALL
SM61	I/O module verification error	Off: Normal On: Error	• This relay switches to on when the state of the I/O module is different from one registered during power-on, and the ON state is maintained even after later recovering to the normal state. • I/O module verification is also done for modules on the remote I/O station. • In a redundant system with redundant extension base unit, this relay is set as follows: · If an I/O verification error occurs in a module on an extension base unit, the on/off state is stored only in the CPU module of the control system, but not stored in the CPU module of the standby system. · When the systems are switched, the state before system switching is held. · When the error is cleared, the CPU module of the system where the error is cleared clears the value.	S (Error)	ALL
SM62	Annunciator	Off: Not detected On: Detected	• This relay switches to on when at least one annunciator is turned on. • This relay returns to off when all the annunciators are turned off.	S (Instruction execution)	ALL
SM80	Detailed information 1: Flag in use	Off: Not used On: In use	This relay switches to on if the detailed information n exists when SM0 switched to on.	S (Status change)	ALL
SM112	Detailed information 2: Flag in use				ALL
SM150	Power-off/power supply voltage drop detection	Off: Power-on/normal power supply voltage On: Power-off/voltage drop detected/power supply module not mounted	• This relay turns on when one or more of the power supply modules whose power has been shut off or power supply voltage has dropped (not including a momentary power failure), or one or more of empty slots for the power supply module are detected on the redundant power supply base unit or a redundant extension base unit. • This relay turns on if causes to turn on any bits of SD150 have been occurred. • This relay turns off if causes to turn on any bits of SD150 have been removed. • In a multiple CPU system, the flags are stored only to the CPU No.1. • In a redundant system with redundant extension base unit, if power-off or power supply voltage drop is detected in a power supply module on an extension base unit, the on/off state is stored only in the CPU module of the control system, but not stored in the CPU module of the standby system.	S (Status change)	Rn*1 RnP RnPSF

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM151	Power supply module failure detection	Off: Not detected/power-off/no power supply module On: Detected	• This relay turns on when one or more power supply module failures have been detected on the redundant power supply base unit or a redundant extension base unit. • This relay turns on if causes to turn on any bits of SD151 have been occurred. • This relay turns off if causes to turn on any bits of SD151 have been removed. • In a multiple CPU system, the flags are stored only to the CPU No.1. • In a redundant system with redundant extension base unit, if a failure of a power supply module on an extension base unit, the on/off state is stored only in the CPU module of the control system, but not stored in the CPU module of the standby system.	S (Status change)	Rn*1 RnP RnPSF
SM152	Momentary power failure detection (power supply module 1)	Off: Not detected On: Detected	• This relay turns on when a momentary power failure of the input power supply to the power supply 1 or 2 is detected one or more times. After turning on, this relay remains on even if the power supply recovers from the momentary power failure. • This register monitors the status of the power supply module mounted on the main base unit and counts the number of momentary power failures. • This relay turns off the flags (SM152 and SM153) of the power supply 1 and 2 when the CPU module starts up. • When one of the two power supply modules is powered off, this relay turns off the corresponding flag to one powered off. • In a multiple CPU system, the flags are stored only to the CPU No.1.	S (Status change)	Rn*1 RnP RnPSF
SM153	Momentary power failure detection (power supply module 2)			S (Status change)	Rn*1 RnP RnPSF
SM154	Invalid power supply module	Off: Valid/power-off/no power supply module On: Invalid	• This relay turns on when one or more invalid power supply modules is detected on the redundant power supply base unit or a redundant extension base unit. • This relay turns on if a factor to turn on any bits of SD154 have been occurred. • This relay turns off if all the factors to turn on any bits of SD154 have been removed. • In a multiple CPU system, the flags are stored only to the CPU No.1. • In a redundant system with redundant extension base unit, if an invalid power supply module is detected on an extension base unit, the on/off state is stored only in the CPU module of the control system, but not stored in the CPU module of the standby system.	S (Status change)	Rn*1 RnP RnPSF

*1 There are restrictions on the firmware version of the supported CPU module. (☞ Page 1139 Added and Enhanced Functions)

A

System information

The following is the list of special relay areas relating to the system information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM203	STOP contact	Off: Other than STOP state On: STOP state	This relay is on in STOP state.	S (Status change)	ALL
SM204	PAUSE contact	Off: Other than PAUSE state On: PAUSE state	This relay is on in PAUSE state. Note that this relay is on during the END processing of the scan which the specified PAUSE contact turns on if PAUSE state is generated at the PAUSE contact.	S (Status change)	ALL
SM210	Clock data set request	Off→On:setting request is detected On→Off:setting is completed	• Clock data stored in SD210 to SD216 is written into the CPU module when this relay is switched from off to on. • This relay switches from on to off when writing of clock data stored in SD210 to SD216 into the clock element is completed.	U/S (Status change)	ALL
SM211	Clock data set error	Off: No error On: Error	This relay switches to on when an error is generated in values from SD210 to SD216, and to off when no error is generated.	S (Request)	ALL
SM213	Clock data read request	Off: Non-processing On: Reading request	Clock data is loaded into SD210 to SD216 when this relay is in the ON state.	U	ALL
SM217	Daylight saving time status flag	Off: Not during daylight saving time On: During daylight saving time	Turns on if during daylight saving time with daylight saving time function. Turns off if not during daylight saving time.	S (Status change)	Rn*1 RnSF*1
SM220	CPU No.1 preparation completed	Off: Not completed On: Completed	• This relay switches to on at the time when access from the CPU module on other CPUs to the CPU module for CPU No. n is enabled during power-on or resetting. • This relay is used as an interlock to access the CPU module for the CPU No. n when the multiple CPU synchronization setting is configured to asynchronous mode.	S (Status change)	ALL
SM221	CPU No.2 preparation completed				ALL
SM222	CPU No.3 preparation completed				ALL
SM223	CPU No.4 preparation completed				ALL
SM230	No.1 CPU error flag	Off: CPU No.n normal On: CPU No.n stop error state	• This relay is off when the CPU module for the CPU No.n is normal (including a continuation error period). • This relay is on when the CPU module for the CPU No.n is in stop error state.	S (Status change)	ALL
SM231	No.2 CPU error flag				ALL
SM232	No.3 CPU error flag				ALL
SM233	No.4 CPU error flag				ALL

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM240	No.1 CPU reset flag	Off: CPU No.n not being reset On: CPU No.n in reset mode	• This relay switches to off when the CPU module of the CPU No.1 is not being reset. • This relay is on while the CPU module of the CPU No.1 is being reset (including the case when the CPU module is removed from the base unit). Other CPUs also enter into reset mode.	S (Status change)	ALL
SM241	No.2 CPU reset flag		• This relay switches to off when the CPU module of the CPU No.2 is not being reset. • This relay is on while the CPU module of the CPU No.2 is being reset (including the case when the CPU module is removed from the base unit). Errors occur in the other CPU modules.	S (Status change)	ALL
SM242	No.3 CPU reset flag		• This relay switches to off when the CPU module of the CPU No.3 is not being reset. • This relay is on while the CPU module of the CPU No.3 is being reset (including the case when the CPU module is removed from the base unit). Errors occur in the other CPU modules.	S (Status change)	ALL
SM243	No.4 CPU reset flag		• This relay switches to off when the CPU module of the CPU No.4 is not being reset. • This relay is on while the CPU module of the CPU No.4 is being reset (including the case when the CPU module is removed from the base unit). Errors occur in the other CPU modules.	S (Status change)	ALL
SM315	Service processing constant wait setting flag	Off: Do not wait for service processing. On: Wait for service processing.	• This relay is turned on when the CPU module is required to accept the service processing requests until the time or rate specified in "Device/Label Access Service Processing Setting" in "CPU parameter" elapses. (The scan time will increase according to the specified time or rate. When "Specifying Method" is set to "Set Processing Counts" or "Execute END Processing between Programs", the CPU module does not wait for the device/label access service processing regardless of the on/off state of this relay.) • This relay is turned off when the CPU module is not required to wait for the device/label access service processing in END processing when there are no requests. (Default: Off) • The device/label access service processing constant wait function is not enabled unless "AFFFH" is stored in SD315 (Service processing constant wait status setting) while SM315 is on, and the CPU module does not wait for the device/ label access service processing.	U (Request)	RnP*1
SM384	System operation setting request	Off: Request accepted On: Request submitted	• This relay is turned from off to on to send a setting request and write/delete data to/ in the setting storage area (system memory) with the value set in SD384. • Regardless of whether it was successful or not, this relay turns off when the setting request is accepted.	U/S (Status change)	Rn*1 RnP*1 RnSF*1*3
SM385	System operation setting error	Off: No error On: Error	• This relay turns on when writing into the setting storage area (system area) fails.	S (Status change)	Rn*1 RnP*1 RnSF*1*3

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM386	Program restoration information write status LED control setting mode	Off: LED flashing On: Without LED flashing	• This relay indicates the LED control setting status of program restoration information.	S (Initial)	Rn[*1] RnP[*1] RnSF[*1*3]
SM387	Program restoration information write status	Off: All written On: Not all written	• This relay indicates the write status of program restoration information in the CPU module. • This relay turns off when all program restoration information is written. • This relay turns on if there is any program whose program restoration information is not in the CPU module.	S (Status change)	Rn[*1] RnP[*1] RnSF[*1*3]
SM388	File batch online change operation setting status	Off: Program file only On: Program file/FB file/ global label setting file	This relay indicates the operating status of the file batch online change.	S (Status change)	Rn[*2] RnP[*1] RnSF[*1]

*1 There are restrictions on the firmware version of the supported CPU module and software version of the engineering tool. (☞ Page 1139 Added and Enhanced Functions)

*2 There are restrictions on the firmware version of the supported CPU module and software version of the engineering tool. (☞ Page 1139 Added and Enhanced Functions)
For the R00CPU, R01CPU, and R02CPU, the operation setting status is always set to "program file/FB file/global label setting file" regardless of whether this relay is on or off.

*3 Only the standard program is supported.

SFC information

The following is a list of special relay areas relating to SFC information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM320	Presence/absence of SFC program	Off: No SFC program On: SFC program	• This relay switches to on if an SFC program has been registered, and switches to off if it has not.	S (Initial)	Rn*1 RnP*1 RnSF*1
SM321	Start/stop SFC program	Off: SFC program not executed (stop) On: SFC program executed (start)	• The same value as SM320 is set for the default value. (This relay automatically turns on if there is an SFC program.) • SFC program execution is stopped if this relay switches from on to off, and execution is restarted when it switches from off to on. • If this relay switches off before SFC program processing, execution of the SFC program is not started.	S (Initial)/U	Rn*1 RnP*1 RnSF*1
SM322	SFC program start status	Off: Initial start On: Resumption	• If the SFC program start mode in the parameters is off for initial start, off is set for the default, and if resumption, on is set.	S (Initial)/U	Rn*1 RnP*1 RnSF*1
SM323	Presence/absence of continuous transition for entire block	Off: No continuous transition On: Continuous transition	• Sets whether there is a continuous transition for blocks for which no SFC information device continuous transition bit has been set. • No continuous transition when off. Continuous transition when on. • There is no effect on operation for blocks for which the continuous transition bit has been set.	U	Rn*1 RnP*1 RnSF*1
SM324	Continuous transition prevention flag	Off: When transition executed On: When there is no transition	• This relay switches to off during operation in modes with continuous transition, or during continuous transition, and switches to on when it is not a continuous transition. • The relay is always on during operation in modes with no continuous transition.	S (Status change)	Rn*1 RnP*1 RnSF*1
SM325	Output mode at block stop	Off: Off On: Hold	• Selects whether to retain coil output for active steps during block stoppages. • The default value when the output mode for block stoppages in the parameters is coil output off is off, and on when coil output is retained. • All coil outputs are turned off when off. Coil outputs are retained when on.	S (Initial)/U	Rn*1 RnP*1 RnSF*1
SM326	SFC device/label clear mode	Off: Device/label clear On: Device/label retain	• Select the device status (all devices and labels (including latch labels) excluding step relay (S)) when the CPU module status changes from STOP → program write → RUN. • This relay is valid only when an SFC program exists after program writing. • This relay is valid not only when an SFC program is written, but also when the program file and the parameter file are written. • When the Process CPU is used, only labels are cleared at the program writing to the programmable controller after data are rebuilt even though this relay is on.	U	Rn*1 RnP*1 RnSF*1

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM327	Output mode at execution of the end step	Off: Hold step output off On: Hold step output retained	• When this relay switches to off, coil output is turned off for steps (SC, SE, ST) for which transition is established and that are on hold reach the END step. • When this relay switches to on, coil output is retained (step becomes inactive) for steps (SC, SE, ST) for which transition is established and that are on hold reach the END step. However, coils are turned off following forced termination.	U	Rn*1 RnP*1 RnSF*1
SM328	Clear processing mode when the sequence reaches the end step	Off: Clear processing performed On: Clear processing not performed	• Selects whether to perform clear processing if an active step other than those retained in the block exists when the END step is reached. When this relay switches to off, all active steps are forcibly terminated, and the block is terminated. When this relay switches to on, block execution is continued as is. When the END step is reached, if no active steps other than those retained exist, all retained steps are terminated, and the block is terminated.	U	Rn*1 RnP*1 RnSF*1
SM329	Online change (SFC block) status flag	Off: Not being executed On: Being executed	This relay turns on while the online change (SFC block) is being executed.	S (Status change)	Rn*1 RnP*1 RnSF*1

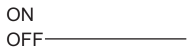
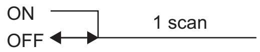
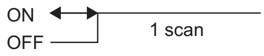
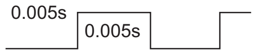
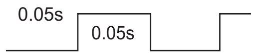
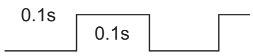
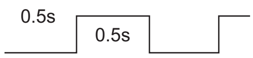
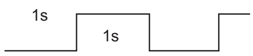
*1 There are restrictions on the firmware version of the supported CPU module and software version of the engineering tool. (☞ Page 1139 Added and Enhanced Functions)

Point

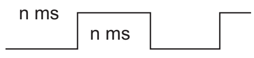
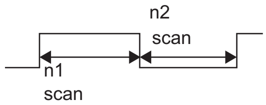
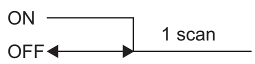
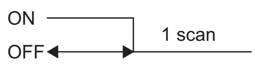
For details on the SFC program, refer to the following.
MELSEC iQ-R Programming Manual (Program Design)

System clock

The following is the list of special relay areas relating to the system clock.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM400	Always On		Always On	S (Power-on to RUN/ STOP to RUN/every END[*3])	ALL
SM401	Always Off		Always Off	S (Power-on to RUN/ STOP to RUN/every END[*3])	ALL
SM402	After RUN, ON for 1 scan only		• This relay is on during only one scan after RUN mode starts. • This relay is enabled only for the scan execution type program.	S (Status change/ every END[*3])	ALL
SM403	After RUN, OFF for 1 scan only		• This relay is off during only one scan after RUN mode starts. • This relay is enabled only for the scan execution type program.	S (Status change/ every END[*3])	ALL
SM409	0.01 second clock		• This relay repeats on/off at 5ms intervals. • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)	ALL
SM410	0.1 second clock		• This relay repeats on/off at certain intervals. • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)	ALL
SM411	0.2 second clock		• This relay repeats on/off at certain intervals. • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)	ALL
SM412	1 second clock		• This relay repeats on/off at certain intervals. • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)	ALL
SM413	2 second clock		• This relay repeats on/off at certain intervals. • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)	ALL
SM414	2n second clock		• This relay repeats on/off at regular intervals specified in SD414 (in units of seconds). (When the value in SD414 is changed, the elapsed time count that has started when the ON/OFF state of SM414 last changed continues, and the ON/OFF state changes when the new specified time is elapsed.[*1]) • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)	ALL

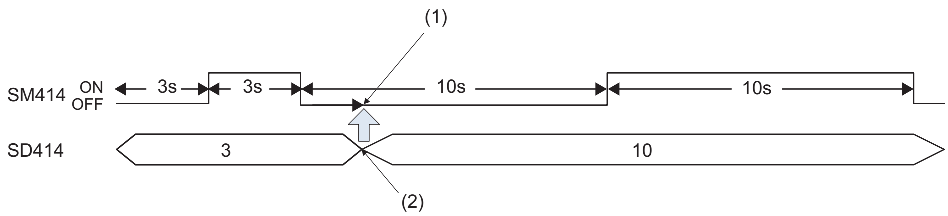
A

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM415	2n millisecond clock		• This relay repeats on/off at regular intervals specified in units of milliseconds on the SD415. (When the value in SD415 is changed, the elapsed time count that has started when the ON/OFF state of SM415 last changed continues, and the ON/OFF state changes when the new specified time is elapsed.[*1]) • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)	ALL
SM420	User timing clock No.0		• This relay repeats on/off at specified scan intervals. • The initial state when the CPU module is powered on or reset is off. • The on/off scan interval is set using the DUTY instruction. (n1: ON scan interval, n2: OFF scan interval)	S (Every END)	ALL
SM421	User timing clock No.1				ALL
SM422	User timing clock No.2				ALL
SM423	User timing clock No.3				ALL
SM424	User timing clock No.4				ALL
SM440	On only initial I44 execution after RUN		• This relay is on during the first execution of the inter-module synchronous interrupt program (I44) after RUN mode starts and off during the second execution onwards. • Note that the relay status while the DI instruction is executed is as follows: On during the first execution of the inter-module synchronous interrupt program (I44) after the DI instruction is cleared and Off during the second execution onwards. (no change after the second DI clear). • This contact is enabled only for the inter-module synchronous interrupt program.	S (Status change)	Rn RnP RnSF[*2]
SM441	On only initial I45 execution after RUN		• This relay is on during the first execution of the multiple CPU synchronous interrupt program (I45) after RUN mode starts and off during the second execution onwards. • Note that the relay status while the DI instruction is executed is as follows: On during the first execution of the multiple CPU synchronous interrupt program (I45) after the DI instruction is cleared and Off during the second execution onwards. (no change after the second DI clear). • This contact is enabled only for the multiple CPU synchronous interrupt program.	S (Status change)	Rn RnP RnSF[*2]
SM1184	System clock (SM400 to SM403) every end update setting request	Off: Request accepted Off→On:Request submitted	• This relay is turned off and on when a setting request of the value stored in SD1184 (System clock (SM400 to SM403) every end update setting) is submitted. • Regardless of whether the setting is successful or not, this relay turns off when the setting request is accepted. • The setting takes effect when the CPU module is powered off and on.	U (Status change)	Rn[*4]
SM1185	System clock (SM400 to SM403) every end update setting error	Off: No error On: Error	This relay turns on when writing of the setting has failed.	S (Status change)	Rn[*4]
SM1186	System clock (SM400 to SM403) every end update setting status	Off: Disabled On: Enabled	This relay indicates the system clock (SM400 to SM403) every end update status.	S (Initial)	Rn[*4]

*1 The following figures show operation examples of SM414/SM415 when a value in SD414/SD415 is changed.

Ex.

When a value in SD414 is changed from 3 to 10:

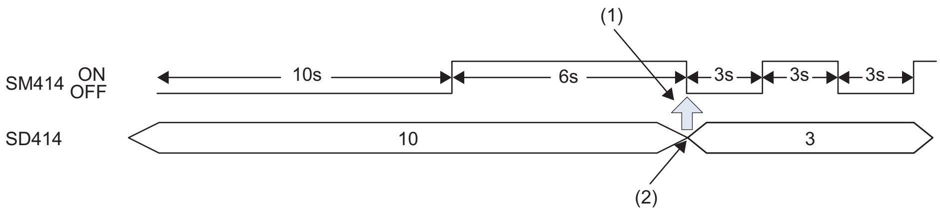


(1) The elapsed time after the ON/OFF state of SM414 changes remains.

(2) Value change

Ex.

When a value in SD414 is changed from 10 to 3:



(1) If the new interval in SD414 has already elapsed after the last change of the ON/OFF state of SM414, the ON/OFF state changes as soon as a value in SD414 is changed.

(2) Value change

*2 There are restrictions on the firmware version of the supported CPU module and software version of the engineering tool. (☞ Page 1139 Added and Enhanced Functions)

*3 To enable the setting timing of "Every END" for these special relay areas, set both SM1184 and SD1184 (System clock (SM400 to SM403) every end update setting).

*4 The programmable controller CPUs with the following firmware versions supports these special relay areas.

· R00CPU, R01CPU, R02CPU: "24" or later

· Programmable controller CPUs other than the R00CPU, R01CPU, R02CPU: "57" or later.

A

Fixed scan function information

The following is the list of special relay areas relating to the fixed scan function information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM480	Cycle error flag for inter-module synchronous interrupt program (I44)	Off: No error for the inter-module synchronization program (Normal) On: Error state for the inter-module synchronization program	• This relay switches to on when the inter-module synchronous interrupt program (I44) has not been completed within the specified inter-module synchronization cycle or the program cannot be executed due to various reasons, such as execution of a higher-priority interrupt program and interrupt disabling by the instruction execution. • The ON state is maintained even after the program is later completed within the specified inter-module synchronization cycle (Clear by turning power off and on or resetting).	S (Status change)	Rn RnP RnSF*1
SM481	Cycle error flag for multiple CPU synchronization interrupt program (I45)	Off: No error for the multiple CPU synchronization program (Normal) On: Error state for the multiple CPU synchronization program	• This relay switches to on when the multiple CPU synchronization program (I45) has not been completed within the specified fixed scan communication cycle or the program cannot be executed due to various reasons, such as execution of a higher-priority interrupt program and interrupt disabling by the instruction execution. • The ON state is maintained even after the program is later completed within the specified fixed scan communication cycle (Clear by turning power off and on or resetting).	S (Status change)	Rn*1 RnP RnSF*1
SM484	Execution section excess error flag for multiple CPU synchronization interrupt program	Off: No execution section excess error for the multiple CPU synchronization interrupt program (Normal) On: Execution section excess error state for the multiple CPU synchronization interrupt program	• This relay switches to on when the program is executed exceeding the program execution section within the specified multiple CPU synchronization cycle. • The ON state is maintained even after the program is later completed within the multiple CPU synchronous interrupt program execution section (cleared by turning power off and on or resetting).	S (Status change)	Rn*1 RnP RnSF*1
SM488	Inter-module synchronization error (out of synchronization was detected on the CPU module)	Off: No error (Normal) On: Error	• This relay switches to on when the inter-module synchronization signal cannot be identified within the inter-module synchronization cycle specified in the parameter or more than one signal has been identified within the same inter-module synchronization cycle. • The ON state is maintained even after the inter-module synchronization signal can be later identified within the specified inter-module synchronization cycle (cleared by turning power off and on or resetting).	S (Status change)	Rn RnP RnSF*1
SM522	Scan time clear request	Off: Do not clear the scan time. On: Clear the scan time.	• This relay clears the maximum/minimum scan time. • Turning off and on this relay clears the scan time. • This relay automatically turns off when the scan time has been cleared.	U/S (Status change)	Rn*1 RnSF*1

*1 There are restrictions on the firmware version of the supported CPU module and software version of the engineering tool. (☞ Page 1139 Added and Enhanced Functions)

Drive information

The following is the list of special relay areas relating to the drive information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM600	Memory card usable flags	Off: Disabled On: Enabled	This relay is on when an SD memory card is enabled (This relay switches to on when a valid SD memory card is inserted and prepared for use).	S (Status change)	ALL[*2]
SM601	Memory card protect flag	Off: Not protected On: Protected	This relay is on when the write protect switch of the SD memory card is set to on.	S (Status change)	ALL[*2]
SM603	Memory card (drive 2) flag	Off: No SD memory card inserted On: SD memory card inserted	This relay is on when an SD memory card is inserted. (This relay switches to on when an SD memory card is inserted regardless of its state (enabled/disabled) and type).	S (Status change)	ALL[*2]
SM604	Memory card in-use flag	Off: Not used On: In use	This relay is on when the SD memory card is being used.	S (Status change)	ALL[*2]
SM605	Memory card remove/insert prohibit flag	Off: Allowed to remove/insert On: Not allowed to remove/insert	Turn on this relay to prohibit the SD memory card from being removed/inserted. When this relay is on, if SM607 is set to on, the system switches it to off.	U/S	ALL[*2]
SM606	SD memory card forced disable instruction	Off: Cancel instruction On: Forced disable instruction	• This relay can be switched to on to issue the instruction that forces the SD memory card to be disabled. However, whenever any function accesses the SD memory card, the disabling process does not start until the access is completed. • This relay can be switched to off to cancel the instruction that forces the SD memory card to be disabled.	U/S (Status change)	ALL[*2]
SM607	SD memory card forced disable status flag	Off: Not disabled On: Disabled	• This relay switches to on when the SD memory card is disabled by switching SM606 to on. • This relay switches to off when the forcibly-disabled SD memory card is enabled by switching SM606 to off.	S (Status change)	ALL[*2]
SM624	Battery-less option cassette insertion flag	Off: Not inserted On: Inserted	This relay is on when the battery-less option cassette is inserted.	S (Status change)	Rn[*4]
SM625	Battery-less function execution flag	Off: Battery-less function not executed On: Battery-less function executed	This relay is on while the battery-less function is being executed.	S (Status change)	Rn[*4]
SM626	Extended SRAM cassette insertion flag	Off: Not inserted On: Inserted	This relay is on when the extended SRAM cassette is inserted.	S (Status change)	ALL[*3]
SM628	Program memory write error	Off: No write operation/ normal On: Write error	This relay switches to on when a write error is detected during write operation to the program memory. This relay switches to off when the write instruction is issued.	S (Writing)	ALL
SM629	Program memory write flag	Off: No write operation On: Executing write operation	This relay is on when the write process to the program memory is in progress. The relay is switched to off when the write process is completed.	S (Writing)	ALL
SM630	Program memory overwrite count error flag	Off: The number of rewrite operations is less than 100000 On: The number of rewrite operations reaches 100000	This relay switches to on when the number of program memory rewriting operations reaches 100000 (CPU module must be replaced).	S (Writing)	ALL
SM632	Data memory write error	Off: No write operation/ normal On: Write error	This relay switches to on when a write error is detected during write operation to the data memory. This relay switches to off when the write instruction is issued.	S (Writing)	ALL
SM633	Data memory write flag	Off: No write operation On: Executing write operation	This relay is on when the write process to the data memory is in progress. The relay is switched to off when the write process is completed.	S (Writing)	ALL

A

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM634	Number of rewriting operations error to data memory flag	Off: The number of rewrite operations is less than 100000 On: The number of rewrite operations reaches 100000	This relay switches to on when the number of data memory rewriting operations reaches 100000 (CPU module must be replaced).	S (Writing)	ALL
SM636	System memory write error	Off: No write operation/ normal On: Write error	This relay switches to on when a write error is detected during write operation to the system memory (Flash ROM)*1. This relay switches to off when the write instruction is issued.	S (Writing)	RnPSF RnSF
SM637	System memory write flag	Off: No write operation On: Executing write operation	This relay switches to on when the write process to the system memory (Flash ROM)*1 is in progress, and switches to off when the write process is complete.	S (Writing)	RnPSF RnSF
SM638	System memory rewrite count error flag	Off: The number of rewrite operations is less than 100000 On: The number of rewrite operations reaches 100000	This relay switches to on when the number of system memory (Flash ROM)*1 rewrite operations reaches 100000. (CPU module must be replaced).	S (Writing)	RnPSF RnSF
SM652	Function memory clear request	Off→On:Clear request exists On→Off:Function memory cleared	• The function memory of the CPU module is initialized (cleared) when this relay is turned off and on while the operating status of the CPU module is in STOP. • The system turns off this relay when the initialization (clear) of the function memory is completed (regardless of whether it was successful or not).	S (Status change)/U	Rn*4
SM653	File transfer to data memory request	Off→On:Transfer executed On→Off:Transfer completed	• Files saved in the function memory (SDRAM) are transferred to the data memory when this relay is changed from off to on. • The system turns off this relay when the file transfer is completed (regardless of whether it was successful or not). Note that the files in the function memory are not deleted.	S (Status change)/U	Rn*4

*1 This is the memory used by the system when the CPU module is executing functions.
*2 The CPU module where the SD memory card can be used supports these special relay areas.
*3 The CPU module where the extended SRAM cassette can be used supports these special relay areas.
*4 There are restrictions on the firmware version of the supported CPU module and software version of the engineering tool. (☞ Page 1139 Added and Enhanced Functions)

Instruction related

The following is the list of special relay areas relating to the instruction-related items.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM699	Dedicated instruction skip flag	Off: Instruction being executed or completed On: Instruction not executed	• This relay indicates whether the built-in Ethernet function instruction (the SP.SOCOPEN/SP.SOCCLOSE/ SP.SOCRCV/S.SOCRCVS/SP.SOCSND/ SP.ECPRTCL/SP.SLMPSND/ SP.FTPPUT/SP.FTPGET instruction), intelligent function module dedicated instruction, or multiple CPU dedicated instruction has been skipped. (Checking this flag immediately after the instruction issue allows to determine whether or not the dedicated instruction is non-processing (skipped) in the internal processing.) • This relay saves/returns while the interrupt program is being executed.	S (Status change)	ALL
SM700	Carry flag	Off: Carry off On: Carry on	• This relay is a carry flag used while the application instruction is executing. • This relay saves/returns while the interrupt program is being executed.	S (Instruction execution)	ALL
SM701	Number of output characters selection	Off: Outputs until reaching NULL code On: Outputs 16 characters	• ASCII codes are output until reaching NULL (00H) code, when SM701 is off. • 16 characters of ASCII codes are output, when SM701 is on. • This relay saves/returns while the interrupt program is being executed.	U	ALL
SM702	Search method	Off: Sequential search On: Dichotomizing search	• The search method in the search instruction can be specified by using this relay. • To use the dichotomizing search, data must be sorted. • This relay saves/returns while the interrupt program is being executed.	U	ALL
SM703	Data sort instruction sort order	Off: Ascending On: Descending	• This relay can be used to specify how to arrange data in the data sort instruction: ascending or descending. • This relay saves/returns while the interrupt program is being executed.	U	ALL
SM704	Block comparison	Off: Mismatch is detected On: Completely match	• This relay switches to on when all the data conditions are met in the block data comparison instruction. • This relay saves/returns while the interrupt program is being executed.	S (Instruction execution)	ALL
SM705	Number of conversion digits selection	Off: Fixed number of digits On: Desired number of digits	• When SM705 is off, the conversion data is specified and output with the fixed number of digits. • When SM705 is on, the conversion data is specified and output with the desired number of digits. • This relay saves/returns while the interrupt program is being executed.	U	Rn[*1]
SM709	DT/TM instruction improper data detection flag	Off: No improper data On: Improper data is detected	• This relay switches to on when the comparison target data is date data or cannot be recognized as clock data, or the comparison target device (three words) exceeds the specified device range in the DT/TM instruction. • This relay saves/returns while the interrupt program is being executed.	S (Instruction execution)/ U	ALL

A

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM752	Dedicated instruction End bit control flag	Off: Automatically controlled On: Not automatically controlled	This relay can be used to set whether the system automatically controls the on/off operation of End bit being used for the dedicated instruction and other (control of End bit which is registered in the End processing of the instruction completion and turned on for only the next one scan (Off: turned on for only one scan after the instruction is completed as well as normal/error End bit, On: turned on when the instruction is completed and the On status is maintained after that)).	U	ALL
SM753	File being accessed	Off: Not in progress On: In progress	• This relay is on during file access by the SP.FWRITE/SP.FREAD/SP.DEVST instruction. • This relay is on while the SP.FTPPUT/SP.FTPGET instruction is being executed. • This relay is on while a database access instruction is being executed.[*1] • This relay is on during access to the SD memory card and data memory.	S (Status change)	ALL
			• This relay is on while a file operation instruction (except the SP.FWRITE/SP.FREAD instruction) is being executed. • This relay is on during access to the SD memory card.		Rn[*3]
			• This relay is on while the SP.SIDRD instruction is being executed.		RnSF
SM754	BIN/DBIN instruction error control flag	Off: Executes error detection On: No execute error detection	This relay can be switched to on when the error detection is not desirable in the BIN/DBIN instruction.	U	ALL
SM755	Scaling data check settings	Off: Performs data check On: Not perform data check	This relay can be used to enable/disable the check whether the scaling data is sorted in ascending order when the SCL/DSCL/SCL2/DSCL2 instruction is being executed.	U	ALL
SM756	Module access completion wait control flag	Off: Not wait the completion On: Waits the completion	This relay can be used to determine whether the system must wait until the access is completed before starting the next instruction when the write access instruction to the buffer memory of another module is being executed.	U	ALL
SM772	CCD/CRC instruction conversion mode (16-bit/8-bit) selection	Off: 16-bit conversion mode On: 8-bit conversion mode	This relay can be used to select 16-bit conversion mode or 8-bit conversion mode with the CCD/CRC instruction.	U	Rn[*2]
SM773	SMOV instruction BCD conversion prohibit flag	Off: BCD conversion enabled On: BCD conversion disabled	This relay switches to on when disabling conversion from BIN data to BCD data with the SMOV instruction.	U	Rn[*2]
SM774	Data table sort instruction sort order	Off: Ascending On: Descending	This relay can be used to specify how to arrange data in the data table sort instruction: ascending or descending.	U	Rn[*3]
SM775	Selection of refresh processing during the COM instruction execution	Off: Executes all the refresh processings On: Executes the refresh processing specified in SD775	This relay can be used to select the refresh processing target when the COM instruction is executed from two options: all refresh processing or only refresh processing specified in SD775.	U	ALL
SM776	Local device setting at CALL	Off: Disables local devices On: Enables local devices	This relay can be used to determine whether the local devices on the subroutine program called when the CALL instruction is being executed are enabled or not.	U	ALL

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM777	Local device setting in interrupt programs	Off: Disables local devices On: Enables local devices	This relay can be used to determine whether the local devices are enabled or not when the interrupt program is being executed. However, for the SIL2 Process CPU and Safety CPU, local devices/safety local devices of a program file in the storage location are always used in the standard program/safety program, regardless of the setting of SM777.	U	ALL
SM792	PID bumpless processing (for the complete differentiation PIDCONT instruction)	Off: Matching On: Not matching	This relay can be used to specify whether to match SV to PV in manual mode.	U	ALL
SM794	PID bumpless processing (for the inexact differential S.PIDCONT instruction)	Off: Matching On: Not matching	This relay can be used to specify whether to match SV to PV in manual mode.	U	ALL
SM796	Number of used blocks information for the multiple CPU dedicated instruction (for CPU No.1)	Off: The specified number of blocks is reserved On: The number of blocks specified in SD796 is not reserved	• This relay switches to on when the number of remaining blocks in the dedicated instruction transfer area to be used in the multiple CPU dedicated instruction (target machine: CPU No.1) drops below the number of blocks specified in SD796. This relay is also on when the instruction is being executed. • This relay switches to off when there exist free blocks during the END processing.	S (Instruction execution/ During END)	Rn*2 RnP RnSF
SM797	Number of used blocks information for the multiple CPU dedicated instruction (for CPU No.2)	Off: The specified number of blocks is reserved On: The number of blocks specified in SD797 is not reserved	• This relay switches to on when the number of remaining blocks in the dedicated instruction transfer area to be used in the multiple CPU dedicated instruction (target machine: CPU No.2) drops below the number of blocks specified in SD797. This relay is also on when the instruction is being executed. • This relay switches to off when there exist free blocks during the END processing.	S (Instruction execution/ During END)	Rn*2 RnP RnSF
SM798	Number of used blocks information for the multiple CPU dedicated instruction (for CPU No.3)	Off: The specified number of blocks is reserved On: The number of blocks specified in SD798 is not reserved	• This relay switches to on when the number of remaining blocks in the dedicated instruction transfer area to be used in the multiple CPU dedicated instruction (target machine: CPU No.3) drops below the number of blocks specified in SD798. This relay is also on when the instruction is being executed. • This relay switches to off when there exist free blocks during the END processing.	S (Instruction execution/ During END)	Rn*2 RnP RnSF
SM799	Number of used blocks information for the multiple CPU dedicated instruction (for CPU No.4)	Off: The specified number of blocks is reserved On: The number of blocks specified in SD799 is not reserved	• This relay switches to on when the number of remaining blocks in the dedicated instruction transfer area to be used in the multiple CPU dedicated instruction (target machine: CPU No.4) drops below the number of blocks specified in SD799. This relay is also on when the instruction is being executed. • This relay switches to off when there exist free blocks during the END processing.	S (Instruction execution/ During END)	Rn*2 RnP RnSF
SM816	Hold mode (S.IN instruction)	Off: Value not held On: Value held	Whether to hold the output value or not is specified when the input value is found to be exceeding the valid range during the range check processing of the S.IN instruction.	U	RnP RnPSF
SM817	Hold mode (S.OUT instruction)	Off: Value not held On: Value held	This relay is used to specify whether or not to hold the output values of the S.OUT1, S.OUT2, and S.DUTY instructions if a sensor error occurs.	U	RnP RnPSF

A

*1 The programmable controller CPU with firmware version "28" or later supports these special relay areas. However, there are no restrictions on the version of the R00CPU, R01CPU, and R02CPU.
*2 There are restrictions on the firmware version of the supported CPU module and software version of the engineering tool. (☞ Page 1139 Added and Enhanced Functions)
*3 The CPU module where this function can be used supports these special relay areas.

Latch area

The following is the list of special relay areas relating to the latch area.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM922	Firmware update completion with/without an error	Off: Update completed without an error (including successful completion) On: Update completed with an error	This relay switches to on when the firmware update function (firmware update using an SD memory card) is completed with an error. (switching to on when SD922 is 0100H to 0300H)	S (Initial)	Rn*1 RnP*1
SM940	Operation setting of the device test with execution conditions	Off: Registration disabled On: Registration not disabled	Set the operation when files relevant to the device test with execution conditions are changed.	U	Rn*1 RnP*1 RnSF*1
SM953	CPU module data backup error check flag	Off: No error On: Error	• This relay turns on if an error occurs at the execution of backup of the CPU module. • This relay turns off at the start of the CPU module data backup.	S (Status change)	Rn*1 RnP*1 RnSF*1
SM959	CPU module data restoration error check flag	Off: No error On: Error	• This relay turns on if an error occurs at the execution of restoration of the CPU module. • This relay turns off at the start of restoration of the CPU module.	S (Status change)	Rn*1 RnP*1 RnSF*1
SM960	Upper limit setting flag for the number of CPU module backup data	Off: Backup continued On: Backup stopped	This relay specifies the operation of backup when the number of backup data of the CPU module reaches the upper limit. (This relay is valid only when bit 5 of SD944 is on.) Off: After deleting the oldest date stamp folder, the backup is continued. On: The backup is not continued if the upper limit of the number of backup data is exceeded. (In this case, the backup is completed with an error.)	U	Rn*1 RnP*1 RnSF*1
SM961	Automatic backup retry failure flag	Off: Retry not executed/ Retry being executed On: Retry failed	This relay turns on when the retry of the automatic backup of the CPU module is failed even after the specified number of retries are attempted. This relay turns off at the start of the automatic backup. (This relay does not turn off when SM1351 is on.)	S (Status change)	Rn*1 RnP*1 RnSF*1

*1 The CPU module where this function can be used supports this special relay area.

Data logging function

The following is the list of special relay areas relating to the data logging function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1200	Auto logging setting file and registration status	Off: Mismatch On: Matching	• This relay is on when the auto logging setting executed (registered) matches with the content of the configuration file stored in the target memory. This relay is off when the setting does not match with the content.	S (Status change)	Rn*1 RnP RnSF
SM1201	SD memory card setting file in use flag	Off: Not used On: In use	• This relay switches to on when the data logging configuration file stored in an SD memory card is being used. This relay switches to on when one or more data logging of the settings No.1 to 10 is registered. • The ON state is maintained even when the data logging later enters into suspend/waiting for start without collection/waiting for RUN without collection mode. However, the relay switches to off when all the data logging stops.	S (Status change)	Rn*1 RnP RnSF
SM1202	Data memory setting file in use flag	Off: Not used On: In use	• This relay switches to on when the data logging configuration file stored in the data memory is being used. This relay switches to on when one or more data logging of the settings No.1 to 10 is registered. • The ON state is maintained even when the data logging later enters into suspend/waiting for start without collection/waiting for RUN without collection mode. However, the relay switches to off when all the data logging stops.	S (Status change)	Rn*1 RnP RnSF
SM1203	Data logging file transfer stop request	Off→On:Stop request exists On→Off:Transfer stopped	• This relay stops the data logging file transfer according to the value specified in SD1203 when this relay is changed from off to on. • This relay turns off when the data logging file transfer is stopped.	U/S (Status change)	Rn*1
SM1210	Data logging setting No.1 Data logging preparation	Off: Not prepared On: Prepared	• This relay switches to on when the data logging preparation is completed. • The ON state is maintained even when the data logging later enters into suspend/waiting for start without collection/waiting for RUN without collection mode. This relay switches to off when the data logging stops.	S (Initial)	Rn*1 RnP RnSF
SM1211	Data logging setting No.1 Data logging start	Off: Suspended/waiting for start On: Start	• This relay switches to on when the data logging starts. The relay turns off when the data logging is in suspend/waiting for start without collection mode. Associated special relay, such as Data logging collection, Data logging end, Data logging trigger, and After data logging trigger, switches to off simultaneously. • This relay also switches to off when the CPU module is moved from RUN to STOP mode and therefore the data collection is halted.	S (Status change)	Rn*1 RnP RnSF
SM1212	Data logging setting No.1 Data logging collection	Off: Not in progress On: In progress	This relay switches to on when the data logging starts data collection.	S (Status change)	Rn*1 RnP RnSF

A

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1213	Data logging setting No.1 Data logging end	Off: Not completed On: Completed	This relay switches to on when the data logging is completed. For continuous logging, the corresponding bit switches to on when writing reaches the maximum number of storage files and data logging is completed (with "Stop" set for the operation at the time when the number of save files exceeds the limit). For trigger logging, the corresponding bit switches to on when data collection for the specified number of records has been completed followed by writing into the SD memory card after the trigger condition was satisfied. The bit also switches to on when an error is generated (except for data logging error caused by online program change) during the data logging execution.	S (Status change)	Rn*1 RnP RnSF
SM1214	Data logging setting No.1 Data logging trigger	Off→On:Triggered	The system switches this relay to on when the specified trigger condition is satisfied.	S (Status change)	Rn*1 RnP RnSF
SM1215	Data logging setting No.1 After data logging trigger	Off: Not post triggering On: Post triggering	• This relay switches to on once data logging triggering occurs. The ON state is maintained even when the data logging is completed. However, this relay is off when the data logging is in suspend/waiting for start without collection/stop mode • This relay also switches to off when the CPU module is moved from RUN to STOP mode and therefore the data collection is halted.	S (Status change)	Rn*1 RnP RnSF
SM1216	Data logging setting No.1 Data logging error	Off: No error On: Error	• This relay switches to on when a data logging function error is generated. • This relay switches to off when the setting is registered or by the stop command from CPU Module Logging Configuration Tool.	S (Status change)	Rn*1 RnP RnSF
SM1217	Data logging setting No.1 Data logging data saving in progress	Off: Not in progress On: In progress	This relay turns on when data in the internal buffer is being saved in the SD memory card or function memory with the data logging.	S (Status change)	Rn*1 RnP RnSF
SM1218	Data logging setting No.1 Logging data storage file switching in progress	Off: Not in progress On: In progress	This relay switches to on when storage file switching is in progress.	S (Status change)	Rn*1 RnP RnSF
SM1219	Data logging setting No.1 Data logging file transfer execution status flag	Off: Not executed On: Being executed	• This relay turns on at the start of the data logging file transfer function or transferring of data logging files to the data memory. • This relay turns off at the completion of the data logging file transfer function or transferring of data logging transfer files to the data memory.	S (Status change)	Rn*1
SM1220 to SM1229	Data logging setting No.2	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1219).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SM1230 to SM1239	Data logging setting No.3	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1219).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SM1240 to SM1249	Data logging setting No.4	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1219).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SM1250 to SM1259	Data logging setting No.5	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1219).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SM1260 to SM1269	Data logging setting No.6	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1219).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1270 to SM1279	Data logging setting No.7	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1219).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SM1280 to SM1289	Data logging setting No.8	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1219).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SM1290 to SM1299	Data logging setting No.9	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1219).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SM1300 to SM1309	Data logging setting No.10	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1219).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SM1312 to SM1321	Data logging setting No.1 to 10 Data logging suspend/ resume flag	Off→On:Suspend On→Off:Resume	• When this relay changes from off to on, the data logging function is suspended. In an off state of the data logging start SM, no processing is performed. • When this relay changes from on to off, the data logging function resumes. In an ON state of the data logging start SM, no processing is performed.	U	Rn*1 RnP RnSF

*1 The CPU module where this function can be used supports these special relay areas.

*2 These CPU modules do not support Data logging file transfer execution status flags for the data logging settings No.2 to No.10 (SM1229, SM1239, SM1249, SM1259, SM1269, SM1279, SM1289, SM1299, SM1309).

A

CPU module data backup/restoration function

The following is the list of special relay areas relating to the CPU module data backup/restoration function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1350	CPU module data backup status flag	Off: Not being executed On: Being executed	This relay turns on during the backup of the CPU module.	S (Status change)	Rn*1 RnP*1 RnSF*1
SM1351	CPU module data backup execution request	Off→On:Backup requested On→Off:Backup completed	• At the timing when this relay turns from off to on, the backup of the CPU module is executed. • This relay turns off at the completion of the backup of the CPU module.	S (Status change)/U	Rn*1 RnP*1 RnSF*1
SM1353	CPU module data restoration status flag	Off: Not being executed On: Being executed	This relay turns on during the restoration of the CPU module.	S (Status change)	Rn*1 RnSF*1
SM1354	CPU module data restoration execution request	Off→On:Restoration requested On→Off:Restoration completed	• At the timing when this relay turns from off to on during STOP, the restoration of the CPU module is executed. • This relay turns off at the completion of the restoration of the CPU module.	S (Status change)/U	Rn*1
SM1356	Retry status flag for CPU module data automatic backup	Off: Not being executed On: Being executed	This relay turns on during the retry of the automatic backup of the CPU module.	S (Status change)	Rn*1 RnP*1 RnSF*1

*1 The CPU module where this function can be used supports these special relay areas.

iQ Sensor Solution data backup/restoration function

The following is the list of special relay areas relating to the iQ Sensor Solution data backup/restoration function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1360	Right-to-use request for iQ Sensor Solution data backup	Off→On:Right to use requested On→Off:Right to use acquired/released	• This relay requests acquiring or releasing the right to use of the data backup/restoration of iQ Sensor Solution. • At the timing when this relay turns from off to on, the relay requests the right to use with the value specified in SD1360. • This relay turns off when the right to use has been acquired or released.	S (Status change)/U	Rn*1 RnSF*1
SM1361	iQ Sensor Solution backup request	Off→On:Backup requested On→Off:Backup completed	• This relay requests the data backup of iQ Sensor Solution. • At the timing when this relay turns from off to on, the backup of the set device(s) is executed. • This relay turns off at the completion of the data backup of iQ Sensor Solution. • This relay turns from on to off when the right to use is acquired.	S (Status change)/U	Rn*1 RnSF*1
SM1362	iQ Sensor Solution backup normal completion	Off: Other than normal completion On: Completed normally	• This relay turns on when the iQ Sensor Solution data backup is completed normally. • When the iQ Sensor Solution data backup is completed with an error, SM1363 turns on and this relay remains off. • This relay turns from on to off when the right to use is acquired or the backup is requested.	S (Status change)	Rn*1 RnSF*1
SM1363	iQ Sensor Solution backup error completion	Off: Other than error completion On: Completed with an error	• This relay turns on when the iQ Sensor Solution data backup is competed with an error. The error code is stored into SD1376 or SD1377. • When the iQ Sensor Solution data backup is competed normally, SM1362 turns on and this relay remains off. • This relay turns from on to off when the right to use is acquired or the backup is requested.	S (Status change)	Rn*1 RnSF*1

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1364	iQ Sensor Solution restoration request	Off→On:Restoration requested On→Off:Restoration completed	• This relay requests the iQ Sensor Solution data restoration. • At the timing when this relay turns from off to on, the restoration of the set device(s) is executed. • This relay turns off at the completion of the iQ Sensor Solution data restoration. • This relay turns from on to off when the right to use is acquired.	S (Status change)/U	Rn*1 RnSF*1
SM1365	iQ Sensor Solution restoration normal completion	Off: Other than normal completion On: Completed normally	• This relay turns on when the iQ Sensor Solution data restoration is completed normally. • When the iQ Sensor Solution data restoration is completed with an error, SM1366 turns on and this relay remains off. • This relay turns from on to off when the right to use is acquired or the restoration is requested.	S (Status change)	Rn*1 RnSF*1
SM1366	iQ Sensor Solution restoration error completion	Off: Other than error completion On: Completed with an error	• This relay turns on when the iQ Sensor Solution data restoration is completed with an error. The error code is stored into SD1376 or SD1377. • When the iQ Sensor Solution data restoration is completed normally, SM1365 turns on and this relay remains off. • This relay turns from on to off when the right to use is acquired or the restoration is requested.	S (Status change)	Rn*1 RnSF*1
SM1367	iQ Sensor Solution backup/ restoration suspend request	Off→On:Suspension requested On→Off:Suspension completed	• This relay requests the suspension of the iQ Sensor Solution data backup/ restoration. • At the timing when this relay turns from off to on, the iQ Sensor Solution data backup/restoration is suspended. • This relay turns off at the suspension completion of the iQ Sensor Solution data backup/restoration. • This relay turns from on to off when the right to use is acquired or the backup/ restoration is requested.	S (Status change)/U	Rn*1 RnSF*1

*1 The CPU module where this function can be used supports these special relay areas.

File transfer function (FTP client)

The following is the list of special relay areas relating to the file transfer function (FTP client).

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1392	FTP client connection status	Off: Not connected (disconnected) On: Connected	This relay turns on when the connection with the FTP server is established. This relay turns off when the connection with the FTP server is cut off.	S (Status change, END processing)	Rn*1 RnP*2

*1 The CPU module where this function can be used supports this special relay area.
*2 The Process CPU with firmware version "13" or later supports this special relay area.

A

Global label assignment information

The following is the list of special relay areas relating to global label assignment information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1398	Operation setting during global label assignment information update	Off: Response available during update On: Response unavailable during update	This relay configures the operation setting while global label assignment information is being updated.	U	Rn*1 RnSF*1

*1 The CPU module where this function can be used supports these special relay areas.

Memory dump function

The following is the list of special relay areas relating to the memory dump function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1472	Memory dump in progress	Off: Memory dump not executed On: Memory dump in progress	Turns on if memory dump is in progress, and turns off if memory dump is not executed.	S (Status change)	Rn*1 RnSF*1
SM1473	Memory dump completed	Off: Not completed On: Completed	Turns on at the time when collection of the data and save to the SD memory card are completed after the establishment of the trigger condition. Thereafter, the ON state remains even if memory dump is executed again. If the trigger condition is established again, On changes to Off.	S (Status change)	Rn*1 RnSF*1

*1 The CPU module where this function can be used supports these special relay areas.

Event history function

The following is the list of special relay areas relating to the event history function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1464	Event history logging restriction status	Off: Not restricted On: Restricted	Turns on when the event history logging has been restricted. The relay remains on even after the restriction is lifted. Restricted modules can be identified in SD1464 to SD1467. SM1466 turns on when the event history logging of the CPU module is restricted and the event category is error (minor error), and SM1467 is turned on when the event history logging of the CPU module is restricted and the event category is information or warning.	S (Status change)	Rn*1 RnP*1 RnSF*1
SM1466	Event history logging restriction status of CPU module (minor error)	Off: Not restricted On: Restricted	Turns on when CPU module event history logging whose event category is error (minor error) was restricted. The relay remains on even after the restriction is lifted.	S (Status change)	Rn*1 RnP*1 RnSF*1
SM1467	Event history logging restriction status of CPU module (information, warning)	Off: Not restricted On: Restricted	Turns on when CPU module event history logging whose event category is information or warning was restricted. The relay remains on even after the restriction is lifted.	S (Status change)	Rn*1 RnP*1 RnSF*1

*1 The CPU module where this function can be used supports these special relay areas.

CPU module database access function

The following is the list of special relay areas relating to the CPU module database access function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1497	Memory card free space flag for CPU module database	Off: 20MB or more free space On: Less than 20MB free space	Turns on when the free space of the SD memory card storing the CPU module database is 20MB or less.	S (Status change)	Rn[*1]
SM1498	CPU module database start-up flag	Off: Before start-up On: After start-up	Turns on only when the CPU module database access function is enabled by the module parameter setting and the database can be accessed.	S (Status change)	Rn[*1]
SM1499	CPU module database start-up failure	Off: Not failed On: Failed	Turns on only when the CPU module database access function is enabled by the module parameter setting but the database start-up is failed.	S (Status change)	Rn[*1]

*1 The CPU module where this function can be used supports these special relay areas.

A

Ethernet function

The following is the list of special relay areas relating to the Ethernet function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1512	FTP server function file path name switching setting Enable	Off: Disabled On: Enabled	• When this relay is changed from off to on, the directory delimiter in a file path name of the CPU module is switched from "\" to "/". The drive number of the file path name can be omitted depending on the stored value in SD1512 when the relay is turned off and on. This setting is enabled while this relay is on. • When the directory delimiter is changed from "\" to "/", the CPU module can receive a command with "\" as the directory delimiter, however, the CPU module sends only a command with "/" as the directory delimiter. • This relay is on when the setting is enabled by executing the FTP command (quote path-delimiter). • Do not turn on and off/off and on this relay during operation. Doing so may cause malfunction.	S (Status change)/U	Rn[*3]
SM1513	FTP server function file path name switching setting error	Off: No error On: Error	• This relay is turned on when FTP server function file path name switching setting is failed to enable and the relay is turned off when the setting is enabled. The error cause is stored in SD1513. • This relay is turned off when SM1512 is changed from on to off, the setting is enabled by executing the FTP command (quote path-delimiter), or the setting is turned off (SM1512 is off).	S (Status change)	Rn[*3]
SM1520	IP address storage area write request	Off→On:Write request exists On→Off:Writing completed	• When this relay is changed from off to on, IP address settings stored in SD1518 to SD1525 are written to the IP address storage area (system memory[*1]) of the CPU module. • Once write to the IP address storage area (system memory[*1]) is completed (regardless of whether it was successful or not), this relay is turned off.	S (Status change)/U	ALL
SM1521	IP address storage area write error	Off: No error On: Error	This relay is turned to on when write to the IP address storage area (system memory[*1]) failed, and off when it was successful.	S (Status change)	ALL
SM1522	IP address storage area clear request	Off→On:Clear request exists On→Off:Storage area cleared	When this relay is changed from off to on, the IP address storage area (system memory[*1]) is cleared. Once clear of the IP address storage area (system memory[*1]) is completed (regardless of whether it was successful or not), this relay is turned off.	S (Status change)/U	ALL
SM1523	IP address storage area clear error	Off: No error On: Error	This relay is turned to on when clear of the IP address storage area (system memory[*1]) failed, and off when it was successful.	S (Status change)	ALL
SM1524	Initial processing successful completion state	Off: Completed with an error On: Completed normally	This relay is turned on when the initial processing for the Ethernet function was completed successfully.[*2] When the initial processing was completed with an error, SM1525 is turned on and SM1524 stays off.	S (Status change)	ALL
SM1525	Initial processing error completion state	Off: Completed normally On: Completed with an error	This relay is turned on when the initial processing for the Ethernet function was completed with an error.[*2] When the initial processing was completed successfully, SM1524 is turned on and SM1525 stays off.	S (Status change)	ALL

*1 This is the memory used by the system when the CPU module is executing functions.

*2 Ethernet function initial processing involves updating parameters set for data communication to Ethernet-equipped modules to enable communication with external devices. Initial processing of Ethernet-equipped modules is performed by setting Ethernet parameters, writing them to the CPU module, and then powering off and on or resetting the CPU module. Note that if Ethernet parameters are not set, initial processing of Ethernet-equipped modules will be performed based on default parameters. (SM1524 (initial processing successful completion status) turns on the moment initial processing is complete, and communication with external devices becomes possible.)

*3 The CPU module where this function can be used supports this special register area.

CC-Link IE Field Network Basic function

The following is the list of special relay areas relating to the CC-Link IE Field Network Basic function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1536	Cyclic transmission status	Off: Not performed On: Being performed	• This relay turns on when the cyclic transmission starts. • This relay is turned off when the cyclic transmission stops.	S (Every END)	Rn*1
SM1540	Data link status	Off: All stations normal On: One or more faulty stations	• This relay turns on when an error exists even in one device station. • The status of each device station can be checked in SD1540 to SD1543.	S (Every END)	Rn*1

*1 The CPU module where this function can be used supports these special relay areas.

Online module change function

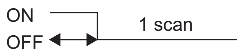
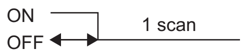
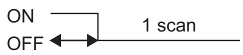
The following is the list of special relay areas relating to the online module change function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1600	Module selection request flag	Off→On:Requested	This relay is turned on to select an online change target module. When changing a module directly, the system turns on this relay upon removal of the target module. The relay can be turned on only when the value set in SD1617 is 0 (Normal operation). The relay turns off upon completion of the online module change processing. If the selection cancel is requested, the relay turns off after the selection is cancelled.	S (Status change)/U (Request)	RnP RnPSF
SM1601	Module selection completion flag	Off: No module selected On: Selected	This relay turns on when an online change target module has been selected. The relay turns off upon completion of the online module change processing.	S (Status change)	RnP RnPSF
SM1602	Module removal request flag	Off→On:Requested	This relay is turned on to request a removal of the selected module. When changing a module directly, the system turns on this relay upon removal of the target module. The relay can be turned on only when the value set in SD1617 is 2 (Module selected). The relay turns off upon completion of the online module change processing.	S (Status change)/U (Request)	RnP RnPSF
SM1603	Module removal ready flag	Off: Not prepared On: Ready	This relay turns on when the selected module is ready to be removed. The relay turns off upon completion of the online module change processing.	S (Status change)	RnP RnPSF
SM1604	Module removal completion flag	Off: Not completed On: Completed	This relay turns on when the selected module has been removed. The relay turns off upon completion of the online module change processing.	S (Status change)	RnP RnPSF
SM1605	Module mounting completion flag	Off: Not completed On: Completed	This relay turns on when a new module has been mounted. The relay turns off upon completion of the online module change processing.	S (Status change)	RnP RnPSF
SM1606	Module recognition request flag	Off→On:Requested	This relay is turned on to request recognition of the newly-mounted module. When changing a module directly, the system turns on this relay upon mounting of the module. The relay can be turned on only when the value set in SD1617 is 6 (Module mounted). The relay turns off upon completion of the online module change processing.	S (Status change)/U (Request)	RnP RnPSF
SM1607	Module recognition completion flag	Off: Not recognized On: Recognized	This relay turns on when the newly-mounted module is recognized by the system. The relay turns off upon completion of the online module change processing.	S (Status change)	RnP RnPSF
SM1608	Module control resumption request flag	Off→On:Requested	This relay turns on to start control of the replaced module. When changing a module directly, the system turns on this relay upon recognition of the module. The relay can be turned on only when the value set in SD1617 is 8 (Module recognized). The relay turns off upon completion of the online module change processing.	S (Status change)/U (Request)	RnP RnPSF
SM1609	Online module change completion flag	Off: Not completed On: Completed	The relay turns on upon completion of the online module change processing. This relay turns off in the next scan.	S (Status change)	RnP RnPSF

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1615	Module selection cancellation request flag	Off→On:Module selection cancellation requested	This relay is turned on to cancel a module selection request. The relay can be turned on only when the value set in SD1617 is 2 (Module selected). The relay turns off after the selection is cancelled.	S (Status change)/U (Request)	RnP RnPSF
SM1616	Online module change availability flag	Off: Disabled On: Enabled	This relay turns on when only the Process CPU is used in a single CPU system. In a multiple CPU system, if any of the CPU modules other than the Process CPU does not support the online module change function, the relay turns off. If all the CPU modules support the function, the relay turns on. Note that the setting details in the direct change setting cannot be checked with SM1616. To check the direct change setting, check the CPU parameters.	S (Initial)	RnP RnPSF
SM1617	Online module change status flag	Off: Function not executed On: Function being executed	This relay turns on when SM1600 is turned on to start the online module change processing. The relay turns off upon completion of the online module change processing.	S (Status change)	RnP RnPSF
SM1618	Online module change error flag	Off: No error On: Error	This relay turns on when an error is detected. This relay turns off when the error cause is eliminated and the online module change related request is executed. An error occurs in selecting a module. Thus, turn off the relay before module selection.	S (Status change)/U (Request)	RnP RnPSF
SM1619	Disable request flag during online module change	Off: No disable request On: Disable request detected	This relay turns on when a disable request is issued during the online module change processing. The relay turns off upon completion of the online module change processing.	S (Status change)	RnP RnPSF

Redundant function

The following is the list of special relay areas relating to the Redundant function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1630	Operation mode identification flag	Off: Redundant system in backup mode, stand-alone system On: Redundant system in separate mode	This relay turns on while a redundant system is operating in separate mode.	S (Every END)	RnP
SM1632	System A identification flag	Off: System B, system not determined On: System A	• This relay distinguishes between the system A and the system B. • The flag status does not change even if tracking cables are disconnected.	S (Initial)	RnP RnPSF
SM1633	System B identification flag	Off: System A, system not determined On: System B	• This relay distinguishes between the system A and the system B. • The flag status does not change even if tracking cables are disconnected.	S (Initial)	RnP RnPSF
SM1634	Control system judgment flag	Off: Standby system, system not determined On: Control system	• This relay indicates operating status of the CPU module. The flag is stored in each system when an initial processing is performed (including when the system is determined while waiting for the start-up of the other system) and when a system switching is completed. • The flag status does not change even if tracking cables are disconnected.	S (Initial/Status change)	RnP RnPSF
SM1635	Standby system judgment flag	Off: Control system, system not determined On: Standby system	• This relay indicates operating status of the CPU module. The flag is stored in each system when an initial processing is performed (including when the system is determined while waiting for the start-up of the other system) and when a system switching is completed. • The flag status does not change even if tracking cables are disconnected.	S (Initial/Status change)	RnP RnPSF
SM1636	Previous control system identification flag		When the previous control system is the system B, this relay turns on during one scan in the system A, following the RUN state after both systems were powered on or the CPU module is reset.	S (Every END)	RnP RnPSF
SM1637	System switching detection (standby system to control system)	Off: Not detected On: Detected	This relay turns on after the standby system has been switched to the control system.	S (Status change)	RnP RnPSF
SM1643	ON for only one scan after system switching (standby system to control system)		• This relay turns on during one scan after the standby system was switched to the control system. • This relay can be executed only in a scan execution type program.	S (Every END)	RnP RnPSF
SM1644	ON for only one scan after system switching (control system to standby system)		• This relay turns on during one scan after the control system was switched to the standby system. • This relay can be executed only in a scan execution type program.	S (Every END)	RnP RnPSF
SM1645	System switching request from a network module	Off: System switching request not issued On: System switching request issued	• This relay turns on when a system switching request from a network module is issued. • The module that issued system switching can be checked in SD1645. • This relay turns off when all bits of SD1645 are off.	S (Every END)	RnP RnPSF

A

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1646	System switching by a user	Off: Disabled On: Enabled	• This relay stores whether to enable system switching operation by a user using an engineering tool or the SP.CONTSW instruction. • Initial value is off: System switching by a user is prohibited.	U	RnP RnPSF
SM1653	Memory copy start	Off: Not started On: Start	• When this relay is turned from off to on, the memory copy from the control system to the standby system starts. Note that the memory copy does not start as long as the I/O number of the copy destination (standby system CPU module: 03D1H) is not stored when SM1653 is turned from off to on. • Initial value is off: The memory copy has not started.	U	RnP
SM1654	Memory copy being executed	Off: Not executed On: Being executed	• This relay is on during the memory copy from the control system to the standby system. • This relay turns off when the memory copy is completed.	S (Status change)	RnP RnPSF
SM1655	Memory copy completion	Off: Not completed On: Completed	• This relay turns on at the completion of the memory copy from the control system to the standby system. • Initial value is off: Memory copy has not completed.	S (Status change)/U	RnP RnPSF
SM1656	Auto memory copy	Off: Disabled On: Enabled	This relay turns on when the automatic memory copy is set to enable.	S (Initial)	RnP
SM1673	Tracking transfer completion flag	Off: Transfer not completed On: Transfer completed	• This flag is stored a result of the tracking transfer operated in the preceding END processing. • The flag turns on if any of the tracking transfers in block 1 to 64 has been normally completed, and turns off if the tracking transfers failed due to causes such as tracking communication error.	S (Status change)	RnP RnPSF
SM1679	Error reset (the other system)	Off→On:Standby system error reset requested On→Off:Standby system error reset completed	• This relay is turned from off to on to clear a continuation error occurred in the standby system. • This relay turns from on to off when the error reset has been completed. • Initial value is off.	U/S (Status change)	RnP RnPSF
SM1680	Error of the other system monitoring	Off: No error On: Error	• This relay turns on if an error occurs on communications with the other system when an initial processing (including when the system is determined while waiting for the other system starts up) or an END processing is performed. (This relay turns on when the bit of SD1648 turns on.) • This relay turns off when an error is cleared.	S (Initial/every END/ system switching)	RnP RnPSF
SM1681	Latest self-diagnostic error (including annunciator ON) (the other system)	Off: No error On: Error	• This relay turns on if a diagnostic error occurs in the CPU module in the other system. (The relay also turns on if an error is detected by an annunciator.) • The SM0 status for the CPU module in the other system is reflected.	S (Every END)	RnP RnPSF
SM1682	Latest self-diagnostic error (not including annunciator ON) (the other system)	Off: No error On: Error	• This relay turns on if a self-diagnostic error occurred in the CPU module in the other system. (The relay remains off if an error is detected by an annunciator.) • The SM1 status for the CPU module in the other system is reflected.	S (Every END)	RnP RnPSF

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1683	Detailed information 1: Flag in use (the other system)	Off: Not used On: In use	• This relay turns on when there is detailed information 1 for an error occurred in the CPU module in the other system. • The SM80 status for the CPU module in the other system is reflected.	S (Every END)	RnP RnPSF
SM1684	Detailed information 2: Flag in use (the other system)	Off: Not used On: In use	• This relay turns on when there is detailed information 2 for an error occurred in the CPU module in the other system. • The SM112 status for the CPU module in the other system is reflected.	S (Every END)	RnP RnPSF
SM1754	Waiting for the start-up of the other system	Off: Started up (own system) On: Waiting for the start-up (the other system)	• This relay turns on while waiting for the start-up of the other system after powered on. • This relay turns off under the following conditions. · Own system starts up as the control system or the standby system after succeeding in tracking communications with the other system. · Own system is started up as the control system by a certain operation while waiting for the start-up of the other system.	S (Status change)	RnP RnPSF
SM1756	Wait timeout for receiving cyclic data after system switching	Off: No timeout On: Timeout	This relay turns on when the receipt of the cyclic data after system switching is not completed within the cyclic data receipt waiting time[*2] while the setting to wait cyclic data receive after system switching is enabled. This relay turns off when the timeout does not occur.	S (At system switching)	RnP[*1] RnPSF[*1]
SM1762	Behavior setting for access from standby system to extension base unit	Off: Error[*3] On: Non-processing	Whether the following operation is handled as an error or not is specified: The execution of an instruction for accessing the buffer memory of a module on an extension base unit from the standby system.	U	RnP[*1]

*1 The CPU module where this function can be used supports this special relay area.

*2 ☞ Page 1087 Waiting time for cyclic data receive after system switching (Twcyc)

*3 The type of the error can be set in the RAS setting of CPU parameters. (Set "Operation Error" of "CPU Module Operation Setting at Error Detected" to "Stop" or "Continue".)

Safety information

The following is a list of special relay areas relating to safety information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1840	Allowed time over flag for continuous RUN in TEST MODE	Off: Within the specified time On: Specified time exceeded	This relay turns on when the continuous RUN time in TEST MODE has exceeded the allowed time set in the CPU parameter, and turns off in the following cases. • When the Safety CPU is powered on or is reset • When an error is cleared	S (Error)	RnSF
SM1888	Safety cycle processing time execution cycle error flag	Off: No error (normal) On: Error	• This relay switches to on if a cycle in which safety cycle processing was not executed is detected. • Once the relay turns on, it remains on even if the safety cycle processing is performed each cycle. (To turn off the relay, power off and on the system or reset the CPU module).	S (Status change)	RnPSF RnSF
SM1904	Safety communication setting (1st module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1904.	S (Initial)	RnPSF RnSF
SM1905	Safety communication setting (2nd module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1905.	S (Initial)	RnPSF RnSF
SM1906	Safety communication setting (3rd module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1906.	S (Initial)	RnPSF RnSF
SM1907	Safety communication setting (4th module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1907.	S (Initial)	RnPSF RnSF
SM1908	Safety communication setting (5th module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1908.	S (Initial)	RnPSF RnSF
SM1909	Safety communication setting (6th module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1909.	S (Initial)	RnPSF RnSF
SM1910	Safety communication setting (7th module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1910.	S (Initial)	RnPSF RnSF

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SM1911	Safety communication setting (8th module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1911.	S (Initial)	RnPSF RnSF

A

Appendix 5 List of Special Register Areas

The following table lists items in the list.

Item	Description
No.	Special register number
Name	Special register name
Data stored	Data stored in the special register
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (program, engineering tool, GOT, or other testing operations from external device) • U/S: User and system <Set timing> • Every END: Data is set every time END processing is performed. • Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN). • Status change: Data is set when the status is changed. • Error: Data is set when an error occurs. • Instruction execution: Data is set when an instruction is executed. • Request: Data is set when requested by a user (using the special relay). • Switch change: Data is set when the switch of the CPU module is changed. • Card insertion/removal: Data is set when an SD memory card is inserted or removed. • Writing: Data is set when a user performs a writing operation. • During END: Data is set when END processing is performed. • System switching: Data is set when two systems are switched (between the control system and the standby system)
CPU	The following shows the supported CPU modules. Each of the CPU module is represented by the following symbols. • Rn: Programmable controller CPU • RnP: Process CPU • RnPSF: SIL2 Process CPU • RnSF: Safety CPU • ALL: All the above CPU modules

Point

Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

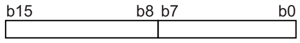
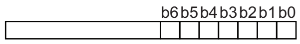
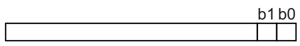
Diagnostic information

The following is the list of special register areas relating to the diagnostic information.

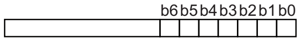
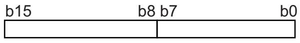
No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD0	Latest self diagnostics error code	Latest self diagnostics error code	Error codes are stored in a hexadecimal value when the diagnostics detects an error.	S (Error)	ALL
SD1	Latest self-diagnostics error time	Latest self-diagnostics error time	The year value (four digits) of the date/time when SD0 data was updated is stored as a BIN code.	S (Error)	ALL
SD2			The month value of the date/time when SD0 data was updated is stored as a BIN code.		ALL
SD3			The day value of the date/time when SD0 data was updated is stored as a BIN code.		ALL
SD4			The hour value of the date/time when SD0 data was updated is stored as a BIN code.		ALL
SD5			The minute value of the date/time when SD0 data was updated is stored as a BIN code.		ALL
SD6			The second value of the date/time when SD0 data was updated is stored as a BIN code.		ALL
SD7			The day of the week value of the date/time when SD0 data was updated is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)		ALL
SD10	Self-diagnostic error number	Self-diagnostic error number 1	The maximum of 16 types of error codes are stored into SD10 onwards when the diagnostics detects an error (The same error code as one already stored in SD10 onwards is not stored). The 17th error code onwards are not stored. Also error codes are not stored when 16 types of error codes have already been stored into SD10 to SD25.	S (Error)	ALL
SD11		Self-diagnostic error number 2			ALL
SD12		Self-diagnostic error number 3			ALL
SD13		Self-diagnostic error number 4			ALL
SD14		Self-diagnostic error number 5			ALL
SD15		Self-diagnostic error number 6			ALL
SD16		Self-diagnostic error number 7			ALL
SD17		Self-diagnostic error number 8			ALL
SD18		Self-diagnostic error number 9			ALL
SD19		Self-diagnostic error number 10			ALL
SD20		Self-diagnostic error number 11			ALL
SD21		Self-diagnostic error number 12			ALL
SD22		Self-diagnostic error number 13			ALL
SD23		Self-diagnostic error number 14			ALL
SD24		Self-diagnostic error number 15			ALL
SD25		Self-diagnostic error number 16			ALL
SD49	Error detection invalidation setting	Error detection invalidation setting	Specify a function for which detection of continuation errors is invalidated. (On: Invalidate (do not detect an error), Off: Do not invalidate (detect an error)) b0 b0: Built-in Ethernet port communication error	U	Rn[*1] RnP[*1]

A

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD53	AC/DC DOWN	The number of AC/DC DOWN detections	The value of this register increments by one and stored as a BIN code, each time input voltage drops to 85% (AC power)/65% (DC power) or less of the nominal value while the CPU module is carrying out an operation. A counting cycle from 0 to 65535 to 0 is repeated. (In a redundant system with redundant extension base unit, if a momentary power failure occurs in a module on an extension base unit, counting will be performed in both systems.)	S (Error)	ALL
SD60	Number of module with blown fuse	Number of module with blown fuse	• The lowest I/O number of module in which a fuse blew is stored. • The fuse blown state check is also done for output modules on the remote I/O station. • In a redundant system with redundant extension base unit, this register is set as follows: · If a fuse blown is detected in an output module on an extension base unit, the I/O number is stored only in the CPU module of the control system, but not stored in the CPU module of the standby system. · When the systems are switched, the state before system switching is held. · When the error is cleared for either system, this register is cleared to 0 only in that system.	S (Error)	ALL
SD61	I/O module verify error module number	I/O module verify error module number	• The lowest I/O number of the module which has an error detected by the I/O module verification is stored. • I/O module verification is also done for modules on the remote I/O station. • In a redundant system with redundant extension base unit, this register is set as follows: · If an I/O verification error is detected in a module on an extension base unit, the I/O number is stored only in the CPU module of the control system, but not stored in the CPU module of the standby system. · When the systems are switched, the state before system switching is held. · When the error is cleared for either system, this register is cleared to 0 only in that system.	S (Error)	ALL
SD62	Annunciator number	Annunciator number	The annunciator number which is detected first is stored.	S (Instruction execution)	ALL
SD63	Number of annunciators	Number of annunciators	The number of detected annunciator is stored.	S (Instruction execution)	ALL
SD64 to SD79	Table of detected annunciator numbers	Detected annunciator number	• Numbers of activated annunciators are sequentially registered into SD64 to SD79, each time an annunciator (F) is turned on by the SET F instruction. • Number of the annunciator deactivated by the RST F instruction is removed from register areas SD64 to SD79, and the numbers of annunciators which were lined up behind the removed one move forward one by one. When the number of detected annunciators has already reached 16, the number of a newly detected annunciator is not stored into SD64 to SD79. For details on the annunciator, refer to the annunciator (F) (☞ Page 379 Annunciator (F))	S (Instruction execution)	ALL

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD80	Detailed information 1 information category	Detailed information 1 information category code	• Detailed information 1 information category code is stored.  b0 to b7: Information category code b8 to b15: Unused (fixed to 0) • The following codes are stored into the information category code. 0: N/A 1: Program position information 2: Drive/file information 4: Parameter information 5: System configuration information 6: Number of times information 7: Time information 24: Failure information 27: System switching information 34: Continuous RUN prevention setting in TEST MODE 37: Data type (tracking transfer) information 38: Tracking transfer trigger information 46: CPU module data backup/restoration folder information 50: Safety station system configuration information 58: Extension cable information	S (Error)	ALL
SD81 to SD111	Detailed information 1	Detailed information 1	• Detailed information 1 corresponding to the error code (SD0) is stored. • The type of the detailed information 1 can be obtained using SD80 (the value of the "Detailed information 1 information category code" stored in SD80 corresponds to the following figures (1), (2), (4) to (7), (24), (27), (34), (37) and (38)). (1) Error location information ■SD81: With or without specification  b0: Argument number b1: SFC block number b2: SFC step number b3: SFC transition number b4: Sequence step number b5: FB number b6: File name ■SD82: Argument number (stored in the range from 1) ■SD83: SFC block number ■SD84 to SD85: SFC step number ■SD86 to SD87: SFC transition number ■SD88 to SD89: Step number ■SD90: FB number ■SD91 to SD98: File name (first eight characters of Unicode string) (2) Drive/file information ■SD81: With or without specification  b0: Drive number b1: File name ■SD82: Drive number ■SD83 to SD90: File name (first eight characters of Unicode string)	S (Error)	ALL

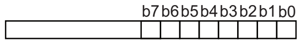
A

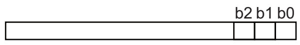
No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD81 to SD111	Detailed information 1	Detailed information 1	(4) Parameter information ■SD81: With or without specification  b0: Parameter type b1: Parameter storage location b2: I/O number b3: Parameter number b4: Network number b5: Station number b6: System information ■SD82  • b0 to b7: Parameter type (stored in the following value) 1: System parameter 2: CPU parameter 3: Module parameter 4: Module extension parameter 5: Memory card parameter 6: Safety system parameter 7: Safety CPU parameter 8: Safety module parameter • b8 to b15: Parameter storage location (2: SD memory card, 4: Data memory) ■SD83: I/O number 0xFFFF is stored when no I/O number is assigned. ■SD84: Parameter number ■SD85: Network number ■SD86: Station number 0 to 120 (0 is stored for the master station.) ■SD87 to SD97: System information	S (Error)	ALL

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD81 to SD111	Detailed information 1	Detailed information 1	(5) System configuration information ■SD81: With or without specification b6 b5 b4 b3 b2 b1 b0 b0: I/O number b1: Slot number b2: Base number b3: Power supply number b4: CPU number b5: Network number b6: Station number ■SD82: I/O number ■SD83 b15 b8 b7 b0 • b0 to b7: Slot number (0 to 11) • b8 to b15: Base number (0: Main base unit, 1 to 7: Extension base unit level 1 to level 7, 8: Higher than level 7) ■SD84 b15 b8 b7 b0 • b0 to b7: Power supply number (1 to 2: Power supply 1 to 2) • b8 to b15: CPU number: (1 to 4: CPU No. 1 to 4) ■SD85: Network number ■SD86: Station number (6) Number of times information When there is no specification, 0 is set to each SD. ■SD81: With or without specification b1 b0 b0: Number of times (set value) b1: Number of times (actual measurement value) ■SD82 to SD83: Number of times (set value) When the number of times (set value) is not specified, 0 is set to each SD. • SD82: Lower word of the number of times (set value) • SD83: Upper word of the number of times (set value) ■SD84 to SD85: Number of times (actual measurement value) When the number of times (actual measurement value) is not specified, 0 is set to each SD. • SD84: Lower word of the number of times (actual measurement value) • SD85: Upper word of the number of times (actual measurement value) (7) Time information When there is no specification, 0 is set to each SD. ■SD81: With or without specification b3 b2 b1 b0 b0: Time (value set) (ms) b1: Time (value set) (μs) b2: Time (actual measurement value) (ms) b3: Time (actual measurement value) (μs)	S (Error)	ALL

A

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD81 to SD111	Detailed information 1	Detailed information 1	■SD82: Time (value set) (ms) ■SD83: Time (value set) (μs) ■SD84: Time (actual measurement value) (ms) ■SD85: Time (actual measurement value) (μs) (24) Failure information Failure information is system information. (27): System switching information ■SD81: With or without specification b3 b2 b1 b0 b0: System switching cause b1: System switching instruction ID number b2: Cause of system switching failure b3: Control system/standby system transition ■SD82: System switching cause 1: Power-off, reset, hardware failure 2: Stop error 3: System switching request from a network module 16: System switching request by using the SP.CONTSW instruction 17: System switching request using an engineering tool ■SD83: System switching instruction ID number ■SD84: Cause of system switching failure 1: Tracking communications disabled 2: Tracking communication timeout 3: Stop error of the standby system 4: Operating status mismatch between both systems 5: Memory copy being executed 6: Online change being executed 7: A failure of a network module detected on the standby system 8: System switching being executed 9: A redundant function module being changed online 10: System switching disabled on the standby system by using the DCONTSW instruction 11: Online module change being executed on a main base unit in a redundant system with redundant extension base unit 14: Safety operation mode mismatch between both systems 15: A failure of safety tracking data detected ■SD85: Control system/standby system transition 1: Control system → Standby system 2: Standby system → Control system (34) TEST MODE continuous RUN prevention setting ■SD81: With or without specification b0 b0: Continuous RUN time value (second) ■SD82 to SD83: Continuous RUN time value (second) • SD82: Lower word of the continuous RUN time value (second) • SD83: Upper word of the continuous RUN time value (second) (37) Data type (tracking transfer) information ■SD81: With or without specification b0 b0: Data type ■SD82: Data type b15 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 b0: Device data b1: Label data b2: Signal flow b3: PID control instruction information b4: SFC information (including step relay (S)) b5: System switching request b6: Operation mode change request b15: System data 0 is stored when each data is not sent, and 1 is stored when each data is being sent in each bit.	S (Error)	ALL

<table>
<tr><th>No.</th><th>Name</th><th>Data stored</th><th>Details</th><th>Set by (setting timing)</th><th>CPU</th></tr>
<tr><td>SD81 to SD111</td><td>Detailed information 1</td><td>Detailed information 1</td><td>(38) Tracking transfer trigger information
■SD81: With or without specification


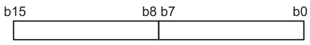
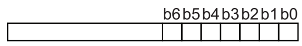
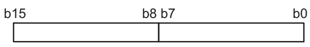
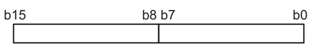
b0: Block No. 1 to 8
b1: Block No. 9 to 16
b2: Block No. 17 to 24
b3: Block No. 25 to 32
b4: Block No. 33 to 40
b5: Block No. 41 to 48
b6: Block No. 49 to 56
b7: Block No. 57 to 64
■SD82 to SD85: Block number
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SD82</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SD83</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SD84</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SD85</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table>
(46) CPU module data backup/restoration folder information
■SD81: With or without specification


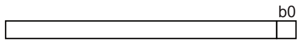
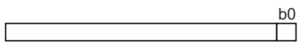
b0: Folder specification
b1: Date folder
b2: Number folder
■SD82: Folder specification
0: Specification allowed
1: Specification not allowed
■SD83 to SD84: Date folder (yyyymmdd)
Date of the folder is stored in BCD code. (yyyy: 0 to 9999, mm: 1 to 12, dd: 1 to 31)
FFFFFFFFH is stored when a folder cannot be specified.
• SD83: Lower word of the date folder (yyyymmdd)
• SD84: Upper word of the date folder (yyyymmdd)
■SD85: Number folder
0 to 32767 (FFFFH is stored when a folder cannot be specified.)</td><td>S (Error)</td><td>ALL</td></tr>
</table>

A

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD81 to SD111	Detailed information 1	Detailed information 1	(50) Safety station system configuration information ■SD81: With or without specification b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 b0: Own station I/O number b1: Slot number on the own station base unit b2: Own station base number b3: Own station power supply number b4: Own station CPU number b5: Network number b6: Station number b7: Connection number b8: Slot number on another station base unit b9: Another station base number ■SD82: Own station I/O number ■SD83: Own station base number/Slot number on the own station base unit b15 b8 b7 b0 • b0 to b7: Slot number on the base unit (0 to 11) • b8 to b15: Base number (0: Main base unit, 1 to 7: Extension base unit level 1 to level 7, 8: Higher than level 7) ■SD84: Own station CPU number/Own station power supply number b15 b8 b7 b0 • b0 to b7: Power supply number (1 to 2: Power supply 1 to 2) • b8 to b15: CPU number: (1 to 4: CPU 1 to 4) ■SD85: Network number ■SD86: Station number ■SD87: Connection number 0 to 120 ■SD88: Another station base number/Slot number on another station base unit b15 b8 b7 b0 • b0 to b7: Slot number on the base unit (0 to 11) • b8 to b15: Base number (0: Main base unit, 1 to 7: Extension base unit level 1 to level 7, 8: Higher than level 7) (58) Extension cable information ■SD81: With or without specification b1 b0 b0: Base number b1: Extension cable connector ■SD82: Base number 1 to 6: Extension level 1 to 6 (extension base units) 11: Main base unit (system A) 12: Main base unit (system B) ■SD83: Extension cable connector 0: OUT (Used when an error occurs in an extension cable between either main base unit (system A/B) and the redundant extension base unit, or between two extension base units (when the extension base unit in the next upper level is one other than the redundant extension base unit) 1: OUT1 2: OUT2	S (Error)	ALL

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD112	Detailed information 2 information category	Detailed information 2 information category code	Detailed information 2 information category code is stored. b15 b8 b7 b0 • b0 to b7: Information category code • b8 to b15: Unused (fixed to 0) The following codes are stored into the information category code. • 0: N/A • 2: Drive/file information • 3: Annunciator number • 4: Parameter information • 5: System configuration information • 25: Process control instruction processing information • 28: Program error information • 29: Error information of other stations (CC-Link IE TSN/CC-Link IE Field) • 57: Pair version information	S (Error)	ALL
SD113 to SD143	Detailed information 2	Detailed information 2	• Detailed information 2 corresponding to the error code (SD0) is stored. • The type of information can be checked in SD112. (The value of the "Detailed information 2 information category code" stored in SD112 corresponds to the following (2) to (5), (25), (28), (29) and (57).) (2) Drive/file information ■SD113: With or without specification b1 b0 b0: Drive number b1: File name ■SD114: Drive number ■SD115 to SD122: File name (first eight characters of Unicode string) (3) Annunciator number ■SD113: With or without specification b0 b0: Annunciator number ■SD114: Annunciator number (4) Parameter information ■SD113: With or without specification b6 b5 b4 b3 b2 b1 b0 b0: Parameter type b1: Parameter storage location b2: I/O number b3: Parameter number b4: Network number b5: Station number b6: System information	S (Error)	ALL

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD113 to SD143	Detailed information 2	Detailed information 2	■SD114  • b0 to b7: Parameter type (stored in the following value) 1: System parameter 2: CPU parameter 3: Module parameter 4: Module extension parameter 5: Memory card parameter 6: Safety system parameter 7: Safety CPU parameter 8: Safety module parameter • b8 to b15: Parameter storage location (2: SD memory card, 4: Data memory) ■SD115: I/O number FFFFH is stored when no I/O number is assigned. ■SD116: Parameter number ■SD117: Network number ■SD118: Station number ■SD119 to SD129: System information (5) System configuration information ■SD113: With or without specification  b0: I/O number b1: Slot number b2: Base number b3: Power supply number b4: CPU number b5: Network number b6: Station number ■SD114: I/O number ■SD115  • b0 to b7: Slot number (0 to 11) • b8 to b15: Base number (0: Main base unit, 1 to 7: Extension base unit level 1 to level 7, 8: Higher than level 7) ■SD116  • b0 to b7: Power supply number (1 to 2: Power supply 1 to 2) • b8 to b15: CPU number: (1 to 4: CPU 1 to 4) ■SD117: Network number ■SD118: Station number	S (Error)	ALL

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD113 to SD143	Detailed information 2	Detailed information 2	(25) Process control instruction processing information ■SD113: With or without specification  b0: Process control instruction processing information ■SD114: Processing description The following are processing blocks stored in SD114 (Processing details). 1: — (Instruction with no processing block) 2: Range check 3: Input limiter 4: Engineering value inverse transformation 5: Digital filter 6: Input addition 7: Variation rate & upper/lower limiter 8: Reset windup 9: Output conversion 10: Output ON time conversion 11: Variation rate check 12: Integrated value calculation 13: Control cycle determination 14: SV setting 15: Tracking 16: Gain (Kp) operation 17: PID operation 18: Deviation check 19: PID operation 1 (operation processing of Bn or Cn) 20: PID operation 2 (operation processing of Dn) 21: PID operation 3 (operation processing of ΔMV) 22: PIDP operation 23: Operating time monitor 24: SPI operation 25: IPD operation 26: BPI operation 27: Engineering value transformation 28: Variation rate limiter 29: Ratio calculation 30: Upper/lower limit check 31: Loop stop 32: MV correction 33: Two-position (on/off) control 34: Three-position (on/off) control 35: Operation constant check 36: SV count-up 37: MVPGS operation 38: Output 39: Input check 40: Timeout determination 41: Step manipulated value set 42: Sampling cycle determination 43: Response waveform observation 44: Identification processing 45: PID constants calculation (28) Program error information ■SD113: With or without specification  b0: Program error code ■SD114: Program error code	S (Error)	ALL

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD113 to SD143	Detailed information 2	Detailed information 2	(29) Error information of other stations (CC-Link IE TSN/CC-Link IE Field) ■SD113: With or without specification bF … b7 b6 b5 b4 b3 b2 b1 b0 b0: Error classification b1: Error code b2: Date (yyyymmdd) b3: Time (hhmmss) b4: Day of the week b5: Detailed error information 1 b6: Detailed error information 2 b7 to bF: Detailed error information 3 to detailed error information 10 ■SD114: Error classification ■SD115: Error code ■SD116 and SD117: Date (yyyymmdd) • SD116: Lower word of date (yyyymmdd) • SD117: Upper word of date (yyyymmdd) ■SD118 and SD119: Time (hhmmss) • SD118: Lower word of time (hhmmss) • SD119: Upper word of time (hhmmss) ■SD120: Day of the week ■SD121: Detailed error information 1 ■SD122: Detailed error information 2 ■SD123 to SD130: Detailed error information 3 to 10 (57) Pair version information ■SD113: With or without specification b1 b0 b0: Pair version (Safety CPU) b1: Pair version (Safety function module) ■SD114: Pair version (Safety CPU) ■SD115: Pair version (Safety function module)	S (Error)	ALL
SD150	Power-off/power supply voltage drop detection status	Power-off/power supply voltage drop detection status (bit pattern) 0: Power-on/ normal voltage 1: Power-off/ voltage drop detected/no power supply module	• For the redundant power supply base unit or redundant extension base unit, the power supply module status (power is shut off, power supply voltage drop (not including a momentary power failure) is detected, or an empty slot for the power-supply module exists) is stored using the following bit pattern. • In a multiple CPU system, the status is stored only to the CPU module No.1. • In a redundant system with redundant extension base unit, if power-off or power supply voltage drop is detected in a power supply module on an extension base unit, the state is stored only in the CPU module of the control system, but not stored in the CPU module of the standby system. (2) (1) b15 b9 b8 b7 b2 b1 b0 (1) Powered-off/power supply voltage dropped (power supply module 1) b0:Main base unit b1 to b7: 1st to 7th level of the extension base unit (2) Powered-off/power supply voltage dropped (power supply module 2) b8: Main base unit b9 to b15: 1st to 7th level of the extension base unit	S (Status change)	Rn*1 RnP RnPSF

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD151	Power supply failure detection status	Power supply failure detection status (bit pattern) 0: Not detected/ power-off/no power supply module 1: Detected	• For the redundant power supply base unit or redundant extension base unit, failure detection status of the power supply module is stored using the following bit pattern. • Bits corresponding to power-off or empty slots for the power supply module turn off. • In a multiple CPU system, the status is stored only to the CPU module No.1. • In a redundant system with redundant extension base unit, if a failure of a power supply module on an extension base unit, the state is stored only in the CPU module of the control system, but not stored in the CPU module of the standby system. (2) (1) b15 b9 b8 b7 b2 b1 b0 (1) Failure detection status of the power supply module 1 b0:Main base unit b1 to b7: 1st to 7th level of the extension base unit (2) Failure detection status of the power supply module 2 b8: Main base unit b9 to b15: 1st to 7th level of the extension base unit	S (Status change)	Rn*1 RnP RnPSF
SD152	Momentary power failure detection count (power supply module 1)	Momentary power failure detection count for power supply module 1	• This register counts the number of momentary power failures. • This register monitors the status of the power supply module mounted on the main base unit and counts the number of momentary power failures. • When the CPU module starts up, the counters of the both power supplies are cleared to 0. • When one of the two power supply is powered off, the corresponding counter to the one powered off is cleared to 0. • The counter is incremented by one upon one momentary power failure on each power supply. A counting cycle from 0 to 65535 to 0 is repeated. • In a multiple CPU system, the status is stored only to the CPU module No.1.	S (Status change)	Rn*1 RnP RnPSF
SD153	Momentary power failure detection count (power supply module 2)	Momentary power failure detection count for power supply module 2		S (Status change)	Rn*1 RnP RnPSF
SD154	Details of the invalid power supply module	Details of the invalid power supply module (bit pattern) 0: Valid/power-off/no power supply module 1: Invalid	• For the redundant power supply base unit or redundant extension base unit, when an invalid power supply module exists, the status is stored using the following bit pattern. • Bits corresponding to power-off or empty slots for the power supply module turn off. • In a multiple CPU system, the status is stored only to the CPU module No.1. • In a redundant system with redundant extension base unit, if an invalid power supply module is detected on an extension base unit, the state is stored only in the CPU module of the control system, but not stored in the CPU module of the standby system. (2) (1) b15 b9 b8 b7 b2 b1 b0 (1) Details of the invalid power supply module 1 b0:Main base unit b1 to b7: 1st to 7th level of the extension base unit (2) Details of the invalid power supply module 2 b8: Main base unit b9 to b15: 1st to 7th level of the extension base unit	S (Status change)	Rn*1 RnP RnPSF

*1 There are restrictions on the firmware version of the supported CPU module. (☞ Page 1139 Added and Enhanced Functions)

A

System information

The following is the list of special register areas relating to the system information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD160	Firmware version	Firmware version	The firmware version is stored.	S (Initial)	Rn*1 RnP*5 RnSF*4
SD164 to SD171	Production information	Production information	This register stores the production information. b15 ··· b8b7 ··· b0 SD164: 2nd digit from the left / 1st digit from the left SD165: 4th digit from the left / 3rd digit from the left ⋮ SD171: 16th digit from the left / 15th digit from the left Example: The production information is 123456789ABCDEFG. b15 ··· b8b7 ··· b0 SD164: 32H (2) / 31H (1) SD165: 34H (4) / 33H (3) ⋮ SD171: 47H (G) / 46H (F)	S (Initial)	Rn*6
SD200	Status of switch	Status of CPU switch	The switch status of the CPU module is stored as follows: 0: RUN, 1: STOP	S (Switch change)	ALL
SD201	LED status	CPU-LED state	This register stores the information that indicates LED status (0: off, 1: on, 2: flashing (fast/slow)) of the CPU module in the following bit patterns. b15 b12 b11 b8 b7 b4 b3 b0 (8) (7) (6) (5) (4) (3) (2) (1) (1) READY (2) ERROR (3) PROGRAM RUN (4) USER (5) BATTERY (6) CARD READY (7) CARD ACCESS (8) FUNCTION • For the R00CPU, (5), (6), and (7) are fixed to 0. • For the R01CPU and R02CPU, (5) is fixed to 0.	S (Status change)	ALL
SD203	Operating status of CPU	Operating status of CPU	The operating status of the CPU module is stored as follows: 0: RUN, 2: STOP, 3: PAUSE	S (Every END)	ALL
SD205	Safety operation mode	Safety operation mode	The safety operation mode is stored. (The applicable bit turns on.) b15 ... b3: 0; b2: 1/0 (3); b1: 1/0 (2); b0: 1/0 (1) (1) TEST MODE (2) SAFETY MODE (3) SAFETY MODE (wait-for-restart)	S (Status change)	RnPSF RnSF
SD206	Pair version	Pair version	The pair version of the SIL2 Process CPU or Safety CPU is stored in decimal notation.	S (Initial)	RnPSF RnSF

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD210	Clock data	Clock data (year)	The year value (four digits) of the clock data is stored as a BIN code.	S/U (Request)	ALL
SD211		Clock data (month)	The month value of the clock data is stored as a BIN code.	S/U (Request)	ALL
SD212		Clock data (day)	The day value of the clock data is stored as a BIN code.	S/U (Request)	ALL
SD213		Clock data (hour)	The hour value of the clock data is stored as a BIN code.	S/U (Request)	ALL
SD214		Clock data (minute)	The minute value of the clock data is stored as a BIN code.	S/U (Request)	ALL
SD215		Clock data (second)	The second value of the clock data is stored as a BIN code.	S/U (Request)	ALL
SD216		Clock data (day of the week)	The day of the week value of the clock data is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)	S/U (Request)	ALL
SD218	Time zone setting value	Time zone (in minutes)	The time zone setting value specified in the parameter is stored in increments of minutes. Example: when the setting value of the time zone is "UTC+9," the result of the following equation is stored into SD218: 9 × 60 (minutes) = 540.	S (Initial)	ALL
SD228	Multiple CPU system information	Number of CPU modules	The number of CPU modules which constitute a multiple CPU system is stored (one to four, including those reserved).	S (Initial)	Rn*2 RnP RnSF
SD229		CPU module number in multiple CPU system	The number of this CPU No. is stored when a multiple CPU system is configured.	S (Initial)	Rn*2 RnP RnSF
SD230		CPU No.1 operating status	The operation information for each CPU No. is stored (the amount of stored information depending on the number of CPU modules indicated in SD228 is stored). b15 b14 … b8 b7 b6 b5 b4 b3 … b0 (4) | | (3) | | (2) | (1) (1) The operating status is stored in b0 to b3. • 0: RUN • 2: STOP • 3: PAUSE • 4: INITIALIZE • FH: RESET (2) The classification is stored in b4 and b5. However, minor or moderate errors will be those set in the CPU parameter RAS settings, system parameter I/O assignment settings, and multiple CPU settings. • 0: Normal • 1: Minor error (error code: 1000H to 1FFFH) • 2: Moderate error (error code: 2000H to 3BFFH) • 3: Major error (error code: 3C00H to 3FFFH) (3) The stop error flag is stored in b7. • 0: No stop error • 1: Stop error (4) The CPU module mounting status is stored in b15. • 0: Not mounted • 1: Mounted	S (During END/Error)	Rn*2 RnP RnSF
SD231		CPU No.2 operating status			Rn*2 RnP RnSF
SD232		CPU No.3 operating status			Rn*2 RnP RnSF
SD233		CPU No.4 operating status			Rn*2 RnP RnSF
SD241	Extension stage number	0: base unit only 1 to 7: number of extension base units	The maximum number of implemented extension base units is stored. (In a redundant system with redundant extension base unit, the number is stored in the CPU module of both systems.)	S (Initial)	Rn RnP RnSF
SD242	Identification for whether or not Q series module can be mounted	Identification of the base type 0: Cannot be mounted (There is no base unit on which the Q series module can be mounted.) 1: Can be mounted (There is a base unit on which the Q series module can be mounted.)	Identify whether or not Q series module can be mounted. When no SD memory card is inserted, the value is fixed to 0. b7 to b2 b1 b0 • b0: Main base unit (fixed to 0) • b1: Extension base unit 1 (Q series modules can be mounted.) • b2: Extension base unit 2 (Q series modules can be mounted.) • b3 to b7: Extension base unit 3 to 7 (Q series modules can be mounted.) • b8 to b15: Fixed to 0	S (Initial)	ALL

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD243	Number of base slots	Number of base slots	The number of slots of the base unit, which is specified in the Base / power supply / extension cable setting of system parameters, is stored. When the number of slots of the base unit is not specified by the system parameter, that of the mounted base unit is stored. (In a redundant system with redundant extension base unit, the number is stored in the CPU module of both systems.) b15 b12 b11 b8 b7 b4 b3 b0 SD243: 3 | 2 | 1 | 0 SD244: 7 | 6 | 5 | 4 0: Main 1 to 7: Extension 1 to 7	S (Initial)	ALL
SD244					ALL
SD250	Loaded maximum I/O	Loaded maximum I/O number	The value obtained by dividing the last I/O number for an implemented unit plus one by 16 is stored. Example 1: Last I/O number 010FH • SD250 = 0011H Example 2: Last I/O number 0FFFH • SD250 = 0100H	S (Initial)	ALL
SD260	Number of points assigned to bit devices	X (L)	The number of points assigned to the X device is stored in 32 bits.	S (Initial)	ALL
SD261		X (H)			ALL
SD262		Y (L)	The number of points assigned to the Y device is stored in 32 bits.	S (Initial)	ALL
SD263		Y (H)			ALL
SD264		M (L)	• The number of points assigned to the M device is stored in 32 bits. • The number of points is stored even when the number of points assigned to M is 32K or less.	S (Initial)	ALL
SD265		M (H)			ALL
SD266		B (L)	• The number of points assigned to the B device is stored in 32 bits. • The number of points is stored even when the number of points assigned to B is 32K or less.	S (Initial)	ALL
SD267		B (H)			ALL
SD268		SB (L)	• The number of points assigned to the SB device is stored in 32 bits. • The number of points is stored even when the number of points assigned to SB is 32K or less.	S (Initial)	ALL
SD269		SB (H)			ALL
SD270		F (L)	The number of points assigned to the F device is stored in 32 bits.	S (Initial)	ALL
SD271		F (H)			ALL
SD272		V (L)	The number of points assigned to the V device is stored in 32 bits.	S (Initial)	ALL
SD273		V (H)			ALL
SD274		L (L)	The number of points assigned to the L device is stored in 32 bits.	S (Initial)	ALL
SD275		L (H)			ALL
SD276		S (L)	The number of points assigned to the S device is stored in 32 bits.	S (Initial)	Rn[*2] RnP[*2]
SD277		S (H)			Rn[*2] RnP[*2]
SD280	Number of points assigned to word devices	D (L)	• The number of points assigned to the D device is stored in 32 bits. • The number of points is stored even when the number of points assigned to D is 32K or less.	S (Initial)	ALL
SD281		D (H)			ALL
SD282		W (L)	• The number of points assigned to the W device is stored in 32 bits. • The number of points is stored even when the number of points assigned to W is 32K or less.	S (Initial)	ALL
SD283		W (H)			ALL
SD284		SW (L)	• The number of points assigned to the SW device is stored in 32 bits. • The number of points is stored even when the number of points assigned to SW is 32K or less.	S (Initial)	ALL
SD285		SW (H)			ALL

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD288	Number of points assigned to timer/ counter devices	T (L)	• The number of points assigned to the T device is stored in 32 bits. • The number of points is stored even when the number of points assigned to T is 32K or less.	S (Initial)	ALL
SD289		T (H)			ALL
SD290		ST (L)	• The number of points assigned to the ST device is stored in 32 bits. • The number of points is stored even when the number of points assigned to ST is 32K or less.	S (Initial)	ALL
SD291		ST (H)			ALL
SD292		C (L)	• The number of points assigned to the C device is stored in 32 bits. • The number of points is stored even when the number of points assigned to C is 32K or less.	S (Initial)	ALL
SD293		C (H)			ALL
SD294		LT (L)	• The number of points assigned to the LT device is stored in 32 bits. • The number of points is stored even when the number of points assigned to LT is 32K or less.	S (Initial)	ALL
SD295		LT (H)			ALL
SD296		LST (L)	• The number of points assigned to the LST device is stored in 32 bits. • The number of points is stored even when the number of points assigned to LST is 32K or less.	S (Initial)	ALL
SD297		LST (H)			ALL
SD298		LC (L)	• The number of points assigned to the LC device is stored in 32 bits. • The number of points is stored even when the number of points assigned to LC is 32K or less.	S (Initial)	ALL
SD299		LC (H)			ALL
SD300	Number of points assigned to the index register	Z	The number of points assigned to the Z device is stored.	S (Initial)	ALL
SD302	Number of points assigned to the long index register	LZ	The number of points assigned to the LZ device is stored.	S (Initial)	ALL
SD306	Number of points assigned to the file register	ZR (L)	• The number of points assigned to the ZR device is stored in 32 bits. • The number of points is stored even when the number of points assigned to ZR is 32K or less.	S (Initial)	ALL
SD307		ZR (H)			ALL
SD308	Number of points assigned to refresh devices	RD (L)	• The number of points assigned to the RD device is stored in 32 bits. • The number of points is stored even when the number of points assigned to RD is 32K or less.	S (Initial)	ALL
SD309		RD (H)			ALL
SD312	File register block number	File register block number	The block number of the file register currently selected is stored.	S (Status change)	ALL
SD315	Service processing constant wait status setting	• Other than AFFFH: Disabled • AFFFH: Enabled	"AFFFH" is stored to enable the device/label access service processing constant wait function. Other than AFFFH: The device/label access service processing wait constant function disabled (Default) AFFFH: The device/label access service processing constant wait function enabled	U (Request)	RnP*2
SD384	System operation setting	• LED control setting for program restoration information write status • Target file setting for the file batch online change	Either of the following two setting can be used. This setting is written to the setting storage area (system area) by turning off and on SM384. ■LED control setting for program restoration information write status Set either of the following to specify whether or not the LED flashes when the program restoration information is not in the CPU module. • AFA0H: LED flashing • AFAFH: Without LED flashing ■Target file setting for the file batch online change Set either of the following to specify the target file of the file batch online change For the R00CPU, R01CPU, and R02CPU, this setting is always "program file/FB file/global label setting file" regardless of the set value of this relay. • AFB0H: Program file only • AFBFH: Program file/FB file/global label setting file	U	Rn*2 RnP*2 RnSF*2*3
SD385	System operation setting error cause	Error cause for the case when failing in writing into setting storage area	An error cause is stored when failing in writing into setting storage area (system memory). (Linked with SM385) • 0H:No error • 100H:A value stored in SD384 is out of the specified range. • 200H:Write failure	S (Status change)	Rn*2 RnP*2 RnSF*2*3

A

*1 The programmable controller CPU with firmware version "28" or later supports these special register areas. However, there are no restrictions on the version of the R00CPU, R01CPU, and R02CPU.
*2 There are restrictions on the firmware version of the supported CPU module and software version of the engineering tool. (☞ Page 1139 Added and Enhanced Functions)
*3 Only the standard program is supported. Only the setting value of "LED control setting for program restoration information write status" can be set for SD384.
*4 The Safety CPU with firmware version "11" or later supports this special register area.
*5 The Process CPU with firmware version "14" or later supports this special register area.
*6 The programmable controller CPUs with the following firmware versions supports these special register areas.
· R00CPU, R01CPU, R02CPU: "24" or later
· Programmable controller CPUs other than the R00CPU, R01CPU, R02CPU: "57" or later.

SFC information

The following is the special register area relating to SFC information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD329	Online change (SFC block) target block number	SFC block number	• A target SFC block number is stored while the online change (SFC block) is being executed (SM329 = ON). • FFFFH is stored when the online change (SFC block) is not executed.	S (Status change)	Rn*1 RnP*1 RnSF*1

*1 There are restrictions on the firmware version of the supported CPU module and software version of the engineering tool. (☞ Page 1139 Added and Enhanced Functions)

System clock

The following is the list of special register areas relating to the system clock.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD412	One second counter	The number of counts that is counted once per second.	• The value in this register increments by one for each second after the CPU module enters in RUN mode. • A counting cycle from 0 to 65535 to 0 is repeated.	S (Status change)	ALL
SD414	2n second clock setting	Unit setting for 2n second clock	• The n value of the 2n second clock is stored (Default: 30). • Configurable range is -32768 to 32767 (0 to FFFFH).	U	ALL
SD415	2n ms clock setting	Unit setting for 2n ms clock	• The n value for the 2n ms clock is stored. (Default: 30). • Configurable range is -32768 to 32767 (0 to FFFFH).	U	ALL
SD420	Scan counter	The number of counts that is counted once for each scan.	• The value in this register increments by one for each scan after the CPU module enters in RUN mode (however, the count is skipped for scans by the initial execution type program). • A counting cycle from 0 to 65535 to 0 is repeated.	S (Every END)	ALL
SD1184	System clock (SM400 to SM403) every end update setting	System clock (SM400 to SM403) every end update setting	A value to enable or disable the system clock (SM400 to SM403) every end update is stored. Turning off and on SM1184 (System clock (SM400 to SM403) every end update setting request) submits a request to set either of the following two values. • AA00H: Disabled • AA55H: Enabled	U	Rn*1
SD1185	System clock (SM400 to SM403) every end update setting error cause	Error cause for the case when failing in writing into setting storage area	This register stores an error cause when the setting value has failed to be written to the setting storage area. (This register operates together with SM1185 (System clock (SM400 to SM403) every end update setting error).) • 0000H: No error • 0100H: Value in SD1184 out of range • 0200H: Setting value failed to be written	S (Status change)	Rn*1

*1 The programmable controller CPUs with the following firmware versions supports these special register areas.
· R00CPU, R01CPU, R02CPU: "24" or later
· Programmable controller CPUs other than the R00CPU, R01CPU, R02CPU: "57" or later.

A

Fixed scan function information

The following is the list of special register areas relating to the fixed scan function information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD480	Number of cycle overrun events for inter-module synchronization cycle program (I44)	0: No cycle overrun event 1 to 65535: Accumulated number of cycle overrun events	The number of events in which the inter-module synchronous interrupt program (I44) has not been completed within the inter-module synchronization cycle or the program cannot be executed due to various reasons, such as execution of a higher-priority interrupt program and interrupt disabling by the instruction execution is stored. When the count exceeds 65535, it returns to 0 and starts a new cycle. The number of cycle overrun events is counted regardless of the setting content for the error check setting of the RAS setting (execution check of the inter-module synchronous interrupt (I44)).	S (Status change)	Rn RnP RnSF*1
SD481	Number of cycle overrun events for multiple CPU synchronization program (I45)	0: No cycle overrun event 1 to 65535: Accumulated number of cycle overrun events	The number of events in which the multiple CPU synchronization program (I45) has not been completed within the fixed scan communication cycle or the program cannot be executed due to various reasons, such as execution of a higher-priority interrupt program and interrupt disabling by the instruction execution is stored. When the count exceeds 65535, it returns to 0 and starts a new cycle. The number of cycle overrun events is counted regardless of the setting content for the error check setting of the RAS setting (execution check of the multiple CPU synchronization program (I45)).	S (Status change)	Rn*1 RnP RnSF*1
SD484	Number of execution section excess errors for multiple CPU synchronization interrupt program	0: No error (Normal) 1 to 65535: Accumulated number of errors	The number of events in which the program is executed exceeding the program execution section within the specified multiple CPU synchronization cycle is stored. When the count exceeds 65535, it returns to 0 and starts a new cycle. Note that the number of error occurrences is counted regardless of the CPU module operation setting for error detections within the RAS setting of the CPU parameter.	S (Status change)	Rn*1 RnP RnSF*1
SD500	Execution program number	Execution program number	The program number which is currently executed is stored as a BIN value.	S (Status change)	ALL
SD518	Initial scan time	Initial scan time (unit: ms)	• The initial scan time is stored into SD518 and SD519 (it is measured in microseconds (μs).) SD518: stores a value in the ms place (storage range: 0 to 65535) SD519: stores a value in the μs place (storage range: 0 to 999) • These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN.	S (Every END)	ALL
SD519		Initial scan time (unit: μs)			ALL
SD520	Current scan time	Current scan time (unit: ms)	• The current scan time is stored into SD520 and SD521 (it is measured in increments of μs). SD520: stores a value in the ms place (storage range: 0 to 65535) SD521: stores a value in the μs place (storage range: 0 to 999) Example: If the current scan time is 23.6ms, the following values are stored: SD520 = 23 SD521 = 600 • These areas are cleared to 0 when the operating status of the CPU module is switched to STOP. • In a redundant system, when the operating status of the CPU module is RUN state, these areas are cleared to 0 at the system switching.	S (Every END)	ALL
SD521		Current scan time (unit: μs)			ALL
SD522	Minimum scan time	Minimum scan time (unit: ms)	• The minimum value of the scan time other than one for the initial execution program is stored into SD522 and SD523 (it is measured in increments of μs). SD522: stores a value in the ms place (storage range: 0 to 65535) SD523: stores a value in the μs place (storage range: 0 to 999) • These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN. • In a redundant system, when the operating status of the CPU module is RUN state, these areas are cleared to 0 at the system switching.	S (Every END)	ALL
SD523		Minimum scan time (unit: μs)			ALL

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD524	Maximum scan time	Maximum scan time (unit: ms)	• The maximum value of the scan time other than one for the initial execution program is stored into SD524 and SD525 (it is measured in increments of μs). SD524: stores a value in the ms place (storage range: 0 to 65535) SD525: stores a value in the μs place (storage range: 0 to 999) • These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN. • In a redundant system, when the operating status of the CPU module is RUN state, these areas are cleared to 0 at the system switching.	S (Every END)	ALL
SD525		Maximum scan time (unit: μs)			ALL
SD526	END processing time	END processing time (unit: ms)	• The time period from completion of a scan program until start of the next scan is stored into SD526 to SD527 (it is measured in increments of μs). SD526: stores a value in the ms place (storage range: 0 to 65535) SD527: stores a value in the μs place (storage range: 0 to 999) • These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN. • In a redundant system, when the operating status of the CPU module is RUN state, these areas are cleared to 0 at the system switching.	S (Every END)	ALL
SD527		END processing time (unit: μs)			ALL
SD528	Constant scan wait time	Constant scan wait time (unit: ms)	• The waiting time specified in the constant scan setting process is stored into SD528 and SD529 (it is measured in increments of μs). SD528: stores a value in the ms place (storage range: 0 to 65535) SD529: stores a value in the μs place (storage range: 0 to 999) • These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN. • In a redundant system, when the operating status of the CPU module is RUN state, these areas are cleared to 0 at the system switching.	S (Every END)	ALL
SD529		Constant scan wait time (unit: μs)			ALL
SD530	Scan program execution time	Scan program execution time (unit: ms)	• The execution time of the scan program for one scan is stored into SD530 and SD531 (it is measured in increments of μs). SD530: stores a value in the ms place (storage range: 0 to 65535) SD531: stores a value in the μs place (storage range: 0 to 999) • These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN. • In a redundant system, when the operating status of the CPU module is RUN state, these areas are cleared to 0 at the system switching.	S (Every END)	ALL
SD531		Scan program execution time (unit: μs)			ALL

*1 There are restrictions on the firmware version of the supported CPU module and software version of the engineering tool. (☞ Page 1139 Added and Enhanced Functions)

A

Drive information

The following is the list of special register areas relating to the drive information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD600	Memory card mounting status	SD memory card type	This register indicates the type of mounted SD memory cards. b15 b8 b7 b4 b3 b0 • b0 to b3: Fixed to 0 • b4 to b7 0: Does not exist 4: SD memory card b8 to b15: Fixed to 0	S (Initial, card insertion/ removal)	ALL[*4]
SD604	SD memory card (drive 2) usage status	SD memory card (drive 2) usage status	Usage status of the SD memory card is stored using the following bit pattern. (On indicates being used.) b0: Event history b1: Module extension parameter[*1] b2: Label communication data b3 to b15: Not used	S (Status change)	ALL[*4]
SD606	SD memory card (drive 2) capacity	SD memory card (drive 2) capacity: the lower digits (unit: K bytes)	The capacity of the SD memory card is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial, card insertion/ removal)	ALL[*4]
SD607		SD memory card (drive 2) capacity: the higher digits (unit: K bytes)	The capacity of the SD memory card is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial, card insertion/ removal)	ALL[*4]
SD610	SD memory card (drive 2) free space	SD memory card (drive 2) free space: the lower digits (unit: K bytes)	The amount of free space of the SD memory card is stored in increments of 1K byte.	S (At change)	ALL[*4]
SD611		SD memory card (drive 2) free space: the higher digits (unit: K bytes)	The amount of free space of the SD memory card is stored in increments of 1K byte.	S (At change)	ALL[*4]
SD614	Device/label memory (drive 3) usage status	Device/label memory (drive 3) usage status	Usage status of the device/label memory is stored using the following bit pattern. (On indicates being used.) b0: File register b1 to b15: Not used	S (Status change)	ALL
SD616	Device/label memory (drive 3) capacity	Device/label memory (drive 3) capacity: the lower digits (unit: K bytes)	The capacity of the device/label memory is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)	ALL
SD617		Device/label memory (drive 3) capacity: the higher digits (unit: K bytes)	The capacity of the device/label memory is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)	ALL
SD618	Device/label memory (file storage area) capacity	Device/label memory (file storage area) (drive 3) capacity: the lower digits (unit: K bytes)	The capacity of the device/label memory (file storage area) is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)	ALL
SD619		Device/label memory (file storage area) (drive 3) capacity: the higher digits (unit: K bytes)	The capacity of the device/label memory (file storage area) is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)	ALL
SD620	Data memory (drive 4) usage status	Data memory (drive 4) usage status	Usage status of the data memory is stored using the following bit pattern. (On indicates being used.) b0: Event history b1: Module extension parameter[*1] b2: Label communication data b3 to b15: Not used	S (Status change)	ALL

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD622	Data memory (drive 4) capacity	Data memory (drive 4) capacity: the lower digits (unit: K bytes)	The capacity of the data memory is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)	ALL
SD623		Data memory (drive 4) capacity: the higher digits (unit: K bytes)	The capacity of the data memory is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)	ALL
SD626	Extended SRAM cassette capacity identification information	Capacity identification information of the Extended SRAM cassette	Capacity identification information of the Extended SRAM cassette is stored. Unmounted: 0, 1M: 1, 2M: 2, 4M: 3, 8M: 4, 16M: 5	S (Initial)	ALL*5
SD629	Program memory write (transfer) status	Write (transfer) status display (percent)	This register displays write (transfer) status to the program memory in percentage (0 to 100%). The initial value is "0". Upon completion of writing, this register is set to "100". It is set to "0" at the time when the write command is issued.	S (Writing)	ALL
SD630	Program memory write count index	Index of the number of write operations up to now	• This register indicates the index value for the number of write operations to the program memory up to now (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value is continued to be counted even when it exceeds 100000). If the index value exceeds 100000, the CPU module must be replaced.	S (Writing)	ALL
SD631					ALL
SD633	Data memory write (transfer) status	Write (transfer) status display (percent)	This register displays write (transfer) status to the data memory in percentage. (0 to 100%). The initial value is "0". Upon completion of writing, this register is set to "100". It is set to "0" at the time when the write command is issued.*2	S (Writing)	ALL
SD634	Index for the number of data memory write operations	Index of the number of write operations up to now	• This register indicates the index value for the number of write operations to the data memory up to now (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value is continued to be counted even when it exceeds 100000). If the index value exceeds 100000, the CPU module must be replaced.	S (Writing)	ALL
SD635					ALL
SD638	Index for the number of system memory write operations	Index of the number of write operations up to now	• This register indicates the index value for the number of write operations to the system memory (Flash ROM)*3 up to now. (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value is continued to be counted even when it exceeds 100000). If the index value exceeds 100000, the CPU module must be replaced.	S (Writing)	RnPSF RnSF
SD639					RnPSF RnSF
SD640	Internal buffer empty area usage status	Internal buffer empty area usage status (usage status of the internal buffer for functions where the internal buffer capacity is not set in the internal buffer capacity setting of the CPU parameters)	This register stores the following in a bit pattern: usage status of the internal buffer for functions where the internal buffer capacity is not set in the internal buffer capacity setting of the CPU parameters. (On indicates being used.) b0: Used for realtime monitor b1 to b15: Not used	S (Status change)	Rn RnSF
SD642	Internal buffer capacity	Internal buffer capacity Low-order (in K bytes)	The capacity of the internal buffer is stored in K bytes.	S (Initial)	Rn RnSF
SD643		Internal buffer capacity High-order (in K bytes)	The capacity of the internal buffer is stored in K bytes.	S (Initial)	Rn RnSF

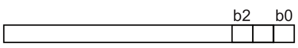
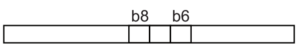
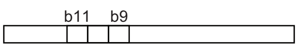
A

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD644	Internal buffer free area space	Internal buffer free area space Low-order (in K bytes)	This register stores the following in K bytes: internal buffer free space that is not set in the internal buffer capacity setting of the CPU parameters.	S (Status change)	Rn RnSF
SD645		Internal buffer free area space High-order (in K bytes)	This register stores the following in K bytes: internal buffer free space that is not set in the internal buffer capacity setting of the CPU parameters.	S (Status change)	Rn RnSF
SD648	Function memory capacity	Function memory capacity Low-order (in K bytes)	The capacity of the function memory is stored in K bytes.	S (Initial)	Rn*6
SD649		Function memory capacity High-order (in K bytes)	The capacity of the function memory is stored in K bytes.	S (Initial)	Rn*6
SD650	Function memory free space capacity	Function memory free space capacity Low-order (in K bytes)	The amount of free space of the function memory is stored in K bytes.	S (Status change)	Rn*6
SD651		Function memory free space capacity High-order (in K bytes)	The amount of free space of the function memory is stored in K bytes.	S (Status change)	Rn*6
SD652	Function memory clear error cause	Function memory clear error cause	An error cause detected at a request for clearing the function memory is stored. 0: No error Other than 0:For details on the values stored when an error occurs, refer to the list of error codes. (☞ Page 805 List of error codes) The register is cleared to 0 at a request for clearing the function memory.	S (Status change)	Rn*6
SD653	File transfer to data memory error cause	File transfer to data memory error cause	The error cause that is detected at file transfer to the data memory is stored. 0: No error Other than 0:For details on the values stored when an error occurs, refer to the list of error codes. (☞ Page 805 List of error codes) The register is cleared to 0 when SM653 (File transfer to data memory request) changes from off to on.	S (Status change)	Rn*6

*1 It is stored when used for the Ethernet function of CPU module.
*2 When the save destination for the event history is the data memory: the event history will be stored at the timing of powering off and on and resetting the CPU module. Accordingly, since the data will be written into the data memory, "100" is stored. (When the save destination for the event history is the SD memory, the initial value remains "0".)
*3 This is the memory used by the system when the CPU module is executing functions.
*4 The CPU module where the SD memory card can be used supports these special register areas.
*5 The CPU module where the extended SRAM cassette can be used supports these special register areas.
*6 There are restrictions on the firmware version of the supported CPU module and software version of the engineering tool. (☞ Page 1139 Added and Enhanced Functions)

Instruction related

The following is the list of special register areas relating to the instruction-related items.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD757	Current interrupt priority	Current interrupt priority	The priority for the interrupt of the interrupt program currently executed is stored. 1 to 8:Priority for the interrupt pointer of the interrupt program currently being executed 0: No interrupt operation (default)	S (Status change)	ALL
SD758	Interrupt disabling for each priority setting value	Interrupt disabling for each priority setting value	The interrupt priority of the interrupt program that the interrupt is disabled by using the DI instructions (Disabling interrupt programs/ Disabling interrupt programs with specified priority or lower) and the EI instruction (Enabling interrupt programs) is stored in this register. Note that safety programs are unaffected by SD758, and are executed based on the safety cycle time. 1: Interrupt programs with priority 1 or lower (all priority levels) disabled (default) 2: Interrupt programs with priority 2 or lower disabled 3: Interrupt programs with priority 3 or lower disabled 4: Interrupt programs with priority 4 or lower disabled 5: Interrupt programs with priority 5 or lower disabled 6: Interrupt programs with priority 6 or lower disabled 7: Interrupt programs with priority 7 or lower disabled 8: Interrupt programs with priority 8 or lower disabled 0: No interrupt program disabled (interrupt programs with any priority level enabled)	S (Status change)	ALL
SD760	Unicode text file faulty area in the DBIMPORT instruction	Row number of the Unicode text file (lower)	If an error in the Unicode text file is detected when the DBIMPORT(P) instruction is executed, the row number of the Unicode text file is stored. The value is stored when the error completion signal of the DBIMPORT(P) instruction is turned on and is cleared to 0 when the DBIMPORT(P) instruction is executed.	S (Status change)	Rn
SD761		Row number of the Unicode text file (upper)		S (Status change)	Rn
SD771	Specification of the number of write instruction executions to data memory	Specification of the number of write instruction executions to data memory	In this register, the maximum number of the data memory writing instruction (SP.DEVST) executions per day is specified. When the number of data memory writing instruction executions exceeds the value specified in this register, an error is generated. Setting range is from 1 to 32767. If a value is set outside the range, an error is generated during the data memory writing instruction execution.	U	ALL
SD774	Execution status of data table sort instructions	Execution status of data table sort instructions	The execution statuses of SORTTBL/SORTTBL2/DSORTTBL2 instructions are stored. The bit indicating the finish of each instruction turns on when the instruction is finished. ■ SORTTBL instruction  b0: Finish (0: Status except execution finished, 1: Execution finished) b2: Execution (0: Not executed, 1: Executed) ■ SORTTBL2 instruction  b6: Finish (0: Status except execution finished, 1: Execution finished) b8: Execution (0: Not executed, 1: Executed) ■ DSORTTBL2 instruction  b9: Finish (0: Status except execution finished, 1: Execution finished) b11:Execution (0: Not executed, 1: Executed)	S (Status change)	Rn*3

<table>
<tr><th>No.</th><th>Name</th><th>Data stored</th><th>Details</th><th>Set by (setting timing)</th><th>CPU</th></tr>
<tr><td>SD775</td><td>Selection of refresh processing during the COM instruction execution</td><td>Selection of refresh processing during the COM instruction execution</td><td>In this register, whether or not each processing is executed during the COM instruction execution is selected (Default: 0).
The specification on the SD775 is effective when SM775 is turned on.
<table><tr><td>b15</td><td>···</td><td>b13</td><td>···</td><td>b6</td><td>···</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>■ Refresh processing (0: Not executed, 1: Executed)
b0: I/O refresh, I/O fetch from a group out of the multiple CPU system
b1: Link refresh of the CC-Link module
b2: Link refresh of the CC-Link IE Controller Network module and MELSECNET/H network module
b3: Intelligent function module refresh
b4: Refresh using the CPU buffer memory of the multiple CPU system (END)
b6: Link refresh of the CC-Link IE Field Network module
b13:Link refresh of CC-Link IE Field Network Basic
■ Device/label access service processing execution (0: Executed, 1: Not executed)
b15:Device/label access service processing (communications with the engineering tool, GOT, or other external devices)</td><td>U</td><td>ALL</td></tr>
<tr><td>SD792 and SD793</td><td>PID limit setting (for complete derivative)</td><td>0: Limit restriction applied
1: No limit restriction</td><td>The limit restriction for each PID loop is specified as follows: (for the PIDCONT instruction)
<table><tr><td></td><td>b15</td><td></td><td>b1</td><td>b0</td></tr><tr><td>SD792</td><td>16</td><td>to</td><td>2</td><td>1</td></tr><tr><td>SD793</td><td>32</td><td>to</td><td>18</td><td>17</td></tr></table>1 to 32: Loop 1 to 32</td><td>U</td><td>ALL</td></tr>
<tr><td>SD794 and SD795</td><td>PID limit setting (for incomplete derivative)</td><td>0: Limit restriction applied
1: No limit restriction</td><td>The limit restriction for each PID loop is specified as follows: (for the S.PIDCONT instruction)
<table><tr><td></td><td>b15</td><td></td><td>b1</td><td>b0</td></tr><tr><td>SD794</td><td>16</td><td>to</td><td>2</td><td>1</td></tr><tr><td>SD795</td><td>32</td><td>to</td><td>18</td><td>17</td></tr></table>1 to 32: Loop 1 to 32</td><td>U</td><td>ALL</td></tr>
</table>

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD796	Maximum number of blocks used for the multiple CPU dedicated instruction (for CPU No.1)	The maximum number of blocks to be used for the dedicated instruction Depending on the number of CPU modules which constitute a multiple CPU system, the range is as follows.*1 When constituting two modules: 2 to 599 When constituting three modules: 2 to 299 When constituting four modules: 2 to 199 (Default: 2).	• The maximum number of blocks used for the multiple CPU dedicated instruction is specified (for CPU No.1). • When executing the multiple CPU dedicated instruction on CPU No.1, if the number of free blocks in the dedicated instruction transfer area is less than the setting value on this register, SM796 is turned on. • This value is used as interlock signal for the continuous executions of the multiple CPU dedicated instruction.	U	Rn*2 RnP RnSF
SD797	Maximum number of blocks setting used for the multiple CPU dedicated instruction (for CPU No.2)		• The maximum number of blocks used for the multiple CPU dedicated instruction is specified (for CPU No.2). • When executing the multiple CPU dedicated instruction on CPU No.2, if the number of free blocks in the dedicated instruction transfer area is less than the setting value on this register, SM797 is turned on. • This value is used as interlock signal for the continuous executions of the multiple CPU dedicated instruction.	U	Rn*2 RnP RnSF
SD798	Maximum number of blocks setting used for the multiple CPU dedicated instruction (for CPU No.3)		• The maximum number of blocks used for the multiple CPU dedicated instruction is specified (for CPU No.3). • When executing the multiple CPU dedicated instruction on CPU No.3, if the number of free blocks in the dedicated instruction transfer area is less than the setting value on this register, SM798 is turned on. • This value is used as interlock signal for the continuous executions of the multiple CPU dedicated instruction.	U	Rn*2 RnP RnSF
SD799	Maximum number of blocks setting used for the multiple CPU dedicated instruction (for CPU No.4)		• The maximum number of blocks used for the multiple CPU dedicated instruction is specified (for CPU No.4). • When executing the multiple CPU dedicated instruction on CPU No.4, if the number of free blocks in the dedicated instruction transfer area is less than the setting value on this register, SM799 is turned on. • This value is used as interlock signal for the continuous executions of the multiple CPU dedicated instruction.	U	Rn*2 RnP RnSF
SD816	Basic period	Execution cycle	An execution cycle (unit: second) of process control instructions is set in real number.	U	RnP RnPSF
SD817					RnP RnPSF
SD818	Bumpless function availability setting for the S.PIDP instruction	0: Enabled 1: Disabled	The availability of the bumpless function for the S.PIDP instruction is set.	U	RnP RnPSF
SD819	Process value output type setting for the S.PHPL2 instruction	0: Decimal 1: Percent	Set the output type of the process value (PV) for the S.PHPL2 instruction of process control instruction.	U	RnP*3
SD820	Dummy device	Dummy device	A dummy device used in process control instructions is set.	U	RnP RnPSF
SD821					RnP RnPSF

*1 When the value out of the range is specified, operation runs while its value is being regarded as max value of each range of multiple CPU system configuration.

*2 There are restrictions on the firmware version of the supported CPU module and software version of the engineering tool. (Page 1139 Added and Enhanced Functions)

*3 The CPU module where this function can be used supports these special register areas.

Firmware update function

The following is the list of special register areas relating to the firmware update function (firmware update using an SD memory card).

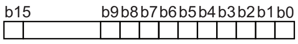
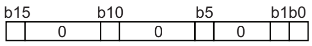
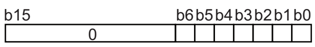
No.	Name	Data stored		Details	Set by (setting timing)	CPU
SD904	Latest firmware update information (network)	History information	Version after the update (network)	The firmware version after the update execution is stored. When the update is completed with an error, 0 is stored.	S (Initial)	Rn[*1] RnP[*1]
SD905			Version before the update (network)	The firmware version before the update execution is stored.	S (Initial)	Rn[*1] RnP[*1]
SD906	Previous firmware update information (network)		Version after the update (network)	The firmware version after the update execution is stored. When the update is completed with an error, 0 is stored.	S (Initial)	Rn[*1] RnP[*1]
SD907			Version before the update (network)	The firmware version before the update execution is stored.	S (Initial)	Rn[*1] RnP[*1]
SD912	Latest firmware update information (CPU)[*2]		Execution time (year)	The year value (four digits) of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)	Rn[*1] RnP[*1]
SD913			Execution time (month)	The month value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)	Rn[*1] RnP[*1]
SD914			Execution time (day)	The day value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)	Rn[*1] RnP[*1]
SD915			Execution time (hour)	The hour value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)	Rn[*1] RnP[*1]
SD916			Execution time (minute)	The minute value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)	Rn[*1] RnP[*1]
SD917			Execution time (second)	The second value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)	Rn[*1] RnP[*1]
SD918			Execution time (day of the week)	The day of the week value of the date/time when the firmware update was executed is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)	S (Initial)	Rn[*1] RnP[*1]
SD919			Version after the update (CPU)	The firmware version after the update execution is stored. When the update is completed with an error, 0 is stored.	S (Initial)	Rn[*1] RnP[*1]
SD920			Version before the update (CPU)	The firmware version before the update execution is stored.	S (Initial)	Rn[*1] RnP[*1]
SD921	Latest firmware update result	Target		The start I/O number of the module where the firmware update was executed is stored. • CPU module: 3FFH	S (Initial)	Rn[*1] RnP[*1]
SD922		Execution result		The execution result of the firmware update is stored. • 0001H: Completed successfully • 0100H: Flash ROM error • 0200H: Model mismatched • 0201H: File invalid • 0202H: Combination invalid • 0203H: Firmware update prohibited state • 0300H: Firmware data error	S (Initial)	Rn[*1] RnP[*1]

No.	Name	Data stored		Details	Set by (setting timing)	CPU
SD923	Previous firmware update information (CPU)[*2]	History information	Execution time (year)	The year value (four digits) of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)	Rn[*1] RnP[*1]
SD924			Execution time (month)	The month value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)	Rn[*1] RnP[*1]
SD925			Execution time (day)	The day value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)	Rn[*1] RnP[*1]
SD926			Execution time (hour)	The hour value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)	Rn[*1] RnP[*1]
SD927			Execution time (minute)	The minute value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)	Rn[*1] RnP[*1]
SD928			Execution time (second)	The second value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)	Rn[*1] RnP[*1]
SD929			Execution time (day of the week)	The day of the week value of the date/time when the firmware update was executed is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)	S (Initial)	Rn[*1] RnP[*1]
SD930			Version after the update (CPU)	The firmware version after the update execution is stored. When the update is completed with an error, 0 is stored.	S (Initial)	Rn[*1] RnP[*1]
SD931			Version before the update (CPU)	The firmware version before the update execution is stored.	S (Initial)	Rn[*1] RnP[*1]
SD932	Previous firmware update result	Target		The start I/O number of the module where the firmware update was executed is stored. • CPU module: 3FFH	S (Initial)	Rn[*1] RnP[*1]
SD933		Execution result		The execution result of the firmware update is stored. • 0001H: Completed successfully • 0100H: Flash ROM error • 0200H: Model mismatched • 0201H: File invalid • 0202H: Combination invalid • 0203H: Firmware update prohibited state • 0300H: Firmware data error	S (Initial)	Rn[*1] RnP[*1]

*1 The CPU module where this function can be used supports these special register areas.
*2 For the RnENCPU, information of the CPU part is displayed.

Latch area

The following is the list of special register areas relating to the latch area.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD940	Stop direction at file change on label specification	Stop direction at file change on label specification	Specify the operation for changing program files or the global label setting file. When the stop direction is given, an error occurs in the corresponding function (setting No.). (On: Stop the function, Off: Do not stop the function (The function continues to operate.))  b0: Data logging setting No.1 b1: Data logging setting No.2 b2: Data logging setting No.3 b3: Data logging setting No.4 b4: Data logging setting No.5 b5: Data logging setting No.6 b6: Data logging setting No.7 b7: Data logging setting No.8 b8: Data logging setting No.9 b9: Data logging setting No.10 b15: Real-time monitor	U	Rn*1 RnSF*1
SD944	Backup function of the CPU module data backup/restoration function	Backup function setting	Set the backup function using the following bit pattern. (Off: Disabled, On: Enabled)  b0: Automatic backup setting by specification of day and time b1: Automatic backup setting by specification of time and day of the week b5: Upper limit value status for the number of the CPU module backup data b10: Retry execution for the automatic backup b15: Automatic backup setting at occurrence of the CPU stop error	U	Rn*1 RnP*1 RnSF*1
SD947		Day and time setting for automatic backup (day)	Store the day when the automatic backup on the specified day and time is executed using the BIN code. • Day (1 to 31)	U	Rn*1 RnP*1 RnSF*1
SD948		Day and time setting for automatic backup (hour)	Store the hour when the automatic backup on the specified day and time is executed using the BIN code. • Hour (0 to 23)	U	Rn*1 RnP*1 RnSF*1
SD949		Day and time setting for automatic backup (minute)	Store the minute when the automatic backup on the specified day and time is executed using the BIN code. • Minute (0 to 59)	U	Rn*1 RnP*1 RnSF*1
SD950		Time and day of the week setting for automatic backup (hour)	Store the hour when the automatic backup on the specified day of the week and time is executed using the BIN code. • Hour (0 to 23)	U	Rn*1 RnP*1 RnSF*1
SD951		Time and day of the week setting for automatic backup (minute)	Store the minute when the automatic backup on the specified day of the week and time is executed using the BIN code. • Minute (0 to 59)	U	Rn*1 RnP*1 RnSF*1
SD952		Time and day of the week setting for automatic backup (day of the week)	Set the day of the week when the automatic backup is executed using the following bit pattern. (Off: Disabled, On: Enabled)  b0: Sunday, b1: Monday, b2: Tuesday, b3: Wednesday, b4: Thursday, b5: Friday, b6: Saturday	U	Rn*1 RnP*1 RnSF*1
SD953		Backup error cause	• The cause of an error that occurred during the CPU module data backup is stored. 0H: No error Other than 0H:For details on the values stored when an error occurs, refer to the list of error codes. (☞ Page 805 List of error codes) • "0" is set at the start of the CPU module data backup.	S (Error)	Rn*1 RnP*1 RnSF*1

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD954	Restoration function of the CPU module data backup/ restoration function	Restoration target data setting	Set the target data to be restored with the CPU module data restoration function. 0: All the target data 1: Device/label data only 2: All the target data except for the device/label data	U	Rn*1 RnP*1 RnSF*1
SD955		Restoration function setting	Set the CPU module data restoration function using the following bit pattern. (Off: Disabled, On: Enabled) b15b14b13 b2 b1 b0 0 b0: Automatic restoration (For the Safety CPU, this bit is set by the system.) b1: Initialization at automatic restoration b2: Automatic restoration with SD CARD OFF button b13: Latest data b14: Special relay and special register b15: Continuous operation with the status at backup	U	Rn*1 RnP*1 RnSF*1
SD956		Restoration target date folder setting	Store the target folder (date folder) of the CPU module data restoration using BCD code. SD957 SD956 b31 b24 b23 b16 b15 b8 b7 b0 (4) (3) (2) (1) (1) Day (1 to 31) (2) Month (1 to 12) (3) Year (last two digits) (0 to 99) (4) Year (first two digits) (0 to 99) [Example] To specify the date folder of June 15 2015, store "H20150615".	U	Rn*1 RnP*1 RnSF*1
SD957				U	Rn*1 RnP*1 RnSF*1
SD958		Restoration target number folder setting	Specify the target folder of the CPU module data restoration. 1 to 32767: Serial number of the backup folder (*****) in a date folder (00001 to 32767)	U	Rn*1 RnP*1 RnSF*1
SD959		Restoration error cause	• The cause of an error that occurred during the CPU module data restoration is stored. 0H: No error Other than 0: For details on the values stored when an error occurs, refer to the list of error codes. (☞ Page 805 List of error codes) • "0" is set at the start of the CPU module data backup.	S (Error)	Rn*1 RnP*1 RnSF*1
SD960	Backup function of the CPU module data backup/restoration function	Upper limit status for the number of CPU module backup data	This register indicates the set value of the upper limit for the number of backup data in accordance with bit 5 of SD944. Bit 5 of SD944 is off: 0 Bit 5 of SD944 is on: 1 to 100	S (Status change)	Rn*1 RnP*1 RnSF*1
SD988	Memory copy completion status (latch)	Memory copy completion status (latch)	This register stores a value indicating the completion status of the memory copy from the control system to the standby system. • The value same as the SD1654 value is stored at the completion or abend of the memory copy from the control system to the standby system. • Since data have been backed up in case of power failure, this register holds the value indicating the latest memory copy completion status from the control system to the standby system. • This register is cleared to 0 by latch clear.	S (Status change)	RnP RnPSF

*1 The CPU module where this function can be used supports these special register areas.

Data logging function

The following is the list of special register areas relating to the data logging function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1200	Data logging Function memory use condition	Data logging Function memory use condition	• The status of the data logging when the function memory is set for the storage location of data logging files are stored in the following bit patterns. b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 b0: Setting No.1 b1: Setting No.2 ⋮ b9: Setting No.10 • Each bit turns on when a relevant data logging starts and turns off when the data logging stops.	S (Status change)	Rn[*1]
SD1203	Data logging file transfer stop information	Data logging file transfer stop information	• To stop data transfer of each data logging setting number, set the relevant setting number with the following bit pattern. (Off: Disabled, On: Enabled) b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 b0: Setting No.1 b1: Setting No.2 ⋮ b9: Setting No.10 • This setting is activated when SM1203 is turned on.	U	Rn[*1]
SD1210	Data logging setting No.1 Latest storage file number	Latest storage file number	• The latest storage file number • This register is cleared to 0 by the stop command from CPU Module Logging Configuration Tool.	S (Status change)	Rn[*1] RnP RnSF
SD1211					Rn[*1] RnP RnSF
SD1212	Data logging setting No.1 Oldest storage file number	Oldest storage file number	• The oldest storage file number • This register is cleared to 0 by the stop command from CPU Module Logging Configuration Tool.	S (Status change)	Rn[*1] RnP RnSF
SD1213					Rn[*1] RnP RnSF
SD1214	Data logging setting No.1 Internal buffer free space	Free space size of the internal buffer (K bytes)	• The amount of free space of the internal buffer is stored in K bytes. The smaller the value, the higher the generating ratio of processing overflow. • For the trigger logging, it refers to the total capacity of internal buffer until the data for the number of records is collected after a trigger occurred. • This register is cleared to 0 by the stop command from CPU Module Logging Configuration Tool.	S (Error)	Rn[*1] RnP RnSF
SD1215	Data logging setting No.1 Number of processing overflow occurrences	Number of processing overflow occurrences	• The number of data logging processing overflow occurrences • When overflow occurs, data is lost. • When the count exceeds 65535, it returns to 0 and starts a new cycle. • With "Stop" set for the operation at the time when the number of save files exceeds the limit, a processing overflow may occur until the stop operation is completed after the collection of data corresponding to the specified number of storage files was completed. • This register is cleared to 0 when the setting is registered or by the stop command from CPU Module Logging Configuration Tool.	S (Error)	Rn[*1] RnP RnSF
SD1216	Data logging setting No.1 Data logging error cause	Data logging error cause	Cause of the error generated during data logging operations is stored. 0: No error Other than 0:For details on the values stored when an error occurs, refer to the list of error codes. (☞ Page 805 List of error codes)	S (Error)	Rn[*1] RnP RnSF

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1217	Data logging setting No.1 Data logging file transfer error cause	Data logging file transfer error cause	An error cause that is detected last in the data logging file transfer function or a transferring of data logging files to the data memory is stored. 0: No error Other than 0:For details on the values stored when an error occurs, refer to the list of error codes. (☞ Page 805 List of error codes) This register is cleared to 0 when data logging is started.	S (Error)	Rn*1
SD1220 to SD1227	Data logging setting No.2	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1217).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SD1230 to SD1237	Data logging setting No.3	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1217).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SD1240 to SD1247	Data logging setting No.4	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1217).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SD1250 to SD1257	Data logging setting No.5	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1217).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SD1260 to SD1267	Data logging setting No.6	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1217).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SD1270 to SD1277	Data logging setting No.7	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1217).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SD1280 to SD1287	Data logging setting No.8	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1217).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SD1290 to SD1297	Data logging setting No.9	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1217).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2
SD1300 to SD1307	Data logging setting No.10	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1217).	Same configuration as the setting No.1	Rn*1 RnP*2 RnSF*2

*1 The CPU module where this function can be used supports these special register areas.

*2 These CPU modules do not support Data logging file transfer error causes for the data logging settings No.2 to No.10 (SD1227, SD1237, SD1247, SD1257, SD1267, SD1277, SD1287, SD1297, SD1307).

CPU module data backup/restoration function

The following is the list of special register areas relating to the CPU module data backup/restoration function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1350	Number of uncompleted folders/ files of CPU module data backup/ restoration	Number of uncompleted folders/ files of CPU module data backup/ restoration	This register indicates the number of folders/files where the backup/restoration of the CPU module is not completed. When the backup/restoration processing is started, the total number of folders and files to be backed up or restored is stored. The number is reduced one each time one folder/file is backed up or restored, and 0 is stored when all the data is backed up or restored.	S (Status change)	Rn*2 RnP*2 RnSF*2
SD1351	Progression status of CPU module data backup/restoration	Progression status of CPU module data backup/restoration	• This register indicates the progression status of the backup or restoration in percentage.*1 Range of the value: 0 to 100 (%) • "0" is set at the start of the CPU module data backup.	S (Status change)	Rn*2 RnP*2 RnSF*2
SD1353	Upper limit value setting for the number of CPU module backup data	Setting of the upper limit value for the number of CPU module backup data	Set the upper limit value for the number of the backup data for the CPU module data backup. (1 to 100)	U	Rn*2 RnP*2 RnSF*2

*1 When program files are restored, the progress in SD1351 stops while data is being written (transferred) to the program memory in the restoration processing because the data is transferred from the program cache memory to the program memory. The progress of data transfer to the program memory can be checked in SD629.

*2 The CPU module where this function can be used supports these special register areas.

Data backup/restoration function of iQ Sensor Solution

The following is the list of special register areas relating to the data backup/restoration function of iQ Sensor Solution.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1360	Right-to-use request number for iQ Sensor Solution data backup/ restoration	Right-to-use request number for data backup/restoration	Set the request number (desired 4-digit number) to acquire/release the right to use. Use the request number as follows depending on the request source. 0000H: Right-to-use release request 1000H to 1FFFH: Request from the program D000H to DFFFH: Request from MELSOFT Navigator E000H to EFFFH: Request from GOT F000H to FFFFH: Request from GX Works3	U	Rn*3 RnSF*3
SD1361	Right-to-use acquisition number for iQ Sensor Solution data backup/ restoration	Right-to-use acquisition number for data backup/ restoration	This register stores the request number of when the right to use for the iQ Sensor Solution data backup/ restoration has been acquired. (0000H is stored when the right to use is not acquired or has been released)	S (Status change)	Rn*3 RnSF*3
SD1362	Target module/ execution unit setting for iQ Sensor Solution data backup/ restoration	Target module/ execution unit setting for data backup/ restoration	Set the target module and execution unit of the iQ Sensor Solution data backup/restoration. • Lower 8 bits (target module) 1H: AnyWireASLINK 2H: CC-Link 3H: Ethernet 4H: CC-Link IE Field Network • Upper 8 bits (execution unit) <AnyWireASLINK> 1H: In module units 2H: In ID units <CC-Link and CC-Link IE Field Network> 1H: In module units 2H: In station units 3H: In station sub ID units <Ethernet> 1H: In module units 2H: In IP address units	U	Rn*3 RnSF*3

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1363	Target folder number setting for iQ Sensor Solution data backup/restoration	Target folder number setting for data backup/restoration	Set a number of folder in which the backup data is to be stored or a folder in which the data to be restored is stored. 00 to 99: Target folder specification FFFEH: Automatic specification (Folder deletion supported) FFFFH (default): Automatic specification	U	Rn*3 RnSF*3
SD1364	Target setting for iQ Sensor Solution data backup/restoration (target module)	Target setting for data backup/restoration (target module)	Set the target module of the iQ Sensor Solution data backup/restoration. I/O number.*1: Module 3FFH: Built-in Ethernet*2	U	Rn*3 RnSF*3
SD1365	Target setting for iQ Sensor Solution data backup/restoration (target device 1)	Target setting for data backup/restoration (target device 1)	Set the target device of the iQ Sensor Solution data backup/restoration. <AnyWireASLINK> ID number <CC-Link and CC-Link IE Field Network> Station number <Ethernet> IP address (lower 16 bits) Example: When the IP address is 192.168.3.40 (expressed in dotted decimal notation*4), 3 = 3H and 40 = 28H. Therefore, the value of the lower 16 bits (0328H) is 808.	U	Rn*3 RnSF*3
SD1366	Target setting for iQ Sensor Solution data backup/restoration (target device 2)	Target setting for data backup/restoration (target device 2)	Set the target device of the iQ Sensor Solution data backup/restoration. <AnyWireASLINK> 0 (Not used) <CC-Link and CC-Link IE Field Network> Station sub-ID number <Ethernet> IP address (upper 16 bits) Example: When the IP address is 192.168.3.40 (expressed in dotted decimal notation*4), 192 = C0H and 168 = A8H. Therefore, the value of the upper 16 bits (C0A8H) is 49320.	U	Rn*3 RnSF*3
SD1367	Operation setting for iQ Sensor Solution data backup/restoration	Operation setting for data backup/restoration	Set the operation of the iQ Sensor Solution data backup/restoration. b15 b8b7 b0 b0 to b7: Operation setting at error occurrence (0: Continue, 1: Stop) b8 to b15: Fixed to 0	U	Rn*3 RnSF*3
SD1368	Communication timeout time of iQ Sensor Solution data backup/restoration	Communication timeout time of data backup/restoration	Specify the communication timeout time of the iQ Sensor Solution data backup/restoration in increments of 100ms. 0: 600000ms (default) 1 to 65535 (×100ms): 100 to 6553500ms	U	Rn*3 RnSF*3
SD1371	Execution status of iQ Sensor Solution data backup/restoration (total number of target devices)	Execution status of data backup/restoration (total number of target devices)	• This register stores the total number of target devices of the data backup/restoration of iQ Sensor Solution for each execution unit. • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change)	Rn*3 RnSF*3
SD1372	Execution status of iQ Sensor Solution data backup/restoration (normal completion count)	Execution status of data backup/restoration (normal completion count)	• This register indicates the execution status (the number of devices where the processing has been completed normally) of the iQ Sensor Solution data backup/restoration. • Among the devices where the operation has been completed, the number of devices where the operation has been completed normally is stored. • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change)	Rn*3 RnSF*3
SD1373	Execution status of iQ Sensor Solution data backup/restoration (error completion count)	Execution status of data backup/restoration (error completion count)	• This register indicates the execution status (the number of devices where the processing has been completed with an error) of the iQ Sensor Solution data backup/restoration. • Among the devices where the operation has been completed, the number of devices where the operation has been completed with an error is stored. • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change)	Rn*3 RnSF*3

A

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1374	Execution status of iQ Sensor Solution data backup/restoration (progress per device)	Execution status of data backup/ restoration (progress per device)	• This register indicates the progression status of the iQ Sensor Solution data backup/restoration of the device in progress in percentage (0 to 100%). Range of the value: 0 to 100 (%) • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change)	Rn*3 RnSF*3
SD1375	Folder number of iQ Sensor Solution data backup	Backup target folder number	• This register stores a number of the target folder where the backup data of the device supporting iQSS is stored. 0 to 99: Folder number FFFFH: Backup data not saved • This register stores FFFFH when the right to use is acquired or the backup/restoration is requested.	S (Status change)	Rn*3 RnSF*3
SD1376	Module error cause of iQ Sensor Solution data backup/ restoration	Module error cause of data backup/ restoration	• This register stores the error cause that is detected in the module during the iQ Sensor Solution data backup/restoration. When errors are detected in multiple devices, the error detected first is stored. (For details on the error cause stored, refer to the manual for the CPU module used or the target device.) • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)	Rn*3 RnSF*3
SD1377	Target device error cause of iQ Sensor Solution data backup/ restoration	Target device error cause of data backup/ restoration	• This register stores the error cause that is detected in the target device during the iQ Sensor Solution data backup/restoration. When errors are detected in multiple devices, the error detected first is stored. (For details on the error cause stored, refer to the manual for the target device.) • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)	Rn*3 RnSF*3
SD1378	Module information/ execution unit information of iQ Sensor Solution data backup/restoration error	Module information/ execution unit information of an error	• At an occurrence of the iQ Sensor Solution data backup/ restoration error, this register stores the information of the module where the error has been occurred and the execution unit information. When errors are detected in multiple devices, the information of a device where the error detected first is stored. • Lower 8 bits (target module) 1H: AnyWireASLINK 2H: CC-Link 3H: Ethernet 4H: CC-Link IE Field Network • Upper 8 bits (execution unit) <AnyWireASLINK> 1H: In module units 2H: In ID units <CC-Link and CC-Link IE Field Network> 1H: In module units 2H: In station units 3H: In station sub ID units <Ethernet> 1H: In module units 2H: In IP address units • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)	Rn*3 RnSF*3
SD1379	Folder number information of iQ Sensor Solution data backup/restoration error	Folder number information of an error	• At an occurrence of the iQ Sensor Solution data backup/ restoration error, this register stores the folder number (0 to 99) of the backup/restoration target. If the folder number cannot be specified or a backup error is detected before creating a backup folder (I/O number_station number), FFFFH is stored. • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)	Rn*3 RnSF*3
SD1380	Details on iQ Sensor Solution data backup/ restoration error (module)	Information of the module where an error has occurred	• This register stores the information of a module where an error of the iQ Sensor Solution data backup/restoration has been occurred. When errors are detected in multiple modules, the information of a module where the error detected first is stored. I/O number.*1: Module 3FFH: Built-in Ethernet*2 • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)	Rn*3 RnSF*3

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1381	Details on iQ Sensor Solution data backup/restoration error (device 1)	Information of the device (device 1) where an error has occurred	• This register stores the information of a device (device 1) where an error of the iQ Sensor Solution data backup/restoration has been occurred. When errors are detected in multiple devices, the information of a device (device 1) where the error detected first is stored. <AnyWireASLINK> ID number <CC-Link and CC-Link IE Field Network> Station number <Ethernet> IP address (lower 16 bits) Example: When the IP address is 192.168.3.40 (expressed in dotted decimal notation*4), 3 = 3H and 40 = 28H. Therefore, the value of the lower 16 bits (0328H) is 808. • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)	Rn*3 RnSF*3
SD1382	Details on iQ Sensor Solution data backup/restoration error (device 2)	Information of the device (device 2) where an error has occurred	• This register stores the information of a device (device 2) where an error of the iQ Sensor Solution data backup/restoration has been occurred. When errors are detected in multiple devices, the information of a device (device 2) where the error detected first is stored. <AnyWireASLINK> 0 (Not used) <CC-Link and CC-Link IE Field Network> Station sub-ID number <Ethernet> IP address (upper 16 bits) Example: When the IP address is 192.168.3.40 (expressed in dotted decimal notation*4), 192 = C0H and 168 = A8H. Therefore, the value of the upper 16 bits (C0A8H) is 49320. • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)	Rn*3 RnSF*3

*1 The start I/O number (first three digits in four-digit hexadecimal representation) of each module
*2 The built-in Ethernet port for the RnENCPU
*3 The CPU module where this function can be used supports these special register areas.
*4 A 32-bit IP address is expressed by converting it into decimal numbers in units of eight bits and separating each with a dot.

A

Interrupt pointer mask pattern

The following is the list of special register areas relating to the mask pattern for interrupt pointers.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1400 to SD1463	Interrupt pointer mask pattern	Mask pattern	The mask pattern for interrupt pointers is stored as shown in the following figure: b15 b1 b0 SD1400 | I15 | to | I1 | I0 SD1401 | I31 | to | I17 | I16 ⋮ SD1463 | I1023 | to | I1009 | I1008	S (at execution)	ALL

Event history function

The following is the list of special relay areas relating to the event history function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1464 to SD1467	Module information on event history logging restriction	Module information on event history logging restriction	Modules on which event history logging is restricted are stored in the following bit patterns. (Off: No event history logging restricted, On: Event history logging restricted) b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SD1464 b0: Slot No.0 b1: Slot No.1 ⋮ b15:Slot No.15 b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SD1465 b0: Slot No.16 b1: Slot No.17 ⋮ b15:Slot No.31 b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SD1466 b0: Slot No.32 b1: Slot No.33 ⋮ b15:Slot No.47 b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SD1467 b0: Slot No.48 b1: Slot No.49 ⋮ b15:Slot No.64 The bits corresponding to the slots in which the target modules are mounted turn on. The bits remain on even after the event history logging restriction is lifted.	S (Status change)	Rn*1 RnP*1 RnSF*1

*1 The CPU module where this function can be used supports these special register areas.

User authentication function

The following is the list of special register area relating to the user authentication function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1468	Number of login users	Number of login users	The number of users logged in to the CPU module is stored. The value is incremented by one every time the login operation to the programmable controller or files is performed. The value is decremented by one every time the logout operation is performed.	S (Status change)	RnSF RnPSF

Memory dump function

The following is the special register area relating to the memory dump function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1472	Memory dump error cause	Memory dump error cause	The cause of the error occurred during the memory dump function is stored. 0: No error Other than 0:For details on the values stored when an error occurs, refer to the list of error codes. (☞ Page 805 List of error codes)	S (upon error)	Rn*1 RnSF*1

*1 The CPU module where this function can be used supports this special register area.

Real-time monitor function

The following is the special register area relating to the real-time monitor function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1484	Real-time monitor internal buffer free space	Free space size of the internal buffer (K bytes)	The amount of free space of the internal buffer is stored in K bytes. The smaller the value, the higher the generating ratio of processing overflow.	S (Status change)	Rn*1 RnSF*1

*1 The CPU module where this function can be used supports this special register area.

Debug function

The following is the special register area relating to the debug function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1488	Debug function usage status	Debug function usage status	The usage status of the debug function is stored using the following bit pattern. b2 b1 b0 b0: External input/output forced on/off*1 Off: Not used On: In use b1: Program restoration information write status*1 Off: All written On: Not all written When SM386 is on (without LED flashing), this bit does not turn on. b2: Registration status of the device test with execution conditions*1 Off: Not registered On: Registered b3 to b15: Empty (fixed to 0)	S (Status change)	Rn RnP RnSF

*1 The CPU module where this function can be used supports this special register area.

CPU module database access function

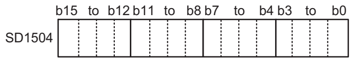
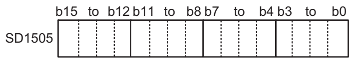
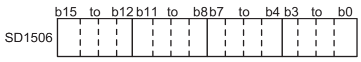
The following is the special register area relating to the CPU module database access function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1498	Start-up status of CPU module database	Start-up status of CPU module database	When the CPU module database access function is enabled with the module parameter, the CPU module database start-up status is stored. 0:No error FFFFH: Starting-up Other than above: Start-up error For the stored value at the start-up error, refer to the list of error codes that occur when a database access instruction is executed. (MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))	S (Status change)	Rn[*1]

*1 The CPU module where this function can be used supports this special register area.

Ethernet function

The following is the list of special register areas relating to the Ethernet function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1504	Open completion signal	In this register, open completion status is stored.	Open statuses of connection No.1 to 16 are stored. (0: Close/Open not completed, 1: Open completed).  b0: Connection 1 b1: Connection 2 b2 to b15: Connection 3 to 16 For details on the on/off timing, refer to the following. MELSEC iQ-R Ethernet User's Manual (Application)	S (Status change)	ALL
SD1505	Open request signal	In this register, open request status is stored.	Open processing statuses of connection No.1 to 16 are stored. (0: No open request, 1: Open request exists).  b0: Connection 1 b1: Connection 2 b2 to b15: Connection 3 to 16 For details on the on/off timing, refer to the following. MELSEC iQ-R Ethernet User's Manual (Application)	S (Status change)	ALL
SD1506	Socket communications receive status signal	In this register, receive status is stored.	Receiving statuses of connections No.1 to 16 are stored. (0: No data received, 1: Data receiving completed)  b0: Connection 1 b1: Connection 2 b2 to b15: Connection 3 to 16 For details on the on/off timing, refer to the following. MELSEC iQ-R Ethernet User's Manual (Application)	S (Status change)	ALL
SD1512	FTP server function file path name switching setting	The detailed of the file path name switching setting of the FTP server function is stored.	The setting details of the file path name switching settings in the FTP server function by executing SM1512 is stored according to the following bit pattern. b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 0 0 0 0 0 0 0 0 (3) 0 0 (2) (1) b0 (1): The directory delimiter of a file path name is set to "/". This delimiter is used for Linux computers. (0: Delimiter "/") b1 (2): The drive number in a file path name is omitted. (0: Not omitted, 1: Omitted) b2 to b3: Empty (fixed to 0) b4 to b7 (3): The target drive number is set when the drive number of the file path name is omitted. (2: SD memory card, 3: Device/label memory, 4: Data memory) When b1 sets to 1, settings in b4 to b7 are enabled. At omitted the drive number, when the CPU module receives a command which is specified the drive number to a file path name received from FTP client, the drive number specified by executing the command is prioritized. b8 to b15: Empty (fixed to 0)	U	Rn*2
SD1513	FTP server function file path name switching setting error cause	An error cause of the file path name switching setting of the FTP server function is stored.	The error cause of the FTP server function file path name switching setting by executing SM1512 is stored. 0H: No error 100H: SD1512 (FTP server function file path name switching setting) is out of the specified range When the FTP command (quote path-delimiter) switching setting is enabled or the switching setting is off, the value is 0H.	S (Status change)	Rn*2

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1514	FTP server function file path name switching setting status	The current setting value of the file path name switching setting of the FTP server function is stored.	The current setting value in SD1512 (FTP server function file path name switching setting) is stored. (This value is also stored when the switching setting is set with the FTP command (quote path-delimiter)). b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 0 0 0 0 0 0 0 0 (3) 0 0 (2) (1) b0 (1): The directory delimiter of the file path name (0: quote path-delimiter) b1 (2): the drive number of the file path name is omitted. (0: Not omitted, 1: Omitted) b4 to b7 (3): The target drive number is set when the drive number of the file path name is omitted. (2: SD memory card, 3: Device/label memory, 4: Data memory) When SM1512 is turned on and off or the switching setting is off by executing the FTP command (quote path-delimiter off), the value is 0H.	S (Status change)	Rn*2
SD1518	IP address setting	Network number	• Set the network number to be stored in the IP address storage area (system memory*1). • Upon completion of writing or clearing the IP address to/from the IP address storage area (system memory*1), the value of the network number stored in the IP address storage area (system memory*1) is stored.	S (Status change)/U	Rn*2
SD1519		Station number	• Set the station number to be stored in the IP address storage area (system memory*1). • Upon completion of writing or clearing the IP address to/from the IP address storage area (system memory*1), the value of the station number stored in the IP address storage area (system memory*1) is stored.	S (Status change)/U	Rn*2
SD1520		IP Address (lower)	• Specify the IP address to be stored into the IP address storage area (system memory*1). Range: 00000001H to DFFFFFFEH (0.0.0.1 to 223.255.255.254) • Upon completion of writing or clearing the IP address to/from the IP address storage area (system memory*1), the value of IP address stored in the IP address storage area (system memory*1) is stored. b15 to b8 / b7 to b0 SD1520: 3 / 4 SD1521: 1 / 2 1 to 4: First to fourth byte	S (Status change)/U	ALL
SD1521		IP Address (upper)			ALL
SD1522		Subnet mask pattern (lower)	• Specify the Subnet mask pattern to be stored into the IP address storage area (system memory*1). Setting range: C0000000H to FFFFFFFCH (192.0.0.0 to 255.255.255.252), 00000000H (no setting) • Upon completion of writing or clearing the IP address to/from the IP address storage area (system memory*1), the value of subnet mask pattern stored in the IP address storage area (system memory*1) is stored. b15 to b8 / b7 to b0 SD1522: 3 / 4 SD1523: 1 / 2 1 to 4: First to fourth byte	S (Status change)/U	ALL
SD1523		Subnet mask pattern (upper)			ALL

<table>
<tr><th>No.</th><th>Name</th><th>Data stored</th><th>Details</th><th>Set by (setting timing)</th><th>CPU</th></tr>
<tr><td>SD1524</td><td rowspan="2">IP address setting</td><td>Default gateway IP address (lower)</td><td rowspan="2">• Specify the default gateway IP address to be stored into the IP address storage area (system memory*1). Setting range: 00000001H to DFFFFFFEH (0.0.0.1 to 223.255.255.254), 00000000H (no setting)
• Upon completion of writing or clearing the IP address to/from the IP address storage area (system memory*1), the value of default gateway IP address stored in the IP address storage area (system memory*1) is stored.
<table><tr><th></th><th>b15 to b8</th><th>b7 to b0</th></tr><tr><td>SD1524</td><td>3</td><td>4</td></tr><tr><td>SD1525</td><td>1</td><td>2</td></tr></table>1 to 4: First to fourth byte</td><td rowspan="2">S (Status change)/U</td><td>ALL</td></tr>
<tr><td>SD1525</td><td>Default gateway IP address (upper)</td><td>ALL</td></tr>
<tr><td>SD1526</td><td>IP address storage area writing error cause</td><td>The error cause for the case when failing in writing into IP address storage area is stored.</td><td>The error cause generated when writing into IP address storage area (system memory*1) will be stored. (Linked with SM1521)
0H: No error
100H:SD1520 to SD1525 are out of the specified range.
200H:Write error
400H:Impossible to execute the write processing because clear processing is in progress.</td><td>S (Status change)</td><td>ALL</td></tr>
<tr><td>SD1527</td><td>IP address storage area clearing error cause</td><td>The cause of the error for the case when failing in clearing IP address storage area is stored.</td><td>The cause of the error generated during writing into IP address storage area (system memory*1) is stored. (Linked with SM1523)
0H: No error
200H:Clear error
400H:Impossible to execute the clear processing because write processing is in progress.</td><td>S (Status change)</td><td>ALL</td></tr>
</table>

*1 This is the memory used by the system when the CPU module is executing functions.

*2 The CPU module where this function can be used supports these special register areas.

A

CC-Link IE Field Network Basic function

The following is the list of special register areas relating to the CC-Link IE Field Network Basic function.

<table>
<tr><th>No.</th><th>Name</th><th>Data stored</th><th>Details</th><th>Set by (setting timing)</th><th>CPU</th></tr>
<tr><td>SD1536 to SD1539</td><td>Cyclic transmission status of each station</td><td>Cyclic transmission status of each station</td><td>The cyclic transmission status of each station is stored using the following bit pattern. (Off: Not performed, On: Being performed)
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SD1536</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SD1537</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SD1538</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SD1539</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table>
The numbers in the figure indicate station numbers.
(Condition)
• Only the bit of the start station number turns on.
• The status is not stored for the reserved stations and the station numbers after the maximum station number.
Use this register as an interlock for cyclic transmission. For details on the interlock program, refer to the following.
CC-Link IE Field Network Basic Reference Manual</td><td>S (Every END)</td><td>Rn[*2]</td></tr>
<tr><td>SD1540 to SD1543</td><td>Data link status of each station</td><td>Data link status of each station</td><td>The data link status of each station is stored using the following bit pattern. (Off: Normally operating station[*1], On: Faulty station)
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SD1540</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SD1541</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SD1542</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SD1543</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table>
The numbers in the figure indicate station numbers.
(Condition)
• Only the bit of the start station number turns on.
• The status is not stored for the reserved stations and the station numbers after the maximum station number.
This register can be used to monitor errors in device stations, connected cables, and a connected hub.</td><td>S (Every END)</td><td>Rn[*2]</td></tr>
</table>

*1 This status includes the case where a device station has not responded to the first request from the master station due to a power-off of the device station. (The device station is not judged as a faulty station because the data link status is not determined.)

*2 The CPU module where this function can be used supports these special register areas.

Online module change function

The following is the list of special register areas relating to the online module change function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1600	Module selection (base unit No.)	Base unit No. where the online change target module is mounted	The base unit number where the online change target module is mounted is specified. 0: Main base unit 1: Extension base unit 1 ⋮ 7: Extension base unit 7 FFFFH: Not specified (Default) The value returns to its default (FFFFH) upon completion of the online module change processing. When changing a module directly, the system stores the corresponding base unit number upon removal of the target module.	S (Status change)/U (Request)	RnP RnPSF
SD1601	Module selection (slot No.)	Slot No. where the online change target module is mounted	The slot number where the online change target module is mounted is specified. 0: Slot No.0 1: Slot No.1 ⋮ 11: Slot No.11 FFFFH: Not specified (Default) The value returns to its default (FFFFH) upon completion of the online module change processing. When changing a module directly, the system stores the corresponding slot number upon removal of the target module.	S (Status change)/U (Request)	RnP RnPSF
SD1602	I/O No. of the module being changed online	I/O No. of the module being changed online	The value of the I/O number of the module being changed online divided by 16 is stored. Other than FFFFH: I/O No. ÷ 16 FFFFH: Not specified (Default) The value returns to its default (FFFFH) upon completion of the online module change processing.	S (Status change)	RnP RnPSF
SD1617	Online module change progress status	Online module change progress status	The online module change progress status is stored. 0: Normal operation 1: Module being selected 2: Module selected 3: Module removal requested 4: Module removal ready 5: Module removed 6: Module mounted 7: Module being recognized 8: Module recognized 9: Module control resumed The value becomes 0 upon completion of the online module change processing.	S (Status change)	RnP RnPSF
SD1618	Online module change error code	0: Normal operation Other than 0: Error code	The corresponding error code is stored when an error is detected during online module change. For the value stored, refer to the list of error codes. (☞ Page 902 Error codes related to the online module change function) This register is cleared to 0 when the error cause is eliminated and the online module change related request is executed. However, the error code needs to be cleared to 0 before module selection since this register is not cleared to 0 if an error occurs in selecting a module.	S (Status change)/U (Request)	RnP RnPSF
SD1619	Disable request error code during online module change	0: Normal operation Other than 0: Error code	The corresponding error code is stored when a disable request is executed during online module change. The error code is cleared to 0 when the error cause is eliminated and the online module change related request is executed. For the value stored, refer to the list of error codes. (☞ Page 902 Error codes related to the online module change function)	S (Status change)	RnP RnPSF

A

System information (Process CPU)

The following is the list of special register areas relating to the system information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1622	Process CPU operation mode	Process CPU operation mode	This register stores the operation mode of the Process CPU 80H: Process mode 81H: Redundant mode	S (Initial)	RnP

Redundant function

The following is the list of special register areas relating to the redundant function.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1642	BACKUP/SEPARATE LED flashing cause	BACKUP/SEPARATE LED flashing cause	Flashing cause of BACKUP/SEPARATE LED is stored using the following bit pattern. b15 b14 ··· b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 Corresponding bits to each cause turns on. b1: Tracking communications disabled b3: Stop error of the standby system b4: Operating status mismatch between both systems b5: Memory copy being executed b6: Online change being executed b7: A failure of a network module detected on the standby system b8: System switching being executed b9: A redundant function module being changed online b10: System switching disabled on the standby system by using the DCONTSW instruction b11: Online module change being executed on a main base unit in a redundant system with redundant extension base unit b14: Safety operation mode mismatch between both systems b15: A failure of safety tracking data detected • 0 is stored as initial value. • When the multiple causes occurred at the same time, bits corresponding to each cause turn on simultaneously. If the tracking communications are disabled or the redundant function module is being changed online, the other causes turn off. • When the redundant function module in the other system is being changed online, "Tracking communications disabled" is stored in own system. • Even if a cause of the system switching failure has occurred, the system is switched when the system switching causes which have higher priority, such as power-off or reset of the CPU module, have occurred. (☞ Page 506 Execution availability of system switching)	S (Status change)	RnP RnPSF
SD1643	System switching cause	System switching cause occurred in own system (normal/abnormal systems witching)	Stores the system switching cause occurred in own system. The system switching cause is stored in this register even if the systems cannot be switched by a cause of system switching failure. This register is initialized with 0 when the CPU module is powered off and on or is reset. 0: Initial value (the control system never been switched) 1: Power-off, reset, hardware failure 2: Stop error 3: System switching request from a network module 16: System switching request by using the SP.CONTSW instruction 17: System switching request using an engineering tool When the system is switched upon the power-off or reset of the CPU module in the control system, 1 is not stored in SD1643 of the new standby system.	S (At system switching)	RnP RnPSF

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1644	Cause of system switching failure	Cause number of system switching failure	• If a system switching is failed because a cause of system switching failure has occurred, either of the following values is stored in this register. 0: System switching completed normally (default) 1: Tracking communications disabled 2: Tracking communication timeout 3: Stop error of the standby system 4: Operating status mismatch between both systems 5: Memory copy being executed 6: Online change being executed 7: A failure of a network module detected on the standby system 8: System switching being executed 9: A redundant function module being changed online or restarted 10: System switching disabled on the standby system by using the DCONTSW instruction 11: Online module change being executed on a main base unit in a redundant system with redundant extension base unit 14: Safety operation mode mismatch between both systems 15: A failure of safety tracking data detected • This register is initialized with 0 when own system is powered on. • Zero is stored in this register upon completion of system switching.	S (At system switching)	RnP RnPSF
SD1645	System switching request status from a network module of own system	System switching request status from a network module of own system	• The system switching request status from a network module of the own system is stored using the following bit pattern. (The bits of the positions where no modules are mounted turn off.) b11 ··· b0 b0 to b11: Slot No.0 to slot No.11 • The system turns off the bit after the error in the corresponding module is removed by a user. • For the system switching request status from a network module of the other system, refer to SD1646.	S (Error/Status change)	RnP RnPSF
SD1646	System switching request status from a network module of the other system	System switching request status from a network module of the other system	• The system switching request status from a network module of the other system is stored using the following bit pattern. (The bits of the positions where no modules are mounted turn off.) b11 ··· b0 b0 to b11: Slot No.0 to slot No.11 • The system turns off the bit after the error in the corresponding module is removed by a user. • For the system switching request status from a network module of the own system, refer to SD1645.	S (Every END)	RnP RnPSF
SD1648	Cause of the other system monitoring error	Cause of the other system monitoring error	• When an error has occurred on the communications with the other system during an initial processing (including when the system is determined while waiting for the other system starts up) or an END processing, either of the following bits turns on. Once the error is cleared, the bit turns off. b15 b2 b1 b0 b0: Communications with the other system disabled (other than a redundant function module hardware failure in own system) b1: Power-off, reset, or hardware failure (the CPU module or the redundant function module) in the other system b2: Stop error in the other system b15: Communications with the other system disabled (a redundant function module hardware failure in own system) • If either of b0, b1, b2 or b15 is on, the other bits are off. • When the communications with the other system is disabled at an initial processing (including when the other system is powered off), b0 turns on. (When power-off or reset of the CPU module in the other system is recognized, b1 turns on.) • The tracking cable status of when one of the above mentioned bits of this register is on can be checked in SD1755.	S (Initial/every END)/system switching	RnP RnPSF

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1649	System switching cause (when the systems are successfully switched)	System switching cause (when the systems are successfully switched)	System switching cause is stored. • System switching cause is stored in SD1649 of both systems when the system is switched. • This register is initialized with 0 when the CPU module is powered off and on or is reset. • Either of the following is stored in this register. 0: Initial value (the control system never been switched) 1: Power-off, reset, hardware failure 2: Stop error 3: System switching request from a network module 16: System switching request by using the SP.CONTSW instruction 17: System switching request using an engineering tool Note that 1 is not stored in SD1649 of the new standby system when the system is switched by the power-off or reset of the CPU module in the control system.	S (At system switching)	RnP RnPSF
SD1650	System switching instruction ID number	System switching instruction ID number	• This register stores the argument (system switching instruction ID number) of the executed SP.CONTSW instruction when the system is switched by using the SP.CONTSW instruction. (The argument is stored in SD1650 of both systems at the system switching.) • This register is valid only while 16 (System switching request by using the SP.CONTSW instruction) is stored in SD1649. • This register is updated only when the system is switched by the SP.CONTSW instruction. • The initial value is 0.	S (At system switching)	RnP RnPSF
SD1653	Memory copy destination I/O number	Memory copy destination I/O number	• Before SM1653 is turned off and on, the I/O number of the memory copy destination (CPU module in the standby system: 03D1H) is stored. • The initial value is 0.	U	RnP RnPSF
SD1654	Memory copy completion status	Memory copy completion status	This register stores the execution status of the memory copy. When the value other than 0 is stored, the memory copy is completed with an error or cannot be executed. 0H: Completed normally Other than 0H: Completed with an error or cannot be executed. For details on the values stored when an error occurs, refer to the error codes. (Page 805 List of error codes)	S (Status change)	RnP RnPSF
SD1662	Tracking transfer data receive completion wait time	Value to be added to the tracking transfer data receive completion wait time	• Specify the value to be added to waiting time for completion of tracking data reception for the CPU module in the control system. • Specify the value in units of ms. (Range is 0 to 2100ms.) If the value out of the range is specified, the system recognized the value as 2100ms. • The initial value is 0ms.	U	RnP RnPSF
SD1664	Tracking transfer error count	Tracking transfer error count	• For the tracking transfer of the device/label data at each scan, the number is added (+1) when the tracking transfer cannot be executed because of disconnection of the tracking cables, power-off, reset, stop error of the CPU module in the standby system, or error in the redundant function module. • A counting cycle from 0 to 65535 to 0 is repeated.	S (upon error)	RnP RnPSF
SD1667 to SD1670	Tracking transfer trigger	Off: No trigger On: Trigger	• When data is transferred based on the tracking transfer setting of redundant settings, specify the target block as trigger. ■SD1667 b0 to b15: Block 1 to block 16 ■SD1668 b0 to b15: Block 17 to block 32 ■SD1669 b0 to b15: Block 33 to block 48 ■SD1670 b0 to b15: Block 49 to block 64 • When "Transfer Automatically" is selected for the tracking block No.1 auto transfer setting, b0 of SD1667 is turned on by the system at power-on or when the CPU module is switched from STOP to RUN. In other cases, turn on bit0 of SD1667 to b15 of SD1670 by the user. • Initial values for b0 of SD1667 to b15 of SD1670 are off (no trigger)	S (Initial)/U	RnP RnPSF

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1673 to SD1676	Tracking transfer completion status	Off: Transfer not completed On: Transfer completed	• This flag is stored a result of the tracking transfer operated in the preceding END processing. • The flag turns on if any of the tracking transfers in the corresponding blocks has been normally completed, and turns off if the tracking transfers failed due to causes such as tracking communication error. ■SD1673 b0 to b15: Block 1 to block 16 ■SD1674 b0 to b15: Block 17 to block 32 ■SD1675 b0 to b15: Block 33 to block 48 ■SD1676 b0 to b15: Block 49 to block 64	S (Status change)	RnP RnPSF
SD1680	CPU module operation information (the other system)	CPU module operation information (the other system)	This register stores the operating status of the CPU module in the other system in the following bit pattern. When communications with the other system are disabled, 00FFH is stored. b7 … b4 b3 … b0 b0 to b3: 0H: RUN, 2H: STOP, 3H: PAUSE, FH: Communications with the other system disabled b4 to b7: 0H: No error, 1H: Continuation error, 2H: Stop error, FH: Communications with other systems disabled Communications with the other system are disabled in the following states. • The CPU module in the other system is powered off or is being reset. • A hardware failure has occurred in own or the other system. • Tracking cables are not connected or disconnected.	S (Initial/every END)/system switching	RnP RnPSF
SD1681	Latest self-diagnostic error code (the other system)	Latest error code (the other system)	This register stores an error code for the error occurred in the other system with hexadecimal. SD0 of the CPU module in the other system is reflected.	S (Every END)	RnP RnPSF
SD1682 to SD1688	Latest self-diagnostic error time (the other system)	Latest self-diagnostic error time (the other system)	• This register stores the error time in the other system. • Data configuration is the same as SD1 to SD7. • The values in SD1 to SD7 of the CPU module in the other system are reflected.	S (Every END)	RnP RnPSF
SD1689	Detailed information 1 information category (the other system)	Information category code of the detailed information 1 (the other system)	• This register stores the category code of the detailed information 1 for the error that has occurred in the other system. • Data configuration is the same as SD80. • The value in SD80 of the CPU module in the other system is reflected.	S (Every END)	RnP RnPSF
SD1690 to SD1720	Detailed information 1 (the other system)	Detailed information 1 (the other system)	• This register stores the detailed information 1 corresponding to the error that has occurred in the other system. • Data configuration is the same as SD81 to SD111. • The values in SD81 to SD111 of the CPU module in the other system are reflected.	S (Every END)	RnP RnPSF
SD1721	Detailed information 2 information category (the other system)	Information category code of the detailed information 2 (the other system)	• This register stores the category code of the detailed information 2 for the error that has occurred in the other system. • Data configuration is the same as SD112. • The value in SD112 of the CPU module in the other system is reflected.	S (Every END)	RnP RnPSF
SD1722 to SD1752	Detailed information 2 (the other system)	Detailed information 2 (the other system)	• This register stores the detailed information 2 corresponding to the error that has occurred in the other system. • Data configuration is the same as SD113 to SD143. • The values in SD113 to SD143 of the CPU module in the other system are reflected.	S (Every END)	RnP RnPSF

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1754	Cause of control system start-up	Cause of start-up as the control system	This register stores the cause that one of the redundant system has been started up as the control system. • This register is initialized with 0 when the CPU module is powered off and on or is reset. b15 … b12 b3 … b0 b0 to b3: Cause of start-up by specific operations to start up the system as the control system 0: No specific operation, 1: Switch operation (RUN→STOP→RUN), 2: Online operation, 4: Contact input (X) b12 to b15: Information when the system is started up as the control system 0: Before start-up, 1: Start-up as the control system by starting up the other system, 2: Start-up as the control system by the specific operation	S (Status change)	RnP RnPSF
SD1755	Tracking cable connection status	Tracking cable connection status	• This register stores the tracking cable connection status when an initial processing (including when the system is determined while waiting for the other system starts up) or an END processing is performed. 0H: Normal 12H: Cable disconnection of the OUT side 13H: Cable insertion error of the OUT side 14H: Communications on the OUT side being established 21H: Cable disconnection of the IN side 22H: Cable disconnection of the IN or OUT side 23H: Cable disconnection of the IN side or cable insertion error of the OUT side 24H: Cable disconnection of the IN side or communications on the OUT side being established 31H: Cable insertion error of the IN side 32H: Cable insertion error of the IN side or cable disconnection of the OUT side 33H: Cable insertion error of the IN or OUT side 34H: Cable insertion error of the IN side or communications on the OUT side being established 41H: Communications on the IN side being established 42H: Communications on the IN side being established or cable disconnection of the OUT side 43H: Communications on the IN side being established or cable insertion error of the OUT side 44H: Communications on the IN or OUT side being established • The tracking cable status can be checked with this register when bit of SD1648 turns on.	S (Initial/every END)	RnP RnPSF
SD1756	Module information on wait timeout for receiving cyclic data after system switching	Timeout occurrence status (bit pattern) Off: No timeout On: Timeout	• When the receipt of the cyclic data after system switching is not completed within the cyclic data receipt waiting time*2 while the setting to wait cyclic data receive after system switching is enabled, this register turns on the bit corresponding to the slot on the main base unit on which a timeout has occurred. (The bits of the positions where no modules are mounted turn off.) • When a timeout does not occur, this register turns off the corresponding bit. b11 … b0 b0 to b11: Slot No.0 to slot No.11	S (At system switching)	RnP*1 RnPSF*1

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1760	Extension cable connection status	Extension cable connection status (bit pattern) Off: Not connected/ extension cable error/connected improperly On: Connected properly	• In a redundant configuration of extension cables, the connection status between each redundant extension base unit (OUT1/ OUT2) in the 1st to 6th extension levels and each next lower level are stored using the following bit pattern. b15 b8 b7 b0 b1 to b6: 1st to 6th extension base units (OUT1) b9 to b14: 1st to 6th extension base units (OUT2) b0, b7, b8, b15: Not used • In a redundant configuration of extension cables, the status is stored in the CPU module of the control system. • If an extension base unit cannot communicate with the CPU module due to an extension cable failure, the bits corresponding to the extension base unit and ones in the lower levels will turn off. • When the systems are switched, the CPU module of the new control system turns off all bits. • In the 2nd and later extension levels, if redundant extension base units and other extension base units are used together, the bits corresponding to extension base units in the lower levels than the redundant extension base unit will turn off.	S (Initial/every END/system switching)	RnP*1
SD1761	Extension cable route information	Extension cable route information (bit pattern) Off: Inactive On: Active	• In a redundant configuration of extension cables, the route status (active/inactive) between each redundant extension base unit (OUT1/OUT2) in the 1st to 6th extension levels and each next lower level are stored using the following bit pattern. b15 b8 b7 b0 b1 to b6: 1st to 6th extension base units (OUT1) b9 to b14: 1st to 6th extension base units (OUT2) b0, b7, b8, b15: Not used • In a redundant configuration of extension cables, the status is stored in the CPU module of the control system. • When a bit of SD1760 is off, the corresponding bit of SD1761 is also off. • When the systems are switched, the CPU module of the new control system turns off all bits.	S (Initial/every END/system switching)	RnP*1

*1 The CPU module where this function can be used supports these special register areas.

*2 ☞ Page 1087 Waiting time for cyclic data receive after system switching (Twcyc)

A

Safety information

The following is the list of special register areas relating to the safety information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1840	Continuous RUN time in TEST MODE	Continuous RUN time in TEST MODE (L)	Continuous RUN time in TEST MODE is stored. (This register stores measurement values within the range from 1 to 2147483647 (measured in increments of 1 second).) However, measurement starts when the operating status of the CPU module is switched from STOP to RUN. (The time is not measured while the CPU module in the STOP state.) Note that measurement continues even if the time limit is exceeded. After stopping measurement, the register is cleared to 0 when the measurement value is cleared. (☞ Page 628 Measuring the continuous RUN time in TEST MODE)	S (Every END)	RnSF
SD1841		Continuous RUN time in TEST MODE (H)			RnSF
SD1844	Number of points assigned to safety bit devices	SA\X (L)	The number of points assigned to the SA\X device is stored in 32 bits.	S (Initial)	RnPSF RnSF
SD1845		SA\X (H)			RnPSF RnSF
SD1846		SA\Y (L)	The number of points assigned to the SA\Y device is stored in 32 bits.	S (Initial)	RnPSF RnSF
SD1847		SA\Y (H)			RnPSF RnSF
SD1848		SA\M (L)	The number of points assigned to the SA\M device is stored in 32 bits.	S (Initial)	RnPSF RnSF
SD1849		SA\M (H)			RnPSF RnSF
SD1850		SA\B (L)	The number of points assigned to the SA\B device is stored in 32 bits.	S (Initial)	RnPSF RnSF
SD1851		SA\B (H)			RnPSF RnSF
SD1864	Number of points assigned to safety word devices	SA\D (L)	The number of points assigned to the SA\D device is stored in 32 bits.	S (Initial)	RnPSF RnSF
SD1865		SA\D (H)			RnPSF RnSF
SD1866		SA\W (L)	The number of points assigned to the SA\W device is stored in 32 bits.	S (Initial)	RnPSF RnSF
SD1867		SA\W (H)			RnPSF RnSF
SD1872	Number of points assigned to safety timer/counter devices	SA\T (L)	The number of points assigned to the SA\T device is stored in 32 bits.	S (Initial)	RnPSF RnSF
SD1873		SA\T (H)			RnPSF RnSF
SD1874		SA\ST (L)	The number of points assigned to the SA\ST device is stored in 32 bits.	S (Initial)	RnPSF RnSF
SD1875		SA\ST (H)			RnPSF RnSF
SD1876		SA\C (L)	The number of points assigned to the SA\C device is stored in 32 bits.	S (Initial)	RnPSF RnSF
SD1877		SA\C (H)			RnPSF RnSF
SD1888	Safety cycle processing time execution cycle error count	0: No safety cycle processing time execution cycle error occurred (normal) 1 to 65535: Cumulative number of execution cycle errors during safety cycle processing time	The number of times that cycles for which safety cycle processing was not executed were detected is stored.	S (Status change)	RnPSF RnSF

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD1890	Current safety cycle processing time	Current safety cycle processing time (unit: ms)	• The current safety cycle processing time is stored (it is measured in increments of μs). SD1890: The ms part is stored. (Range: 0 to 65535) SD1891: The μs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.	S (Status change)	RnPSF RnSF
SD1891		Current safety cycle processing time (unit: μs)			RnPSF RnSF
SD1892	Minimum safety cycle processing time	Minimum safety cycle processing time (unit: ms)	• The minimum safety cycle processing time is stored (it is measured in increments of μs). SD1892: The ms part is stored. (Range: 0 to 65535) SD1893: The μs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.	S (Status change)	RnPSF RnSF
SD1893		Minimum safety cycle processing time (unit: μs)			RnPSF RnSF
SD1894	Maximum safety cycle processing time	Maximum safety cycle processing time (unit: ms)	• The maximum safety cycle processing time is stored (it is measured in increments of μs). SD1894: The ms part is stored. (Range: 0 to 65535) SD1895: The μs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.	S (Status change)	RnPSF RnSF
SD1895		Maximum safety cycle processing time (unit: μs)			RnPSF RnSF
SD1903	Total standard/safety shared label usage capacity	Total standard/safety shared label usage capacity (unit: words)	The total usage capacity for labels defined in the standard/safety shared label setting file is stored in 4-word units.	S (Status change)	RnPSF RnSF
SD1904	Start I/O number of Safety communication target station (1st module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)	RnPSF RnSF
SD1905	Start I/O number of Safety communication target station (2nd module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)	RnPSF RnSF
SD1906	Start I/O number of Safety communication target station (3rd module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)	RnPSF RnSF
SD1907	Start I/O number of Safety communication target station (4th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)	RnPSF RnSF
SD1908	Start I/O number of Safety communication target station (5th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)	RnPSF RnSF
SD1909	Start I/O number of Safety communication target station (6th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)	RnPSF RnSF
SD1910	Start I/O number of Safety communication target station (7th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)	RnPSF RnSF
SD1911	Start I/O number of Safety communication target station (8th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)	RnPSF RnSF

System monitoring information

The following is the list of special register areas relating to the system monitoring information.

Point

These special register areas can be written only by the system.
The stored values, however, can be read by monitoring systems or SLMP-compatible devices. When reading a value using SLMP commands, use the following commands: 0401 (Read), 0403 (Read Random), and 0406 (Read Block). Do not use subcommands 008□ (Read by device extension specification).
Also, do not use the following to read the stored values (if used, "-1" is read):

- Programs, the circuit monitor function, or the watch function
- Index modification, indirect specification, or bit number specification

<table>
<tr><th>No.</th><th>Name</th><th>Data stored</th><th>Details</th><th>Set by (setting timing)</th><th>CPU</th></tr>
<tr><td>SD2000</td><td>Firmware version (monitor)</td><td>Firmware version</td><td>This register reads the firmware version.</td><td>S (Initial)</td><td>Rn*1</td></tr>
<tr><td>SD2001 to SD2008</td><td>Production information (monitor)</td><td>Production information</td><td>This register reads the production information.
<table>
<tr><td></td><td>b15 ··· b8</td><td>b7 ··· b0</td></tr>
<tr><td>SD2001</td><td>2nd digit from the left</td><td>1st digit from the left</td></tr>
<tr><td>SD2002</td><td>4th digit from the left</td><td>3rd digit from the left</td></tr>
<tr><td></td><td>⋮</td><td></td></tr>
<tr><td>SD2008</td><td>16th digit from the left</td><td>15th digit from the left</td></tr>
</table>
Example: The production information is 123456789ABCDEFG.
<table>
<tr><td></td><td>b15 ··· b8</td><td>b7 ··· b0</td></tr>
<tr><td>SD2001</td><td>32H (2)</td><td>31H (1)</td></tr>
<tr><td>SD2002</td><td>34H (4)</td><td>33H (3)</td></tr>
<tr><td></td><td>⋮</td><td></td></tr>
<tr><td>SD2008</td><td>47H (G)</td><td>46H (F)</td></tr>
</table></td><td>S (Initial)</td><td>Rn*1</td></tr>
<tr><td>SD2009</td><td rowspan="2">IP address (monitor)</td><td>IP address (lower)</td><td rowspan="2">This register reads the IP address of the built-in Ethernet port which is set with parameters or the IP address change function.
<table>
<tr><td></td><td>b15 ··· b8</td><td>b7 ··· b0</td></tr>
<tr><td>SD2009</td><td>3rd byte</td><td>4th byte</td></tr>
<tr><td>SD2010</td><td>1st byte</td><td>2nd byte</td></tr>
</table>
Example: The IP address is 192.168.3.40 (C0A80328H).
<table>
<tr><td></td><td>b15 ··· b8</td><td>b7 ··· b0</td></tr>
<tr><td>SD2009</td><td>03H (3)</td><td>28H (40)</td></tr>
<tr><td>SD2010</td><td>C0H (192)</td><td>A8H (168)</td></tr>
</table></td><td rowspan="2">S (Initial)</td><td rowspan="2">Rn*1</td></tr>
<tr><td>SD2010</td><td>IP address (upper)</td></tr>
<tr><td>SD2011</td><td rowspan="2">Subnet mask pattern (monitor)</td><td>Subnet mask pattern (lower)</td><td rowspan="2">This register reads the subnet mask pattern of the built-in Ethernet port which is set with parameters or the IP address change function.
<table>
<tr><td></td><td>b15 ··· b8</td><td>b7 ··· b0</td></tr>
<tr><td>SD2011</td><td>3rd byte</td><td>4th byte</td></tr>
<tr><td>SD2012</td><td>1st byte</td><td>2nd byte</td></tr>
</table>
Example: The subnet mask pattern is 255.255.255.0 (FFFFFF00H).
<table>
<tr><td></td><td>b15 ··· b8</td><td>b7 ··· b0</td></tr>
<tr><td>SD2011</td><td>FFH (255)</td><td>00H (0)</td></tr>
<tr><td>SD2012</td><td>FFH (255)</td><td>FFH (255)</td></tr>
</table></td><td rowspan="2">S (Initial)</td><td rowspan="2">Rn*1</td></tr>
<tr><td>SD2012</td><td>Subnet mask pattern (upper)</td></tr>
</table>

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SD2013	Default gateway IP address (monitor)	Default gateway IP address (lower)	This register reads the default gateway IP address of the built-in Ethernet port which is set with parameters or the IP address change function. b15 ⋯ b8b7 ⋯ b0 SD2013: 3rd byte | 4th byte SD2014: 1st byte | 2nd byte Example: The default gateway IP address is 192.168.3.254 (C0A803FEH). b15 ⋯ b8b7 ⋯ b0 SD2013: 03H (3) | FEH (254) SD2014: C0H (192) | A8H (168)	S (Initial)	Rn[*1]
SD2014		Default gateway IP address (upper)			
SD2015	MAC address (monitor)	MAC address (5th octet and 6th octet)	This register reads the MAC address. b15 ⋯ b8b7 ⋯ b0 SD2015: 5th octet | 6th octet SD2016: 3rd octet | 4th octet SD2017: 1st octet | 2nd octet Example: The MAC address is 123456789ABC. b15 ⋯ b8b7 ⋯ b0 SD2015: 9AH | BCH SD2016: 56H | 78H SD2017: 12H | 34H	S (Initial)	Rn[*1]
SD2016		MAC address (3rd octet and 4th octet)			
SD2017		MAC address (1st octet and 2nd octet)			
SD2018	Network number (monitor)	Network number	This register reads the network number of the built-in Ethernet port which is set with parameters or the IP address change function. 0: Not set 1 to 239: Network number	S (Initial)	Rn[*1]
SD2019	Station number (monitor)	Station number	This register reads the station number of the built-in Ethernet port which is set with parameters or the IP address change function. 0: Not set 1 to 120: Station number	S (Initial)	Rn[*1]
SD2020	Firmware hash value (monitor)	Firmware hash value (3rd and 4th bytes)	This register reads the firmware hash value. b15 ⋯ b8b7 ⋯ b0 SD2020: 3rd byte | 4th byte SD2021: 1st byte | 2nd byte Example: The firmware hash value is 12345678H. b15 ⋯ b8b7 ⋯ b0 SD2020: 56H | 78H SD2021: 12H | 34H	S (Status change)	Rn[*1]
SD2021		Firmware hash value (1st and 2nd bytes)		S (Status change)	Rn[*1]
SD2022	Parameter file(s) hash value (monitor)	Parameter file(s) hash value (3rd and 4th bytes)	This register reads the parameter file(s) hash value calculated from the following files: • System parameter file • CPU parameter file • Module parameter file b15 ⋯ b8b7 ⋯ b0 SD2022: 3rd byte | 4th byte SD2023: 1st byte | 2nd byte Example: The parameter file(s) hash value is 12345678H. b15 ⋯ b8b7 ⋯ b0 SD2022: 56H | 78H SD2023: 12H | 34H	S (Status change)	Rn[*1]
SD2023		Parameter file(s) hash value (1st and 2nd bytes)		S (Status change)	Rn[*1]

A

<table>
<tr><th>No.</th><th>Name</th><th>Data stored</th><th>Details</th><th>Set by (setting timing)</th><th>CPU</th></tr>
<tr><td>SD2024</td><td rowspan="2">Program file(s) and global label setting file hash value (monitor)</td><td>Program file(s) and global label setting file hash value (3rd and 4th bytes)</td><td rowspan="2">This register reads the parameter file(s) hash value calculated from the following files:
• Program files (All the files set in the program settings of the CPU parameter)
• FB files (All the files set in the FB/FUN file settings of the CPU parameter)
• Global label setting file

b15 ··· b8b7 ··· b0
SD2024: 3rd byte | 4th byte
SD2025: 1st byte | 2nd byte

Example: The program file(s) and global label setting file hash value is 12345678.

b15 ··· b8b7 ··· b0
SD2024: 56H | 78H
SD2025: 12H | 34H</td><td>S (Status change)</td><td>Rn*1</td></tr>
<tr><td>SD2025</td><td>Program file(s) and global label setting file hash value (1st and 2nd bytes)</td><td>S (Status change)</td><td>Rn*1</td></tr>
<tr><td>SD2026</td><td>CPU operating status (monitor)</td><td>Operating status of the CPU module</td><td>This register reads the operating status of the CPU module.
0: RUN
2: STOP
3: PAUSE</td><td>S (Status change)</td><td>Rn*1</td></tr>
<tr><td>SD2027</td><td rowspan="2">Initial scan time (monitor)</td><td>Initial scan time (unit: ms)</td><td rowspan="2">• This register reads the initial scan time.
SD2027: The number of ms is read. (Range: 0 to 65535)
SD2028: The number of μs is read. (Range: 0 to 999)
• These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN.</td><td>S (Status change)</td><td>Rn*1</td></tr>
<tr><td>SD2028</td><td>Initial scan time (unit: μs)</td><td>S (Status change)</td><td>Rn*1</td></tr>
<tr><td>SD2029</td><td rowspan="2">Current scan time (monitor)</td><td>Current scan time (unit: ms)</td><td rowspan="2">• This register reads the current scan time.
SD2029: The number of ms is read. (Range: 0 to 65535)
SD2030: The number of μs is read. (Range: 0 to 999)
Example:If the current scan time is 23.6ms, the following values are read:
SD2029 = 23
SD2030 = 600
• These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN.</td><td>S (Status change)</td><td>Rn*1</td></tr>
<tr><td>SD2030</td><td>Current scan time (unit: μs)</td><td>S (Status change)</td><td>Rn*1</td></tr>
<tr><td>SD2031</td><td rowspan="2">Minimum scan time (monitor)</td><td>Minimum scan time (unit: ms)</td><td rowspan="2">• The minimum value of the scan times except the scan time of the initial execution program is read.
SD2031: The number of ms is read. (Range: 0 to 65535)
SD2032: The number of μs is read. (Range: 0 to 999)
• These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN.</td><td>S (Status change)</td><td>Rn*1</td></tr>
<tr><td>SD2032</td><td>Minimum scan time (unit: μs)</td><td>S (Status change)</td><td>Rn*1</td></tr>
<tr><td>SD2033</td><td rowspan="2">Maximum scan time (monitor)</td><td>Maximum scan time (unit: ms)</td><td rowspan="2">• The maximum value of the scan times except the scan time of the initial execution program is read.
SD2033: The number of ms is read. (Range: 0 to 65535)
SD2034: The number of μs is read. (Range: 0 to 999)
• These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN.</td><td>S (Status change)</td><td>Rn*1</td></tr>
<tr><td>SD2034</td><td>Maximum scan time (unit: μs)</td><td>S (Status change)</td><td>Rn*1</td></tr>
<tr><td>SD2035</td><td rowspan="2">Scan program execution time (monitor)</td><td>Scan program execution time (unit: ms)</td><td rowspan="2">• The execution time of the scan program in a scan is read.
SD2035: The number of ms is read. (Range: 0 to 65535)
SD2036: The number of μs is read. (Range: 0 to 999)
• These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN.</td><td>S (Status change)</td><td>Rn*1</td></tr>
<tr><td>SD2036</td><td>Scan program execution time (unit: μs)</td><td>S (Status change)</td><td>Rn*1</td></tr>
</table>

*1 There are restrictions on the firmware version of the supported CPU module. (☞ Page 1139 Added and Enhanced Functions)

Appendix 6 List of Safety Special Relay Areas

The following table lists items in the list.

Item	Description
No.	Safety special relay number
Name	Safety special relay name
Description	Data stored in the safety special relay and its meaning
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (safety program, engineering tool)*1 • U/S: User and system <Set timing> • Every END: Data is set every time END processing is performed. • Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN). • Status change: Data is set when the status is changed. • Error: Data is set when an error occurs. • Instruction execution: Data is set when an instruction is executed. • Request: Data is set when requested by a user (using the safety special relay). • Writing: Data is set when a user performs a writing operation. • During END: Data is set when END processing is performed. • Power-on to RUN or STOP to RUN: Data is set when the operating status changes from power-on to RUN or from STOP to RUN.
CPU	The following shows the supported CPU modules. Each of the CPU module is represented by the following symbols. • Rn: Programmable controller CPU • RnP: Process CPU • RnPSF: SIL2 Process CPU • RnSF: Safety CPU • ALL: All the above CPU modules

*1 As an operation from the GOT or other external devices, only data read is available.

Point

Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

A

System clock

The following is the list of safety special relay areas relating to the system clock.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SA\SM400	Always on	ON OFF	This relay is always on.	S (Power-on to RUN or STOP to RUN)	RnPSF RnSF
SA\SM401	Always off	ON OFF	This relay is always off.	S (Power-on to RUN or STOP to RUN)	RnPSF RnSF
SA\SM444	On at the first safety program execution	ON OFF First time only	This relay turns on at the first safety program execution.	S (Power-on to RUN or STOP to RUN)	RnPSF RnSF
SA\SM1800	ON for only the first safety program execution after system switching (standby system to control system)	ON OFF First time only	This relay turns on only the first safety cycle processing after the standby system was switched to the control system.	S (Status change)	RnPSF

Safety information

The following is the list of safety special relay areas relating to safety information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SA\SM1008	Safety refresh communication status of each module (1st module)	Off: Normal On: Communication error	The safety refresh communication status of the first RJ71GN11-T2 or the first CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1904) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1008 to SA\SD1015.)	S (Status change)	RnPSF RnSF
SA\SM1016	Safety refresh communication status of each module (2nd module)	Off: Normal On: Communication error	The safety refresh communication status of the second RJ71GN11-T2 or the second CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1905) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1016 to SA\SD1023.)	S (Status change)	RnPSF RnSF
SA\SM1024	Safety refresh communication status of each module (3rd module)	Off: Normal On: Communication error	The safety refresh communication status of the third RJ71GN11-T2 or the third CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1906) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1024 to SA\SD1031.)	S (Status change)	RnPSF RnSF
SA\SM1032	Safety refresh communication status of each module (4th module)	Off: Normal On: Communication error	The safety refresh communication status of the fourth RJ71GN11-T2 or the fourth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1907) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1032 to SA\SD1039.)	S (Status change)	RnPSF RnSF
SA\SM1040	Safety refresh communication status of each module (5th module)	Off: Normal On: Communication error	The safety refresh communication status of the fifth RJ71GN11-T2 or the fifth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1908) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1040 to SA\SD1047.)	S (Status change)	RnPSF RnSF
SA\SM1048	Safety refresh communication status of each module (6th module)	Off: Normal On: Communication error	The safety refresh communication status of the sixth RJ71GN11-T2 or the sixth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1909) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1048 to SA\SD1055.)	S (Status change)	RnPSF RnSF

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SA\SM1056	Safety refresh communication status of each module (7th module)	Off: Normal On: Communication error	The safety refresh communication status of the seventh RJ71GN11-T2 or the seventh CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1910) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1056 to SA\SD1063.)	S (Status change)	RnPSF RnSF
SA\SM1064	Safety refresh communication status of each module (8th module)	Off: Normal On: Communication error	The safety refresh communication status of the eighth RJ71GN11-T2 or the eighth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1911) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1064 to SA\SD1071.)	S (Status change)	RnPSF RnSF
SA\SM1088	Module switch request for safety communication information	Off: Module specification completed On: Module specification requested	The SA\SD1088 content is updated when this relay changes from off to on. This relay turns off when SA\SD1090 to SA\SD1097 and SA\SD1104 to SA\SD1223 data update is complete.	U (Request)/S (Status change)	RnPSF RnSF

A

Appendix 7 List of Safety Special Register Areas

The following table lists items in the list.

Item	Description
No.	Safety special register number
Name	Safety special register name
Description	Data stored in the safety special register
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (safety program, engineering tool)*1 • U/S: User and system <Set timing> • Every END: Data is set every time END processing is performed. • Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN). • Status change: Data is set when the status is changed. • Error: Data is set when an error occurs. • Instruction execution: Data is set when an instruction is executed. • Request: Data is set when requested by a user (using the safety special relay). • Switch change: Data is set when the switch of the CPU module is changed. • Card insertion/removal: Data is set when an SD memory card is inserted or removed. • Writing: Data is set when a user performs a writing operation. • During END: Data is set when END processing is performed.
CPU	The following shows the supported CPU modules. Each of the CPU module is represented by the following symbols. • Rn: Programmable controller CPU • RnP: Process CPU • RnPSF: SIL2 Process CPU • RnSF: Safety CPU • ALL: All the above CPU modules

*1 As an operation from the GOT or other external devices, only data read is available.

Point

Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

System information

The following is the list of safety special register areas relating to system information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SA\SD205	Safety operation mode	Safety operation mode	The safety operation mode is stored. (The applicable bit turns on.) b15 … b3: 0; b2: 1/0 (3); b1: 1/0 (2); b0: 1/0 (1) (1) TEST MODE (2) SAFETY MODE (3) SAFETY MODE (wait-for-restart)	S (Status change)	RnPSF RnSF

Safety information

The following is the list of safety special register areas relating to safety information.

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SA\SD1008 to SA\SD1015	Safety refresh communication status of each safety connection (1st module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the first RJ71GN11-T2 or the first CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1904) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223.	S (Status change)	RnPSF RnSF

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SA\SD1008	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SA\SD1009	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
SA\SD1010	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
SA\SD1011	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
SA\SD1012	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65
SA\SD1013	96	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81
SA\SD1014	112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97
SA\SD1015	—	—	—	—	—	—	—	—	120	119	118	117	116	115	114	113

1 to 120: Safety connection number
—: Fixed to 0

No.	Name	Data stored	Details	Set by (setting timing)	CPU
SA\SD1016 to SA\SD1023	Safety refresh status of each safety connection (2nd module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the second RJ71GN11-T2 or the second CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1905) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)	RnPSF RnSF
SA\SD1024 to SA\SD1031	Safety refresh status of each safety connection (3rd module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the third RJ71GN11-T2 or the third CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1906) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)	RnPSF RnSF
SA\SD1032 to SA\SD1039	Safety refresh status of each safety connection (4th module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the fourth RJ71GN11-T2 or the fourth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1907) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)	RnPSF RnSF

<table>
<tr><th>No.</th><th>Name</th><th>Data stored</th><th>Details</th><th>Set by (setting timing)</th><th>CPU</th></tr>
<tr><td>SA\SD1040
to
SA\SD1047</td><td>Safety refresh status of each safety connection (5th module)</td><td>0: Safety communications normal, safety connection not used, own station
1: Communication error</td><td>• The communication status of safety stations connected to the fifth RJ71GN11-T2 or the fifth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1908) is stored.
• A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1048
to
SA\SD1055</td><td>Safety refresh status of each safety connection (6th module)</td><td>0: Safety communications normal, safety connection not used, own station
1: Communication error</td><td>• The communication status of safety stations connected to the sixth RJ71GN11-T2 or the sixth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1909) is stored.
• A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1056
to
SA\SD1063</td><td>Safety refresh status of each safety connection (7th module)</td><td>0: Safety communications normal, safety connection not used, own station
1: Communication error</td><td>• The communication status of safety stations connected to the seventh RJ71GN11-T2 or the seventh CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1910) is stored.
• A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1064
to
SA\SD1071</td><td>Safety refresh communication status of each safety connection (8th module)</td><td>0: Safety communications normal, safety connection not used, own station
1: Communication error</td><td>• The communication status of safety stations connected to the eighth RJ71GN11-T2 or the eighth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1911) is stored.
• A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1088</td><td>Module setting for safety communication information</td><td>1 to 8:Target module</td><td>• The RJ71GN11-T2 or the CC-Link IE Field Network master/local module targeted for safety communication status check is specified.
• To reflect the value set in this register, turn on SA\SM1088.
• If an unmounted module is specified, 0 is stored in SA\SD1090 to SA\SD1097 and SA\SD1104 to SA\SD1223.</td><td>U (Request)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1089</td><td>Target module for safety communication information</td><td>1 to 8:Target module</td><td>• The module number specified in SA\SD1088 is stored.
• Data for the module displayed in this register is stored in SA\SD1090 to SA\SD1097 and SA\SD1104 to SA\SD1223.</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1090
to
SA\SD1097</td><td>Safety communication setting of each safety connection (1st to 8th module)</td><td>0: Not set
1: Set</td><td>The safety communication setting status of safety connections of the module stored in SA\SD1089 is stored.
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SA\SD1090</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SA\SD1091</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SA\SD1092</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SA\SD1093</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
<tr><td>SA\SD1094</td><td>80</td><td>79</td><td>78</td><td>77</td><td>76</td><td>75</td><td>74</td><td>73</td><td>72</td><td>71</td><td>70</td><td>69</td><td>68</td><td>67</td><td>66</td><td>65</td></tr>
<tr><td>SA\SD1095</td><td>96</td><td>95</td><td>94</td><td>93</td><td>92</td><td>91</td><td>90</td><td>89</td><td>88</td><td>87</td><td>86</td><td>85</td><td>84</td><td>83</td><td>82</td><td>81</td></tr>
<tr><td>SA\SD1096</td><td>112</td><td>111</td><td>110</td><td>109</td><td>108</td><td>107</td><td>106</td><td>105</td><td>104</td><td>103</td><td>102</td><td>101</td><td>100</td><td>99</td><td>98</td><td>97</td></tr>
<tr><td>SA\SD1097</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>120</td><td>119</td><td>118</td><td>117</td><td>116</td><td>115</td><td>114</td><td>113</td></tr>
</table>
1 to 120: Safety connection number
—: Fixed to 0</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1104
to
SA\SD1223</td><td>Safety communication status of each safety connection (1st to 8th module)
Safety connection No. 1 to 120</td><td>Safety communication status of safety connection numbers 1 to 120</td><td>• The safety communication status of safety connections of the module stored in SA\SD1089 is stored.
• 0 is stored if not used in SA\SD1104 to SA\SD1223, or if own station.
· 0H: Performing safety refresh communications
· 10H to 15H: Performing safety initial communications[*2]
· 30H: Safety communication stopped (This value is stored when the safety communication error occurs and the systems are switched in interlock status.)
· 1000H or later:For details on the values stored when an error occurs, refer to the list of error codes. (☞ Page 805 List of error codes)</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
</table>

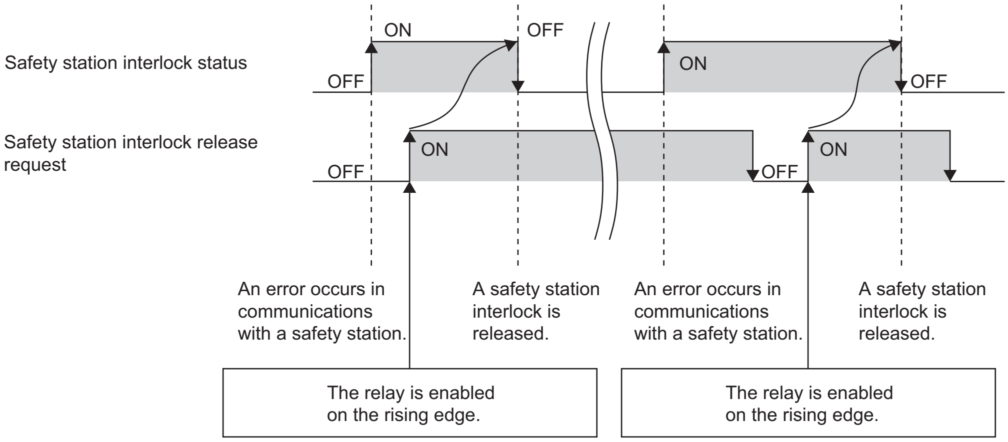
<table>
<tr><th>No.</th><th>Name</th><th>Data stored</th><th>Details</th><th>Set by (setting timing)</th><th>CPU</th></tr>
<tr><td>SA\SD1232 to SA\SD1239</td><td>Interlock status of each safety connection (1st module)</td><td>0: Not interlocked
1: Interlocked</td><td>After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the safety connection turns on.
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SA\SD1232</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SA\SD1233</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SA\SD1234</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SA\SD1235</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
<tr><td>SA\SD1236</td><td>80</td><td>79</td><td>78</td><td>77</td><td>76</td><td>75</td><td>74</td><td>73</td><td>72</td><td>71</td><td>70</td><td>69</td><td>68</td><td>67</td><td>66</td><td>65</td></tr>
<tr><td>SA\SD1237</td><td>96</td><td>95</td><td>94</td><td>93</td><td>92</td><td>91</td><td>90</td><td>89</td><td>88</td><td>87</td><td>86</td><td>85</td><td>84</td><td>83</td><td>82</td><td>81</td></tr>
<tr><td>SA\SD1238</td><td>112</td><td>111</td><td>110</td><td>109</td><td>108</td><td>107</td><td>106</td><td>105</td><td>104</td><td>103</td><td>102</td><td>101</td><td>100</td><td>99</td><td>98</td><td>97</td></tr>
<tr><td>SA\SD1239</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>120</td><td>119</td><td>118</td><td>117</td><td>116</td><td>115</td><td>114</td><td>113</td></tr>
</table>
1 to 120: Safety connection number
—: Fixed to 0</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1240 to SA\SD1247</td><td>Interlock release request for each safety connection (1st module)</td><td>0: Do not release the interlock.
1: Release the interlock.</td><td>Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SA\SD1240</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SA\SD1241</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SA\SD1242</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SA\SD1243</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
<tr><td>SA\SD1244</td><td>80</td><td>79</td><td>78</td><td>77</td><td>76</td><td>75</td><td>74</td><td>73</td><td>72</td><td>71</td><td>70</td><td>69</td><td>68</td><td>67</td><td>66</td><td>65</td></tr>
<tr><td>SA\SD1245</td><td>96</td><td>95</td><td>94</td><td>93</td><td>92</td><td>91</td><td>90</td><td>89</td><td>88</td><td>87</td><td>86</td><td>85</td><td>84</td><td>83</td><td>82</td><td>81</td></tr>
<tr><td>SA\SD1246</td><td>112</td><td>111</td><td>110</td><td>109</td><td>108</td><td>107</td><td>106</td><td>105</td><td>104</td><td>103</td><td>102</td><td>101</td><td>100</td><td>99</td><td>98</td><td>97</td></tr>
<tr><td>SA\SD1247</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>120</td><td>119</td><td>118</td><td>117</td><td>116</td><td>115</td><td>114</td><td>113</td></tr>
</table>
1 to 120: Safety connection number
—: Fixed to 0</td><td>U (Request)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1248 to SA\SD1255</td><td>Interlock status of each safety connection (2nd module)</td><td>0: Not interlocked
1: Interlocked</td><td>After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the safety connection turns on. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1256 to SA\SD1263</td><td>Interlock release request for each safety connection (2nd module)</td><td>0: Do not release the interlock.
1: Release the interlock.</td><td>Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>U (Request)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1264 to SA\SD1271</td><td>Interlock status of each safety connection (3rd module)</td><td>0: Not interlocked
1: Interlocked</td><td>After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the safety connection turns on. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1272 to SA\SD1279</td><td>Interlock release request for each safety connection (3rd module)</td><td>0: Do not release the interlock.
1: Release the interlock.</td><td>Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>U (Request)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1280 to SA\SD1287</td><td>Interlock status of each safety connection (4th module)</td><td>0: Not interlocked
1: Interlocked</td><td>After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the safety connection turns on. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1288 to SA\SD1295</td><td>Interlock release request for each safety connection (4th module)</td><td>0: Do not release the interlock.
1: Release the interlock.</td><td>Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>U (Request)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1296 to SA\SD1303</td><td>Interlock status of each safety connection (5th module)</td><td>0: Not interlocked
1: Interlocked</td><td>After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the safety connection turns on. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1304 to SA\SD1311</td><td>Interlock release request for each safety connection (5th module)</td><td>0: Do not release the interlock.
1: Release the interlock.</td><td>Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>U (Request)</td><td>RnPSF
RnSF</td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Data stored</th><th>Details</th><th>Set by (setting timing)</th><th>CPU</th></tr>
<tr><td>SA\SD1312 to SA\SD1319</td><td>Interlock status of each safety connection (6th module)</td><td>0: Not interlocked
1: Interlocked</td><td>After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the safety connection turns on.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1320 to SA\SD1327</td><td>Interlock release request for each safety connection (6th module)</td><td>0: Do not release the interlock.
1: Release the interlock.</td><td>Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>U (Request)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1328 to SA\SD1335</td><td>Interlock status of each safety connection (7th module)</td><td>0: Not interlocked
1: Interlocked</td><td>After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the safety connection turns on.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1336 to SA\SD1343</td><td>Interlock release request for each safety connection (7th module)</td><td>0: Do not release the interlock.
1: Release the interlock.</td><td>Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>U (Request)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1344 to SA\SD1351</td><td>Interlock status of each safety connection (8th module)</td><td>0: Not interlocked
1: Interlocked</td><td>After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the safety connection turns on.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1352 to SA\SD1359</td><td>Interlock release request for each safety connection (8th module)</td><td>0: Do not release the interlock.
1: Release the interlock.</td><td>Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>U (Request)</td><td>RnPSF
RnSF</td></tr>
<tr><td>SA\SD1376 to SA\SD1379</td><td>Safety I/O refresh status</td><td>0: Normal, safety I/O refresh not used, own module
1: Safety I/O refresh error</td><td>The safety I/O refresh status is stored.
A safety I/O refresh error means that an error has occurred or safety I/O refresh initialization is in progress in the I/O module with safety functions.
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SA\SD1376</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SA\SD1377</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SA\SD1378</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SA\SD1379</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table></td><td>S (Status change)</td><td>RnSF</td></tr>
<tr><td>SA\SD1384 to SA\SD1387</td><td>Safety I/O refresh setting</td><td>0: Not set
1: Set</td><td>Availability of the safety I/O refresh setting is indicated.
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SA\SD1384</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SA\SD1385</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SA\SD1386</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SA\SD1387</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table></td><td>S (Status change)</td><td>RnSF</td></tr>
<tr><td>SA\SD1392 to SA\SD1395</td><td>Safety I/O refresh interlock status</td><td>0: Not interlocked
1: Interlocked</td><td>After safety I/O refresh error is detected and safety I/O refresh is interlocked, the bit corresponding to the slot number is set to 1.
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SA\SD1392</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SA\SD1393</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SA\SD1394</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SA\SD1395</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table></td><td>S (Status change)</td><td>RnSF</td></tr>
<tr><td>SA\SD1400 to SA\SD1403</td><td>Safety I/O refresh interlock release request</td><td>0: Do not release the interlock.
1: Release the interlock.</td><td>The number to release the safety I/O refresh interlock for each safety I/O refresh is changed from 0 to 1. (The number does not automatically return to 0 after execution is complete.)
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SA\SD1400</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SA\SD1401</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SA\SD1402</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SA\SD1403</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table></td><td>U (Request)</td><td>RnSF</td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Data stored</th><th>Details</th><th>Set by (setting timing)</th><th>CPU</th></tr>
<tr><td>SA\SD1600 to SA\SD1607</td><td>Safety I/O hold status of each safety connection (1st module)</td><td>0: Not holding safety input/output
1: Holding safety input/output</td><td>The safety input/output hold status of safety stations connected to the 1st CC-Link IE Field Network module (the one whose start I/O number is stored in SD1904) is stored.
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SA\SD1600</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SA\SD1601</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SA\SD1602</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SA\SD1603</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
<tr><td>SA\SD1604</td><td>80</td><td>79</td><td>78</td><td>77</td><td>76</td><td>75</td><td>74</td><td>73</td><td>72</td><td>71</td><td>70</td><td>69</td><td>68</td><td>67</td><td>66</td><td>65</td></tr>
<tr><td>SA\SD1605</td><td>96</td><td>95</td><td>94</td><td>93</td><td>92</td><td>91</td><td>90</td><td>89</td><td>88</td><td>87</td><td>86</td><td>85</td><td>84</td><td>83</td><td>82</td><td>81</td></tr>
<tr><td>SA\SD1606</td><td>112</td><td>111</td><td>110</td><td>109</td><td>108</td><td>107</td><td>106</td><td>105</td><td>104</td><td>103</td><td>102</td><td>101</td><td>100</td><td>99</td><td>98</td><td>97</td></tr>
<tr><td>SA\SD1607</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>120</td><td>119</td><td>118</td><td>117</td><td>116</td><td>115</td><td>114</td><td>113</td></tr>
</table>
1 to 120: Safety connection number
—: Fixed to 0</td><td>S (Status change)</td><td>RnPSF</td></tr>
<tr><td>SA\SD1608 to SA\SD1615</td><td>Safety I/O hold status of each safety connection (2nd module)</td><td>0: Not holding safety input/output
1: Holding safety input/output</td><td>The safety input/output hold status of safety stations connected to the 2nd CC-Link IE Field Network module (the one whose start I/O number is stored in SD1905) is stored.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF</td></tr>
<tr><td>SA\SD1616 to SA\SD1623</td><td>Safety I/O hold status of each safety connection (3rd module)</td><td>0: Not holding safety input/output
1: Holding safety input/output</td><td>The safety input/output hold status of safety stations connected to the 3rd CC-Link IE Field Network module (the one whose start I/O number is stored in SD1906) is stored.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF</td></tr>
<tr><td>SA\SD1624 to SA\SD1631</td><td>Safety I/O hold status of each safety connection (4th module)</td><td>0: Not holding safety input/output
1: Holding safety input/output</td><td>The safety input/output hold status of safety stations connected to the 4th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1907) is stored.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF</td></tr>
<tr><td>SA\SD1632 to SA\SD1639</td><td>Safety I/O hold status of each safety connection (5th module)</td><td>0: Not holding safety input/output
1: Holding safety input/output</td><td>The safety input/output hold status of safety stations connected to the 5th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1908) is stored.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF</td></tr>
<tr><td>SA\SD1640 to SA\SD1647</td><td>Safety I/O hold status of each safety connection (6th module)</td><td>0: Not holding safety input/output
1: Holding safety input/output</td><td>The safety input/output hold status of safety stations connected to the 6th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1909) is stored.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF</td></tr>
<tr><td>SA\SD1648 to SA\SD1655</td><td>Safety I/O hold status of each safety connection (7th module)</td><td>0: Not holding safety input/output
1: Holding safety input/output</td><td>The safety input/output hold status of safety stations connected to the 7th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1910) is stored.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF</td></tr>
<tr><td>SA\SD1656 to SA\SD1663</td><td>Safety I/O hold status of each safety connection (8th module)</td><td>0: Not holding safety input/output
1: Holding safety input/output</td><td>The safety input/output hold status of safety stations connected to the 8th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1911) is stored.
The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)</td><td>S (Status change)</td><td>RnPSF</td></tr>
</table>

*1 A safety station interlock release request is executed when the bit rises, and the interlock status is then released. To release the interlock again, it is necessary to turn the request off and on again. After eliminating the cause of any communication errors, turn on the request with the CPU module. When performing safety communications between CPU modules, perform this operation with both CPU modules.

A



*2 If the safety communications do not start when the safety communication status keep performing safety initial communications (10H to 15H), check the safety communications partner station status or network communication status.

Appendix 8 Buffer Memory

The buffer memory is memory used with the following applications.

Module	Application
CPU module	Stores values such as Ethernet function setting values.
SIL2 function module and safety function module	Stores the status of the SIL2 function module and safety function module (such as diagnostics information and system information). (☞ Page 966 List of Special Register Areas)

The buffer memory content returns to its default status (initial value) when the CPU module is powered off or is reset.

List of buffer memory addresses

CPU module

The following table lists the CPU module buffer memory addresses.

■Ethernet function

For a list of buffer memory addresses as well as details relating to the Ethernet function, refer to the following.
(📖 MELSEC iQ-R Ethernet User's Manual (Application))

■CC-Link IE Field Network Basic function

The following table lists buffer memory addresses relating to the CC-Link IE Field Network Basic function. (Programmable controller CPU only) (☞ Page 1036 Details on buffer memory addresses)

Address	Name	Data stored
Un\G1024	Total number of connected stations	The total number of connected stations set in parameter is stored.
Un\G1025	Reserved station specification status	The reserved station specification status of the device station specified in parameter is stored.
Un\G1026 to Un\G1029	Reserved station specification status of each station	The reserved station specification status is stored.
Un\G1030 to Un\G1032, Un\G1034 to Un\G1036, Un\G1038 to Un\G1040, Un\G1042 to Un\G1044	Link scan information	The link scan time during cyclic transmission is stored.
Un\G1050	Diagnostic information display request	When a station number (1 to 16) to be diagnosed is specified in 'Diagnostic request information' (Un\G1051) and the bit 0 of 'Diagnostic information display request' (Un\G1050) is turned off and on, the diagnostic information of the specified device station is stored in 'Diagnostic information 1' (Un\G1053 to Un\G1064) and 'Diagnostic information 2' (Un\G1068 to Un\G1077).
Un\G1051	Diagnostic request information	
Un\G1052	Diagnostic information status flag	
Un\G1053 to Un\G1064	Diagnostic information 1	
Un\G1068 to Un\G1077	Diagnostic information 2	

SIL2 function module and safety function module

The following table lists buffer memory addresses of the SIL2 function module and safety function module.

Address	Name	Data stored
Un\G0	Latest self-diagnostics error code	Error codes are stored in a hexadecimal value when the self-diagnostics detects an error.
Un\G1	Latest self-diagnostics error time	The year value (four digits) for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G2		The month value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G3		The day value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G4		The hour value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G5		The minute value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G6		The second value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G7		The day value for the date/time when Un\G0 data was updated is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)
Un\G10 to Un\G25	Self-diagnostics error code 1 to 16	The maximum of 16 types of error codes are stored into Un\G10 onwards when the self-diagnostics detects an error. (The same error code as one already stored in Un\G10 onwards is not stored.) The 17th error code onwards are not stored. Also, error codes are not stored when 16 types of error codes have already been stored in Un\G10 to Un\G25.
Un\G50	Error clear	Set 1 in Un\G50 to clear the error. After the error is cleared, 0 is automatically set in Un\G50 by the system.
Un\G52	LED status	This buffer memory area stores the information that indicates the LED status (0: off, 1: on, 2: flashing) in the following bit patterns. • b0 to b1: READY • b2 to b3: ERROR • b4 to b5: PROGRAM RUN • b6 to b7: SAFETY COM RUN • b8 to b9: SAFETY COM ERR • b10 to b11: TEST
Un\G54	Safety cycle processing time execution cycle error count	The detected number of times that cycles for which safety cycle processing was not executed is stored.
Un\G60	Pair version	The pair version is stored in decimal notation.
Un\G62	Current safety cycle time (ms unit)	• The current safety cycle processing time is stored (it is measured in increments of µs). Un\G62: The ms part is stored. (Range: 0 to 65535) Un\G63: The µs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.
Un\G63	Current safety cycle time (µs unit)	
Un\G64	Minimum safety cycle time (ms unit)	• The minimum safety cycle processing time value is stored (it is measured in increments of µs). Un\G64: The ms part is stored. (Range: 0 to 65535) Un\G65: The µs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.
Un\G65	Minimum safety cycle time (µs unit)	
Un\G66	Maximum safety cycle time (ms unit)	• The maximum safety cycle processing time value is stored (it is measured in increments of µs). Un\G66: The ms part is stored. (Range: 0 to 65535) Un\G67: The µs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.
Un\G67	Maximum safety cycle time (µs unit)	
Un\G70[*2]	Firmware version	The firmware version is stored.
Un\G80	Detailed information 1 information category	The detailed information 1 information category code is stored.
Un\G81 to Un\G111	Detailed information 1	• Detailed information 1 corresponding to the error code (Un\G0) is stored. • The stored information contains the following eight types. 0: N/A 1: Program position information 2: Drive number and file name 4: Parameter information 5: System configuration information 6: Number of times information 7: Time information 24: Failure information 34: Continuous RUN prevention setting in TEST MODE • The type of detailed information 1 can be obtained using Un\G80. (The bit patterns for values in the "Detailed information 1 information category code" stored in Un\G80 correspond to those in SD81 to SD111.) (☞ Page 966 List of Special Register Areas)
Un\G112	Detailed information 2 information category	The detailed information 2 information category code is stored.

Address	Name	Data stored
Un\G113 to Un\G143	Detailed information 2	• Detailed information 2 corresponding to the error code (Un\G0) is stored. • The stored information contains the following six types. 0: N/A 2: Drive number and file name 3: Annunciator number 4: Parameter information 5: System configuration information 25: Process control instruction processing information 29: Other station error information (CC-Link IE Field) • The type of detailed information 1 can be obtained using Un\G112. (The bit patterns for values in the "Detailed information 2 information category code" stored in Un\G112 correspond to those in SD113 to SD143.) (☞ Page 966 List of Special Register Areas)
Un\G150	Program memory writing	• b0: This relay switches to on when a write error is detected during write operation to the program memory. This relay switches to off when the write instruction is issued. • b1: This relay is on when the write process to the program memory is in progress. The relay is switched to off when the write process is completed. • b2: This relay switches to on when the number of program memory rewriting operations reaches 100000. (The module must be replaced.)
Un\G152	Program memory write (transfer) status	This register displays the write (transfer) status to the program memory with a percentage (0 to 100%). The initial value is "0". Upon completion of writing, this register is set to "100". It is set to "0" at the time when the write command is issued.
Un\G153 to Un\G154	Program memory write count index	• This register indicates the index value for the number of write operations to the program memory up to now (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value continues to be counted even when it exceeds 100000). If the index value exceeds 100000, the module must be replaced.
Un\G155	Data memory writing	• b0: This relay switches to on when a write error is detected during write operation to the data memory. This relay switches to off when the write instruction is issued. • b1: This relay is on when the write process to the data memory is in progress. The relay is switched to off when the write process is completed. • b2: This relay switches to on when the number of data memory rewriting operations reaches 100000. (The module must be replaced.)
Un\G157	Data memory write (transfer) status	This register displays write (transfer) status to the data memory with a percentage. (0 to 100%). The initial value is "0". Upon completion of writing, this register is set to "100". It is set to "0" at the time when the write command is issued.
Un\G158 to Un\G159	Index for the number of data memory write operations	• This register indicates the index value for the number of write operations to the data memory up to now (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value continues to be counted even when it exceeds 100000). If the index value exceeds 100000, the module must be replaced.
Un\G160	System memory writing	• b0: This relay switches to on when a write error is detected during write operation to the system memory (Flash ROM)*1. This relay switches to off when the write instruction is issued. • b1: This relay switches to on when the write process to the system memory (Flash ROM)*1 is in progress, and switches to off when the write process is complete. • b2: This relay switches to on when the number of system memory (Flash ROM)*1 rewrite operations reaches 100000. (The SIL2 function module or safety function module must be replaced.)
Un\G163 to Un\G164	Index for the number of system memory write operations	• This register indicates the index value for the number of write operations to the system memory (Flash ROM)*1 up to now. (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value continues to be counted even when it exceeds 100000). If the index value exceeds 100000, the module must be replaced.

*1 This memory is used by the system when the SIL2 function module or safety function module is executing functions.
*2 The safety function module with firmware version "11" or later supports this buffer memory area.

Details on buffer memory addresses

The following table lists items in the list.

Item	Description
Address	Buffer memory address of the CPU module
Name	Buffer memory name of the CPU module
Data stored	Data stored in the CPU module buffer memory and its meaning
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (program, engineering tool, GOT, or other testing operations from external device) • U/S: User and system <Set timing> • Every END: Data is set every time END processing is performed. • Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN). • Status change: Data is set when the status is changed. • Error: Data is set when an error occurs. • Instruction execution: Data is set when an instruction is executed. • Request: Data is set when requested by a user (using the special relay). • Switch change: Data is set when the switch of the CPU module is changed. • Card insertion/removal: Data is set when an SD memory card is inserted or removed. • Writing: Data is set when a user performs a writing operation. • During END: Data is set when END processing is performed.

Point

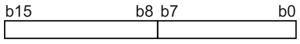
Do not change the data set by the system in a program or by a device test. Doing so may cause an unintended operation.

CC-Link IE Field Network Basic function

The following table lists buffer memory areas of the CPU module relating to the CC-Link IE Field Network Basic function. (Programmable controller CPU only)

<table>
<tr><th>Address</th><th>Name</th><th>Data stored</th><th>Details</th><th>Set by (setting timing)</th></tr>
<tr><td>Un\G1024</td><td>Total number of connected stations</td><td>Total number of connected stations</td><td>The total number of connected stations set in parameter is stored.
Range: 1 to 16</td><td>S (Initial)</td></tr>
<tr><td>Un\G1025</td><td>Reserved station specification status</td><td>Reserved station specification status</td><td>The reserved station specification status of the device station specified in parameter is stored. (0: Not specified, 1: Specified)
b0
b0: Reserved station specification status
b1 to b15: Empty (fixed to 0)
The station number that is specified as a reserved station can be checked in 'Reserved station specification status of each station' (Un\G1026).</td><td>S (Initial)</td></tr>
<tr><td>Un\G1026 to Un\G1029</td><td>Reserved station specification status of each station</td><td>Reserved station specification status of each station</td><td>The reserved station specification status is stored using the following bit pattern. (Off: Other than the reserved station, On: Reserved station)
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>Un\G1026</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>Un\G1027</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>Un\G1028</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>Un\G1029</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table>
The numbers in the figure indicate station numbers.
(Condition)
• Only the bit of the start station number turns on.
• The status is not stored for the station numbers after the maximum station number.</td><td>S (Initial)</td></tr>
<tr><td>Un\G1030</td><td rowspan="12">Link scan information</td><td>Group No.1 maximum link scan</td><td>The maximum link scan time value during cyclic transmission is stored. (Unit: ms)</td><td>S (Status change)</td></tr>
<tr><td>Un\G1031</td><td>Group No.1 minimum link scan</td><td>The minimum link scan time value during cyclic transmission is stored. (Unit: ms)</td><td>S (Status change)</td></tr>
<tr><td>Un\G1032</td><td>Group No.1 current link scan</td><td>The current link scan time value during cyclic transmission is stored. (Unit: ms)</td><td>S (Status change)</td></tr>
<tr><td>Un\G1034</td><td>Group No.2 maximum link scan</td><td>The maximum link scan time value during cyclic transmission is stored. (Unit: ms)</td><td>S (Status change)</td></tr>
<tr><td>Un\G1035</td><td>Group No.2 minimum link scan</td><td>The minimum link scan time value during cyclic transmission is stored. (Unit: ms)</td><td>S (Status change)</td></tr>
<tr><td>Un\G1036</td><td>Group No.2 current link scan</td><td>The current link scan time value during cyclic transmission is stored. (Unit: ms)</td><td>S (Status change)</td></tr>
<tr><td>Un\G1038</td><td>Group No.3 maximum link scan</td><td>The maximum link scan time value during cyclic transmission is stored. (Unit: ms)</td><td>S (Status change)</td></tr>
<tr><td>Un\G1039</td><td>Group No.3 minimum link scan</td><td>The minimum link scan time value during cyclic transmission is stored. (Unit: ms)</td><td>S (Status change)</td></tr>
<tr><td>Un\G1040</td><td>Group No.3 current link scan</td><td>The current link scan time value during cyclic transmission is stored. (Unit: ms)</td><td>S (Status change)</td></tr>
<tr><td>Un\G1042</td><td>Group No.4 maximum link scan</td><td>The maximum link scan time value during cyclic transmission is stored. (Unit: ms)</td><td>S (Status change)</td></tr>
<tr><td>Un\G1043</td><td>Group No.4 minimum link scan</td><td>The minimum link scan time value during cyclic transmission is stored. (Unit: ms)</td><td>S (Status change)</td></tr>
<tr><td>Un\G1044</td><td>Group No.4 current link scan</td><td>The current link scan time value during cyclic transmission is stored. (Unit: ms)</td><td>S (Status change)</td></tr>
<tr><td>Un\G1050</td><td>Diagnostic information display request</td><td>Diagnostic information display request</td><td>After the END instruction of the scan where the bit 0 is turned off and on is executed, the diagnostic information of a device station specified in 'Diagnostic request information' (Un\G1051) is read to Un\G1052 to Un\G1077.
When reading of the diagnostic information has completed at END processing, 0 is stored.
b0
b0: Diagnostic information display request
b1 to b15: Empty (fixed to 0)</td><td>S (Status change)*1/U</td></tr>
</table>

A

Address	Name	Data stored	Details	Set by (setting timing)
Un\G1051	Diagnostic request information	Diagnostic request information	Specify a device station number whose diagnostic information is to be displayed. Range: 1 to 64 • For the programmable controller CPU with firmware version earlier than "28", the range is from 1 to 16.	U
Un\G1052	Diagnostic information status flag	Diagnostic information status flag	After the END instruction of the scan where the bit 0 of 'Diagnostic information display request' (Un\G1050) is turned off and on is executed, the status (valid or invalid) of diagnostic information (Diagnostic information 1, Diagnostic information 2) of the device station specified in 'Diagnostic request information' (Un\G1051) is stored. (Valid: 1, Invalid: 0)  b0 to b7: Diagnostic information 1 b8 to b15: Diagnostic information 2 • If the device number of the device station that is specified in 'Diagnostic request information' (Un\G1051) is the start station number of the occupied stations and the cyclic transmission is performed for the device station, 1 is stored in b0 to b7 and b8 to b15. (If the specified device station is a reserved station, 0 is stored in b8 to b15.) • If the station number of the device station that is specified in 'Diagnostic request information' (Un\G1051) is other than the start station number of the occupied stations or the cyclic transmission is not performed for the device station, 0 is stored in b0 to b7 and b8 to b15. • When b0 to b7 are valid, the number of occupied stations, group number, IP address, the accumulated number of timeouts, and the accumulated number of disconnection detection are stored in 'Diagnostic information 1' (Un\G1053 to Un\G1064). When invalid, 0 is stored in 'Diagnostic information 1' (Un\G1053 to Un\G1064). • When b8 to b15 are valid, the Manufacturer code, model code, device version, module information, error code, and detailed module information are stored in 'Diagnostic information 2' (Un\G1068 to Un\G1077). When invalid, 0 is stored in 'Diagnostic information 2' (Un\G1068 to Un\G1077).	S (Status change)

<table>
<tr><th>Address</th><th>Name</th><th>Data stored</th><th>Details</th><th>Set by (setting timing)</th></tr>
<tr><td>Un\G1053 to Un\G1064</td><td>Diagnostic information 1[*3]</td><td>Diagnostic information 1</td><td>When 1 (valid) is stored in b0 to b7 of Un\G1052, the number of occupied stations, group number, IP address, the accumulated number of timeouts, and the accumulated number of disconnection detection are stored. When 0 (invalid) is stored in b0 to b7 of Un\G1052, 0 is stored.
■Un\G1053: Number of occupied stations
■Un\G1054: Group number
■Un\G1055: IP address (lower)
■Un\G1056: IP address (upper)
<table><tr><th></th><th>b15 to b8</th><th>b7 to b0</th></tr><tr><td>Un\G1055</td><td>3</td><td>4</td></tr><tr><td>Un\G1056</td><td>1</td><td>2</td></tr></table>1 to 4: First octet to fourth octet
When the IP address has not been set in the parameter, 0 is stored.
■Un\G1063: Accumulated number of timeouts
After the END instruction of the scan where the bit 0 of 'Diagnostic information display request' (Un\G1050) is turned off and on is executed, the accumulated number of timeouts that occurred in a device station specified in 'Diagnostic request information' (Un\G1051) is stored.
• 0: No timeouts
• 1 to 65535: Number of timeouts (accumulated number)[*2]
■Un\G1064: Accumulated number of disconnection detection
After the END instruction of the scan where the bit 0 of 'Diagnostic information display request' (Un\G1050) is turned off and on is executed, the accumulated number of disconnections that detected in a device station specified in 'Diagnostic request information' (Un\G1051) is stored.
• 0: No disconnections
• 1 to 65535: Number of disconnection detection (accumulated number)[*2]</td><td>S (Status change)</td></tr>
<tr><td>Un\G1068 to Un\G1077</td><td>Diagnostic information 2[*3]</td><td>Diagnostic information 2</td><td>When Diagnostic information 2 is valid (1 is stored in b8 to b15 of Un\G1052), the manufacturer code, model code, device version, module information, error code, and detailed module information are stored. When Diagnostic information 2 is invalid (0 is stored in b8 to b15 of Un\G1052), 0 is stored.
■Un\G1068: Manufacturer code
■Un\G1070: Model code (lower)
■Un\G1071: Model code (upper)
■Un\G1072: Device version
■Un\G1074: Module information
■Un\G1075: Error code
■Un\G1076: Detailed module information (lower)
■Un\G1077: Detailed module information (upper)</td><td>S (Status change)</td></tr>
</table>

*1 Data is stored at END processing where the value is changed.
*2 When the count exceeds 65535, counting is continued from 1 again.
*3 A device station specified is being disconnected, the information immediately before the disconnection is stored.

Appendix 9 Processing Time

The scan time of the CPU module is the sum of the instruction execution time, the program execution time, and the END processing time. Each of the processing time that constitutes the scan time is as follows.

Point

For the availability of functions depending on the CPU module, refer to the description of each function.

Instruction execution time

The instruction execution time is the total of each instruction processing time used for the program executed by the CPU module. For the processing time of each instruction, refer to the following.

MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)
MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions)

Program execution time

The program execution time is the processing time when multiple programs are executed. When the interrupt program is executed, the overhead time during the interrupt program being executed is also included.

Processing time when multiple programs are executed (program switching time)

When multiple programs are executed, the scan time becomes longer.

■When multiple programs run

The processing time [μs] with multiple programs run is given by: the number of program files × 1.2.

Overhead time at execution of interrupt/fixed scan execution type program

This section describes each overhead time when each of the interrupt programs and fixed scan execution type programs is executed.

Point

The processing time for the interrupt program and fixed scan execution type program is as shown below.
Instruction processing time for each instruction + Overhead time + Each refresh processing time (when setting)

■Overhead time when executing the interrupt program

The overhead time when executing the interrupt program includes the one before executing the interrupt program and the other when the interrupt program finished.

- Overhead time before executing the interrupt program

Interrupt factor	Condition	Overhead time
Internal timer interrupt (I28 to I31)	Saves/stores the file register (R) block number.	19.5μs
	Not save/store the file register (R) block number.	12.9μs
Inter-module synchronous interrupt (I44)	Saves/stores the file register (R) block number.	25.4μs
	Not save/store the file register (R) block number.	17.0μs
Multiple CPU synchronous interrupt (I45)	Saves/stores the file register (R) block number.	24.2μs
	Not save/store the file register (R) block number.	16.1μs
High-speed internal timer interrupt 2 (I48), high-speed internal timer interrupt 1 (I49)	Saves/stores the file register (R) block number.	25.8μs
	Not save/store the file register (R) block number.	18.1μs
Interrupt from module (I0 to I15, I50 to I1023)	Saves/stores the file register (R) block number.	19.1μs
	Not save/store the file register (R) block number.	12.8μs

- Overhead time when finishing the interrupt program

Interrupt factor	Condition	Overhead time
Internal timer interrupt (I28 to I31)	Saves/stores the file register (R) block number.	16.0μs
	Not save/store the file register (R) block number.	8.5μs
Inter-module synchronous interrupt (I44)	Saves/stores the file register (R) block number.	19.0μs
	Not save/store the file register (R) block number.	11.4μs
Multiple CPU synchronous interrupt (I45)	Saves/stores the file register (R) block number.	18.7μs
	Not save/store the file register (R) block number.	10.5μs
High-speed internal timer interrupt 2 (I48), high-speed internal timer interrupt 1 (I49)	Saves/stores the file register (R) block number.	19.4μs
	Not save/store the file register (R) block number.	10.3μs
Interrupt from module (I0 to I15, I50 to I1023)	Saves/stores the file register (R) block number.	15.7μs
	Not save/store the file register (R) block number.	8.5μs

■Overhead time when executing the fixed scan execution type program

The following table lists the overhead times when executing the fixed scan execution type program.

Condition	Overhead time
Saves/stores the file register (R) block number.	37.9μs
Not save/store the file register (R) block number.	20.6μs

Overhead time when executing safety cycle processing

The safety cycle processing execution time indicates the overhead time when executing safety cycle processing.

■Overhead time before starting safety cycle processing

Overhead time before starting safety cycle processing [μs] = Total standard/safety shared label usage capacity*1 × KM1 + KM2

*1 The capacity indicates the total usage capacity for labels defined in the standard/safety shared label setting file. This can be verified in SD1903. (☞ Page 966 List of Special Register Areas)
This is not the capacity set in the Safety CPU parameter Standard/Safety Shared Label Area Capacity.

Constant		Constant value	
		SIL2 Process CPU	Safety CPU
KM1	Without an extended SRAM cassette	0.15	0.15
	With an extended SRAM cassette	0.22	0.22
KM2		170	165

■Overhead time when ending safety cycle processing

The safety cycle processing execution time of the overhead time [μs] when ending safety cycle processing

Overhead time when ending safety cycle processing [μs]	
SIL2 Process CPU	Safety CPU
85	80

Processing time of safety communications

This section describes the processing time of safety communications.

■Input processing time of safety communications

(1) When using the following module: a CC-Link IE Field Network master/local module with the safety protocol version set only to "1" in the safety communication setting*1

*1 The safety protocol version of the safety communication setting can be set in the engineering tool whose version is "1.095Z" or later. For the engineering tool with its version earlier than "1.095Z", the safety protocol version of the safety communication setting cannot be set. In that case, read "CC-Link IE Field Network master/local module with the safety protocol version set only to "1" in the safety communication setting" as "CC-Link IE Field Network master/local module".

T1 = (N $\times$ KM1) + (M $\times$ KM2) + KM3[μs]

- T1: Input processing time of safety communications for (1)
- N: Number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting
- M: Number of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting

Constant		Constant value
KM1		11
KM2	Main base unit	15
	Extension base unit	15
KM3		110

(2) When using any of the following modules: a CC-Link IE TSN master/local module, the RJ71GN11-EIP, or a CC-Link IE Field Network master/local module with the safety protocol version set only to "2" in the safety communication setting alone or combined

T2 = (N $\times$ KM1) + (M $\times$ KM2) + KM3[μs]

- T2: Input processing time of safety communications for (2)
- N: Number of connections of the CC-Link IE TSN master/local modules or the RJ71GN11-EIP for the safety communication setting, or the number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting
- M: Number of the CC-Link IE TSN master/local modules set for the safety communication setting, the RJ71GN11-EIPs, or the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting

Constant		Constant value
KM1		13
KM2	Main base unit	15
	Extension base unit	15
KM3		110

(3) When using any of the following modules: a CC-Link IE TSN master/local module, the RJ71GN11-EIP, a CC-Link IE Field Network master/local module with the safety protocol version set only to "2" in the safety communication setting, or a single CC-Link IE Field Network master/local module or multiple CC-Link IE Field Network master/local modules with the safety protocol version to "1" and "2" simultaneously in the safety communication setting[*2]

*2 The safety protocol version of the safety communication setting can be set in the engineering tool whose version is "1.095Z" or later. For the engineering tool with its version earlier than "1.095Z", the safety protocol version of the safety communication setting cannot be set. In that case, read "CC-Link IE Field Network master/local module with the safety protocol version set only to "1" in the safety communication setting" as "CC-Link IE Field Network master/local module".

T3 = (N1 × KM1) + (N2 × KM2) + ((M1 + M2) × KM3) + (M3 × KM4) + KM5 + KM6[μs]

- T3: Input processing time of safety communications for (3)
- N1: Number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting
- N2: Number of connections of the CC-Link IE TSN master/local modules or the RJ71GN11-EIPs for the safety communication setting, or the number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting
- M1: Number of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting
- M2: Number of the CC-Link IE TSN master/local modules set for the safety communication, the RJ71GN11-EIPs, or the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting
- M3: Total number of the mounted CC-Link IE TSN master/local modules set for the safety communication, the RJ71GN11-EIPs, or CC-Link IE Field Network master/local modules

Constant		Constant value	
		CPU module with firmware version earlier than 30	CPU module with firmware version 30 or later
KM1		11	11
KM2		13	13
KM3	Main base unit	5	5
	Extension base unit	5	5
KM4	Main base unit	10	10
	Extension base unit	10	10
KM5		110	110
KM6		5	69

■Output processing time of safety communications

(1) When using the following module: a CC-Link IE Field Network master/local module with the safety protocol version set only to "1" in the safety communication setting[*1]

*1 The safety protocol version of the safety communication setting can be set in the engineering tool whose version is "1.095Z" or later. For the engineering tool with its version earlier than "1.095Z", the safety protocol version of the safety communication setting cannot be set. In that case, read "CC-Link IE Field Network master/local module with the safety protocol version set only to "1" in the safety communication setting" as "CC-Link IE Field Network master/local module".

T1 = (N × KM1) + (M × KM2) + KM3[μs]

- T1: Output processing time of safety communications for (1)
- N: Number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting
- M: Number of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting

Constant		Constant value
KM1		6
KM2	Main base unit	15
	Extension base unit	15
KM3		89

(2) When using any of the following modules: a CC-Link IE TSN master/local module, the RJ71GN11-EIP, or a CC-Link IE Field Network master/local module with the safety protocol version set only to "2" in the safety communication setting alone or combined

T2 = (N × KM1) + (M × KM2) + KM3[μs]

- T2: Output processing time of safety communications for (2)
- N: Number of connections of the CC-Link IE TSN master/local modules or the RJ71GN11-EIP for the safety communication setting, or the number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting
- M: Number of the CC-Link IE TSN master/local modules or RJ71GN11-EIPs with their safety communication setting set, or the number of the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting

Constant		Constant value
KM1		7
KM2	Main base unit	15
	Extension base unit	15
KM3		89

(3) When using any of the following modules: a CC-Link IE TSN master/local module, the RJ71GN11-EIP, a CC-Link IE Field Network master/local module with the safety protocol version set only to "2" in the safety communication setting, or a single CC-Link IE Field Network master/local module or multiple CC-Link IE Field Network master/local modules with the safety protocol version to "1" and "2" simultaneously in the safety communication setting[*2]

*2 The safety protocol version of the safety communication setting can be set in the engineering tool whose version is "1.095Z" or later. For the engineering tool with its version earlier than "1.095Z", the safety protocol version of the safety communication setting cannot be set. In that case, read "CC-Link IE Field Network master/local module with the safety protocol version set only to "1" in the safety communication setting" as "CC-Link IE Field Network master/local module".

T3 = (N1 × KM1) + (N2 × KM2) + ((M1 + M2) × KM3) + (M3 × KM4) + KM5 + KM6[μs]

- T3: Output processing time of safety communications for (3)
- N1: Number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting
- N2: Number of connections of the CC-Link IE TSN master/local modules or the RJ71GN11-EIPs for the safety communication setting, or the number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting
- M1: Number of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting
- M2: Number of the CC-Link IE TSN master/local modules set for the safety communication, the RJ71GN11-EIPs, or the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting
- M3: Total number of the mounted CC-Link IE TSN master/local modules set for the safety communication, the RJ71GN11-EIPs, or CC-Link IE Field Network master/local modules

Constant		Constant value	
		CPU module with firmware version earlier than 30	CPU module with firmware version 30 or later
KM1		6	6
KM2		7	7
KM3	Main base unit	8	8
	Extension base unit	8	8
KM4	Main base unit	7	7
	Extension base unit	7	7
KM5		89	89
KM6		0	103

A

Safety I/O refresh processing time

Safety I/O refresh processing time = input processing time for safety I/O refresh*1 + output processing time for safety I/O refresh*2

*1 This value is added even when only an output module with safety functions is used.

*2 This value is added even when only an input module with safety functions is used.

■Input processing for safety I/O refresh

$T = (\alpha T + \beta T) + KM2$

$\alpha T = KM1 \times M$

$\beta T = (KM1 + (KM3 \times n1) \times M)^{*1} + (KM1 + (KM3 \times n2) \times M)^{*1} + \cdots + (KM1 + (KM3 \times n7) \times M)$

*1 Only the number of the extension base units that have the I/O modules with safety functions mounted is added.

- T: Input processing time for safety I/O refresh (μs)
- αT: Input processing time for safety I/O refresh on the main base unit (μs)
- βT: Input processing time for safety I/O refresh on the extension base unit (μs)
- M: Number of the installed I/O modules with safety functions
- n1 to n7: Processing time increase coefficient per extension base unit (1 to 7)
- KM1, KM2, KM3: Constants

Constant	Constant value
KM1	28
KM2	110
KM3	1

Ex.

When I/O modules with safety functions are mounted as follows: two modules on the main base unit, three modules on the first level of the extension base unit, and four modules on the fifth level of the extension base unit

$T = (KM1 \times 2) + ((KM1 + (KM3 \times 1)) \times 3) + ((KM1 + (KM3 \times 5)) \times 4) + KM2$

$= (28 \times 2) + ((28 + (1 \times 1)) \times 3) + ((28 + (1 \times 5)) \times 4) + 110$

$= 385$

■Output processing for safety I/O refresh

$T = (\alpha T + \beta T) + KM2$

$\alpha T = KM1 \times M$

$\beta T = (KM1 + (KM3 \times n1) \times M)^{*1} + (KM1 + (KM3 \times n2) \times M)^{*1} + \cdots + (KM1 + (KM3 \times n7) \times M)$

*1 Only the number of the extension base units that have the I/O modules with safety functions mounted is added.

- T: Output processing time for safety I/O refresh (μs)
- αT: Output processing time for safety I/O refresh on the main base unit (μs)
- βT: Output processing time for safety I/O refresh on the extension base unit (μs)
- M: Number of the installed I/O modules with safety functions
- n1 to n7: Processing time increase coefficient per extension base unit (1 to 7)
- KM1, KM2, KM3: Constants

Constant	Constant value
KM1	16
KM2	105
KM3	0.6

Processing time for when safety communications and safety I/O refresh are mixed

T = T1 + T2 - KM4

- T: Total processing time for safety communications and safety I/O refresh (μs)
- T1: Safety communications processing time
- T2: Safety I/O refresh processing time
- KM4: Constants

Constant	Constant value
KM4	199

Processing time when multiple programs are executed (safety program switching time)

The processing time [μs] with multiple programs run is given by: the number of program files × 7.5.

A

END processing time

The END processing time includes the following:

- Common processing time
- I/O refresh processing time
- Link refresh processing time for the network module
- Link refresh processing time for CC-Link IE Field Network Basic
- Intelligent function module refresh processing time
- Multiple CPU refresh processing time
- Prolonged time of END processing when executing each function
- Device/label access service processing time
- Processing time of the identification check for safety data
- Tracking transfer time (☞ Page 1078 Increase in the scan time due to tracking transfer)

Common processing time

The following table lists the common processing time of each CPU module to be treated by the system.

Condition	Common processing time				
	Programmable controller CPU	Process CPU (process mode)	Process CPU (redundant mode)	SIL2 Process CPU	Safety CPU
One unit of CPU module (module is not mounted)	85μs	100μs	190μs	0.88 to 97.80ms*1	0.50 to 56.00ms*2

*1 When the scan time is around 20ms, the processing time will be approximately 10.60ms. When the scan time is around 100ms, the processing time will be approximately 17.50ms.

*2 When the scan time is around 20ms, the processing time will be approximately 2.00ms. When the scan time is around 100ms, the processing time will be approximately 6.10ms.

I/O refresh processing time

The I/O refresh processing time for module mounted on main base unit and extension base unit can be calculated by the following formula.

I/O refresh processing time [μs] = ((Number of input refresh points*1 × KM1) + (Number of unit having number of input points × KM2) + KM3*3) + ((Number of output refresh points*2 × KM4) + (Number of unit having number of output points × KM5) + KM6*3)

*1 The value indicates the numeric value that is obtained through dividing the number of input points by 16.

*2 The value indicates the numeric value that is obtained through dividing the number of output points by 16.

*3 When the number of I/O points is 0, this value is handled as 0.

Condition				Constant value
MELSEC iQ-R series module	Main base unit	Input	KM1	0.04
			KM2	0.56
			KM3	11.2
		Output	KM4	0.02
			KM5	0.94
			KM6	6.30
	Extension base unit	Input	KM1	0.04
			KM2	0.56
			KM3	11.2
		Output	KM4	0.02
			KM5	0.94
			KM6	6.30
Q series module	RQ extension base unit	Input	KM1	1.15
			KM2	1.33
			KM3	24.0
		Output	KM4	0.03
			KM5	1.20
			KM6	6.21
	Q series extension base unit	Input	KM1	1.85
			KM2	1.98
			KM3	39.4
		Output	KM4	0.02
			KM5	1.29
			KM6	6.01

Link refresh processing time for the network module

This section describes the link refresh processing time for the network module.

■Link refresh processing time for the CC-Link IE TSN master/local module

The link refresh processing time between the CPU module and the CC-Link IE TSN master/local module on the main base unit or the extension base unit is calculated by the following formulas.

$\alpha T1, \alpha R1$ [ms] = $KM1 + KM2 \times ((RX + RY + SB) \div 16 + RWr + RWw + SW) + \alpha U$

$\alpha T2, \alpha R2$ [ms] = $KM1 + KM2 \times (LB \div 16 + LW)$

αU [ms] = $KM3 \times (SBU \div 16 + SWU)$

- $\alpha T1$: RX/RY/RWr/RWw/SB/SW link refresh time (sending side)
- $\alpha T2$: LB/LW link refresh time (sending side)
- $\alpha R1$: RX/RY/RWr/RWw/SB/SW link refresh time (receiving side)
- $\alpha R2$: LB/LW link refresh time (receiving side)
- αU: Unit label (SB/SW) refresh time
- RX: Total number of points of remote input (RX) refreshed by the master station/local station[*1]
- RY: Total number of points of remote output (RY) refreshed by the master station/local station[*1]
- LB: Total number of points of link relay (LB) refreshed by the master station/local station station[*1]
- RWw: Total number of points of remote register (RWw) refreshed by the master station/local station[*1]
- RWr: Total number of points of remote register (RWr) refreshed by the master station/local station[*1]
- LW: Total number of points of link register (LW) refreshed by the master station/local station station[*1]
- SB: Number of points of link special relay (SB)[*2]
- SW: Number of points of link special register (SW)[*2]
- SBU: Number of points of link special relay (SB)[*3]
- SWU: Number of points of link special register (SW)[*3]

Condition		Constant value
KM1 ($\times 10^{-3}$)		33
KM2 ($\times 10^{-3}$)	Main base unit[*4]	0.01
	Extension base unit[*4]	0.01
KM3 ($\times 10^{-3}$)	Main base unit[*4]	0.02
	Extension base unit[*4]	0.02

*1 Total number of points here indicates the one of link devices that have been set in "Refresh Setting" and "Network Configuration Settings". The number of points that has been assigned in the reserved station is excluded in the link relay (LB) and the link register (LW).

*2 Number of points here indicates the one of when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.

*3 Number of points here indicates the one of when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.

*4 The base units here indicates the type of base unit where the network module targeted for refresh is mounted.

■Link refresh processing time for the CC-Link IE Controller Network module

The link refresh processing time between the CPU module and the CC-Link IE Controller Network module on the main base unit or the extension base unit is calculated by the following formulas.

αT, αR [ms] = KM1 + KM2 × ((LB + LX + LY + SB) ÷ 16 + LW + SW) + αU

αU [ms] = KM3 × (SBU ÷ 16 + SWU)

- αT: Link refresh time (sending side)
- αR: Link refresh time (receiving side)
- αU: Unit label (SB/SW) refresh time
- LB: Total number of points of link relay (LB) that is refreshed by the station*1
- LW: Total number of points of link register (LW) that is refreshed by the station*1
- LX: Total number of points of link input (LX) that is refreshed by the station*1
- LY: Total number of points of link output (LY) that is refreshed by the station*1
- SB: Number of points of link special relay (SB)*2
- SW: Number of points of link special register (SW)*2
- SBU: Number of points of link special relay (SB)*3
- SWU: Number of points of link special register (SW)*3

Condition		Constant value
KM1 ($\times 10^{-3}$)		33
KM2 ($\times 10^{-3}$)	Main base unit*4	0.01
	Extension base unit*4	0.01
KM3 ($\times 10^{-3}$)	Main base unit*4	0.02
	Extension base unit*4	0.02

*1 Total number of points here indicates the one of link devices that have been set in "Refresh Setting" and "Network Configuration Settings". Note that the number of points that has been assigned in the reserved station is excluded.
*2 Number of points here indicates the one of when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.
*3 Number of points here indicates the one of when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.
*4 The base units here indicates the type of base unit where the network module targeted for refresh is mounted.

A

■Link refresh processing time for the CC-Link IE Field Network module

The link refresh processing time between the CPU module and the CC-Link IE Field Network module on the main base unit or the extension base unit is calculated by the following formulas.

αT, αR [ms] = KM1 + KM2 $\times$ ((RX + RY + SB) $\div$ 16 + RWr + RWw + SW) + αU

αU [ms] = KM3 $\times$ (SBU $\div$ 16 + SWU)

- αT: Link refresh time (sending side)
- αR: Link refresh time (receiving side)
- αU: Unit label (SB/SW) refresh time
- RX: Total number of points of remote input (RX) refreshed by the master station/local station*1
- RY: Total number of points of remote output (RY) refreshed by the master station/local station*1
- RWw: Total number of points of remote register (RWw) refreshed by the master station/local station*1
- RWr: Total number of points of remote register (RWr) refreshed by the master station/local station*1
- SB: Number of points of link special relay (SB)*2
- SW: Number of points of link special register (SW)*2
- SBU: Number of points of link special relay (SB)*3
- SWU: Number of points of link special register (SW)*3

Condition		Constant value
KM1 ($\times 10^{-3}$)		33
KM2 ($\times 10^{-3}$)	Main base unit*4	0.01
	Extension base unit*4	0.01
KM3 ($\times 10^{-3}$)	Main base unit*4	0.02
	Extension base unit*4	0.02

*1 Total number of points here indicates the one of link devices that have been set in "Refresh Setting" and "Network Configuration Settings".

*2 Number of points here indicates the one of when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.

*3 Number of points here indicates the one of when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.

*4 The base units here indicates the type of base unit where the network module targeted for refresh is mounted.

■Link refresh processing time for the CC-Link module

The link refresh processing time between the CPU module and the CC-Link module on the main base unit or the extension base unit is calculated by the following formulas. (Remote net Ver.1 mode, Remote net Ver.2 mode)

αT, αR [ms] = KM1 + KM2 $\times$ ((RX + RY + SB) $\div$ 16 + RWr + RWw + SW) + αU

αU [ms] = KM3 $\times$ (SBU $\div$ 16 + SWU)

- αT: Link refresh time (sending side)
- αR: Link refresh time (receiving side)
- αU: Unit label (SB/SW) refresh time
- RX: Total number of points of remote input (RX) refreshed by the master station/local station*1
- RY: Total number of points of remote output (RY) refreshed by the master station/local station*1
- RWw: Total number of points of remote register (RWw) refreshed by the master station/local station*1
- RWr: Total number of points of remote register (RWr) refreshed by the master station/local station*1
- SB: Number of points of link special relay (SB)*2
- SW: Number of points of link special register (SW)*2
- SBU: Number of points of link special relay (SB)*3
- SWU: Number of points of link special register (SW)*3

Condition		Constant value
KM1 ($\times 10^{-3}$)		33
KM2 ($\times 10^{-3}$)	Main base unit*4	0.01
	Extension base unit*4	0.01
KM3 ($\times 10^{-3}$)	Main base unit*4	0.02
	Extension base unit*4	0.02

*1 Total number of points here indicates the one of link devices that have been set in "Refresh Setting" and "Network Configuration Settings".

*2 Number of points here indicates the one of when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.

*3 Number of points here indicates the one of when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.

*4 The base units here indicates the type of base unit where the network module targeted for refresh is mounted.

■Link refresh processing time for the MELSECNET/H network module

The link refresh processing time between the CPU module and the MELSEC iQ-R series MELSECNET/H network module on the main base unit or the extension base unit is calculated by the following formulas.

αT, αR [ms] = KM1 + KM2 × ((LB + LX + LY + SB) ÷ 16 + LW + SW) + αU

αU [ms] = KM3 × (SBU ÷ 16 + SWU)

- αT: Link refresh time (sending side)
- αR: Link refresh time (receiving side)
- αU: Unit label (SB/SW) refresh time
- LB: Total number of points of link relay (LB) that is refreshed by the station*1
- LW: Total number of points of link register (LW) that is refreshed by the station*1
- LX: Total number of points of link input (LX) that is refreshed by the station*1
- LY: Total number of points of link output (LY) that is refreshed by the station*1
- SB: Number of points of link special relay (SB)*2
- SW: Number of points of link special register (SW)*2
- SBU: Number of points of link special relay (SB)*3
- SWU: Number of points of link special register (SW)*3

Condition		Constant value
KM1 ($\times 10^{-3}$)		33
KM2 ($\times 10^{-3}$)	Main base unit*4	0.01
	Extension base unit*4	0.01
KM3 ($\times 10^{-3}$)	Main base unit*4	0.02
	Extension base unit*4	0.02

*1 Total number of points here indicates the one of link devices that have been set in "Refresh Setting" and "Network Configuration Settings". Note that the number of points that has been assigned in the reserved station is excluded.
*2 Number of points here indicates the one of when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.
*3 Number of points here indicates the one of when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.
*4 The base units here indicates the type of base unit where the network module targeted for refresh is mounted.

The link refresh processing time between the CPU module and the MELSEC-Q series MELSECNET/H network module on the RQ extension base unit or the Q series extension base unit is calculated by the following formulas.

αT, αR [ms] = KM1 + KM2 × ((LB + LX + LY + SB) ÷ 16 + LW + SW)

- αT: Link refresh time (sending side)
- αR: Link refresh time (receiving side)
- LB: Total number of points of link relay (LB) that is refreshed by the station*5
- LW: Total number of points of link register (LW) that is refreshed by the station*5
- LX: Total number of points of link input (LX) that is refreshed by the station*5
- LY: Total number of points of link output (LY) that is refreshed by the station*5
- SB: Number of points of link special relay (SB)
- SW: Number of points of link special register (SW)

Condition		Constant value
KM1 ($\times 10^{-3}$)		65
KM2 ($\times 10^{-3}$)	RQ extension base unit (RQ6☐B)*6	0.41
	Q series extension base unit (Q5☐B/Q6☐B)*6	0.92

*5 Total number of points here indicates the one of link devices that have been set in "Refresh Setting" and "Network Configuration Settings". Note that the number of points that has been assigned in the reserved station is excluded.
*6 The base units here indicates the type of base unit where the network module targeted for refresh is mounted.

Link refresh processing time for CC-Link IE Field Network Basic

The link refresh processing time for CC-Link IE Field Network Basic (the increase in END processing time for the CPU module) can be calculated by the following formula.

αT [μs] = KM1 + KM2 $\times$ (((RX + RY) $\div$ 16) + RWw + RWr) + αE

αE [μs] = KM3 + KM4 $\times$ ((RX + RY) $\div$ 16) + RWw + RWr)

- αT: Link refresh time
- αE: Link refresh time when the file register (R, ZR) is used[*1]
- RX: Total number of points of remote input (RX) refreshed by the master station[*2]
- RY: Total number of points of remote output (RY) refreshed by the master station[*2]
- RWw: Total number of points of remote register (RWw) refreshed by the master station[*2]
- RWr: Total number of points of remote register (RWr) refreshed by the master station[*2]

Condition		Constant value
KM1		12.5
KM2		0.02
KM3		14.5
KM4	When the extended SRAM cassette or battery-less option cassette is not used	0.04
	When the extended SRAM cassette or battery-less option cassette is used	0.11

*1 This is added when file register (R, ZR) is used.

*2 This value is determined according to the number of device stations to be connected and the number of occupied stations.

A

Intelligent function module/RJ71GN11-EIP (EtherNet/IP part) refresh processing time

The following calculation formula shows the refresh processing time for the intelligent function module or the RJ71GN11-EIP (EtherNet/IP part) mounted on the main base unit and the extension base unit.

Point

The refresh processing time described in this manual is for the case when the "Target" is set to "Device" in the refresh setting. For the refresh processing time of when the refreshing target is set to the module label or refresh data register (RD), refer to the manual for the module used.

Refresh processing time of one module [μs] = Read refresh time*1 + Write refresh time*1

*1 When the number of settings of read refresh (Module → CPU module) or write refresh (CPU module → Module) is 0, each processing time is 0.

■Read refresh time [μs]

Number of read refresh settings × KM1 + Refresh time for the first item (A) + Refresh time for the second item (A) + ··· + Refresh time for the nth item (A) + KM2

■Write refresh time [μs]

Number of write refresh settings × KM4 + Refresh time for the first item (B) + Refresh time for the second item (B) + ··· + Refresh time for the nth item (B) + KM5

- A: KM3 × Number of refresh transfer (word) [μs]
- B: KM6 × Number of refresh transfer (word) [μs]
- n: Number of blocks for refresh settings*1
- KM1 to KM6: Constant value as shown below

Condition			Constant value
MELSEC iQ-R series module	KM1	Module on the main base unit → CPU module	0.98
		Module on the extension base unit → CPU module	0.98
	KM2	Module on the main base unit → CPU module	11.6
		Module on the extension base unit → CPU module	11.6
	KM3	Module on the main base unit → CPU module	0.05
		Module on the extension base unit → CPU module	0.05
	KM4	CPU module → Main base unit	0.58
		CPU module → Extension base unit	0.58
	KM5	CPU module → Main base unit	9.10
		CPU module → Extension base unit	9.10
	KM6	CPU module → Main base unit	0.01
		CPU module → Extension base unit	0.01
Q series module	KM1	Module on the RQ extension base unit → CPU module	1.47
		Module on the Q series extension base unit → CPU module	2.92
	KM2	Module on the RQ extension base unit → CPU module	21.2
		Module on the Q series extension base unit → CPU module	20.2
	KM3	Module on the RQ extension base unit → CPU module	0.38
		Module on the Q series extension base unit → CPU module	0.91
	KM4	CPU module → Module on the RQ extension base unit	0.83
		CPU module → Module on the Q series extension base unit	1.20
	KM5	CPU module → Module on the RQ extension base unit	15.8
		CPU module → Module on the Q series extension base unit	15.0
	KM6	CPU module → Module on the RQ extension base unit	0.43
		CPU module → Module on the Q series extension base unit	0.97

*1 The number of blocks for refresh settings can be checked in "Auto Refresh Setting Total Counts" of "Module Parameter List" window. For details, refer to the manual for the module used.

Multiple CPU refresh processing time

This section describes the Multiple CPU refresh processing time.

Refresh processing time [μs] = Send refresh time + Receive refresh time

Send refresh time [μs] = KM1 + KM2 × Number of points of send word

Receive refresh time [μs] = KM3 + KM4 × Number of other CPU modules + KM5 × Number of points of receive word

■Refresh (when END)

The following table lists the constant values when refresh is executed in END processing at CPU buffer memory area.

Constant	Constant value
KM1	6
KM2	0.01
KM3	5
KM4	14
KM5	0.011

■Refresh (when I45 is executed)

The following table lists the constant values when refresh is executed in Multiple CPU synchronous interrupt (I45) at fixed scan communication area.

Constant	Constant value
KM1	20
KM2	0.007
KM3	20
KM4	4
KM5	0.007

A

Prolonged time of END processing when executing each function

This section describes the prolonged time of END processing when executing each function.

■Latch processing time

If the latch function is used and the range for the latch time setting is effective, the scan time becomes longer. Also when the time setting is configured for the latch time setting, the scan time may be prolonged in the next END processing after the specified time has passed. The increase in scan time when the latch range is set can be calculated by the following formula.

Increase in scan time[μs] = (KM1 × Number of settings of latch range setting[*1]) + (KM2 × (Number of points of bit device with latch specified ÷ 16 + Number of points of word device with latch specified + Number of points of double word device with latch specified × 2)) + KM3

*1 The latch range (1) and the latch range (2) are counted as different device types.

Condition			Constant value
When set to "each scan"	KM1		1.00
	KM2	When the battery-less option cassette is not used	0.09
		When the battery-less option cassette is used	0.1
	KM3		1.20
When set to "hour"	KM1		1.0
	KM2		0.004
	KM3		17.5

■Data logging function

When the data logging function is executed, the scan time becomes longer. The increase in scan time can be calculated by the following formula.

Increase in scan time [μs] = KM1 + (KM2 × Number of data logging settings) + (KM3 × Number of device points of internal device[*1])

*1 This indicates the total number of points of data logging setting No.1 to No.10.

Condition		Constant value			
		Programmable controller CPU	Process CPU (process mode)	Process CPU (redundant mode)	Safety CPU
When all the columns are output under the condition of file format of Unicode text file	KM1	34	35		34
	KM2	34	37		34
	KM3	0.08	0.35		0.08

■Data logging file transfer function

In the data logging file transfer function, the scan time does not increase.

■Processing time when file register is used

When "Use File Register of Each Program" has been set through setting the file register, the scan time becomes longer. The increase in the scan time [ms] is given by: the number of program files × 0.016.

Point

When "Use Common File Register in All Programs" has been set, the scan time may not become longer.

■Memory dump function

The execution of the memory dump function results in an increase in the scan time. The increase in scan time can be calculated by the following formula.

Increase in scan time [μs] = (KM1 × Number of points[*1]) + KM2

Constant	Constant value		
	R00CPU, R01CPU, R02CPU, R04CPU	Other programmable controller CPUs	Safety CPU
KM1	102.00	61.00	61.00
KM2	10.00	25.00	25.00

*1 This indicates the total number of points (total number of words) of devices that are set in the device area, file storage area, and refresh data register area. (unit: K words).
When the total number of points exceeds the value specified in the internal buffer capacity setting, change the total number to that value (unit: K words).

■Real-time monitor function

The execution of the real-time monitor function results in an increase in the scan time. The increase in scan time can be calculated by the following formula.

Increase in scan time [μs] = KM1 + (KM2 × Number of points of word device) + (KM3 × Number of points of bit device)

- In monitoring global devices (X, Y, M, L, B, F, SB, V, T, ST, C, LT, LST, LC, D, W, SW, FX, FY, SM, FD, SD, BLn\S)/global labels

Constant	Constant value
KM1	149.00
KM2	3.20
KM3	0.80

- In monitoring file registers (R, ZR)

Constant	Constant value
KM1	149.00
KM2	3.50
KM3	1.10

- In monitoring module access devices (Un\G, U3En\G, U3En\HG)/link direct devices (Jn\W, Jn\X, Jn\SW, Jn\Y, Jn\SB, Jn\B)

Constant	Constant value	
	R00/R01/R02CPU	Other CPU modules
KM1	180.00	156.00
KM2	11.30	11.60
KM3	12.80	8.90

- In monitoring local devices (M, V, T, ST, C, LT, LST, LC, D)/local labels

Constant	Constant value	
	R00/R01/R02CPU	Other CPU modules
KM1	149.00	149.00
KM2	8.80	6.20
KM3	5.80	3.10

■Sequence scan synchronization sampling function

When the sequence scan synchronization sampling function is executed, the scan time becomes longer. The increase in scan time can be calculated by the following formula.

Increase in scan time [μs] = (KM1 × Number of processing times[*1]) + (KM2 × Total number of points of device that has been set[*2]) + (KM3 × Number of settings[*2]) + (KM4 × Number of modules[*3]) + KM5

*1 The number of processing times indicates the number of times to perform the processing for 64 points of the device that can be processed at a time. The number of times is calculated by the total number of points of device that has been set n ÷ 64. (The decimal point is rounded up.)
*2 For the device types and number of settings that can be set, refer to the manual for the module used.
*3 The number of modules indicates the total number of modules which has executed this function.

Constant	Constant value
KM1	8.00
KM2	0.42
KM3	0.20
KM4	24.00
KM5	85.00

■File batch online change

When the file batch online change is executed, the scan time becomes longer. The increase in scan time [ms] can be calculated by the following formula.

Condition*1	Calculation formula	Constant value
When SM388 is off (Program file only)	(KM1 × Number of program files*2) + (KM2 × Number of steps in the largest program*3) + KM3	KM1: 0.018 KM2: 0.06×10^{-3} KM3: 0.9
When SM388 is on (Program file/FB file/global label setting file)	(KM4 × Number of program files to be written*4) + (KM5 × Total number of steps in programs*5)	KM4: 2.34 KM5: 0.06×10^{-3}

*1 For R00CPU, R01CPU, R02CPU, calculate with calculation formula for "when SM388 is on" in the above table. For R00CPU, R01CPU, R02CPU, the operation setting status is set to "program file/FB file/global label setting file" regardless of the SM388 setting.
*2 This indicates the number of program files that are written to the CPU module.
*3 This indicates the number of steps in the largest program file among the program files to which the file batch online change is performed.
*4 This indicates the number of program files to which the file batch online change is performed.
*5 This indicates the total number of steps in the program files to which the file batch online change is performed.

For the Safety CPU, the increase in scan time [ms] can be calculated by the following formula.

Condition	Calculation formula	Constant value
When SM388 is off (Standard/safety program file only)	(KM1 × Number of standard program files*6) + (KM2 × Number of steps in the largest standard program*7) + KM3 + (KM4 × Number of safety program files*8) + (KM5 × Number of steps in the largest safety program*9)	KM1: 0.02 KM2: 0.062×10^{-3} KM3: 0.9 KM4: 0.4 KM5: 0.8×10^{-3}
When SM388 is on (Standard/safety program file, standard/safety FB file, standard/safety global label setting file, standard/safety shared label setting file)	(KM6 × Number of standard program files to be written*10) + (KM7 × Total number of steps in standard programs*11) + (KM8 × Number of safety program files to be written*12) + (KM9 × Total number of steps in safety programs*13)	KM6: 2.38 KM7: 0.062×10^{-3} KM8: 0.4 KM9: 1.20×10^{-3}

*6 This indicates the number of standard program files that are written to the CPU module.
*7 This indicates the number of steps in the largest standard program file among the standard program files to which the file batch online change is performed.
*8 This indicates the number of safety program files that are written to the CPU module.
*9 This indicates the number of steps in the largest safety program file among the safety program files to which the file batch online change is performed.
*10 This indicates the number of standard program files to which the file batch online change is performed.
*11 This indicates the total number of steps in the standard program files to which the file batch online change is performed.
*12 This indicates the number of safety program files to which the file batch online change is performed.
*13 This indicates the total number of steps in the safety program files to which the file batch online change is performed.

Point

The number of steps for each program file can be checked [Confirm Memory Size (Offline)] in [Tool] of the engineering tool.

■Simple CPU communication function

The processing time of the simple CPU communication function (the increase in END processing time for the CPU module) can be calculated by the following formula.

Processing time of the simple CPU communication function [μs] = KM1 + αT_1 + αT_2 + ... + αT_n

- αT_n [μs] = KM2 + αB_n + αW_n
- αB_n [μs] = KM3 × (number of bit points to be communicated ÷ 16)
- αW_n*1 [μs] = KM3 × (number of word points to be communicated)

*1 αE_n in the following is used when the file register (R, ZR) is used.
αE_n [μs] = KM4 + KM5 × (number of word points to be communicated)

- αT_n: Simple CPU communication processing time per setting
- αB_n: Simple CPU communication processing time per setting when the bit device is used
- αW_n: Simple CPU communication processing time per setting when using a word device (except for the file register (R, ZR))
- αE_n: Simple CPU communication processing time when using the file register (R, ZR)
- n: Number of settings

Condition		Constant value
KM1		60.00
KM2		0.90
KM3		0.024
KM4		0.70
KM5	When an extended SRAM cassette is not used	0.05
	When an extended SRAM cassette is used	0.13

Device/label access service processing time

This section describes the device/label access service processing time when "Processing time = One time" in the device/label access service processing setting.

Condition			Device/label access service processing time
Connected via USB	Ladder block change during RUN (online program change)	100 steps are inserted into the head of program of 40K steps.	1.0ms maximum*1
	Monitor data registration	Data register (D) (Number of device points = 32 points) is registered to monitor.	0.07ms maximum
When connecting the Ethernet port (TCP)	Ladder block change during RUN (online program change)	100 steps are inserted into the head of program of 40K steps.	1.0ms maximum*1
	Monitor data registration	Data register (D) (Number of device points = 32 points) is registered to monitor.	0.07ms maximum
When connecting the Ethernet port (UDP)	Ladder block change during RUN (online program change)	100 steps are inserted into the head of program of 40K steps.	1.0ms maximum*1
	Monitor data registration	Data register (D) (Number of device points = 32 points) is registered to monitor.	0.07ms maximum

*1 When data are written to the program using a pointer (P) during RUN, the processing time is extended depending on the pointer number used. For example, when the data are written to the program using P8191 during RUN, the processing time is extended up to 3.0ms.

Processing time of the identification check for safety data

The extended scan time when executing the identification check for safety data is 800 μs under the following conditions.

- Safety programs: 20K steps
- Create safety global label setting file (32 word definition)
- Create standard/safety shared label setting file (32 word definition)
- Safety cycle time: 10ms
- Perform safety communications for 3 connections
- Scan time: 1.5ms (when the identification check for safety data not executed)

Data logging function processing time

This section describes the processing time taken to store the data when executing the data logging function. (The values shown in the tables below are the minimum time values that allow the CPU module to collect data without missing any data while the continuous logging is executed.)

When the file format is Unicode text file

■When a global device or global label is specified

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- When specifying a global label in the Unicode text file format, check the firmware version of the CPU module. (☞ Page 1139 Added and Enhanced Functions) Note that the global label specification is not available for the Process CPU.

The following table lists the collection intervals at which data can be collected under the following conditions.

- Scan time = 1.5ms (up to 3ms)*1
- Internal buffer capacity setting = 128K bytes per setting (default setting)
- Collection setting = Time specification (data collection at time interval)
- Data setting = when a global device is specified: Data register (D) (Data-type: word-signed (decimal number type)), when a global label is specified: Global1 (Data-type: word-signed (decimal number type))
- Output setting = Day/time column (output format is default) and index column are outputted.
- Save setting = file switching timing: 1024K bytes (when the data storage destination memory is the function memory) or 10000 records (when the data storage destination memory is the SD memory card), operation when the number of saved files exceeded: overwriting
- When the data storage destination memory is the SD memory card, functions that require access to files in the SD memory card except for the data logging function are not active.
- The default setting value is used for the file transfer server setting of the CPU Module Logging Configuration Tool (however, the IP address, login user name, and password can be customized).
- Parameters are used only for required settings that can be connected to the FTP server (IP address settings of the module parameter only) and default setting values are used for other settings.

*1 3ms (to 4.5ms) for the Process CPU (redundant mode)

Number of points		File transfer		Collection interval at which data can be collected						
				When the function memory is used	When the SD memory card is used					
					NZ1MEM-2GBSD			NZ1MEM-4/8/16GBSD		
					Programmable controller CPU	Process CPU	Safety CPU	Programmable controller CPU	Process CPU	Safety CPU
8 points	(8 points × 1 setting)	Not performed		—	1.0ms	1.0ms	1.0ms	1.0ms	1.0ms	1.0ms
		Performed	No file deletion specification at transfer completion	—	1.0ms	—	—	1.0ms	—	—
			File deletion specification at transfer completion	1.0ms	1.0ms	—	—	1.0ms	—	—
16 points	(16 points × 1 setting)	Not performed		—	1.0ms	1.0ms	1.0ms	1.0ms	2.0ms	1.0ms
		Performed	No file deletion specification at transfer completion	—	1.0ms	—	—	1.0ms	—	—
			File deletion specification at transfer completion	1.0ms	1.0ms	—	—	1.0ms	—	—

Number of points		File transfer		Collection interval at which data can be collected						
				When the function memory is used	When the SD memory card is used					
					NZ1MEM-2GBSD			NZ1MEM-4/8/16GBSD		
					Programmable controller CPU	Process CPU	Safety CPU	Programmable controller CPU	Process CPU	Safety CPU
64 points	(64 points × 1 setting)	Not performed		—	1.0ms	2.0ms	1.0ms	2.0ms	5.0ms	2.0ms
		Performed	No file deletion specification at transfer completion	—	1.0ms	—	—	2.0ms	—	—
			File deletion specification at transfer completion	1.0ms	1.0ms	—	—	3.0ms	—	—
128 points	(128 points × 1 setting)	Not performed		—	1.0ms	4.0ms	1.0ms	4.0ms	7.0ms	4.0ms
		Performed	No file deletion specification at transfer completion	—	2.0ms	—	—	5.0ms	—	—
			File deletion specification at transfer completion	1.0ms	2.0ms	—	—	6.0ms	—	—
256 points	(128 points × 2 setting)	Not performed		—	4.0ms	7.0ms	4.0ms	7.0ms (For the R04(EN)CPU: 5.0ms)	13.0ms	7.0ms
		Performed	No file deletion specification at transfer completion	—	4.0ms	—	—	8.0ms	—	—
			File deletion specification at transfer completion	1.0ms	4.0ms	—	—	8.0ms	—	—
1280 points	(128 points × 10 setting)	Not performed		—	18.0ms	24.0ms	18.0ms	20.0ms	26.0ms	20.0ms
		Performed	No file deletion specification at transfer completion	—	19.0ms	—	—	23.0ms	—	—
			File deletion specification at transfer completion	6.0ms	19.0ms	—	—	24.0ms (For the R04(EN)CPU: 23.0ms)	—	—

■When a local device or local label is specified

RnCPU RnENCPU ~~RnPCPU (Process)~~ ~~RnPCPU (Redundant)~~ ~~RnPSFCPU (Standard)~~ ~~RnPSFCPU (Safety)~~ RnSFCPU (Standard) ~~RnSFCPU (Safety)~~

• When specifying a local device/local label in the Unicode text file format, check the firmware version of the CPU module. (☞ Page 1139 Added and Enhanced Functions)

The following table lists the collection intervals at which data can be collected under the following conditions.

- Scan time = 1.5ms (up to 5ms)
- Internal buffer capacity setting = 128K bytes per setting (default setting)
- Collection setting = Time specification (data collection at time interval)
- Data setting = when a local device is specified: MAIN/#D0 (Data-type: word-signed (decimal number type)), when a local label is specified: Program name "MAIN", ProgPou/local1 (Data-type: word-signed)
- Output setting = Day/time column (output format is default) and index column are output.
- Save setting = file switching timing: 1024K bytes (when the data storage destination memory is the function memory) or 10000 records (when the data storage destination memory is the SD memory card), operation when the number of saved files exceeded: overwriting
- When the data storage destination memory is the SD memory card, functions that require access to files in the SD memory card except for the data logging function are not active.
- The default setting value is used for the file transfer server setting of the CPU Module Logging Configuration Tool (however, the IP address, login user name, and password can be customized).
- Parameters are used only for required settings that can be connected to the FTP server (IP address settings of the module parameter only) and default setting values are used for other settings.

Number of points		File transfer		Collection interval at which data can be collected		
				When the function memory is used	When the SD memory card is used	
					NZ1MEM-2GBSD	NZ1MEM-4/8/16GBSD
8 points	(8 points × 1 setting)	Not performed		—	1.0ms	1.0ms
		Performed*1	No file deletion specification at transfer completion	—	1.0ms	1.0ms
			File deletion specification at transfer completion	1.0ms	1.0ms	1.0ms
16 points	(16 points × 1 setting)	Not performed		—	1.0ms	1.0ms
		Performed*1	No file deletion specification at transfer completion	—	1.0ms	1.0ms
			File deletion specification at transfer completion	1.0ms	1.0ms	1.0ms
64 points	(64 points × 1 setting)	Not performed		—	1.0ms	2.0ms
		Performed*1	No file deletion specification at transfer completion	—	1.0ms	3.0ms
			File deletion specification at transfer completion	1.0ms	1.0ms	3.0ms
128 points	(128 points × 1 setting)	Not performed		—	3.0ms	5.0ms
		Performed*1	No file deletion specification at transfer completion	—	4.0ms	6.0ms
			File deletion specification at transfer completion	1.0ms	4.0ms	6.0ms
256 points	(128 points × 2 setting)	Not performed		—	6.0ms	6.0ms
		Performed*1	No file deletion specification at transfer completion	—	7.0ms	7.0ms
			File deletion specification at transfer completion	1.0ms	7.0ms	7.0ms
1280 points	(128 points × 10 setting)	Not performed		—	25.0ms	30.0ms
		Performed*1	No file deletion specification at transfer completion	—	30.0ms	35.0ms
			File deletion specification at transfer completion	6.0ms	30.0ms	35.0ms

*1 The file transfer (data logging file transfer function) is not available for the Safety CPU.

When the file format is CSV

■When a global device or global label is specified

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

• When specifying a global label in the CSV file format, check the firmware version of the CPU module. Note that the global label specification is not available for the Process CPU. (☞ Page 1139 Added and Enhanced Functions)

The following table lists the collection intervals at which data can be collected under the following conditions.

- Scan time = 1.5ms (up to 3ms)
- Internal buffer capacity setting = 128K bytes per setting (default setting)
- Collection setting = Time specification (data collection at time interval)
- Data setting = when a global device is specified: Data register (D) (Data-type: word-signed (decimal number type)), when a global label is specified: Global1 (Data-type: word-signed (decimal number type))
- Output setting = Day/time column (output format is default) and index column are output.
- Save setting = file switching timing: 1024K bytes (when the data storage destination memory is the function memory) or 10000 records (when the data storage destination memory is the SD memory card), operation when the number of saved files exceeded: overwriting
- When the data storage destination memory is the SD memory card, functions that require access to files in the SD memory card except for the data logging function are not active.
- The default setting value is used for the file transfer server setting of the CPU Module Logging Configuration Tool (however, the IP address, login user name, and password can be customized).
- Parameters are used only for required settings that can be connected to the FTP server (IP address settings of the module parameter only) and default setting values are used for other settings.

Number of points		File transfer		Collection interval at which data can be collected		
				When the function memory is used	When the SD memory card is used: NZ1MEM-2GBSD	When the SD memory card is used: NZ1MEM-4/8/16GBSD
8 points	(8 points × 1 setting)	Not performed		—	1.0ms	1.0ms
		Performed	No file deletion specification at transfer completion	—	1.0ms	1.0ms
			File deletion specification at transfer completion	1.0ms	1.0ms	1.0ms
16 points	(16 points × 1 setting)	Not performed		—	1.0ms	1.0ms
		Performed	No file deletion specification at transfer completion	—	1.0ms	1.0ms
			File deletion specification at transfer completion	1.0ms	1.0ms	1.0ms
64 points	(64 points × 1 setting)	Not performed		—	1.0ms	2.0ms
		Performed	No file deletion specification at transfer completion	—	1.0ms	2.0ms
			File deletion specification at transfer completion	1.0ms	1.0ms	3.0ms
128 points	(128 points × 1 setting)	Not performed		—	1.0ms	4.0ms
		Performed	No file deletion specification at transfer completion	—	2.0ms	5.0ms
			File deletion specification at transfer completion	1.0ms	2.0ms	6.0ms
256 points	(128 points × 2 setting)	Not performed		—	4.0ms	7.0ms (For the R04(EN)CPU: 5.0ms)
		Performed	No file deletion specification at transfer completion	—	4.0ms	8.0ms
			File deletion specification at transfer completion	1.0ms	4.0ms	8.0ms

Number of points		File transfer		Collection interval at which data can be collected		
				When the function memory is used	When the SD memory card is used	
					NZ1MEM-2GBSD	NZ1MEM-4/8/16GBSD
1280 points	(128 points × 10 setting)	Not performed		—	18.0ms	20.0ms
		Performed	No file deletion specification at transfer completion	—	19.0ms	23.0ms
			File deletion specification at transfer completion	6.0ms	19.0ms	24.0ms (For the R04(EN)CPU: 23.0ms)

■When a local device or local label is specified

RnCPU RnENCPU ~~RnPCPU (Process)~~ ~~RnPCPU (Redundant)~~ ~~RnPSFCPU (Standard)~~ ~~RnPSFCPU (Safety)~~ ~~RnSFCPU (Standard)~~ ~~RnSFCPU (Safety)~~

- When specifying a local device/global label in the CSV file format, check the firmware version of the CPU module. (☞ Page 1139 Added and Enhanced Functions)

The following table lists the collection intervals at which data can be collected under the following conditions.

- Scan time = 1.5ms (up to 5ms)
- Internal buffer capacity setting = 128K bytes per setting (default setting)
- Collection setting = Time specification (data collection at time interval)
- Data setting = when a local device is specified: MAIN/#D0 (Data-type: word-signed (decimal number type)), when a local label is specified: Program name "MAIN", ProgPou/local1 (Data-type: word-signed)
- Output setting = Day/time column (output format is default) and index column are output.
- Save setting = file switching timing: 1024K bytes (when the data storage destination memory is the function memory) or 10000 records (when the data storage destination memory is the SD memory card), operation when the number of saved files exceeded: overwriting
- When the data storage destination memory is the SD memory card, functions that require access to files in the SD memory card except for the data logging function are not active.
- The default setting value is used for the file transfer server setting of the CPU Module Logging Configuration Tool (however, the IP address, login user name, and password can be customized).
- Parameters are used only for required settings that can be connected to the FTP server (IP address settings of the module parameter only) and default setting values are used for other settings.

Number of points		File transfer		Collection interval at which data can be collected		
				When the function memory is used	When the SD memory card is used	
					NZ1MEM-2GBSD	NZ1MEM-4/8/16GBSD
8 points	(8 points × 1 setting)	Not performed		—	1.0ms	1.0ms
		Performed	No file deletion specification at transfer completion	—	1.0ms	1.0ms
			File deletion specification at transfer completion	1.0ms	1.0ms	1.0ms
16 points	(16 points × 1 setting)	Not performed		—	1.0ms	1.0ms
		Performed	No file deletion specification at transfer completion	—	1.0ms	1.0ms
			File deletion specification at transfer completion	1.0ms	1.0ms	1.0ms
64 points	(64 points × 1 setting)	Not performed		—	1.0ms	2.0ms
		Performed	No file deletion specification at transfer completion	—	1.0ms	3.0ms
			File deletion specification at transfer completion	1.0ms	1.0ms	3.0ms
128 points	(128 points × 1 setting)	Not performed		—	3.0ms	5.0ms
		Performed	No file deletion specification at transfer completion	—	4.0ms	6.0ms
			File deletion specification at transfer completion	1.0ms	4.0ms	6.0ms
256 points	(128 points × 2 setting)	Not performed		—	6.0ms	6.0ms
		Performed	No file deletion specification at transfer completion	—	7.0ms	7.0ms
			File deletion specification at transfer completion	1.0ms	7.0ms	7.0ms
1280 points	(128 points × 10 setting)	Not performed		—	25.0ms	30.0ms
		Performed	No file deletion specification at transfer completion	—	30.0ms	35.0ms
			File deletion specification at transfer completion	6.0ms	30.0ms	35.0ms

When the file format is binary file

■When a global device or global label is specified

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) ~~RnPSFCPU (Standard)~~ ~~RnPSFCPU (Safety)~~ RnSFCPU (Standard) ~~RnSFCPU (Safety)~~

• When specifying a global label in the binary file format, check the firmware version of the CPU module. (☞ Page 1139 Added and Enhanced Functions) Note that the global label specification is not available for the Process CPU.

The following table lists the collection intervals at which data can be collected under the following conditions.

- Scan time = 1.5ms (up to 3ms)*1
- Internal buffer capacity setting = 128K bytes per setting (default setting)
- Collection setting = Time specification (data collection at time interval)
- Data setting = when a global device is specified: Data register (D) (Data-type: word-signed (decimal number type)), when a global label is specified: Global1 (Data-type: word-signed (decimal number type))
- Output setting = Day/time column (output format is default) and index column are output.
- Save setting = file switching timing: 1024K bytes (when the data storage destination memory is the function memory) or 10000 records (when the data storage destination memory is the SD memory card), operation when the number of saved files exceeded: overwriting
- When the data storage destination memory is the SD memory card, functions that require access to files in the SD memory card except for the data logging function are not active.
- The default setting value is used for the file transfer server setting of the CPU Module Logging Configuration Tool (however, the IP address, login user name, and password can be customized).
- Parameters are used only for required settings that can be connected to the FTP server (IP address settings of the module parameter only) and default setting values are used for other settings.

*1 3ms (to 4.5ms) for the Process CPU (redundant mode)

Number of points		File transfer		Collection interval at which data can be collected						
				When the function memory is used	When the SD memory card is used					
					NZ1MEM-2GBSD			NZ1MEM-4/8/16GBSD		
					Programmable controller CPU	Process CPU	Safety CPU	Programmable controller CPU	Process CPU	Safety CPU
8 points	(8 points × 1 setting)	Not performed		—	1.0ms	1.0ms	1.0ms	1.0ms	1.0ms	1.0ms
		Performed	No file deletion specification at transfer completion	—	1.0ms	—	—	1.0ms	—	—
			File deletion specification at transfer completion	1.0ms	1.0ms	—	—	1.0ms	—	—
16 points	(16 points × 1 setting)	Not performed		—	1.0ms	1.0ms	1.0ms	1.0ms	2.0ms	1.0ms
		Performed	No file deletion specification at transfer completion	—	1.0ms	—	—	1.0ms	—	—
			File deletion specification at transfer completion	1.0ms	1.0ms	—	—	1.0ms	—	—

Number of points		File transfer		Collection interval at which data can be collected						
				When the function memory is used	When the SD memory card is used					
					NZ1MEM-2GBSD			NZ1MEM-4/8/16GBSD		
					Programmable controller CPU	Process CPU	Safety CPU	Programmable controller CPU	Process CPU	Safety CPU
64 points	(64 points × 1 setting)	Not performed		—	1.0ms	2.0ms	1.0ms	1.0ms	5.0ms	1.0ms
		Performed	No file deletion specification at transfer completion	—	1.0ms	—	—	1.0ms	—	—
			File deletion specification at transfer completion	1.0ms	1.0ms	—	—	2.0ms (For the R04(EN)CPU: 1.0ms)	—	—
128 points	(128 points × 1 setting)	Not performed		—	1.0ms	4.0ms	1.0ms	3.0ms	7.0ms	3.0ms
		Performed	No file deletion specification at transfer completion	—	2.0ms	—	—	4.0ms (For the R04(EN)CPU: 3.0ms)	—	—
			File deletion specification at transfer completion	1.0ms	2.0ms	—	—	4.0ms (For the R04(EN)CPU: 3.0ms)	—	—
256 points	(128 points × 2 setting)	Not performed		—	4.0ms	7.0ms	4.0ms	5.0ms	13.0ms	5.0ms
		Performed	No file deletion specification at transfer completion	—	4.0ms	—	—	6.0ms	—	—
			File deletion specification at transfer completion	1.0ms	4.0ms	—	—	7.0ms (For the R04(EN)CPU: 6.0ms)	—	—
1280 points	(128 points × 10 setting)	Not performed		—	13.0ms	24.0ms	13.0ms	18.0ms	26.0ms	18.0ms
		Performed	No file deletion specification at transfer completion	—	17.0ms	—	—	20.0ms (For the R04(EN)CPU: 18.0ms)	—	—
			File deletion specification at transfer completion	5.0ms	17.0ms	—	—	21.0ms (For the R04(EN)CPU: 18.0ms)	—	—

A

■When a local device or local label is specified

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

• When specifying a local device/local label in the binary file format, check the firmware version of the CPU module. (☞ Page 1139 Added and Enhanced Functions)

The following table lists the collection intervals at which data can be collected under the following conditions.

- Scan time = 1.5ms (up to 5ms)
- Internal buffer capacity setting = 128K bytes per setting (default setting)
- Collection setting = Time specification (data collection at time interval)
- Data setting = when a local device is specified: MAIN/#D0 (Data-type: word-signed (decimal number type)), when a local label is specified: Program name "MAIN", ProgPou/local1 (Data-type: word-signed)
- Output setting = Day/time column (output format is default) and index column are output.
- Save setting = file switching timing: 1024K bytes (when the data storage destination memory is the function memory) or 10000 records (when the data storage destination memory is the SD memory card), operation when the number of saved files exceeded: overwriting
- When the data storage destination memory is the SD memory card, functions that require access to files in the SD memory card except for the data logging function are not active.
- The default setting value is used for the file transfer server setting of the CPU Module Logging Configuration Tool (however, the IP address, login user name, and password can be customized).
- Parameters are used only for required settings that can be connected to the FTP server (IP address settings of the module parameter only) and default setting values are used for other settings.

Number of points		File transfer		Collection interval at which data can be collected		
				When the function memory is used	When the SD memory card is used	
					NZ1MEM-2GBSD	NZ1MEM-4/8/16GBSD
8 points	(8 points × 1 setting)	Not performed		—	1.0ms	1.0ms
		Performed*1	No file deletion specification at transfer completion	—	1.0ms	1.0ms
			File deletion specification at transfer completion	1.0ms	1.0ms	1.0ms
16 points	(16 points × 1 setting)	Not performed		—	1.0ms	1.0ms
		Performed*1	No file deletion specification at transfer completion	—	1.0ms	1.0ms
			File deletion specification at transfer completion	1.0ms	1.0ms	1.0ms
64 points	(64 points × 1 setting)	Not performed		—	1.0ms	1.0ms
		Performed*1	No file deletion specification at transfer completion	—	1.0ms	2.0ms
			File deletion specification at transfer completion	1.0ms	1.0ms	2.0ms
128 points	(128 points × 1 setting)	Not performed		—	2.0ms	5.0ms
		Performed*1	No file deletion specification at transfer completion	—	3.0ms	6.0ms
			File deletion specification at transfer completion	1.0ms	3.0ms	6.0ms
256 points	(128 points × 2 setting)	Not performed		—	4.0ms	6.0ms
		Performed*1	No file deletion specification at transfer completion	—	5.0ms	7.0ms
			File deletion specification at transfer completion	1.0ms	5.0ms	7.0ms
1280 points	(128 points × 10 setting)	Not performed		—	25.0ms	30.0ms
		Performed*1	No file deletion specification at transfer completion	—	25.0ms	35.0ms
			File deletion specification at transfer completion	5.0ms	25.0ms	35.0ms

*1 The file transfer (data logging file transfer function) is not available for the Safety CPU.

Memory dump function processing time

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

This section describes the time (unit: seconds) taken to save all of the data when the memory dump function is in execution.

The following table shows the time taken for data save under the following conditions:

- Scan time = 1.5ms (up to 3ms)
- Internal buffer capacity setting = 256K bytes (by default)
- SD memory card: NZ1MEM-2/4/8/16GBSD used

Number of points*1	Time taken for data save when the memory dump function is in execution	
	NZ1MEM-2GBSD	NZ1MEM-4/8/16GBSD
550.4K points*2	17.00 seconds	25.00 seconds
1224K points*3	35.00 seconds	56.00 seconds
5320K points*4	145.00 seconds	240.00 seconds
9416K points*5	244.00 seconds	404.00 seconds

*1 This indicates the total number of points (total number of words) of devices that are set in the device area, file storage area, and refresh data register area.

*2 This indicates the number of device points (user device: 38.4K points (words), refresh data register: 512K points (words)) with the CPU parameters set to default values.

*3 Settings for this number of points (with no extended SRAM cassette inserted) are as follows.
· User device: 40K points (words), file register: 160K points (words), and refresh data register: 1024K points (words)
· "Use Common File Register in All Programs" is selected in the file register setting.

*4 Settings for this number of points (with an 8M extended SRAM cassette inserted) are as follows.
· Total number of points (total number of words) of user devices, local devices, and file registers: 4296K points (words), and refresh data register: 1024K points (words)
· "Use File Register of Each Program" is selected in the file register setting.
· The number of program files is 10.

*5 Settings for this number of points (with a 16M extended SRAM cassette inserted) are as follows.
· Total number of points (total number of words) of user devices, local devices, and file registers: 8392K points (words), and refresh data register: 1024K points (words)
· "Use File Register of Each Program" is selected in the file register setting.
· The number of program files is 10.

Real-time monitor function processing time

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

This section shows the minimum value (unit: ms) of monitor interval that allows monitoring under the following conditions without data loss during execution of the real-time monitor function.

- Scan time = 1.5ms (up to 3ms)
- Monitor target data = "Number of points" column in the table below (The value of monitor interval does not change when a local device[*1] or label[*1] is specified.)
- Timing setting in the monitor condition setting = Time specification (deselect "Sample data at the next END processing after the specified time has elapsed")
- Internal buffer capacity = 256K bytes (default setting)
- Performance of the system of personal computer = CPU: Intel® Core™ i9-10900K (3.7GHz), Memory: 32GB, OS: Windows®10 Enterprise 22H2 64 bits

Condition	Number of points (in word conversion)	Value of monitor interval
Connected via USB	1 point	1.0ms
	8 points	1.0ms
	16 points	1.0ms
	32 points[*1]	2.0ms
When connecting the Ethernet port (TCP)	1 point	1.0ms
	8 points	1.0ms
	16 points	1.0ms
	32 points[*1]	2.0ms
When connecting the Ethernet port (UDP)	1 point	1.0ms
	8 points	1.0ms
	16 points	1.0ms
	32 points[*1]	2.0ms

*1 When specifying it, check the firmware version of the CPU module. (☞ Page 1139 Added and Enhanced Functions)

Database function processing time

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

Database operation by database access instruction

The table below lists the processing time under the following conditions:

Condition 1	Condition 2
• Number of records: 10000 • Number of fields: 16 The data type of the field is WORD (6 fields), LREAL (6 fields), WSTRING (124 characters) (3 fields), and BOOL (1 field). • Main key: Provided • Functions that require access to files in the SD memory card except for the database function are not active.	• Number of records: 10000 • Number of fields: 64 The data type of the field is WORD (20 fields), LREAL (20 fields), WSTRING (124 characters) (6 fields), and BOOL (18 fields). • Main key: Provided • Functions that require access to files in the SD memory card except for the database function are not active.

This section shows the time taken for the completion signal of the completion device to turn on from the execution of each instruction of the database access instructions.

Instruction			Time taken to complete the execution of instruction	
			Condition 1	Condition 2
Database import	DBIMPORT		155s	255s
Database export	DBEXPORT		75s	195s
Database connection	DBOPEN	First time[*1]	150ms	150ms
		Second time or later	105ms	105ms
Database disconnection	DBCLOSE		30ms	30ms
Database record search	DBSELECT		700ms	750ms
Database record addition	DBINSERT		700ms	800ms
Database record update	DBUPDATE		550ms	600ms
Database record deletion	DBDELETE		700ms	1000ms

*1 After power-on, initial database access involves the diagnostics to check whether the database is corrupted, and thus it takes time to execute the DBOPEN instruction.

■For the programmable controller CPU with firmware version earlier than "28"

The table below lists the processing time under the following conditions:

Condition
• Number of tables: 1 • Number of records: 10000 • Number of fields: 4 The data type of the field is specified as WORD, DWORD, WSTRING (16 characters), and WSTRING (32 characters). • Index: specified • Transaction: Provided (The DBINSERT/DBUPDATE/DBSELECT/DBDELETE instructions only are subject to this condition.) • Functions that require access files in to the SD memory card except for the database access instruction are not active.

This section shows the time taken for the completion signal of the completion device to turn on from the execution of each instruction of the database access instructions.

Instruction			Time taken to complete the execution of instruction
Database import	DBIMPORT		320s
Database export	DBEXPORT		220s
Database connection	DBOPEN	First time*1	590ms
		Second time or later	140ms
Database disconnection	DBCLOSE		3ms
Database record search	DBSELECT		180ms
Database record addition	DBINSERT		210ms
Database record update	DBUPDATE		260ms
Database record deletion	DBDELETE		300ms

*1 After power-on, initial database access involves the diagnostics to check whether the database is corrupted, and thus it takes time to execute the DBOPEN instruction.

Processing time of the CPU module database access (from external device) function

The table below lists the processing time under the following conditions:

Condition 1	Condition 2
• Number of records: 10000 • Number of fields: 16 The data type of the field is specified as WORD (6 fields), DOUBLE PRECISION (6 fields), NLSCHAR(UNICODE) [124 characters] (3 fields), and BOOLEAN (1 field). • Main key: Provided • Functions that require access to files in the SD memory card except for the CPU module database access function are not active.	• Number of records: 10000 • Number of fields: 64 The data type of the field is specified as WORD (20 fields), DOUBLE PRECISION (20 fields), NLSCHAR(UNICODE) [124 characters] (6 fields), and BOOLEAN (18 fields). • Main key: Provided • Functions that require access to files in the SD memory card except for the CPU module database access function are not active.

The following table lists the processing time until the database operation is completed with the CPU module database access function.

Item	Time until the operation is completed	
	Condition 1	Condition 2
Connect to the database of the CPU module from the application (Microsoft Access).	5s	
Select a record in the database of the CPU module from the application (Excel). (SELECT)	3s	3s
Insert 1000 records in the database of the CPU module from the application (Microsoft Access). (INSERT)	230s	330s
Update a record in the database of the CPU module from the application (Microsoft Access). (UPDATE)	1s	1s
Delete a record in the database of the CPU module from the application (Microsoft Access). (DELETE)	1s	1s

Process control function processing time

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

The process control function processing time is the time required to process the control loop when the process control function is executed. The following table lists the processing time of some control loop examples.

Loop type	Configuration	Processing time
Two-degree-of-freedom PID control (S2PID)	S.IN, S.PHPL, S.2PID, and S.OUT1 instructions	160μs
PID control (SPID)	S.IN, S.PHPL, S.PID, and S.OUT1 instructions	151μs
Monitor (SMON)	S.IN and S.PHPL instructions	73μs

Point

For the processing time of each process control instruction, refer to the following.
MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions)

SFC program processing time

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

This section describes the time required for SFC program processing.

Point

For details on the SFC program, refer to the following.
MELSEC iQ-R Programming Manual (Program Design)

SFC program processing performance

The SFC program execution time can be calculated with the following formula.

- SFC program execution time = (A) + (B) + (C)

Item		Description
(A)	SFC processing time	This is the total time shown in the following table.
(B)	Operation output processing time for all steps	This is the total processing time for each instruction used for operation output for all steps in the active status.
(C)	Processing time for all transition conditions	This is the total processing time for each instruction used for transition conditions associated with each step in the active status.

*1 For the processing time for each instruction, refer to the following.
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

The following table lists the types of the SFC processing time (A).

- SFC processing time (A) = (a) + (b) + (c) + (d) + (e) + (f) + (g) + (h)

Item		Processing time calculation (unit: μs)	Description
(a)	Active block processing time	Active block processing time coefficient × Number of active blocks	This is the system processing time required to execute active blocks.
(b)	Inactive block processing time	Inactive block processing time coefficient × Number of inactive blocks	This is the processing time required to execute inactive blocks.
(c)	Nonexistent block processing time	Nonexistent block processing time coefficient × Number of nonexistent blocks	This is the system processing time required to execute blocks that have not been created.
(d)	Active step processing time	Active step processing time coefficient × Number of active steps	This is the time required to execute active steps.
(e)	Active transition processing time	Active transition processing time coefficient × Number of active transitions	This is the system processing time required to execute active transitions.
(f)	Transition establishment step processing time	Transition establishment step processing time coefficient × Number of transitions	This is the time required to turn off active steps when transitions are established.
(g)	SFC END processing time	SFC END processing time = SFC END processing time	This is the system processing time required for SFC END processing.
(h)	Operation output processing time	Action processing time coefficient × Number of actions	This is the system processing time required to process operation outputs.

The following table lists the coefficient values for each processing time.

Item		Coefficient value	
		R00CPU, R01CPU, R02CPU	Other CPU modules
Active block processing time coefficient		3.35	4.4
Inactive block processing time coefficient		2.8	2.7
Nonexistent block processing time coefficient		0.25	0.23
Active step processing time coefficient		12.6	6.8
Active transition processing time coefficient		0.21	0.12
Transition establishment step processing time coefficient	Hold step	30.0	22.5
	Normal step	45.0	31.0
SFC END processing time		45.0	56.5
Operation output processing time coefficient		2.5	0.94

SFC program switching

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

This section describes the processing time required to switch the SFC program from the standby status to the scan execution type.

- Switching processing time [μs] = (Number of blocks created × Km) + (Number of steps created × Kn) + (SFC program capacity × Kp) + Kq

Constant	Constant value	
	R00CPU, R01CPU, R02CPU	Other CPU modules
Km	7.90	7.90*1
Kn	1.85	1.55*1
Kp	0.21	0.21*1
Kq	2550	2500

*1 For the programmable controller CPU with firmware version "30" or earlier and for the Process CPU with firmware version "12" or earlier, the values are as follows.
· R04CPU, R04ENCPU, R08CPU, R08ENCPU, R08PCPU, R16CPU, R16ENCPU, R16PCPU, R32CPU, R32ENCPU, R32PCPU:Km = 3.97, Kn = 0.41, Kp = 0.39
· R120CPU, R120ENCPU, R120PCPU:Km = 3.97, Kn = 0.41, Kp = 0.32

When the SFC program capacity is the following models or conditions, the processing time required to switch the SFC program from the standby status to the scan execution type will be fixed.

Model	Condition	Processing time (constant value)
• The R04CPU, R04ENCPU, R08CPU, R08ENCPU, R16CPU, R16ENCPU, R32CPU, and R32ENCPU with firmware version "30" or earlier • R08PCPU, R16PCPU, R32PCPU	28K steps or less	14ms
• The R120CPU and R120ENCPU with firmware version "30" or earlier • R120PCPU	48K steps or less	18ms

When the SFC information device is set, the processing time required to switch the SFC program from the standby status to the scan execution type will become longer.

A

Redundant function processing time

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

This section describes the redundant function processing time in the Process CPU (redundant mode) or SIL2 Process CPU.

Increase in the scan time due to tracking transfer

The following describes the increase in the scan time of the CPU module due to tracking transfer.

The increase in the scan time determined by the following calculation formula is a rough standard for a system start-up. Check the increase in the scan time with the actual systems.

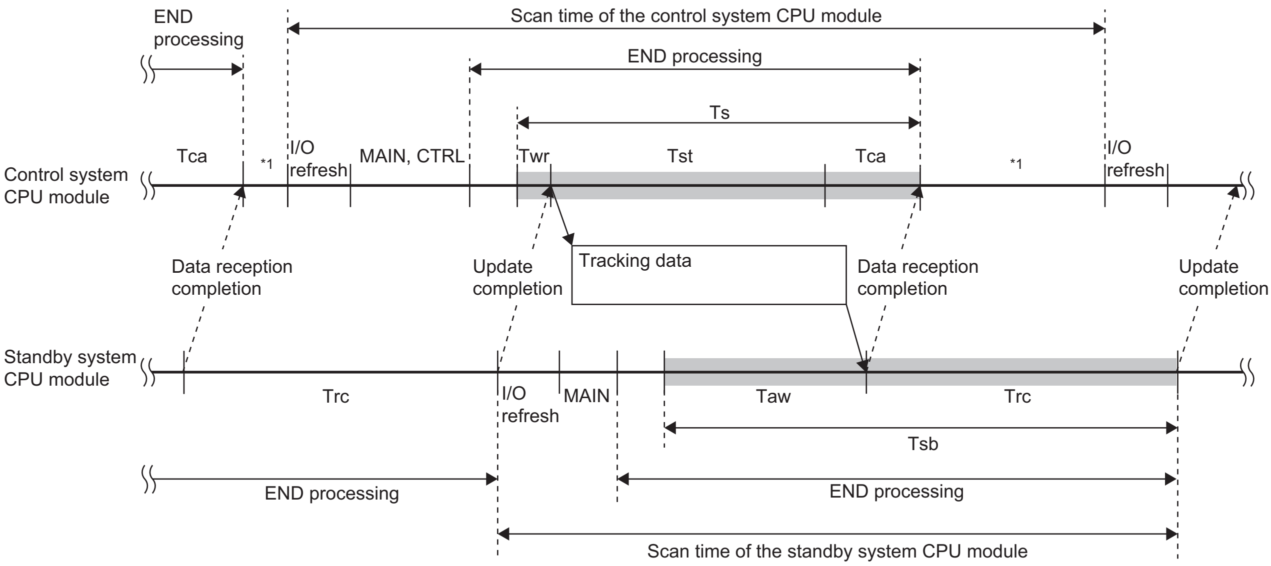
Point

The scan time can be checked in SD520 and SD521 (Current scan time). (☞ Page 986 Fixed scan function information)

Item	Increase in the scan time	Reference
Increase in the scan time of the CPU module in the control system (Ts)	Ts = Twr + Tst + Tca + α [ms]	Twr: Page 1079 Waiting time for completion of the previous tracking data reflection (Twr) Tst: Page 1079 Tracking data send time (Tst) Tca: Page 1081 Waiting time for completion of tracking data reception (Tca) α: Page 1081 Other extended times (α)
Increase in the scan time of the CPU module in the standby system (Tsb)	Tsb = Taw + Trc + α[ms]	Taw: Page 1082 Waiting time for tracking data reception (Taw) Trc: Page 1083 Tracking data reflection time (Trc) α: Page 1083 Other extended times (α)

Ex.

When CTRL is a control system execution program and MAIN is a program executed in both systems (in configuration with the main base unit only*2)



*1 If the constant scan is used, waiting time for the constant scan is generated.

*2 In a redundant system with redundant extension base unit, the processing order in the scan is different (☞ Page 459 Scan Configuration), but the operation is the same as in a configuration with main base units only.

■Waiting time for completion of the previous tracking data reflection (Twr)

The following describes the waiting time (Twr) for completion of the previous tracking data reflection in the CPU module of the control system.

Twr = 1 + Tdrm - Toth [ms]

- Tdrm: Maximum time taken for reflection of tracking data in the CPU module of the standby system
- Toth: Scan time excluding Ts (Increase in the scan time) in the CPU module of the control system

Tdrm is determined as follows.

- When no extended SRAM cassette is inserted: 1 + (No. of tracking blocks $\times$ 300.0 $\times$ 10^{-6}) + (Total size of tracking data [word] $\times$ 45.0 $\times$ 10^{-6})
- When an extended SRAM cassette is inserted: 1 + (No. of tracking blocks $\times$ 300.0 $\times$ 10^{-6}) + (Total size of tracking data [word] $\times$ 106.0 $\times$ 10^{-6})

If Tdrm - Toth is smaller than 0, Twr is handled as 1.

Twr is not generated in the first scan in which the tracking transfer is started.

■Tracking data send time (Tst)

The following describes the tracking data send time (Tst) in the CPU module of the control system. The calculation method differs depending on whether an extended SRAM cassette is inserted or not.

CPU module	Extended SRAM cassette inserting status	Tracking data send time
Process CPU (redundant mode)	Not inserted	0.5 + (26.7 $\times$ 10^{-6}) $\times$ D1 + (43.5 $\times$ 10^{-6}) $\times$ D2 + (1.5 $\times$ 10^{-3}) $\times$ D3 + E1 + F1[ms]
	Inserted	0.5 + (26.7 $\times$ 10^{-6}) $\times$ D1 + (113.5 $\times$ 10^{-6}) $\times$ D2 + (1.5 $\times$ 10^{-3}) $\times$ D3 + E1 + F1[ms]
SIL2 Process CPU	Not inserted	0.58 + (26.7 $\times$ 10^{-6}) $\times$ D4 + (43.5 $\times$ 10^{-6}) $\times$ D5 + (1.5 $\times$ 10^{-3}) $\times$ D6 + (66.0 $\times$ 10^{-6}) $\times$ D7 + F1[ms]
	Inserted	0.58 + (26.7 $\times$ 10^{-6}) $\times$ D4 + (113.5 $\times$ 10^{-6}) $\times$ D5 + (1.5 $\times$ 10^{-3}) $\times$ D6 + (66.0 $\times$ 10^{-6}) $\times$ D7 + F1[ms]

- D1: Size [word] of tracking data in the system data, signal flow memory, refresh data register (RD), and module label (extension base unit)
- D2: Size [word] of tracking data of global devices, local devices, global labels, and local labels
- D3: Number of tracking transfer settings of global devices
- D4: Size [word] of tracking data in the system data, signal flow memory (standard/safety), and refresh data register (RD)
- D5: Size [word] of tracking data of global devices (standard/safety), local devices (standard/safety), global labels (standard/safety), local labels (standard/safety), and standard/safety shared labels
- D6: Number of tracking transfer settings of global devices (standard)
- D7: Total size [word] of tracking data in the system data (safety)*1, signal flow memory (safety), local devices (safety), global labels (safety), local labels (safety), and global devices (safety)
- E1: Additional time when the SFC program is used = 3.5[ms] (Only when the SFC program is used)
- F1: Additional time when the PID control instruction is used = 0.02[ms] (Only when the PID control instruction is used)

*1 For the size of the system data (safety), refer to the value for D4 "■Variable depending on safety communication status" in "System data".

D1 is determined as follows.

Item	Size
System data	■Fixed data • In configuration with the main base unit only: 8225 • In redundant system with redundant extension base unit: 16361 ■Data that varies depending on the tracking device/label setting 16 $\times$ (No. of tracking settings of T, ST, C, and LC) + 8 $\times$ (No. of global device tracking settings of other devices) ■Data that varies depending on the number of CC-Link modules mounted on the extension base unit*2 3131 $\times$ (Number of CC-Link modules mounted on the extension base unit)
Signal flow memory	(Total number of steps of control system execution programs, or total number of steps of rising/falling instruction in an FB)/16 (The digit after the decimal point is round up.)
Refresh data register (RD)	Follows the tracking transfer settings. (☞ Page 520 Tracking transfer setting)
Module label (extension base unit)	■In configuration with the main base unit only 0 ■In redundant system with redundant extension base unit • When transferring the module label of modules on the extension base unit*3 Number of units to be refreshed to the module label of modules on the extension base unit $\times$ 1.7KW (fixed value) • When not transferring the module label of modules on the extension base unit*4 0

A

*2 The CC-Link module that was set the parameter is the target.
*3 This applies when the tracking device/label setting for the redundant settings of the CPU parameter is set to "Transfer collectively" (default) or when the tracking device/label setting is set to "Detailed setting" and "Module Label (Extension) Setting" is set to "Transfer".
*4 This applies when the tracking device/label setting of the redundant settings of the CPU parameter is set to "Detailed setting" and "Module Label (Extension) Setting" is set to "Do Not Transfer".

D2 is determined as follows.

Item	Size*5
Global device (other than the refresh data register (RD))	Follows the tracking transfer settings. (☞ Page 520 Tracking transfer setting)
Local device	Total size of local devices of the device/label memory area detailed setting × the number of programs on the program setting*7
Global label*6	Follows the global label settings.
Local label*6	Follow the local label settings.

*5 The size of the devices/labels to be transferred (tracked) can be checked by clicking the [Size Calculation] button in "Detailed Setting" of "Device/Label Detailed Setting". (☞ Page 523 Detailed setting)
*6 Even though global labels/local labels are deleted by the online change, the size of the tracking data of the global labels/local labels are the same as the one before the online change. The size is updated when the program and global label setting file to which all data is rebuilt (reassigned) are written to the programmable controller and the operating status of the CPU module is switched from STOP to RUN.
*7 This indicates the number of programs where local devices are set to use in the "Setting of Device/File Use Or Not" window of the program setting of the CPU parameter.

D4 is determined as follows.

Item	Size
System data	■Fixed data 7602 ■Data that varies depending on the tracking device/label setting 16 × (No. of tracking settings of T, ST, C, and LC) + 8 × (No. of global device tracking settings of other devices) ■Variable depending on safety communication status*8 136 + (29 × (Number of connections set in the safety communication setting of all the modules*9)
Signal flow memory (standard/safety)	(Total number of steps of control system execution programs, or total number of steps of rising/falling instruction in an FB)/16 (The digit after the decimal point is round up.)
Refresh data register (RD)	Follows the tracking transfer settings. (☞ Page 727 Tracking transfer setting)

*8 When calculating D7, use this value for "System data (safety)".
*9 Since the maximum number of mountable CC-Link IE Field Network module is 8, the maximum number of connections that can be set in the safety communication setting is 960 (120 connections × 8 modules).

D5 is determined as follows.

Item	Size*10
Global device (standard) (other than the refresh data register (RD))	Follows the tracking transfer settings. (☞ Page 727 Tracking transfer setting)
Global device (safety)	Follows the safety device area in the device/label memory are settings.
Local device (standard)	Total size of local devices of the device/label memory area detailed setting × the number of programs on the program setting*12
Local device (safety)	Safety device/label area capacity minus safety device area capacity and safety label area capacity specified in the device/label memory area settings
Global label (standard)*11	Follows the global label settings.
Global label (safety)	Follows the safety global label settings.
Local label (standard)*11	Follows the local label settings of each standard program.
Local label (safety)	Safety label area capacity specified in the device/label memory area setting minus the total size of labels defined in the safety global label setting file
Standard/safety shared label	Follows the standard/safety shared label setting file.

*10 The size of the devices/labels to be transferred (tracked) can be checked by clicking the [Size Calculation] button in "Detailed Setting" of "Device/Label Detailed Setting". (☞ Page 523 Detailed setting)
*11 Even though global labels/local labels are deleted by the online change, the size of the tracking data of the global labels/local labels are the same as the one before the online change. The size is updated when the program and global label setting file to which all data is rebuilt (reassigned) are written to the programmable controller and the operating status of the CPU module is switched from STOP to RUN.
*12 This indicates the number of programs where local devices are set to use in the "Setting of Device/File Use Or Not" window of the program setting of the CPU parameter.

■Waiting time for completion of tracking data reception (Tca)

The following describes the waiting time for completion of tracking data reception (Tca) in the CPU module of the control system.

Tca = 2[ms]

■Other extended times (α)

The following describes other delay time (α) in the CPU module of the control system.

α = 0.6 + α1[ms]

When an error occurs during a tracking transfer, the scan time may be extended (α1).α 1 is determined as follows.

Item	Time for α1
Disconnection, pulling out, inserting of the other tracking cable	6[ms]
Momentary power failure or power-off in the standby system	16 to 56 [ms]
Hardware failure in the standby system	50 to Total size of tracking data [word] $\times$ 26.7 $\times$ 10^{-6} + 10*1 [ms]

*1 Maximum extended time for the redundant function module

Point

When using the constant scan, take one of the following measures.

- The setting time of the constant scan is set by adding the above extended time (α1) when an error occurs. (☞ Page 43 Setting constant scan)
- When one of the above errors has occurred and a continuation error occurs due to the excess of constant scan time, clear the error. (☞ Page 145 Error Clear)

A

■Waiting time for tracking data reception (Taw)

The following describes the waiting time for tracking data reception (Taw) in the CPU module of the standby system.

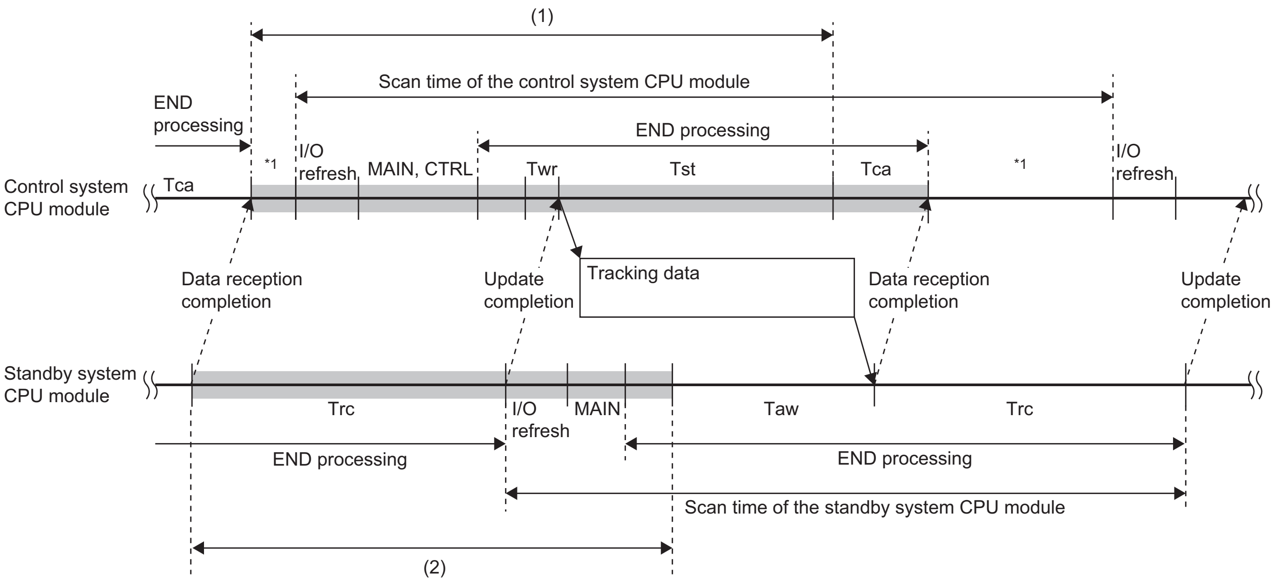
The waiting time for tracking data reception is determined by two types of period; (1) the period from when the control system CPU module receives a receive completion notification until next time it receives the notification and (2) the period when the standby system CPU module sends a receive completion notification until it starts receiving tracking data.

When (1) ≤ (2): 0 [ms]

When (1) > (2): (1) - (2) [ms]

Ex.

When CTRL is a control system execution program and MAIN is a program executed in both systems (in configuration with the main base unit only[*2])



*1 If the constant scan is used, waiting time for the constant scan is generated.

*2 In a redundant system with redundant extension base unit, the processing order in the scan is different (☞ Page 459 Scan Configuration), but the operation is the same as in a configuration with main base units only.

- For the I/O refresh processing time for (1) and (2) and END processing time for others, refer to the following.

☞ Page 1048 END processing time

- If the values of each item for (1) and (2) vary, the waiting time for tracking data reception also varies.

■Tracking data reflection time (Trc)

The following describes the tracking data reflection time (Trc) in the CPU module of the standby system. The calculation method differs depending on whether an extended SRAM cassette is inserted or not.

CPU module	Extended SRAM cassette inserting status	Tracking data reflection time
Process CPU (redundant mode)	Not inserted	$1+ (20.0 \times 10^{-6}) \times D1 + (37.2 \times 10^{-6}) \times D2 + E1 + F1 + (300.0 \times 10^{-6}) \times G1$[ms]
	Inserted	$1+ (20.0 \times 10^{-6}) \times D1 + (106.0 \times 10^{-6}) \times D2 + E1 + F1 + (300.0 \times 10^{-6}) \times G1$[ms]
SIL2 Process CPU	Not inserted	$1+ (20.0 \times 10^{-6}) \times D4 + (37.2 \times 10^{-6}) \times D5 + (682.3 \times 10^{-6}) \times D7 + F1 + (300.0 \times 10^{-3}) \times G1$[ms]
	Inserted	$1+ (20.0 \times 10^{-6}) \times D4 + (106.0 \times 10^{-6}) \times D5 + (682.3 \times 10^{-6}) \times D7 + F1 + (300.0 \times 10^{-3}) \times G1$[ms]

- D1: Size [word] of tracking data in the system data, signal flow memory, refresh data register (RD), and module label (extension base unit)
- D2: Size [word] of tracking data of global devices, local devices, global labels, and local labels
- D4: Size [word] of tracking data in the system data, signal flow memory (standard/safety), and refresh data register (RD)
- D5: Size [word] of tracking data of global devices (standard/safety), local devices (standard/safety), global labels (standard/safety), local labels (standard/safety), and standard/safety shared labels
- D7: Total size [word] of tracking data in the system data (safety), signal flow memory (safety), local devices (safety), global labels (safety), local labels (safety), and global devices (safety)
- E1: Additional time when the SFC program is used = 4.0[ms] (Only when the SFC program is used)
- F1: Additional time when the PID control instruction is used = 0.02[ms] (Only when the PID control instruction is used)
- G1: Number of transferred blocks

For the calculation formula for the size of tracking data, refer to the following.

☞ Page 1079 Tracking data send time (Tst)

■Other extended times (α)

When an error occurs during a tracking transfer, the scan time may be extended (α).[*1]

Item	Time for α
In a redundant system with redundant extension base unit, when disconnection of the tracking cable, disconnection of the redundant function module, or redundant function module failure has occurred	Maximum of 60[ms]

*1 This does not apply in the configuration with the main base unit only.

Increase in the scan time in a redundant system with redundant extension base unit

In a redundant system with redundant extension base unit, the scan time becomes longer (END processing) because the CPU module of the control system checks the connection status of the extension cable. The following shows the increase in the scan time caused by checking the connection status of the extension cable.

- Control system: 10μs
- Standby system: 0μs

Extended time required for detecting a system switching cause

The following describes the extended time required from occurrence of a system switching cause for the CPU module or redundant function module in the control system to the detection of a system switching request, or the extended time required for the CPU module to detect a system switching request sent from the network module.

System switching cause		Extended time required for detecting a system switching cause
Power-off		19.26 to 38.83ms
CPU module hardware failure, CPU module stop error	When the tracking communication cannot be made	10ms
	When the tracking communication can be made	0ms
System switching request from a network module		0ms
System switching request by using the SP.CONTSW instruction		0ms
System switching request from the engineering tool		0ms

A

Time required for system switching

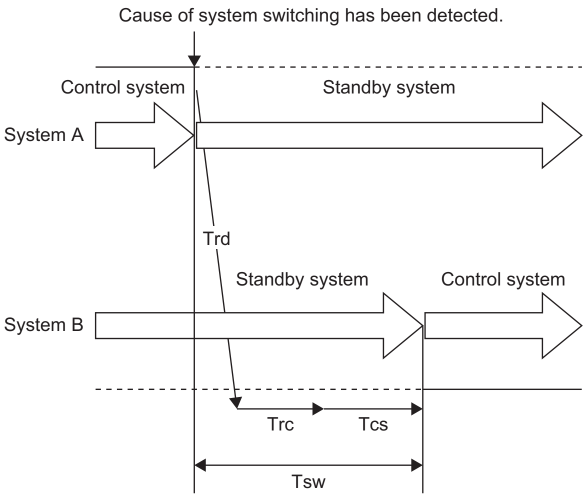
The following describes the time required for the CPU module in the new control system to start operating as the new control system after the detection of a system switching cause in the control system.

The time required for system switching determined by the following calculation formula is a rough standard for a system start-up. Check the time with the actual systems.

Tsw = Trd + Trc + Tcs [ms]

- Tsw: Time required for system switching[*1]
- Trd: Delay until the CPU module in the standby system receives the system switching request sent from the control system
- Trc: Time required for reflection of tracking data in the CPU module of the standby system
- Tcs: Setup time required for the CPU module in the standby system to start operating as the new control system

*1 This is the maximum system switching time.



System switching cause	Processing time per each system switching cause			
	Trd	Trc	Tcs	
• Power supply module failure[*2], power-off • Reset	■When the standby system scan time is longer than 19.26ms[*1] 0.517 + Standby system scan time - 19.26 [ms] ■When the standby system scan time is 19.26ms or shorter[*1] 0.517ms	0ms	• Process CPU (redundant mode): 1ms • SIL2 Process CPU: 4ms	The following time is added according to the parameter setting and system configuration. • When the signal flow memory is tracked: 0ms • When the signal flow memory is not tracked: 2ms • When an extension base unit for the redundant system is used: 5.8ms • When a fault occurs in the extension cable: 2.5ms
• CPU module failure • Base unit failure[*3] • CPU module stop error (major) • Redundant function module failure[*3]	■When the standby system scan time is longer than 10ms[*1] 0.517 + Standby system scan time - 10 [ms] ■When the standby system scan time is 10ms or shorter[*1] 0.517ms			
CPU module stop error (moderate)[*5]	Increase in the scan time of the CPU module in the control system (Ts) by tracking transfer[*4] - Waiting time for completion of the previous tracking data reflection (Twr)	☞ Page 1085 Tracking data reflection time (Trc)[*4]	• Process CPU (redundant mode): 5ms • SIL2 Process CPU: 8ms	
System switching request from a network module	Increase in the scan time of the CPU module in the control system (Ts) by tracking transfer - Waiting time for completion of the previous tracking data reflection (Twr)	☞ Page 1085 Tracking data reflection time (Trc)		
System switching request by using the SP.CONTSW instruction				
System switching request from the engineering tool				

*1 The standby system scan time is the total of "Execution time of the standby system program after a system switching cause is received + Tracking data reflection time".

*2 The systems may not be switched when the CC-Link IE Field Network module has not been mounted, when the extension base unit for the redundant system is not being used, or depending on the state of the power supply module failure. To switch the systems regardless of the state of the power supply module failure, mount the CC-Link IE Field Network module or use the extension base unit for the redundant system. Note, in this case, that Trd is determined by the following:
· Via the CC-Link IE Field Network module: Trd = 400ms (maximum) + Standby system scan time
· Via the redundant extension base unit: Trd = 57.5ms (maximum)

*3 When the CC-Link IE Field Network module has not been mounted or the extension base unit for the redundant system is not being used, and a communication error has been detected in the communications between the redundant function module and the CPU module, the systems may not be switched. To switch the systems even though the redundant function module has been removed from the base unit or a failure has occurred in the base unit, mount the CC-Link IE Field Network module or use the extension base unit for the redundant system. Note, in this case, that Trd is determined by the following:
Trd = Standby system scan time
· Via the CC-Link IE Field Network module: Trd = Standby system scan time
· Via the redundant extension base unit: Trd = 57.5ms (maximum)

*4 When a system switching occurs due to a stop error of the CPU module, only the system data is transferred and the other data (such as the signal flow memory, devices and labels) are not transferred.

*5 The CPU module stop error (moderate) also occurs when the extension cable between the main base unit and the extension base unit or on the active side between extension base units is disconnected (fails).

■Tracking data reflection time (Trc)

The following describes the time required for the CPU module of the standby system to reflect tracking data. The calculation method differs depending on whether an extended SRAM cassette is inserted or not.

CPU module	Extended SRAM cassette inserting status	Tracking data reflection time
Process CPU (redundant mode)	Not inserted	$1+ (20.0 \times 10^{-6}) \times D1 + (37.2 \times 10^{-6}) \times D2 + (300.0 \times 10^{-6}) \times F1$[ms]
	Inserted	$1+ (20.0 \times 10^{-6}) \times D1 + (106.0 \times 10^{-6}) \times D2 + (300.0 \times 10^{-6}) \times F1$[ms]
SIL2 Process CPU	Not inserted	$1+ (20.0 \times 10^{-6}) \times D4 + (37.2 \times 10^{-6}) \times D5 + (682.3 \times 10^{-6}) \times D7 + (300.0 \times 10^{-6}) \times F1$[ms]
	Inserted	$1+ (20.0 \times 10^{-6}) \times D4 + (106.0 \times 10^{-6}) \times D5 + (682.3 \times 10^{-6}) \times D7 + (300.0 \times 10^{-6}) \times F1$[ms]

- D1: Size [word] of tracking data in the system data, signal flow memory, and refresh data register (RD)
- D2: Size [word] of tracking data of global devices, local devices, global labels, and local labels
- D4: Size [word] of tracking data in the system data, signal flow memory (standard/safety), and refresh data register (RD)
- D5: Size [word] of tracking data of global devices (standard/safety), local devices (standard/safety), global labels (standard/safety), local labels (standard/safety), and standard/safety shared labels
- D7: Total size [word] of tracking data in the system data (safety), signal flow memory (safety), local devices (safety), global labels (safety), local labels (safety), and global devices (safety)
- F1: Number of transferred blocks

For the calculation formula for D1 to D7, refer to the following.

☞ Page 1079 Tracking data send time (Tst)

A

Delay time until initial output after system switching (Tjo)

This section describes the delay time until initial output after system switching (Tjo).

■Time required until values are output to the network module

The following describes the time required for the CPU module that has started operating as the new control system to output values to the network module for the first time.

- Standard control

Tjo = (Sc + Twcyc) - Twc - Ts - Toref [ms]

- Tjo: Extended time from completion of a system switching to the first output
- Sc: Scan time of the CPU module in the control system
- Twcyc: Waiting time for cyclic data receive after system switching (☞ Page 1087 Waiting time for cyclic data receive after system switching (Twcyc))
- Twc: Constant scan waiting time
- Ts: Increase in the scan time of the CPU module in the control system by tracking transfer
- Toref: Output refresh of the intelligent function module (CPU module → Intelligent function module)

- Safety control

Item	Calculation formula
Normal value	Tso = (Msw or Lsw) + Tsc + 1.5 × TM + 4.5 × RM + Tsio[ms]
Maximum value	Tso = (Msw or Lsw) + Tsc + 2 × TM + 9 × RM + Tsio[ms]

- Tso: Extended time from completion of a system switching to the first output
- Msw: Master station switching time
- Lsw: Line switching time
- Tsc: Safety cycle time
- Tsio: Safety cycle processing time
- TM: Transmission interval monitoring time
- RM: Safety refresh monitoring time

■Time required until values are output to modules on the extension base unit

The following describes the time required for the CPU module that has started operating as the new control system to output values to modules on the extension base unit for the first time.

Tjo = (Sc + Twcyc) - Ts - Toref[ms]

- Tjo: Extended time from completion of a system switching to the first output
- Sc: Scan time of the CPU module in the control system
- Twcyc: Waiting time for cyclic data receive after system switching (☞ Page 1087 Waiting time for cyclic data receive after system switching (Twcyc))
- Ts: Increase in the scan time of the CPU module in the control system by tracking transfer
- Toref: Total time for each input (input refresh (X), link input refresh, and intelligent input refresh) from the module installed in the extension base unit

I/O holding time in a redundant system with redundant extension base unit

This section describes the time required for the new control system to input/output values from/to modules on the extension base unit when systems are switched in a redundant system with redundant extension base unit.

Th = Sc + Tdt + Tsw +Tjo + Y[ms]

- Th: I/O holding time
- Tdt: Extended time required for detecting a system switching cause (☞ Page 1083 Extended time required for detecting a system switching cause)
- Tsw: Time required for system switching (☞ Page 1084 Time required for system switching)
- Tjo: Extended time from completion of a system switching to the first output (☞ Page 1086 Delay time until initial output after system switching (Tjo))
- Sc: Scan time of the CPU module in the control system
- Y: Response time of the mounted module (📖 Manual for each module)

Waiting time for cyclic data receive after system switching (Twcyc)

The following shows the calculation formula for waiting time for cyclic data receive after system switching.

When 'Setting to wait cyclic data receive after system switching' of the CPU parameter is enabled in the redundant line structure of the CC-Link IE Field Network module, the waiting time for cyclic data after system switching (Twcyc) is added to the delay time until initial output after system switching (Tjo). When the setting is disabled (default), 0[ms] is added.

Twcyc = Lsw + 2LS[ms]

- Lsw: Line switching time of the CC-Link IE Field Network module
- LS: Link scan time of the CC-Link IE Field Network module

For the calculation formula for Lsw and LS, refer to the following.

MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

Memory copy time

The following table shows the time taken for the memory copy under the following conditions:

■Example of Process CPU (redundant mode)

[Condition]

- CPU module: R120PCPU
- Sequence scan time: 5ms
- The SD memory card of the standby system CPU module has been formatted.
- Data in the program memory, data memory, and SD memory card are mismatched between the control system and standby system.

Memory copy target memory and size of data to be transferred	Memory copy time
Size of data transferred from the data memory and program memory: 144K bytes Size of data transferred from the SD memory card: 128K bytes	30s
Size of data transferred from the data memory and program memory: 282K bytes Size of data transferred from the SD memory card: 512K bytes	40s

Memory copy time depends on the data to be copied and the use of the SD memory card. Use the time above as a rough standard for memory copy.

■Example of SIL2 Process CPU

[Condition]

- CPU module: R120PSFCPU
- Sequence scan time: 5ms
- The SD memory card of the standby system CPU module has been formatted.
- Data in the SD memory card is mismatched between the control system and standby system.

Memory copy target memory and size of data to be transferred	Memory copy time
Size of data transferred from the data memory and program memory: 154K bytes Size of data transferred from the SD memory card: 128K bytes	50s
Size of data transferred from the data memory and program memory: 292K bytes Size of data transferred from the SD memory card: 512K bytes	70s

Memory copy time depends on the data to be copied and the use of the SD memory card. Use the time above as a rough standard for memory copy.

Processing time until the file operation is completed

This section describes the processing time from the start of the file operation instruction until the completion of the file operation.

Changes in the processing time according to the number of files

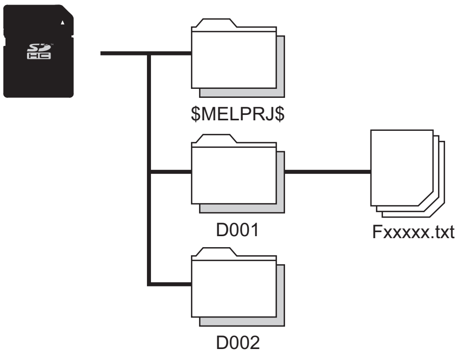
The processing time changes according to the number of files stored in folders. The table below lists the processing time under the following conditions:

[Condition]

- Folder/file structure (drive 2: SD memory card)
- SD memory card: NZ1MEM-2/4/8/16GBSD used
- Size of each file to be operated: 1K byte
- The following table lists the instruction arguments of each file operation instruction.

Instruction name	First argument	Second argument	Third argument	Fourth argument	Fifth argument	Sixth argument	Seventh argument
SP.FDELETE	U0	K2	D0	"D001"	M0	—	—
SP.FCOPY	U0	D0	K2	"D001"	K2	"D002"	M0
SP.FMOVE	U0	D0	K2	"D001"	K2	"D002"	M0
SP.FRENAME	U0	K2	D0	"D001"	"D003"	M0	—
SP.FSTATUS	U0	K2	D0	"D001"	D10	M0	—

- File/folder structures except those shown below do not exist.



- Overwriting setting: Not overwrite (SP.FCOPY, SP.FMOVE only)
- Target type setting: Folder specification
- Empty folder deletion setting: Delete folders even when they are not empty (SP.FDELETE only)

■Processing time

Instruction name	Number of operated files in the folder (D001)			
	1	10	100	1000
SP.FDELETE	38ms	173ms	1417ms	16768ms
SP.FCOPY	694ms	1169ms	6045ms	80977ms
SP.FMOVE	53ms	53ms	53ms	53ms
SP.FRENAME	32ms	32ms	32ms	32ms
SP.FSTATUS	9ms	9ms	9ms	9ms

Changes in the processing time according to the file size

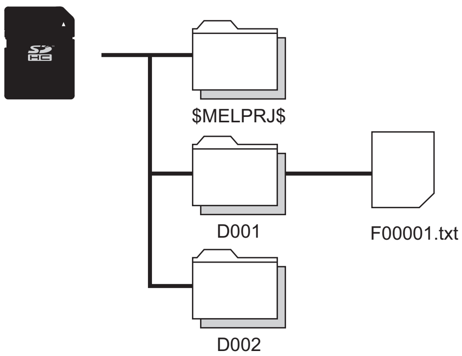
The processing time changes according to the size of the files stored in the folder. The table below lists the processing time under the following conditions:

[Condition]

- Folder/file structure (drive 2: SD memory card)
- SD memory card: NZ1MEM-2/4/8/16GBSD used
- Number of files in the folder: 1
- The following table lists the instruction arguments of each file operation instruction.

Instruction name	First argument	Second argument	Third argument	Fourth argument	Fifth argument	Sixth argument	Seventh argument
SP.FDELETE	U0	K2	D0	"D001\F00001.txt"	M0	—	—
SP.FCOPY	U0	D0	K2	"D001\F00001.txt"	K2	"D002"	M0
SP.FMOVE	U0	D0	K2	"D001\F00001.txt"	K2	"D002"	M0
SP.FRENAME	U0	K2	D0	"D001\F00001.txt"	"F00002.txt"	M0	—
SP.FSTATUS	U0	K2	D0	"D001\F00001.txt"	D10	M0	—

- File/folder structures except those shown below do not exist.



- Overwriting setting: Not overwrite (SP.FCOPY, SP.FMOVE only)
- Target type setting: File specification

■Processing time

Instruction name	Size of operated files in the folder (D001)			
	1MB	16MB	64MB	256MB
SP.FDELETE	40ms	40ms	130ms	270ms
SP.FCOPY	450ms	5450ms	21450ms	85400ms
SP.FMOVE	60ms	60ms	60ms	60ms
SP.FRENAME	50ms	50ms	50ms	50ms
SP.FSTATUS	10ms	10ms	10ms	10ms

A

Appendix 10 Parameter List

This appendix lists the parameter items and parameter numbers.

System parameters

The following is the list of system parameter items and parameter numbers.

Item			Parameter No.
I/O Assignment	Base/Power/Extension Cable Setting	Slot	0201H
		Base, Power Supply Module, Extension Cable	0203H
	I/O Assignment Setting	Points, Start XY, Module Status Setting	0200H
		Control PLC Settings	0202H
		Module Name	0203H
		Number of CPU Modules	0301H
		Host CPU Setting	3900H
		CPU Module Operation Setting at Error Detection	3600H
	Points Occupied by Empty Slot Batch Setting		0100H
	Redundant Module Group Setting		0400H
Multiple CPU Setting	Communication Setting between CPUs	CPU Buffer Memory Setting	0303H
		CPU Buffer Memory Setting (at refresh END)	0304H
		CPU Buffer Memory Setting (at execution of refresh I45)	0308H
		PLC Unit Data	0309H
		Fixed Scan Communication Area Setting	0307H
	Fixed Scan Communication Setting	Fixed Scan Interval Setting of Fixed Scan Communication	0306H
		Fixed Scan Communication Function and Inter-module Synchronization Function	
	Operation Mode Setting	Stop Setting	0302H
		Synchronous Startup Setting	030AH
	Other PLC Control Module Setting	I/O Setting Outside Group	0305H
Inter-module Synchronization Setting	Select Synchronous Target Unit between Unit		0101H
	Fixed Scan Interval Setting of Inter-module Synchronization		
	Inter-module Synchronous Master Setting		0102H

CPU Parameters

The following is the list of CPU parameter items and parameter numbers.

Item			Parameter No.
Name Setting	Title Setting		3100H
	Comment Setting		3101H
Operation Related Setting	Timer Limit Setting		3200H
	RUN-PAUSE Contact Setting		3201H
	Remote Reset Setting		3202H
	Output Mode Setting of STOP to RUN		3203H
	Module Synchronous Setting		3207H
	Clock Related Setting		3209H
Interrupt Settings	Fixed Scan Interval Setting		3A00H
	Fixed Scan Execution Mode Setting		
	Interrupt Enable Setting in Executing Instruction		
	Block No. Save/Recovery Setting		
	Interrupt Priority Setting from Module		3A01H
Service Processing Setting	Device/Label Access Service Processing Setting		3B00H
File Setting	File Register Setting		3300H
	Initial Value Setting		3301H
	Label Initial Value Reflection Setting		3302H
	File Setting for Device Data Storage		3303H
Memory/Device Setting	Cassette Setting	Extended SRAM Cassette	3404H
		Battery-less Option Cassette Setting	340CH
	Device/Label Memory Area Setting	Device/Label Memory Area Capacity Setting (Standard Device/Label Area)	3400H, 3409H
		Device/Label Memory Area Capacity Setting (Safety Device/Label Area)	5A00H*2, 5A09H*2
		Device Points	3401H
		Safety Device Point Quantity Setting	5A01H*2
		Local Device	3405H
		Safety Local Device Setting	5A05H*2
		Latch Range Setting	3407H
		Device write-protect setting	340FH
		Latch Type Setting of Latch Type Label	3408H
	Index Register Setting		3402H
	Refresh Memory Setting		3403H
	Device Latch Interval Setting		3406H
	Pointer Setting		340BH
	Internal Buffer Capacity Setting		340AH
	Link Direct Device Setting		340DH
	Buffer Area Setting for Data Sampling		340EH
RAS Setting	Scan Time Monitoring Time (WDT) Setting		3500H
	Constant Scan Setting		3503H
	Error Detections Setting		3501H
	CPU Module Operation Setting at Error Detected		
	LED Display Setting		3502H
	Event History Setting	Save Destination	3504H*1, 3507H*1
		Storage Capacity Setting per File	
		Save Device/Label Operations	3507H
	Online module change function setting	Direct change setting	3505H
	RnSFCPU Operation Mode Setting		3506H

Item			Parameter No.
Program Setting	Program Setting	Program Name	3700H, 5A10H[*2], 5A13H[*2]
		Execution Type	3700H, 5A10H[*2]
		Type	3700H, 3701H
		Refresh Group Setting	3700H
		Device/File Use or not	
		Both Systems Program Executions Setting	
	FB/FUN File Setting		3702H, 5A12H[*2]
Refresh Setting between Multiple CPU	Refresh Setting (At the END)		3901H
	Refresh Setting (At I45 Exe.)		3902H
Routing Setting	Routing Setting		3800H
SFC Setting	SFC Program Start Mode Setting		3C00H
	Start Condition Setting		
	Output Mode at Block Stop Setting		
Redundant System Settings	Redundant Behavior Setting		5000H
	Tracking Setting	Signal Flow Memory Tracking Setting	5001H
		Tracking Device/Label Setting	
		Safety Tracking Setting	5A30H[*2]
	Redundant system with extension base unit	Extension cable redundant error detection setting at startup	5002H
		Automatic standby system recovery function	
Safety Function Setting	Safety Cycle Time		5A20H[*2]
	Safety I/O Hold Time		5A21H[*2]
User-defined Setting	User-defined Setting		6F00H

*1 When the device/label operation save setting is set to "Do not save", the value is 3504H, and when the device/label operation save setting is set to "Save", the value is 3507H.

*2 Saved in the safety CPU parameter file.

Memory card parameters

The following is the list of memory card parameter items and parameter numbers.

Item		Parameter No.
Boot Setting	Operation Setting at CPU Built-in Memory Boot	2000H
	Boot File Setting	
Setting of File/Data Use or Not in Memory Card	Data for Label Communication	2010H
	Module Extended Parameter	
	Device Station Parameter	

Module parameters

The following is the list of module parameter items and parameter numbers.

Built-in Ethernet function

The following is the list of module parameter items and parameter numbers relating to the built-in Ethernet function.

Item				Parameter No.
Basic Settings	Own Node Settings	Parameter Setting Method		7100H
		IP Address Setting	IP Address	A012H
			Subnet Mask	
			Default Gateway	A013H
		Enable/Disable Online Change		A030H
		Communication Data Code		
		Opening Method		A031H
	External Device Configuration			A031H, A032H
	Transmission Port Settings	MELSOFT Transmission Port UDP/IP		7A30H
		MELSOFT Transmission Port (TCP/IP)		
		Auto-open UDP Port		
Application Settings	FTP Server Settings			A037H
	Simple CPU Communication Setting			7A10H
	Time Setting			A039H
	Timer Settings for Data Communication			A038H
	Security	IP Filter		A03AH
		Disable Direct Connection with MELSOFT		A034H
		Do Not Respond to CPU Module Search		
	Gateway Parameter Settings			A013H
	IP Packet Transfer Setting			A030H

A

I/O modules, intelligent function modules, and network modules

The following is the list of module parameter items and parameter numbers relating to I/O modules, intelligent function modules, and network modules.

For details on items and parameter numbers relating to intelligent function modules and network modules, refer to the manual for the module used.

Item			Parameter No.
Module Information (Common)	Module Name		7000H
	Start I/O No.		7002H
	Mounting Base, Mounting Slot No.		
	Comment		7001H
I/O Module	Input Response Time Setting		7102H
	Interrupt Settings		7800H
	Output Mode upon CPU Error		7101H
	Refresh Settings		7400H
Intelligent Function Module	Basic Settings		7100H
	Application Settings		7200H
	Interrupt Settings		7800H
	Refresh Settings		7400H
Network Module	Required Settings	Required Settings	7100H
		Station Type	7700H
	Basic Settings	Basic Settings	7300H, 7301H, 7302H, 7303H
		Refresh Settings	7401H
	Application Settings	Application Settings	7300H, 7301H, 7302H, 7303H
		Interrupt Settings	7800H
		Safety Communication Setting	7100H[*1], A017H[*1], A018H[*1], A01BH[*1]
I/O Module (With Safety Functions)	Basic Settings		7C21H[*1]
	Application Settings		
	Refresh Settings		7400H[*1]

*1 Saved in the safety module parameter file.

Appendix 11 Target List and Operation Details of the Device/Label Access Service Processing Setting

Target list

This section describes the targets of the device/label access service processing setting.

Applicable function	Description
Functions to perform read/write to files accessed by programs	When read/write is performed to the relevant files during execution of a program, file inconsistency may occur. Therefore, such operation is performed during the END processing.
Functions to perform write to devices/labels	When write is performed to devices/labels during execution of a program, the result of operation processing may become inconsistent. Therefore, such operation is performed during the END processing.

Communication functions via the SLMP/MC protocol

Among communication functions using the SLMP/MC protocol, those to which the device/label access service processing setting can be applied are as follows.

Function			Command
Device memory	Multiple blocks batch read		0406 (00□0)
	Multiple blocks batch write		1406 (00□0)
	Batch read	In units of bits	0401 (00□1)
		In units of words	0401 (00□0)
	Batch write	In units of bits	1401 (00□1)
		In units of words	1401 (00□0)
	Random read		0403 (00□0)
	Test (Random write)	In units of bits	1402 (00□1)
		In units of words	1402 (00□0)
	Monitor		0802 (00□0)
Files	New file creation[*1]		1820 (0000)
	File copy[*1]	File password function incompatible	1824 (0000)
		File password function compatible	1824 (0004)
	File open[*1]	File password function incompatible	1827 (0000)
		File password function compatible	1827 (0004)
	File read[*1]		1828 (0000)
	File write[*1]		1829 (0000)
	File close[*1]		182A (0000)
Programmable controller CPU monitoring	Registration[*2]		0630 (0000)
	Clear[*2]		0631 (0000)

*1 Only the file register file can support it. (Other file access is always executed asynchronously from the program.)
*2 When registration/clear command is executed, it does not affect the scan time. However, since the serial communication module will periodically access into the device memory of the CPU module when Function is enabled, it becomes the target for Device/Label access service processing setting.

Communication function using an engineering tool

Of the communication functions using an engineering tool, the functions targeted for the setting of the Device/Label access service processing are indicated.

Function	
Writing data to the programmable controller	File register file
	Device data (Local device also included)
	Global label and local label data
Reading data from the programmable controller	File register file
	Device data (Local device also included)
	Global label and local label data
	Device data storage file
Monitor function	Circuit monitor
	Device/buffer memory batch monitor
	Label batch monitor (local label included)
	Buffer memory monitor
Ethernet function	File transfer (FTP server)*1
	File transfer (FTP client)*1

*1 Only the file register file can support it. (Other file access is always executed asynchronously from the program.)

Operation details

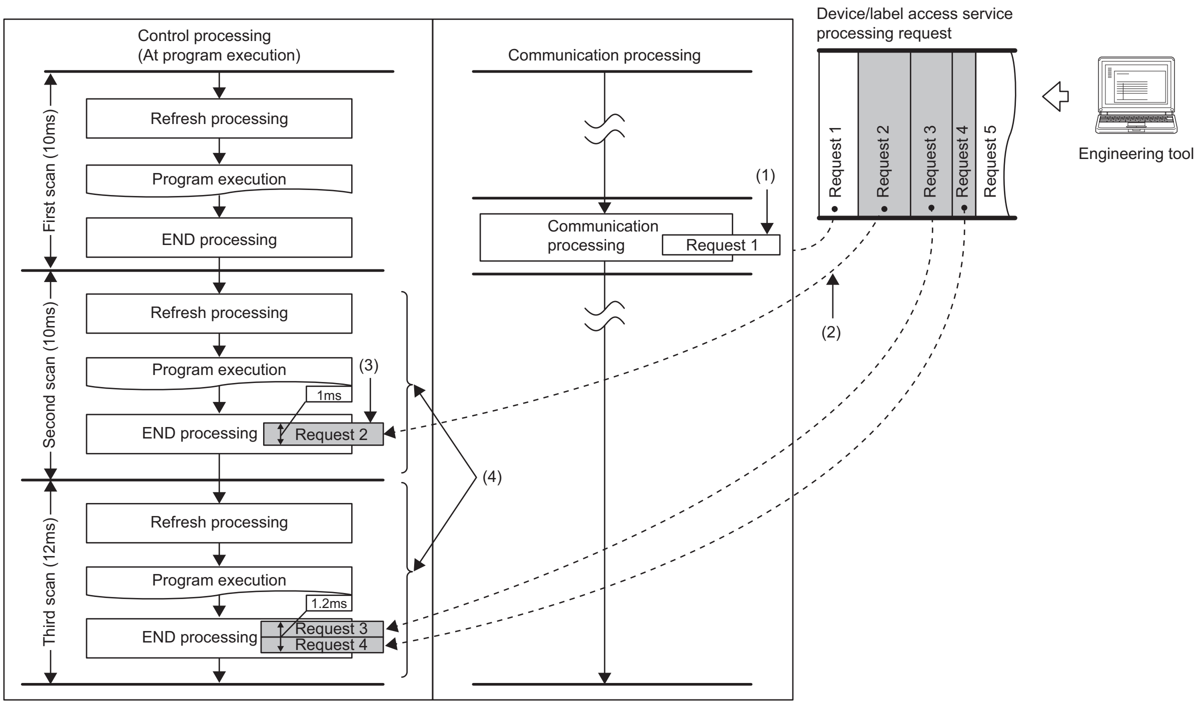
Operations enabled by setting details of the device/label access service processing setting are as follows.

Execute the process as scan time proceeds

This setting is useful to execute the device/label access service processing in a way commensurate with the system size. It allows the system to be designed without considering the device/label access service processing time because it is determined as a function of the scan time.

Ex.

When "Scan time ratio = 10%" is set



(1) Synchronization with program is unnecessary.
(2) Multiple requests are processed until the specified device/label access service processing ratio (10%) is exceeded. When the specified device/label access service processing ratio is exceeded, the remaining requests are processed in the END processing of the next scan. Also, for scan which has the device/label access service processing time shorter than 0.1ms, it is assumed as "The device/label access service processing time per scan = 0.1ms".
(3) Operations such as access to device, which are synchronized with the program, are processed in the END processing.
(4) The maximum device/label access service processing time available varies because the scan time (program execution time) varies.

Point

For operations such as access to devices, which are synchronized with the program, adjust the time by this setting because they are processed during the END processing. If no request data for the device/label access service processing exists, the scan time is shortened by the specified ratio as the CPU module proceeds to the next scan without waiting for requests.
However, when the device/label access service processing constant wait function[*1] is enabled, until the ratio set for the device/label access service processing setting of the CPU parameters is reached, the CPU module waits for requests even if requests for service processing do not exist.

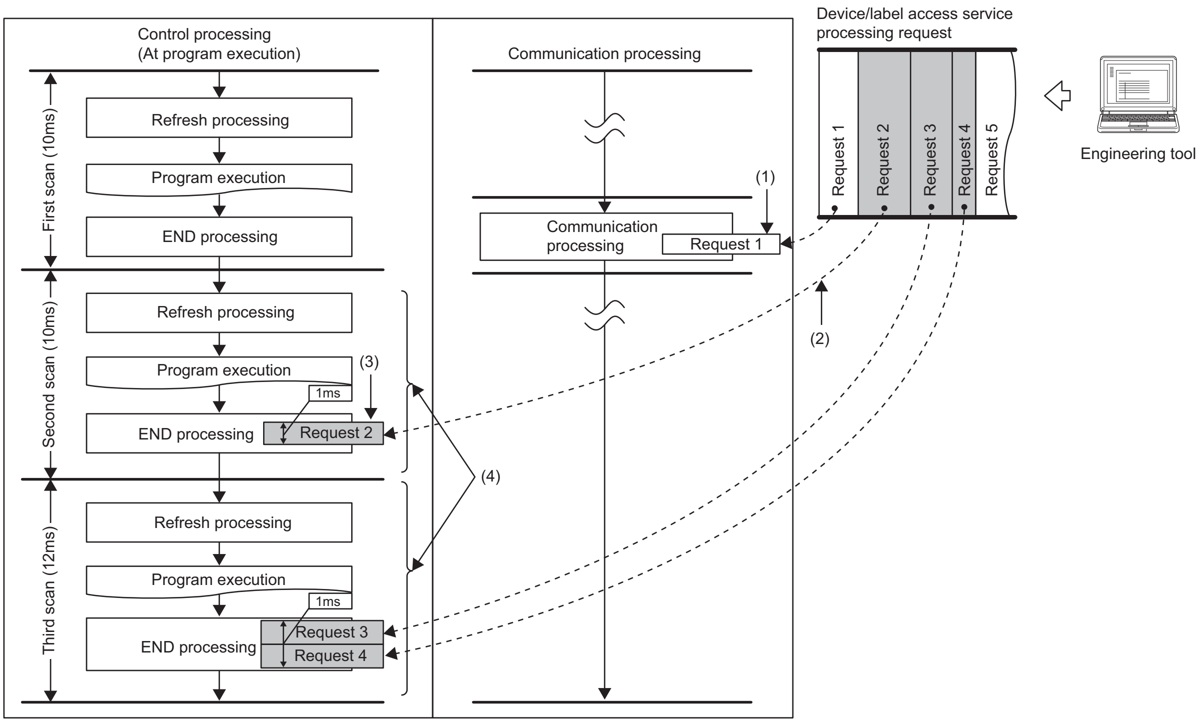
*1 For details on the compatible CPU modules, refer to the following.
☞ Page 1139 Added and Enhanced Functions

Set Processing Time

This setting is useful to give priority to the device/label access service processing. It allows for stable communication because the CPU module can always process a constant amount of the device/label access service processing without affecting the scan time.

Ex.

When "Processing time = 1ms" is set



(1) Synchronization with program is unnecessary.

(2) Multiple requests are processed until the specified processing time (1ms) is exceeded. If the specified processing time is exceeded, the remaining requests are processed in the END processing of the next scan.

(3) Operations such as access to device, which are synchronized with the program, are processed in the END processing.

(4) The maximum device/label access service processing time available is the same even when the scan time (program execution time) varies.

Point

For operations such as access to devices, which are synchronized with the program, adjust the time by this setting because they are processed during the END processing. If no request data for the device/label access service processing exists, the scan time is shortened by the specified ratio as the CPU module proceeds to the next scan without waiting for requests.

However, when the device/label access service processing constant wait function[*1] is enabled, until the time set for the device/label access service processing setting of the CPU parameters is reached, the CPU module waits for requests even if requests for service processing do not exist.

*1 For details on the compatible CPU modules, refer to the following.

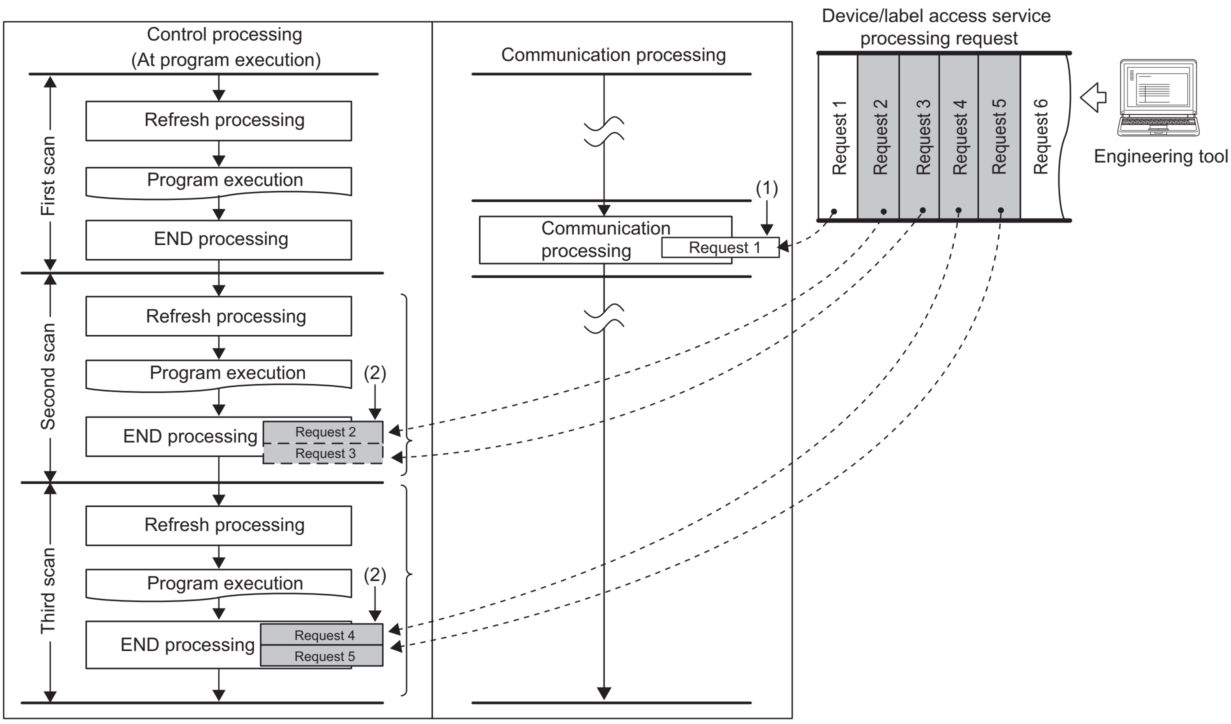
☞ Page 1139 Added and Enhanced Functions

Set Processing Counts

This setting is useful to stably execute the device/label access service processing in a system where requests come from multiple peripherals. It provides stable communication in a system where multiple peripherals exist because the CPU module can execute the device/label access service processing based on the number of request sources.

Ex.

When "Processing counts = 2" is set



(1) Synchronization with program is unnecessary.

(2) Two requests are processed in one END processing independently of the requested processing time.

Point

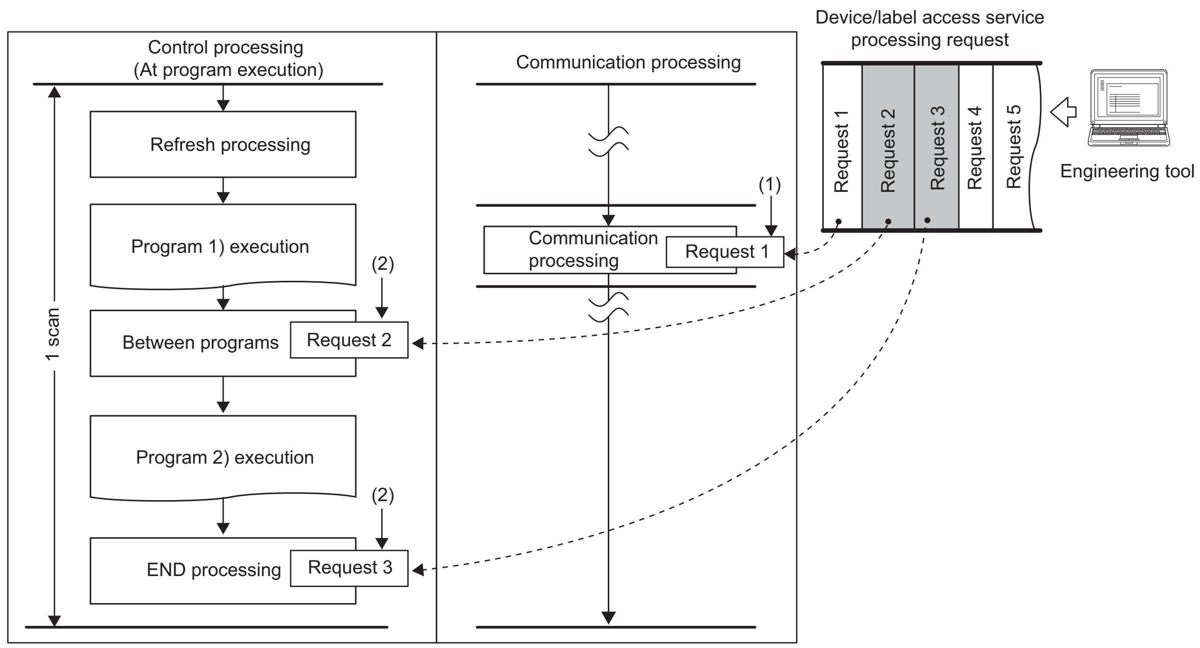
For operations such as access to devices, which are synchronized with the program, adjust the number of executions by this setting because they are processed during the END processing. If no request data for the device/label access service processing exists, the CPU module proceeds to the next scan without waiting for requests.

Execute END Processing between Programs

This setting is useful to give priority to the device/label access service processing in a system with a large number of programs. Because operations such as access to devices are performed between program executions and during the END processing, requests as many as the number of programs can be processed during a single scan. As a result, it can reduce the response for the device/label access service processing.

Ex.

When "Execute END Processing between Programs" is enabled



(1) Synchronization with program is unnecessary.

(2) Requests for operations such as access to devices are processed between program executions and during the END processing.

Appendix 12 Program Restoration Information Write Selection

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnPSFCPU (Standard) RnPSFCPU (Safety) RnSFCPU (Standard) RnSFCPU (Safety)

- If using the setting for the programmable controller CPU (except for the R00CPU, R01CPU, or R02CPU), Process CPU, or Safety CPU, check the versions of the CPU module and engineering tool. (☞ Page 1139 Added and Enhanced Functions)

Program restoration information includes the information required to read a program from the programmable controller with the engineering tool. (☞ Page 101 Data allocation and procedure of read/write operations)

Generally, use the CPU module with program restoration information written. However, the method that the program restoration information is not written during writing to the programmable controller and online change[*1*2] can be selected. Disabling the writing of program restoration information can shorten the time required for writing. This setting is useful to change and write programs repeatedly in the short term such as system start-up and program debugging.

*1 Even if the program restoration information writing is disabled, an empty file of the same size as the program restoration information is created in the data memory.

*2 In the process CPU, the method that only the program restoration information is written can be selected. (☞ Page 1139 Added and Enhanced Functions)

For the setting method and operating procedure, refer to the following.

📖 GX Works3 Operating Manual

Restriction

When the program restoration information is not written, data cannot be read from the programmable controller or the detailed verification result window cannot be displayed for verifying data with the programmable controller. Therefore, always write program restoration information after the completion of work such as system start-up and program debugging.

Before writing the program restoration information to a CPU module with no program restoration information by using the online change, select "Write in Background" in the following option.

[Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Online Program Change" ⇨ "Operational Setting" ⇨ "Write Program Restore Information"

Checking the program restoration information write status

The write status of the program restoration information can be checked in the following ways.

Item	Description	Reference
SM387 (Program restoration information write status)	Off: All written[*1] On: Not all written[*2]	Page 934 System information
SD1488 (Debug function usage status)	b1: Program restoration information write status Off: All written[*1] On: Not all written[*2]	Page 1005 Debug function
FUNCTION LED	Off: All written[*1] Flashing: Not all written[*2]	—
Monitor status of GX Works3	 All written[*1] (Data can be read from the programmable controller.)	GX Works3 Operating Manual
	 Not all written[*2] (Data cannot be read from the programmable controller.)	

*1 Program restoration information of all programs is written.

*2 There are one or more programs whose restoration information is not written.

Point

The records of the program restoration information write status are not retained in the event history. Check the write status using the items described above.

A

FUNCTION LED

The FUNCTION LED indication follows the priority order shown below.

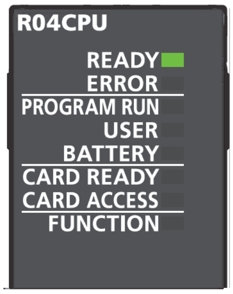
Priority	Description	Remarks
High	When program restoration information is not written, when the external input/output forced on/off function is executed (in registration), when the device tests with execution conditions are registered	Same priority
Low	Functions set in "Function to use FUNCTION LED" of "LED Display Setting" (such as the data logging function)	☞ Page 143 LED display setting

When program restoration information is not written, the operating status of other functions cannot be checked with the FUNCTION LED indication only. However, the operating status of the above functions can be checked with the FUNCTION LED by changing the LED control setting for program restoration information write status.

■Procedures for disabling the FUNCTION LED flashing

The following describes the procedures for disabling the FUNCTION LED flashing when program restoration information is not written.

1. Check that SM386 (Program restoration information write status LED control setting mode) is off (LED flashing).
2. Set "AFAFH" to SD384 (System operation setting).
3. Turn on SM384 (System operation setting request). SM384 is automatically turned off. If writing has failed, SM385 (System operation setting error) is turned on and an error is stored in SD385 (System operation setting error cause).



4. Check that SM385 is off, and turn off or reset the CPU module.
5. The FUNCTION LED turns off and SM386 turns on (without LED flashing). b1 of SD1488 (Debug function usage status) is turned off.

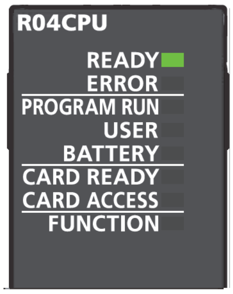
When other functions relating to the FUNCTION LED indication are used, the LED indication follows the execution status.

Point

When using the Process CPU (redundant mode), perform the above procedure for both systems.

■Procedures for returning the FUNCTION LED to the flashing state

The following describes the procedures for returning the FUNCTION LED to the flashing state when program restoration information is not written.



1. Check that SM386 (Program restoration information write status LED control setting mode) is turned on (without LED flashing).
2. Set "AFA0H" to SD384 (System operation setting).
3. Turn on SM384 (System operation setting request). SM384 is automatically turned off. If writing has failed, SM385 (System operation setting error) is turned on and an error is stored in SD385 (System operation setting error cause).
4. Check that SM385 is off, and turn off or reset the CPU module.
5. The FUNCTION LED flashes and SM386 turns off (LED flashing). b1 of SD1488 (Debug function usage status) is turned on.

Precautions

The following describes the precautions for the program restoration information write selection.

Precautions when using the boot operation

- When using the boot operation, always write program restoration information. When "No" is selected in "Program Restore Information", the SD memory card cannot be specified for the write destination.
- If online change is executed during boot operation with "No" being selected in "Program Restore Information", the change is not reflected in a program in the transfer source SD memory card.

Precautions when writing only program restoration information

The initial global label value file and initial local label value file are also written at writing only program restoration information.

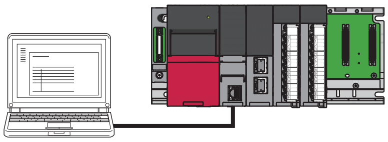
Appendix 13 Specifications of the Data Logging Function

This section describes the system configuration, setting items, and procedures for the data logging when using the data logging function.

System configuration

This section describes the system configuration when using the data logging function.

Overall system configuration



Software

This section describes the software used for the data logging function.

Name	Description
CPU Module Logging Configuration Tool	This software is used for configuration and maintenance of the data logging.
GX LogViewer	This software graphically displays data collected by the data logging function.

Point

GX LogViewer is not an essential tool for using the data logging function. Use this tool for various objectives, such as graphically displaying data logged by this function. GX LogViewer is beyond the scope of this manual unless it specifically relates to the topic of this manual. For details on GX LogViewer, refer to the following:

📖 GX LogViewer Version 1 Operating Manual

■Supported versions of CPU Module Logging Configuration Tool

Each CPU module can be used with CPU Module Logging Configuration Tool of the supported version (listed in the table below) or any later version.

Version of CPU Module Logging Configuration Tool	Compatible CPU module
"1.35M" or later	R04CPU, R08CPU, R16CPU, R32CPU, R120CPU
"1.43V" or later	RnPCPU (process mode)
"1.49B" or later	RnENCPU, RnSFCPU
"1.58L" or later	RnPCPU (redundant mode)
"1.76E" or later	R01CPU, R02CPU

■Compatible OS

For details on the OSs supported by CPU Module Logging Configuration Tool, refer to following manual, which is stored in the installer.

📖 CPU Module Logging Configuration Tool/GX LogViewer Installation Instructions (BCN-P5999-0506)

■Operating environment of CPU Module Logging Configuration Tool

For details on the operating environment for CPU Module Logging Configuration Tool, refer to following manual which is stored in the installer.

📖 CPU Module Logging Configuration Tool/GX LogViewer Installation Instructions (BCN-P5999-0506)

■Display language change

The CPU Module Logging Configuration Tool supports multiple languages, and can be used by changing the display language for menus and so on at the same computer.[*1]

*1 CPU Module Logging Configuration Tool with version "1.46Y" or later supports this function.

Operating procedure

[View] ⇨ [Switch Display Language]

Precautions

Text may be cut off if the OS and set display language differ.

Communication route

To connect the CPU module to a personal computer, use the following methods:

■Connection through a USB port

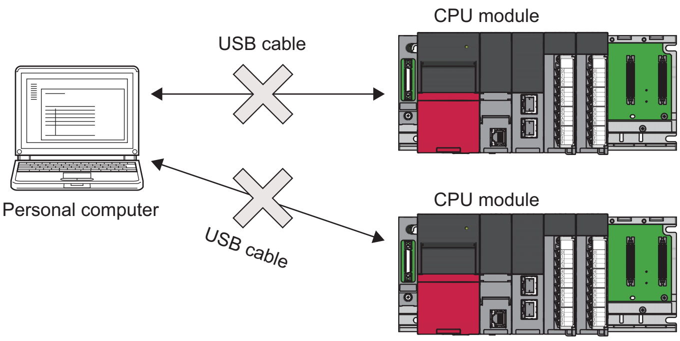
Use a USB cable. The following types of USB cables can be used:

- USB cable (USB A type—USB miniB type)
- USB cable (USB B type—USB miniB type)

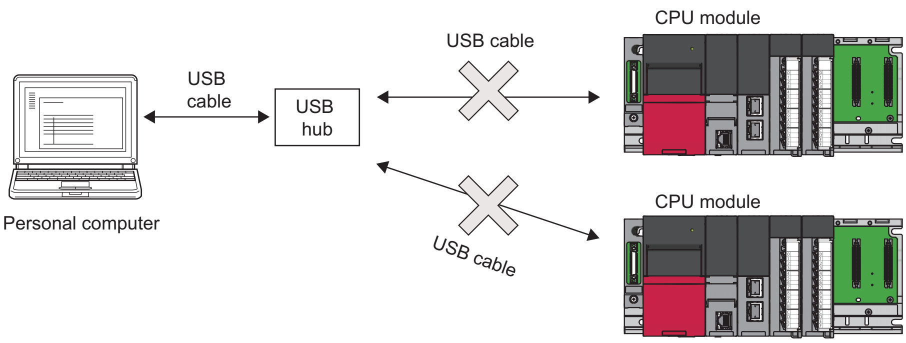
Restriction

Only one CPU module can be connected to the same personal computer simultaneously. The following connection configurations are prohibited.

- Connecting a personal computer with multiple USB ports to more than one CPU module



- Connecting to more than one CPU module via a USB hub



■Connection through an Ethernet port

• Connection via a hub

Connect the CPU module via a hub to a personal computer on the same local network.[*1] Note that IP address of the CPU module must be specified. Also the personal computer should have the same network address as the CPU module.

*1 If the CPU module is RnENCPU, use the RnENCPU (CPU part) Ethernet port to connect. Connection is not possible from the RnENCPU (network part) Ethernet port.

Restriction

Only local area network can be used for connections. Connections via the Internet are not allowed.

• Direct connection

One-to-one direct connection with an Ethernet crossing cable is possible. This method requires no hub.[*2] Note that IP address of the CPU module is not required to be specified with this method (broadcast address is used for this communication).

*2 If the CPU module is RnENCPU, use the RnENCPU (CPU part) Ethernet port to connect. Connection is not possible from the RnENCPU (network part) Ethernet port.

■Precautions about Ethernet connection

- Do not directly connect to a personal computer via LAN line. Load imposed on the LAN line adversely affects communications by other devices.
- Do not configure the direct connection setting when using one-to-one connection via a hub between the CPU module and a personal computer.
- If the following conditions are met, the direct connection communication may be disabled. If the communication is disabled, review the settings of the CPU module and personal computer.

Ex.

When all the bits of the CPU module-side IP address that correspond to 0 part of the personal computer-side subnet mask are on or off:

CPU module-side IP address: 64.64.255.255

Personal computer-side IP address: 64.64.1.1

Personal computer-side subnet mask: 255.255.0.0

Ex.

When all the bits of the CPU module-side IP address that correspond to the host address of each class of the personal computer-side IP address are on or off[*1*2]:

CPU module-side IP address: 64.64.255.255

Personal computer-side IP address: 192.168.0.1

Personal computer-side subnet mask: 255.255.0.0

*1 The IP address for each class is as follows:
·Class A: 0.x.x.x to 127.x.x.x
·Class B: 128.x.x.x to 191.x.x.x
·Class C: 192.x.x.x to 223.x.x.x

*2 The host address for each class is 0 part in the following addresses:
·Class A: 255.0.0.0
·Class B: 255.255.0.0
·Class C: 255.255.255.0

Ex.

When the CPU module-side IP address is obtained through DHCP:

- Disable Windows firewall setting, if enabled.
- Do not set direct connection method in a configuration in which more than one IP address is enabled simultaneously as described in the following list.
  - An IP address is assigned to each Ethernet port (network device) of the personal computer with more than one port.
  - Wireless LAN setting is enabled as well as the Ethernet port on the personal computer.
  - More than one IP address is assigned to a single Ethernet port of the personal computer.

■Connection to the Process CPU (redundant mode)

When the Process CPU (redundant mode) is connected, select "Connect to RnPCPU (Redundant Mode)" and select a target system for "Transfer Setup System".

Setting items

This section describes the setting items for the data logging function.

Setting item list

The following table lists the setting items.

Item			Specifications
Number of data logging settings			10[*1]
Data storage location			• SD memory card • Data memory • CPU built-in memory (function memory)[*2*3]
Logging type			• Continuous logging • Trigger logging
Data collection	Collection interval		• Every scan operation • Time specification[*3] • Interrupt occurrence[*3] • Condition specification (device/label specification, step No. specification)[*4]
	Number of points for collection		Maximum of 1280 (128 per setting)
	AND conjunction		AND conjunction of the device/label specification and step No. specification is enabled.
Data processing	Trigger logging	Trigger condition	• Condition specification (device/label change specification, step No. specification) • When trigger instruction executed
		AND conjunction	AND conjunction of the device/label change specification and step No. specification is enabled.
		Trigger logging range	Number of records specified before and after the trigger establishment
		Number of trigger establishments (number of events that can be handled as trigger)	one
		Number of records	■SD memory card Maximum of 1000000 ■CPU built-in memory (function memory)[*3] Maximum of 50000
File output	File name		Additional information plus file number
	File storage format		• Unicode text file • CSV file[*3] • Binary file
	Data type		• Bit • Word (signed) • Double word (signed) • Word (unsigned) • Double word (unsigned) • Single-precision real number • Double-precision real number • String • Numeric string • Time
	Data output format	Unicode text file	• Decimal format • Hexadecimal format • Decimal fraction format • Exponential format
		CSV file[*3]	• Decimal format • Hexadecimal format • Decimal fraction format • Exponential format
		Binary file	• Word (signed) • Double word (signed) • Word (unsigned) • Double word (unsigned) • Single-precision real number • Double-precision real number

Item			Specifications
Output file handling	Storage file switching	File switching timing	• Number of records • File size • Condition specification[*3]
		Maximum number of storage files	1 to 65535
Auto logging			When inserting an SD memory card, which holds data logging setting, into the CPU module, the data logging automatically starts based on the data logging setting information on the SD memory card.
Operation settings when entering into RUN mode			This function sets data logging operations when entering into RUN mode after the data logging setting is registered.
Data logging file transfer function[*3]			This function automatically transfers data logging files to the FTP server.
SD memory card replacement			SD memory cards can be replaced using the SD memory card forced disable function even when the data logging is in progress.

*1 When the storage destination of the data logging files is the function memory, up to 2 data loggings can be executed simultaneously even though 10 data loggings can be set. However, depending on the firmware version and production information of the CPU module, the maximum number of data loggings that can be executed simultaneously will be ten. (☞ Page 1139 Added and Enhanced Functions)

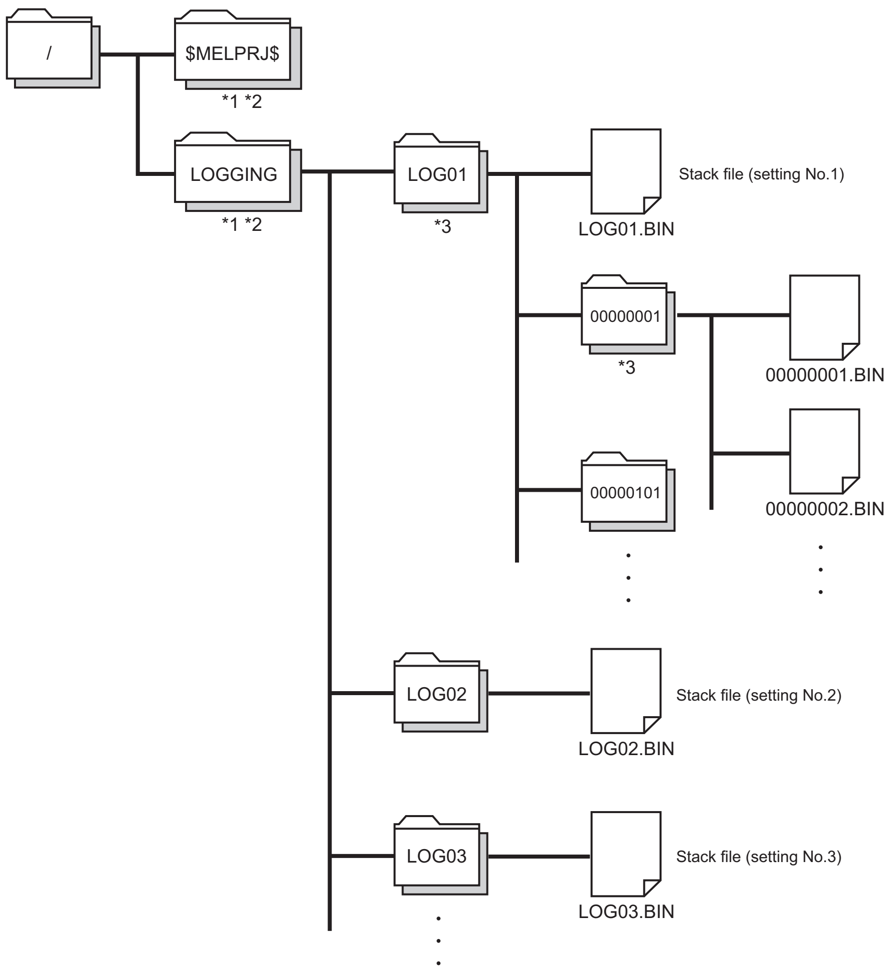
*2 If the data logging file transfer function is not set, the files are stored in the data memory.

*3 Check the firmware version of the CPU module and the version of the CPU Module Logging Configuration Tool. (☞ Page 1139 Added and Enhanced Functions)

*4 The items cannot be specified in the Safety CPU.

Folder configuration

The following figure shows the folder configuration of the SD memory card attaching to a CPU module.



*1 Folder names cannot be modified.

*2 Do not create folders/files under the $MELPRJ$ and LOGGING folders using a personal computer and other devices.

*3 To remove unnecessary folders, use the following methods:

· Use a personal computer.

· Logging file operation

Data output type

This section describes each of file output types.

■Unicode text file output type

This section describes the format specifications of the Unicode text file output type and output content of each data.

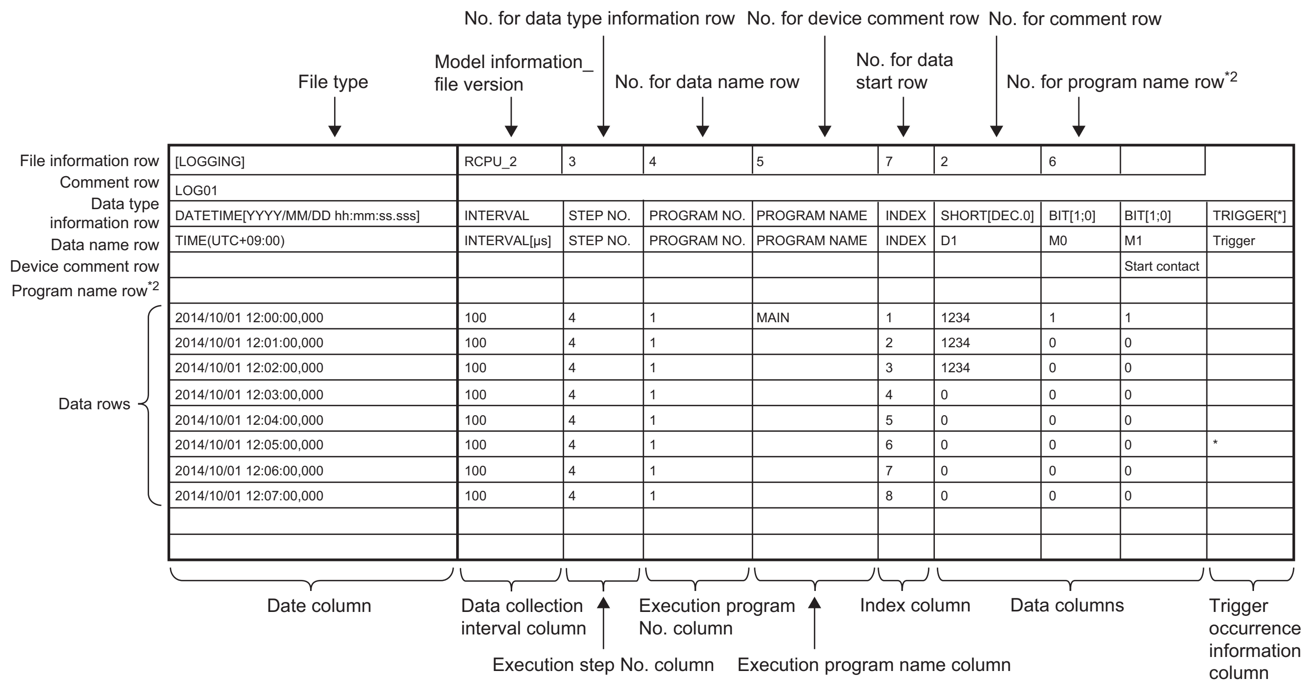
- Format specifications

Item	Description
Delimiter	Tab
Return code	CRLF (0x0D, 0x0A)
Character code	Unicode
Character encoding schema	UTF-16 (Little-Endian)
Field data	Not enclosed by double quotation marks (" ") Double quotation marks (" ") and commas (,) cannot be used in each data.[*1]

*1 When double quotation marks (" ") and/or commas (,) are included in a device comment, the following processing occurs:
· When the data includes commas (,), whole data is enclosed by double quotation marks (" ").
· When the data includes double quotation marks (" "), another set of double quotation marks will be appended to each of them.

- File format example

Output items can be specified



File information row	[LOGGING]	RCPU_2	3	4	5	7	2	6		
Comment row	LOG01									
Data type information row	DATETIME[YYYY/MM/DD hh:mm:ss.sss]	INTERVAL	STEP NO.	PROGRAM NO.	PROGRAM NAME	INDEX	SHORT[DEC.0]	BIT[1;0]	BIT[1;0]	TRIGGER[*]
Data name row	TIME(UTC+09:00)	INTERVAL[μs]	STEP NO.	PROGRAM NO.	PROGRAM NAME	INDEX	D1	M0	M1	Trigger
Device comment row									Start contact	
Program name row*2										
Data rows	2014/10/01 12:00:00,000	100	4	1	MAIN	1	1234	1	1	
	2014/10/01 12:01:00,000	100	4	1		2	1234	0	0	
	2014/10/01 12:02:00,000	100	4	1		3	1234	0	0	
	2014/10/01 12:03:00,000	100	4	1		4	0	0	0	
	2014/10/01 12:04:00,000	100	4	1		5	0	0	0	
	2014/10/01 12:05:00,000	100	4	1		6	0	0	0	*
	2014/10/01 12:06:00,000	100	4	1		7	0	0	0	
	2014/10/01 12:07:00,000	100	4	1		8	0	0	0	

*2 These items are displayed when a local device/local label is specified. The items are not output when a global device/global label is specified. (The corresponding columns and rows become blank.)

A

• Output content for each data

<File information row>

File-related information is displayed.

Item	Description	Size
File type	[LOGGING] is output.	14 bytes
Model information_file version	When a local device or local label is specified, "RCPU_2" is displayed in the file version, which shows the model information. When a global device or global label is specified, "RCPU_1" is displayed in the file version, which shows the model information.	12 bytes
No. for data type information row	Numerical value indicating the position of the data type information row from the top of the file is placed.	2 bytes
No. for data name row	Numerical value indicating the position of the data name row from the top of the file is placed.	2 bytes
No. for device comment row	Numerical value indicating the position of the device comment row from the top of the file is placed.	2 bytes
No. for data start row	Numerical value indicating the starting position of the data row from the top of the file is placed.	2 bytes
No. for comment row	Numerical value indicating the position of the comment row from the top of the file is placed. When the comment row is not output, this field is blank.	0 to 2 bytes
No. for program name row	Numerical value indicating the position of the program name row from the top of the file is placed.	2 bytes

Ex.

The total size of the file information row can be obtained by the following equation: (when comment is output)

14(file type) + 12(model information_file version) + 2(data type information row number) + 2(data name row number) + 2(device comment row number) + 2(data start row number) + 2(comment row number) + 10(the number of tabs) + 4(CR + LF) = 50 bytes

<Comment row>

Comments are displayed.

Item	Description	Size
Comment	Comment specified in CPU Module Logging Configuration Tool is output (the comment can contain up to 256 characters.[*3] When no comment is set, a blank row is output).	0 to 512 bytes

*3 Double quotation marks (" "), commas (,), and semicolons (;) cannot be used.

Ex.

The total size of the comment row can be obtained by the following equation:

Character size of the specified comment (depending on the specified character string) + 4(CR + LF)

<Data type information row>

The data type for each column is displayed. This information is output in the following format: (Data type)[(Additional information)].

Item	"Data type" output content	Size	"Additional information" output content	Size
Date column	DATETIME	16 bytes	Format is output. [YYYY/MM/DD hh:mm:ss.sss]	6 to 68 bytes
Data collection interval column	INTERVAL	16 bytes	No additional information	0 byte
Execution step No. column	STEP NO.	16 bytes		
Execution program number column	PROGRAM NO.	22 bytes		
Execution program name column	PROGRAM NAME	24 bytes		
Index column	INDEX	10 bytes		
Data column	Bit type: BIT	6 bytes	Bit type: [1;0]	10 bytes
	16-bit integer (unsigned): USHORT	12 bytes	For decimal format: [DEC.0]	14 bytes
	16-bit integer (signed): SHORT	10 bytes		
	32-bit integer (unsigned): ULONG	10 bytes	For hexadecimal format: [HEX]	10 bytes
	32-bit integer (signed): LONG	8 bytes		
	Single-precision floating point (32-bit): FLOAT	10 bytes	For exponent expression: [EXP.(number of digits of decimal part)]	14 to 16 bytes
	Double-precision floating point (64-bit): DOUBLE	12 bytes		
	Character string type: STRING	12 bytes	Character string type, numeric string type: the specified data length value (unit: bytes) is output.	6 to 10 bytes
	Numeric string type: RAW	6 bytes		
	Time: TIME	8 bytes	No additional information	0 byte
Trigger occurrence information column	TRIGGER	14 bytes	[(string occurred)] is output (semicolons (;), double quotation marks (" "), and commas (,) cannot be used).	6 to 516 bytes

Ex.

The size of the data type information row is determined by the following equation when data logging of 128 points of data (signed 16-bit integer, decimal format) is performed (The following sections in the "Output" window are set to be output: "Date" (the output format is YYYY/MM/DD hh:mm:ss.sss), "Data sampling interval", "Execution step No.", "Execution program name or execution program No.", and "Index").

(16 + 50)(date column) + 16(data collection interval column) + 16(execution step No. column) + 22(execution program number column) + 24(execution program name column) + 10(index column) + (10 + 14) × 128(data column) + 264(the number of tabs) + 4(CR + LF) = 3494 bytes

<Data name row>

The data name for each column is displayed.

Item	Description	Size
Date column	TIME (time zone) is output.	28 bytes
Data collection interval column	INTERVAL[us] is output.	24 bytes
Execution step No. column	STEP NO. is output.	16 bytes
Execution program number column	PROGRAM NO. is output.	22 bytes
Execution program name column	PROGRAM NAME is output.	24 bytes
Index column	INDEX is output.	10 bytes
Data column	The specified data name is output.	1 to 512 bytes*4
Trigger occurrence information column	Trigger is output.	14 bytes

*4 When the data column is specified and the multidimensional array label is specified, if the index of the array includes comma (,), the data size will increase than the data name string size.

Ex.

The size of the data name row is determined by the following equation when data logging of 128 data points from D100 to D227 is performed (The following sections in the "Output" window are set to be output: "Date", "Data sampling interval", "Execution step No.", "Execution program name or execution program No.", and "Index").

28(date column) + 24(data collection interval column) + 16(execution step No. column) + 22(execution program number column) + 24(execution program name column) + 10(index column) + (6 × 128)(data column) + 264(the number of tabs) + 4(CR + LF) = 1138 bytes

<Program name row>

The program name row for each column is displayed. (These items are displayed when a local device/local label is specified. The items are not output when a global device/global label is specified. (The corresponding rows become blank.))

Item	Description	Size
Date column	No output (blank).	0 byte
Data collection interval column		
Execution step No. column		
Execution program number column		
Execution program name column		
Index column		
Data column	When the collected data is a local device or local label, the specified program name (no extension) is displayed. When the collected data is a global device or global label, this field is blank.	0 to 120 bytes
Trigger occurrence information column	No output (blank).	0 byte

Ex.

The size of the program name row is determined by the following equation when data logging of 128 data points from D100 to D227 is performed (The following sections in the "Output" window are set to be output: "Date", "Data sampling interval", "Execution step No.", "Execution program name or execution program No.", and "Index". Also, devices from D200 to D214 are specified as the local device of the program No.1 of "MAIN").

0(date column) + 0(data collection interval column) + 0(execution step number column) + 0(execution program number column) + 0(execution program name column) + 0(index column) + ($0 \times 100 + 8 \times 15$)(data column) + 264(the number of tabs) + 4(CR + LF)

= 388 bytes

<Device comment row>

The device comment row for each column is displayed.

Item	Description	Size
Date column	No output (blank).	0 byte
Data collection interval column		
Execution step No. column		
Execution program number column		
Execution program name column		
Index column		
Data column	When "Output device comment" is selected in the output settings, the comment with the specified comment number is displayed. When "Output each program device comment" is selected, the comment with the specified comment number in the device comment file for each program is output only for the local device. When the device comment file for each program or the comment of the corresponding program does not exist, the comment of the device comment file is output. When comments are set not to be output, or a device comment to be displayed does not exist, this field is blank.	0 to 2048 bytes[*5]
Trigger occurrence information column	No output (blank).	0 byte

*5 When "Output device comment" is selected in "Device comment output" of the output settings and double quotation marks (" ") and/or commas (,) are included in a device comment, the size of data in use becomes larger than the size of the character string of the device comments.

Ex.

The size of the device comment row is determined by the following equation when data logging of 128 data points from D100 to D227 is performed (The following sections in the "Output" window are set to be output: "Date" (the output format is YYYY/MM/DD hh:mm:ss.sss), "Data sampling interval", "Execution step No.", "Execution program name or execution program No.", "Index", and "Device comment output" (only for devices from D200 to D227)).

0(date column) + 0(data collection interval column) + 0(execution step number column) + 0(execution program number column) + 0(execution program name column) + 0(index column) + ($0 \times 100 + 14 \times 28$)(data column) + 264(the number of tabs) + 2(CR + LF) = 660 bytes

<Data row>

The collected data value is displayed. All the data items collected during a single collection is displayed in a single row.

Item	Description	Size
Date column	Information is output according to the data row output character string specified in the format.	2 to 64 bytes
Data collection interval column	The time interval from the previous collection time to the current collection time is output. If the maximum display range is exceeded, the count returns to 1 and starts again to output a new time interval (unit: μs, display range: 1 to 100000000000).	2 to 24 bytes
Execution step No. column	The step No. executed on the engineering tool at the time interval and timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the step No. at the time of execution of END instruction in the last executed program on the scan. With the collection condition "Interrupt occurrence" specified, the resulting output is the step No. at the time of execution of IRET instruction in the interrupt program. If the system operation (such as system interrupt) is running during collection interval and collection timing or the FB program is in running, "0" is output.	2 to 12 bytes
Execution program number column	The program No. executed on the engineering tool at the time interval and timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the program number of the last executed program in the scan. If two or more system operations, such as collection timing and system interrupt, occur simultaneously, "—" is output.	2 to 6 bytes
Execution program name column	The program name (no extension) executed on the engineering tool at the time interval and timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the program name of the last executed program in the scan. If a program name with the same program number has already been in the file, a null value is output. If two or more system operations, such as collection timing and system interrupt, occur simultaneously, "* SYSTEM" is output.	0 to 200 bytes
Index column	A value which increments in ascending order from 1 is output. When it exceeds the upper limit, it returns to 1 and increments again (range: 1 to 4294967295).	2 to 20 bytes
Data column	When bits are specified: bit On = 1 and bit Off = 0 are output.	2 bytes
	When unsigned/signed word type is specified: data value is output according to the specified output type.	• Decimal format: 2 to 22 bytes • Hexadecimal format: 2 to 16 bytes • Decimal fraction format: 2 to 42 bytes • Exponential format: 10 to 42 bytes
	When unsigned/signed double word type, single-precision real number, or double-precision real number is specified: data value is output according to the specified output type.	• Decimal format: 2 to 22 bytes*6 • Hexadecimal format: 2 to 16 bytes • Decimal fraction format: 2 to 52 bytes • Exponential format: 10 to 44 bytes
	When character string is specified: the specified character string is output.	2 to 256 bytes
	When numeric string is specified: the character string which represented by the hexadecimal in increments of a byte is output without clearance.	4 to 1024 bytes
	When time is specified: T#-24d20h31m23s648ms to T#24d20h31m23s647ms is displayed.	26 to 40 bytes
Trigger occurrence information column	The specified character string is output when the trigger occurs. In other cases, no character string is output.	0 to 512 bytes

*6 When single-precision real number or double-precision real number is specified, if the numerical value to be output does not fall into the range -2147483648.0 to 4294967295.0, it is displayed in an equivalent format to "exponential format and the number of decimal part digits is nine".

Ex.

The size of the data type information row is determined by the following equation when data logging of 128 points of data from D100 to D227 (unsigned word type, decimal format) is performed (The following sections in the "Output" window are set to be output: "Date" (the output format is YYYY/MM/DD hh:mm:ss.sss), "Data sampling interval", "Execution step No.", "Execution program name or execution program No.", and "Index").

46(date column) + 24(data collection interval column) + 12(execution step number column) + 6(execution program number column) + 200(execution program name column) + 20(index column) + (12 × 128)(data column) + 264(the number of tabs) + 4(CR + LF) = 2112 bytes

■CSV file output format

This section describes the format specifications of the CSV file output type and output content of each data.

- Format specifications

Item	Description
Delimiter	Comma (,)
Return code	CRLF (0x0D, 0x0A)
Character code	ASCII
Character encoding schema	Shift-JIS
Field data	Not enclosed by double quotation marks (" ") Double quotation marks (" ") and commas (,) cannot be used in each data.*1

*1 When double quotation marks (" ") and/or commas (,) are included in a device comment, the following processing occurs:
· When the data includes commas (,), whole data is enclosed by double quotation marks (" ").
· When the data includes double quotation marks (" "), another set of double quotation marks will be appended to each of them.

- File format example

Output items can be specified

File type / Model information_ file version / No. for data type information row / No. for data name row / No. for device comment row / No. for data start row / No. for comment row / No. for program name row*2

File information row	[LOGGING]	RCPU_2	3	4	5	7	2	6		
Comment row	LOG01									
Data type information row	DATETIME[YYYY/MM/DD hh:mm:ss.sss]	INTERVAL	STEP NO.	PROGRAM NO.	PROGRAM NAME	INDEX	SHORT[DEC.0]	BIT[1;0]	BIT[1;0]	TRIGGER[*]
Data name row	TIME(UTC+09:00)	INTERVAL[μs]	STEP NO.	PROGRAM NO.	PROGRAM NAME	INDEX	D1	M0	M1	Trigger
Device comment row									Start contact	
Program name row*2										
Data rows	2014/10/01 12:00:00,000	100	4	1	MAIN	1	1234	1	1	
	2014/10/01 12:01:00,000	100	4	1		2	1234	0	0	
	2014/10/01 12:02:00,000	100	4	1		3	1234	0	0	
	2014/10/01 12:03:00,000	100	4	1		4	0	0	0	
	2014/10/01 12:04:00,000	100	4	1		5	0	0	0	
	2014/10/01 12:05:00,000	100	4	1		6	0	0	0	*
	2014/10/01 12:06:00,000	100	4	1		7	0	0	0	
	2014/10/01 12:07:00,000	100	4	1		8	0	0	0	

Date column / Data collection interval column / Execution step No. column / Execution program No. column / Execution program name column / Index column / Data columns / Trigger occurrence information column

*2 These items are displayed when a local device/local label is specified. The items are not output when a global device/global label is specified. (The corresponding columns and rows become blank.)

A

• Output content for each data

<File information row>

File-related information is displayed.

Item	Description	Size
File type	[LOGGING] is output.	9 bytes
Model information_file version	When a local device or local label is specified, "RCPU_2" is displayed in the file version, which shows the model information. When a global device or global label is specified, "RCPU_1" is displayed in the file version, which shows the model information.	6 bytes
No. for data type information row	Numerical value indicating the position of the data type information row from the top of the file is placed.	1 byte
No. for data name row	Numerical value indicating the position of the data name row from the top of the file is placed.	1 byte
No. for device comment row	Numerical value indicating the position of the device comment row from the top of the file is placed.	1 byte
No. for data start row	Numerical value indicating the starting position of the data row from the top of the file is placed.	1 byte
No. for comment row	Numerical value indicating the position of the comment row from the top of the file is placed. When the comment row is not output, this field is blank.	0 to 1 byte
No. for program name row	Numerical value indicating the position of the program name row from the top of the file is placed.	1 byte

Ex.

The total size of the file information row can be obtained by the following equation. (when comment is output)

9(file type) + 6(model information_file version) + 1(data type information row number) + 1(data name row number) + 1(device comment row number) + 1(data start row number) + 1(comment row number) + 6(the number of commas) + 2(CR + LF)

= 29 bytes

<Comment row>

Comments are displayed.

Item	Description	Size
Comment	Comment specified in CPU Module Logging Configuration Tool is output (the comment can contain up to 256 characters.*3 When no comment is set, a blank row is output). Since the comment is stored in Unicode, the data is converted to ASCII and Shift-JIS. If any characters that cannot be converted to ASCII or Shift-JIS, the conversion fails and those characters will be output as periods (.).	0 to 512 bytes

*3 Double quotation marks (" "), commas (,), and semicolons (;) cannot be used.

Ex.

The total size of the comment row can be obtained by the following equation:

Character size of the specified comment (depending on the specified character string) + 2(CR + LF)

<Data type information row>

The data type for each column is displayed. This information is output in the following format: (Data type)[(Additional information)].

Item	"Data type" output content	Size	"Additional information" output content	Size
Date column	DATETIME	8 bytes	Format is output. [YYYY/MM/DD hh:mm:ss.sss]	3 to 34 bytes
Data collection interval column	INTERVAL	8 bytes	No additional information	0 byte
Execution step No. column	STEP NO.	8 bytes		
Execution program number column	PROGRAM NO.	11 bytes		
Execution program name column	PROGRAM NAME	12 bytes		
Index column	INDEX	5 bytes		
Data column	Bit type: BIT	3 bytes	Bit type: [1;0]	5 bytes
	16-bit integer (unsigned): USHORT	6 bytes	For decimal format: [DEC.0]	7 bytes
	16-bit integer (signed): SHORT	5 bytes		
	32-bit integer (unsigned): ULONG	5 bytes	For hexadecimal format: [HEX]	5 bytes
	32-bit integer (signed): LONG	4 bytes		
	Single-precision floating point (32-bit): FLOAT	5 bytes	For exponent expression: [EXP.(number of digits of decimal part)]	7 to 8 bytes
	Double-precision floating point (64-bit): DOUBLE	6 bytes		
	Character string type: STRING	6 bytes	Character string type, numeric string type: The specified data length value (unit: bytes) is output.	3 to 5 bytes
	Numeric string type: RAW	3 bytes		
	Time: TIME	4 bytes	No additional information	0 byte
Trigger occurrence information column	TRIGGER	7 bytes	[(string occurred)] is output (semicolons (;), double quotation marks (" "), and commas (,) cannot be used). Since the string occurred is stored in Unicode, the data is converted to ASCII and Shift-JIS. If any characters that cannot be converted to ASCII or Shift-JIS, the conversion fails and those characters will be output as asterisks (*).	3 to 514 bytes

Ex.

The size of the data type information row is determined by the following equation when data logging of 128 points of data (signed 16-bit integer, decimal format) is performed (The following sections in the "Output" window are set to be output: "Date" (the output format is YYYY/MM/DD hh:mm:ss.sss), "Data sampling interval", "Execution step No.", "Execution program number" "Execution program name", and "Index").

(8 + 25)(date column) + 8(data collection interval column) + 8(execution step No. column) + 11(execution program number column) + 12(execution program name column) + 5(index column) + (5 + 7) × 128(data column) + 132(the number of commas) + 2(CR + LF)

= 1745 bytes

<Data name row>

The data name for each column is displayed.

Item	Description	Size
Date column	TIME (time zone) is output.	14 bytes
Data collection interval column	INTERVAL[us] is output.	12 bytes
Execution step No. column	STEP NO. is output.	8 bytes
Execution program number column	PROGRAM NO. is output.	8 bytes
Execution program name column	PROGRAM NAME is output.	12 bytes
Index column	INDEX is output.	5 bytes
Data column	The specified data name is output. Since the data name is stored in Unicode, the data is converted to ASCII and Shift-JIS. If any characters that cannot be converted to ASCII or Shift-JIS, the conversion fails and those characters will be output as periods (.).	1 to 512 bytes*4
Trigger occurrence information column	Trigger is output.	7 bytes

*4 When the data column is specified and the multidimensional array label is specified, if the index of the array includes comma (,), the data size will increase than the data name string size.

Ex.

The size of the data name row is determined by the following equation when data logging of 128 data points from D100 to D227 is performed (The following sections in the "Output" window are set to be output: "Date", "Data sampling interval", "Execution step No.", "Execution program number" "Execution program name", and "Index").

14(date column) + 12(data collection interval column) + 8(execution step No. column) + 8(execution program number column) + 12(execution program name column) + 5(index column) + (4 × 128)(data column) + 132(the number of commas) + 2(CR + LF)

= 705 bytes

<Program name row>

The program name row for each column is displayed. (These items are displayed when a local device/local label is specified. The items are not output when a global device/global label is specified. (The corresponding rows become blank.))

Item	Description	Size
Date column	No output (blank).	0 byte
Data collection interval column		
Execution step No. column		
Execution program number column		
Execution program name column		
Index column		
Data column	When the collected data is a local device or local label, the specified program name (no extension) is displayed. When the collected data is a global device or global label, this field is blank. Since the program name is stored in Unicode, the data is converted to ASCII and Shift-JIS. If any characters that cannot be converted to ASCII or Shift-JIS, the conversion fails and those characters will be output as periods (.).	0 to 120 bytes
Trigger occurrence information column	No output (blank).	0 byte

Ex.

The size of the program name row is determined by the following equation when data logging of 128 data points from D100 to D227 is performed (The following sections in the "Output" window are set to be output: "Date", "Data sampling interval", "Execution step No.", "Execution program name or execution program No.", and "Index". Also, devices from D200 to D227 are specified as the local device of the program No.1 of "MAIN").

0(date column) + 0(data collection interval column) + 0(execution step No. column) + 0(execution program number column) + 0(execution program name column) + 0(index column) + (0 × 100 + 4 × 15)(data column) + 132(the number of commas) + 2(CR + LF)

= 194 bytes

<Device comment row>

The device comment row for each column is displayed.

Item	Description	Size
Date column	No output (blank).	0 byte
Data collection interval column		
Execution step No. column		
Execution program number column		
Execution program name column		
Index column		
Data column	When "Output device comment" is selected in the output settings, the comment with the specified comment number is displayed. When "Output each program device comment" is selected, the comment with the specified comment number in the device comment file for each program is output only for the local device. When the device comment file for each program or the comment of the corresponding program does not exist, the comment of the device comment file is output. When comments are set not to be output, or a device comment to be displayed does not exist, this field is blank. Since the device comment is stored in Unicode, the data is converted to ASCII and Shift-JIS. If any characters that cannot be converted to ASCII or Shift-JIS, the conversion fails and those characters will be output as periods (.).	0 to 2048 bytes*5
Trigger occurrence information column	No output (blank).	0 byte

*5 When "Output device comment" is selected in "Device comment output" of the output settings and double quotation marks (" ") and/or commas (,) are included in a device comment, the size of data in use becomes larger than the size of the character string of the device comments.

Ex.

The size of the device comment row is determined by the following equation when data logging of 128 data points from D100 to D227 is performed (The following sections in the "Output" window are set to be output: "Date" (the output format is YYYY/MM/DD hh:mm:ss.sss), "Data sampling interval", "Execution step No.", "Execution program name or execution program No.", "Index", and "Device comment output" (only for devices from D200 to D227)).

0(date column) + 0(data collection interval column) + 0(execution step No. column) + 0(execution program number column) + 0(execution program name column) + 0(index column) + $(0 \times 100 + 7 \times 15)$(data column) + 132(the number of commas) + 2(CR + LF)

= 239 bytes

A

<Data row>

The collected data value is displayed. All the data items collected during a single collection is displayed in a single row.

Item	Description	Size
Date column	Information is output according to the data row output character string specified in the format.	2 to 64 bytes
Data collection interval column	The time interval from the previous collection time to the current collection time is output. If the maximum display range is exceeded, the count returns to 1 and starts again to output a new time interval (unit: µs, display range: 1 to 100000000000).	1 to 12 bytes
Execution step No. column	The step No. executed on the engineering tool at the time interval and timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the step No. at the time of execution of END instruction in the last executed program on the scan. With the collection condition "Interrupt occurrence" specified, the resulting output is the step No. at the time of execution of IRET instruction in the interrupt program. If the system operation (such as system interrupt) is running during collection interval and collection timing or the FB program is in running, "0" is output.	1 to 6 bytes
Execution program number column	The program No. executed on the engineering tool at the time interval and timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the program number of the last executed program in the scan. If two or more system operations, such as collection timing and system interrupt, occur simultaneously, — is output.	1 to 3 bytes
Execution program name column	The program name (no extension) executed on the engineering tool at the time interval and timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the program name of the last executed program in the scan. If a program name with the same program number has already been in the file, a null value is output. If two or more system operations, such as collection timing and system interrupt, occur simultaneously, "* SYSTEM" is output. Since the program name is stored in Unicode, the data is converted to ASCII and Shift-JIS. If any characters that cannot be converted to ASCII or Shift-JIS, the conversion fails and those characters will be output as periods (.).	0 to 120 bytes
Index column	A value which increments in ascending order from 1 is output. When it exceeds the upper limit, it returns to 1 and increments again (range: 1 to 4294967295).	1 to 10 bytes
Data column	When bits are specified: bit On = 1 and bit Off = 0 are output.	1 byte
	When unsigned/signed word type is specified: data value is output according to the specified output type.	• Decimal format: 1 to 6 bytes • Hexadecimal format: 1 to 4 bytes • Decimal fraction format: 1 to 21 bytes • Exponential format: 3 to 21 bytes
	When unsigned/signed double word type, single-precision real number, or double-precision real number is specified: data value is output according to the specified output type.	• Decimal format: 1 to 11 bytes*6 • Hexadecimal format: 1 to 8 bytes • Decimal fraction format: 1 to 26 bytes • Exponential format: 3 to 22 bytes
	When character string is specified: the specified character string is output.	1 to 256 bytes
	When numeric string is specified: the character string which represented by the hexadecimal in increments of a byte is output without clearance.	1 to 256 bytes
	When time is specified: T#-24d20h31m23s648ms to T#24d20h31m23s647ms is displayed.	13 to 20 bytes
Trigger occurrence information column	The specified character string is output when the trigger occurs. In other cases, no character string is output. Since the trigger occurrence information is stored in Unicode, the data is converted to ASCII and Shift-JIS. If any characters that cannot be converted to ASCII or Shift-JIS, the conversion fails and those characters will be output as asterisks (*).	0 to 512 bytes

*6 When single-precision real number or double-precision real number is specified, if the numerical value to be output does not fall into the range -2147483648.0 to 4294967295.0, it is displayed in an equivalent format to "exponential format and the number of decimal part digits is nine".

Ex.

The size of the data type information row is determined by the following equation when data logging of 128 points of data from D100 to D227 (unsigned word type, decimal format) is performed (The following sections in the "Output" window are set to be output: "Date" (the output format is YYYY/MM/DD hh:mm:ss.sss), "Data sampling interval", "Execution step No.", "Execution program name or execution program No.", and "Index").

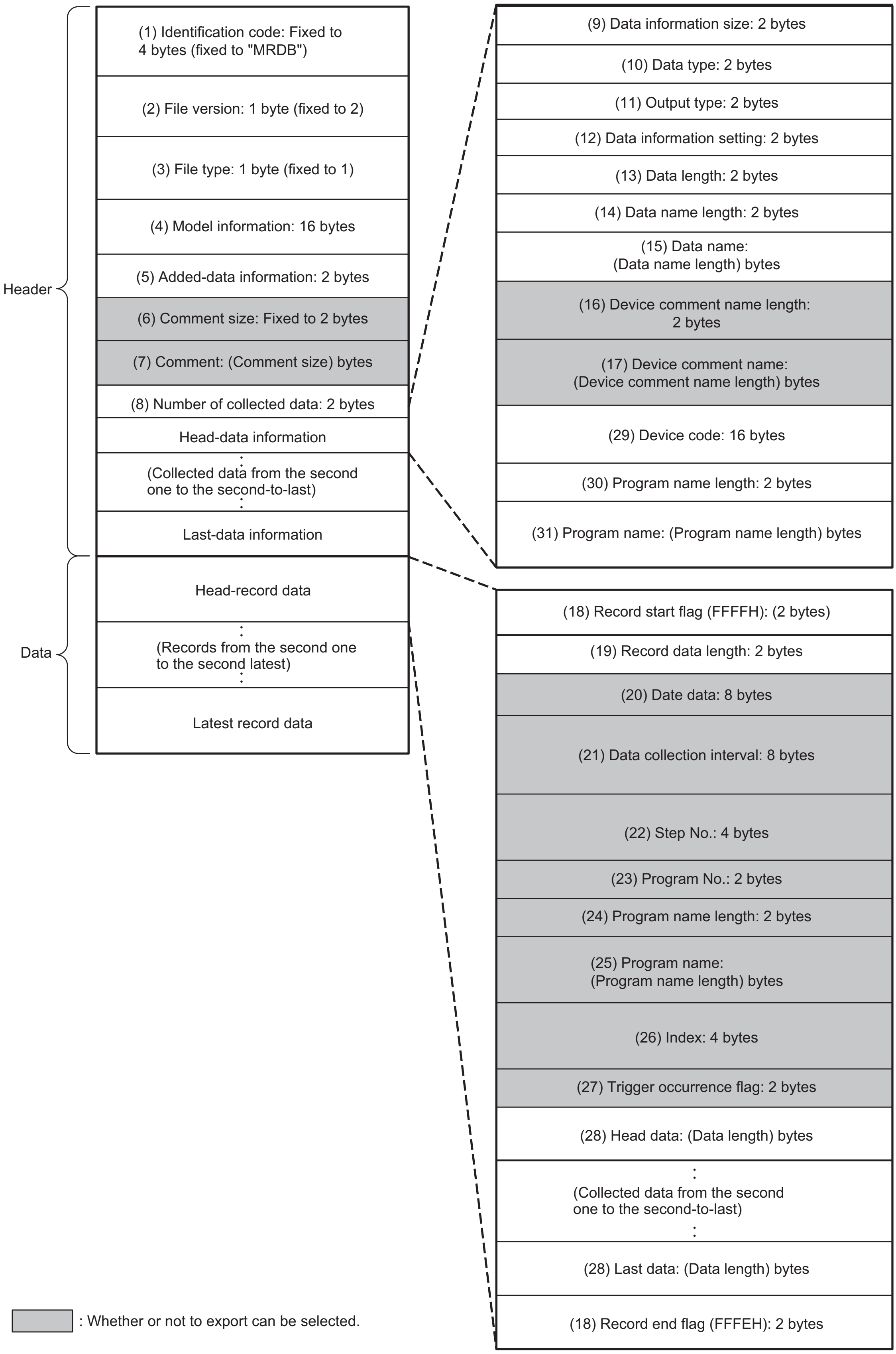
23(date column) + 12(data collection interval column) + 6(execution step No. column) + 3(execution program number column) + 100(execution program name column) + 10(index column) + (6 × 128)(data column) + 132(the number of commas) + 2(CR + LF)

= 1056 bytes

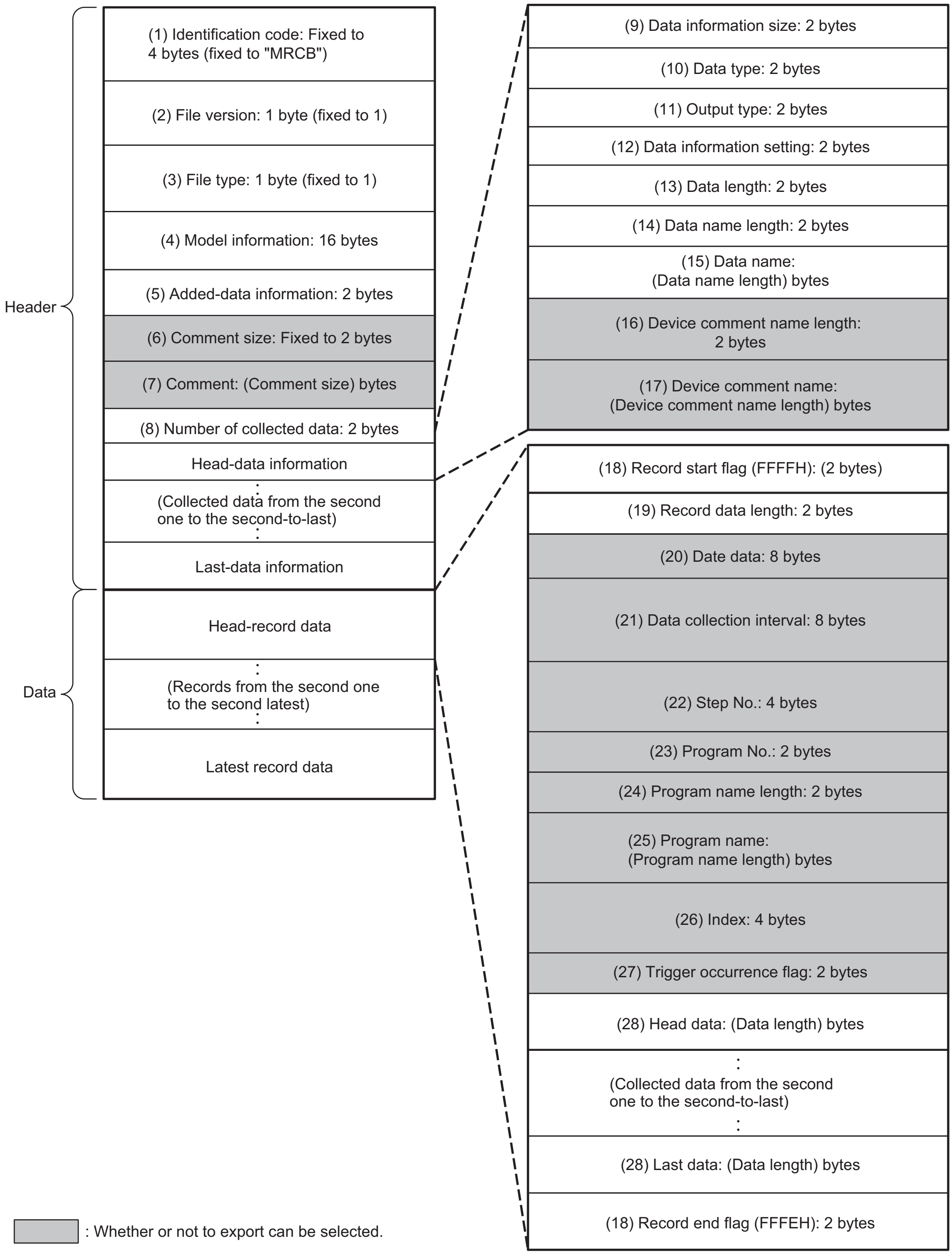
■Binary file output format

The following figure shows the configuration of the binary format and details of each data.

- When a local device or local label is specified



- When a global device or global label is specified



- Details of each data

No.	Item	Description	Size (byte)
(1)	Identification code	When a local device or local label is specified, "MRDB" is always output to identify the file. When a global device or global label is specified, "MRCB" is always output to identify the file.	4
(2)	File version	When a local device or local label is specified, "2" is displayed as the file version. When a global device or global label is specified, "1" is displayed as the file version.	1
(3)	File type	The file type is output. (fixed to 1: Continuous/trigger logging)	1
(4)	Model information	The module model name that outputted binary file is output. R□CPU is output to the first eight bytes and 00H is output to the last eight bytes.[*1]	16
(5)	Added-data information	The output selection setting for the data that can be output is output. b9 b7 b6 b5 b4 b3 b2 b1 b0 b0 1: Output date and time data. 0: Do not output date and time data. b1 1: Output a data collection interval. 0: Do not output a data collection interval. b2 1: Output an execution step No. 0: Do not output an execution step No. b3 1: Output an execution program name and No. 0: Do not output an execution program name or No. b4 1: Output a trigger flag. 0: Do not output a trigger flag. b5 1: Output index. 0: Do not output index. b6 1: Output a device comment. (When using the Safety CPU, this bit can be specified for the module with firmware version "34" or later.) 0: Do not output device comments. b7 1: Output comments. 0: Do not output comments. b9 1: Output device comments for each program. (When using the Safety CPU, this bit can be specified for the module with firmware version "34" or later.) 0: Do not output device comments for each program.	2
(6)	Comment size	The comment length of (7) Comment is output.	2
(7)	Comment	The comment specified in the setting is output in Unicode.	2 to 512
(8)	Number of collected data	The number of data points of the data information ((9) to (17) and (29) to (31)) for data logging is output.	2
(9)	Data information size	The total size of the data information ((9) to (17) and (29) to (31)) for data logging is output.	2
(10)	Data type	The numeric value shown below is output depending on the data type. 0000H: Bit 0001H: Word (signed) 0002H: Double word (unsigned) 0003H: Word (signed) 0004H: Double word (unsigned) 0005H: Single-precision real number 0006H: Double-precision real number 0007H: String 0008H: Numeric string 0009H: Time	2
(11)	Output type	The same numerical value as the value in (10) Data type is output. However, when the data type is bit, character string, numeric string, or time, FFFFH is output because the output type cannot be specified.	2
(12)	Data information setting	The data-related information is output. b4 b3 b2 b2 1: Device comment specified (When using the Safety CPU, this bit can be specified for the module with firmware version "34" or later.) 0: Device comment not specified b3 1: Device code specified 0: Device code not specified b4 1: Program name length/program name specified 0: Program name length/program name not specified	2
(13)	Data length	The data length of data is output. When the data type is the bit type, it will be output as two bytes.	2
(14)	Data name length	The length of the data name specified in the setting is output.	2
(15)	Data name	The data name specified in the setting is output in Unicode.	2 to 512

No.	Item	Description	Size (byte)
(16)	Device comment name length	The length of the device comment name specified in the setting is output. When "Output device comment" is selected and the corresponding device comment exists, the device comment is output. When the corresponding device comment does not exist, "0" is output.	2
(17)	Device comment name	The device comment name specified in the setting is output in Unicode. When "Output each program device comment" is selected and the corresponding device comment exists, each program device comment of a specified comment number is output to the local device. When the corresponding device comment does not exist, no data is output. If a device comment is set for the target device while each program device comment of the corresponding program does not exist or a comment for the target device is not set, the device comment set for the target device is output. When the label is specified as the data to be logged, data is not output.	0 to 2048
(18)	Record start flag, record end flag	The flags for identifying the start and end of the record are output. The FFFFH is output for record start while the FFFEH is output for record end as the fixed flag.	2
(19)	Record data length	The total size of (20) Day and time data to (28) Last data is output.	2
(20)	Date data	The Day and time data is output. b15 to b0 Year | Month — Year: Last 2 digits of the year, Month: 1 to 12 Day | Time — Day: 1 to 31, Time: 0 to 23 Minute | Second — Minute: 0 to 59, Second: 0 to 59 Millisecond — Millisecond: 0 to 999	8
(21)	Data collection interval	The time interval from the previous collection time to the current collection time is output. (Unit: μs, Display range: 1 to 100000000000 (When it exceeds the max value, it returns to "1" and incrementing runs again.)) After logging collection is started, 0 is stored at the first collection.	8
(22)	Step No.	The step No. executed with the engineering tool at the timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the step No. at the time of execution of END instruction in the last executed program on the scan. With the collection condition "Interrupt occurrence" specified, the resulting output is the step No. at the time of execution of IRET instruction in the interrupt program. If the system operation (such as system interrupt) is running during collection interval and collection timing or the FB program is in running, "0" is output.	4
(23)	Program No.	The program No. (FB File No) executed on the engineering tool at the timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the program number of the last executed program in the scan. If system operation (such as system interrupt) is running at collection interval and collection timing, "0" is output.	2
(24)	Program name length	The name length of a program that is executed at the time interval and timing in which data was collected is output. If the same program number or program name has already been in the file, "0" is output.	2
(25)	Program name	The program name (no extension) that is executed at the time interval and timing in which data was collected is output in Unicode. With the collection condition "Each scanning cycle" specified, the resulting output is the program name of the last executed program in the scan. If system operation (such as system interrupt) is running at collection interval and collection timing, "* SYSTEM" is output.	0 to 200
(26)	Index	The index number ranging from 1 to 4294967295 of data, which was collected by the data logging function, is output. When it exceeds the max value, it returns to "1" and incrementing runs again. If missing occurs in processing data, index will be reassigned from 1 again.	4
(27)	Trigger occurrence flag	The trigger occurrence information is output. b0 b0 1: A trigger has occurred. 0: No trigger has occurred.	2
(28)	Data	Data collected by the data logging function is output corresponding to (13) Data length and (10) Data type. • When bits are specified: bit On = 1 and bit Off = 0 are output. • When word type (signed/unsigned) or double-word type (signed/unsigned) is specified: the data values are output in the specified unit. • When single-precision real number or double-precision real number is specified: data value is output in the specified unit. (☞ Page 1125 Numerical value range for each output type) • When character string type is specified: the character string with the specified size is output. If the character string terminator "0" exists in the middle of data, NULL is generated on from said point onward until the terminator of the specified size. • When numeric string type is specified: the data value with the specified size is output. • When time is specified: T#-24d20h31m23s648ms to T#24d20h31m23s647ms is output.	• Bit: 2 • Word (signed/ unsigned): 2 • Double word (signed/ unsigned): 4 • Single-precision real number: 4 • Double-precision real number: 8 • String/numeric string: 1 to 256 • Time: 4
(29)	Device code	The device code of the device/label specified in the setting is output.	16
(30)	Program name length	The length of the program name specified in the setting is output. (When a global device or global label is specified, "0" is output.)	2
(31)	Program name	The specified program name (no extension) is output in Unicode. (When a global device or global label is specified, the program name is not output.)	0 to 120

*1 The following capacity values are output to □ based on the model.
Example: R08CPU, R08ENCPU, R08PCPU, R08SFCPU: □ = 08

Numerical value range for each output type

This section describes the numerical value ranges that can be output for each output type.

■Integer type

The following table lists the numerical value ranges that can be expressed for each integer type.

Output format	Lower limit	Upper limit
Word (unsigned)	0	65535
Word (signed)	-32768	32767
Double word (unsigned)	0	4294967295
Double word (signed)	-2147483648	2147483647

■Real number type

The following table lists the numerical value ranges that can be expressed for each real number type.

Output format	Negative value		Positive value	
	Lower limit	Upper limit	Lower limit	Upper limit
Single-precision real number	-3.4028235E+38	-1.401298E-45	1.401298E-45	3.4028235E+38
Double-precision real number	-1.79769313486231570E+308	-4.94065645841246544E-324	4.94065645841246544E-324	1.79769313486231570E+308

When the value of the data exceeds the numerical value range, the following rule is applied to the output.

- When the value exceeds the upper limit of the positive value, +Inf is output.
- When the value falls below the lower limit of the negative value, -Inf is output.
- When the value is between the upper limit of the negative value and the lower limit of the positive value, 0 is output.

Output format	-Inf	0	+Inf
Single-precision real number	0xff800000	0x00000000	0x7f800000
Double-precision real number	0xfff0000000000000	0x0000000000000000	0x7ff0000000000000

A

Data logging procedure

This section describes the data logging procedure.

1. Install CPU Module Logging Configuration Tool.
2. Launch CPU Module Logging Configuration Tool.
3. When specifying a label as the target data, import the project of the engineering tool to CPU Module Logging Configuration Tool.
4. Configure the data logging setting.
5. If auto logging is used, configure the common settings (auto logging common settings).
6. When the storage destination of the data logging files is the SD memory card, insert an SD memory card into the CPU module, and power up the module.
7. Connect the CPU module to a personal computer.
8. Write the data logging setting to the SD memory card or data memory.
9. Switch the CPU module to RUN state to start the data logging.
10. Stop the data logging and read the data logging file.
11. Check the file into which data has been read.

Appendix 14 List of Available SQL Commands for CPU Module Database Access Function

This section describes the available SQL commands for the CPU module database access function.

Data Definition

CREATE TABLE

This command creates a table in the database.

Syntax
CREATE TABLE [table name] ([field name] [data type] [option], [field name] [data type] [option], ···, [field name] [data type] [option]);

■Option

Item	Description	Syntax
Primary key constraint	Maintains the uniqueness of a value stored in the field. (NULL is not available.) Only one primary key constraint can be set per table. Key names must be up to 16 single-byte alphabetical characters and are case sensitive.	PRIMARY KEY [key name]
Foreign key constraint	Refers to the value of a field of another table. Only the field with the name where the primary key constraint of another table is set can be set as the reference field. (Always set the same field name to the reference field and own field.) Key names must be up to 16 single-byte alphabetical characters and are case sensitive.	FOREIGN KEY [key name] REFERENCES [reference table name] ([reference field name])
NOT NULL constraint	Disables storing NULL values. (Always store a value other than NULL.)	NOT NULL

■Application example

- To create table1 (fld1 (integer type, primary key (key name: pk1)), fld2 (integer type), fld3 (character string type (120 characters, NOT NULL)))

CREATE TABLE "table1" ("fld1" INT PRIMARY KEY "pk1", "fld2" INT, "fld3" NLSCHAR (120) NOT NULL);

- To create table2 (fld1 (integer type, foreign key (key name: fk1, reference: table1 fld1)), fld2 (BOOLEAN type), fld3 (single-precision real number type))

CREATE TABLE "table2" ("fld1" INT FOREIGN KEY "fk1" REFERENCES "table1" ("fld1"), "fld2" BOOLEAN, "fld3" REAL);

■Precautions

- To define the NLSCHAR field, check the data size of the field.
- Define the NLSCHAR type with the size specified in NLSCHAR = (n+1) × 3*1 since it needs to be defined with the data size. For example, the definition is 39 characters for "fld3" NLSCHAR (120) of table1 in the application example.

*1 n indicates the number of characters.

DROP TABLE

This command deletes the specified table in the database.

Syntax
DROP TABLE [table name];

■Application example

- To delete table 1

DROP TABLE "table1";

A

CREATE INDEX

This command creates an index in the specified field of the specified table in the database. Index names must be up to 16 single-byte alphabetical characters and are case sensitive.

Syntax
CREATE INDEX [index name] ON [table name] (field name);

■Application example

- To set the index "idx1" to fld2 of table1

CREATE INDEX "idx1" ON "table1" ("fld2");

DROP INDEX

This command deletes the specified index in the database.

Syntax
DROP INDEX [index name] ON [table name];

■Application example

- To delete the "idx1" index in table1

DROP INDEX "idx1" ON "table1";

EMPTY TABLE

This command deletes all records of the specified table in the database.

Syntax
EMPTY [table name];

■Application example

- To delete all records in table1

EMPTY "table1";

Data Operation

INSERT

This command adds a record to the specified table in the database.

Syntax*1*2
• To add records to all fields INSERT INTO [table name] VALUES (field 1 value, field 2 value, ···, field X value); • To add records to the specified fields INSERT INTO [table name] (field specification 1, field specification 2, ···, field specification X) VALUES (field 1 value, field 2 value, ···, field X value);

*1 Input null not to store the field value.
*2 When the field value is a character string, enclose it in single quotation marks (').

■Application example

• To add records to all fields

INSERT INTO "table1" VALUES (11, null, 'abc');

• To add records to the specified fields

INSERT INTO "table1" ("fld1", "fld3") VALUES (12, 'efg');

SELECT

This command outputs a value of the specified record from the specified table in the database.

Syntax
• To output all values SELECT [output target field or numeric expression] FROM [table name] [option]; • To output a value excluding overlapped values SELECT DISTINCT [output target field or numeric expression] FROM [table name] [option 1] [option 2];

■Option 1

Item	Description	Syntax
INNER JOIN	Performs inner join. (Records that exist in the specified both tables are output.)	SELECT [output target field or numeric expression] FROM [table name 1] INNER JOIN [table name 2] ON [combining condition];
LEFT JOIN	Performs outer join. (All records that exist in [table name 1] are output.)	SELECT [output target field or numeric expression] FROM [table name 1] LEFT JOIN [table name 2] ON [combining condition];
RIGHT JOIN	Performs outer join. (All records that exist in [table name 2] are output.)	SELECT [output target field or numeric expression] FROM [table name 1] RIGHT JOIN [table name 2] ON [combining condition];
FULL JOIN	Performs outer join. (All records that exist in both of specified tables are output.)	SELECT [output target field or numeric expression] FROM [table name 1] FULL JOIN [table name 2] ON [combining condition];

■Option 2*1

Item	Description
WHERE	Specifies the condition to obtain a specific record. (☞ Page 1131 WHERE).
GROUP BY	Specifies the grouping condition of the record to be obtained. (☞ Page 1131 GROUP BY).
HAVING	Specifies the condition to narrow down a search using the result of the aggregate function. (☞ Page 1131 HAVING).
ORDER BY	Specifies the sorting condition of the record to be obtained. (☞ Page 1132 ORDER BY).

*1 Multiple options can be used.

■Application example

• To output all record values in table1 (all fields of table1)

SELECT * FROM table1;

• To output all record values in table1 (fld1 and fld2 in table1)

SELECT "fld1", "fld2" FROM table1;

• To output all records excluding the overlapped records in fld1 of table1

SELECT DISTINCT "fld1" FROM table1;

UPDATE

This command updates the value of the specified record of the specified table in the database.

Syntax*1
UPDATE [table name] SET [field name = assigned value], [field name = assigned value], ···, [field name = assigned value] [option];

*1 When the field value is a character string, enclose it in single quotation marks (').

■Option

Item	Description	Syntax
WHERE	Specifies the condition to obtain (process) a specific record (☞ Page 1131 WHERE).	WHERE

■Application example

- To store a value in "fld1" and "fld3" of the record of "fld2" field value = 935 of table1 (storing fld1 = 20, fld3 = abc)

UPDATE table1 SET "fld1" = 20, "fld3" = 'abc' WHERE "fld2" = 935;

DELETE

This command deletes a record of the specified table in the database.

Syntax*1
DELETE [table name] [option];

*1 When the field value is a character string, enclose it in single quotation marks (').

■Option

Item	Description	Syntax
WHERE	Specifies the condition to obtain (process) a specific record (☞ Page 1131 WHERE).	WHERE

■Application example

- To delete all records in table1

DELETE "table1";

- To delete the record of the fld3 value = abc in table1

DELETE "table1" WHERE "fld3" = 'abc';

Clause

WHERE

This command conducts search with conditions.

Syntax

- To specify a condition

WHERE [condition]

- To specify an exception

WHERE NOT [condition];

- To specify two conditions with AND

WHERE [condition 1] AND [condition 2];

- To specify two conditions with OR

WHERE [condition 1] OR [condition 2];

■Operators that can be used in [condition]

Item	Description	Syntax
=	Equals	[numeric expression 1] = [numeric expression 2]
!=	Not equal to	[numeric expression 1] != [numeric expression 2]
<>	Not equal to	[numeric expression 1] <> [numeric expression 2]
>	Greater than	[numeric expression 1] > [numeric expression 2]
>=	Greater than or equal to	[numeric expression 1] >= [numeric expression 2]
<	Less than	[numeric expression 1] < [numeric expression 2]
<=	Less than or equal to	[numeric expression 1] <= [numeric expression 2]
IN	Brings the conditions to be set together.	IN ([numeric expression 1], [numeric expression 2], ···, [numeric expression 3])
BETWEEN	Specifies the range to be output.	[numeric expression 1] BETWEEN [numeric expression 2] AND [numeric expression 3]

■Application example

- To output a record with the fld1 value of 1 to 100 of table1

SELECT * FROM "table1" WHERE "fld1" BETWEEN 1 AND 100;

- To output fld1 and fld3 of the record with the fld3 value other than abc of table1

SELECT "fld1", "fld3" FROM "table1" WHERE NOT "fld3" = 'abc';

GROUP BY

This command groups elements.

Syntax

GROUP BY [numeric expression]

■Application example

- To group Table1 according to the value of fld1 and aggregate the number of records

SELECT COUNT(*) FROM "table1" GROUP BY "fld1";

HAVING

This command narrows down a search using the result of the aggregate function.

Syntax

HAVING [numeric expression]

■Application example

- To output only the groups whose total value is 50 or less

SELECT SUM ("fld2") FROM "table1" GROUP BY "fld1" HAVING SUM ("fld2") < 50;

ORDER BY

This command sorts output.

Syntax
• Ascending ORDER BY [numeric expression] ASC • Descending ORDER BY [numeric expression] DESC

■Application example

• To sort and output fld1 and fld2 of all records in ascending order of the fld1 value from table1

SELECT "fld1", "fld2" FROM "table1" ORDER BY "fld1" ASC;

Aggregate Function

AVG

This command obtains a mean value.

Syntax
AVG ([numeric expression])

■Application example

• To obtain the mean value of fld1 from table1 (including overlapped values)

SELECT AVG ("fld1") FROM "table1";

• To obtain the mean value of fld1 from table1 (excluding overlapped values)

SELECT AVG (DISTINCT "fld1") FROM "table1";

COUNT

This command aggregates the number of records.

Syntax
COUNT [target]

■Application example

• To obtain the number of records of fld1 from table1 (including overlapped values)

SELECT COUNT ("fld1") FROM "table1";

• To obtain the number of records of fld1 from table1 (excluding overlapped values)

SELECT COUNT (DISTINCT "fld1") FROM "table1";

MAX

This command obtains a maximum value.

Syntax
MAX ([numeric expression])

■Application example

• To obtain the maximum value of fld1 from table1

SELECT MAX ("fld1") FROM "table1";

MIN

This command obtains a minimum value.

Syntax
MIN ([numeric expression])

■Application example

• To obtain the minimum value of fld1 from table1

SELECT MIN ("fld1") FROM "table1";

SUM

This command obtains the total.

Syntax
SUM ([numeric expression])

■Application example

• To obtain the total of fld1 from table1 (including overlapped values)

SELECT SUM ("fld1") FROM "table1";

• To obtain the total of fld1 from table1 (excluding overlapped values)

SELECT SUM (DISTINCT "fld1") FROM "table1";

Arithmetic Function

ABS

This command obtains an absolute value.

Syntax
ABS ([numeric expression])

■Application example

- To obtain the absolute value of fld1 from table1

SELECT ABS ("fld1") FROM "table1";

ACOS

This command obtains an arc cosine.

Syntax
ACOS ([numeric expression])

■Application example

- To obtain the arc cosine of fld1 from table1

SELECT ACOS ("fld1") FROM "table1";

ASIN

This command obtains an arc sine.

Syntax
ASIN ([numeric expression])

■Application example

- To obtain the arc sine of fld1 from table1

SELECT ASIN ("fld1") FROM "table1";

ATAN

This command obtains an arc tangent.

Syntax
ATAN ([numeric expression])

■Application example

- To obtain the arc tangent of fld1 from table1

SELECT ATAN ("fld1") FROM "table1";

ATAN2

This command obtains an arc tangent.

Syntax
ATAN2 ([numeric expression 1], [numeric expression 2])*1

*1 This syntax is ATAN2 (y, x).

■Application example

- To obtain an arc tangent of fld1/fld2 from table1

SELECT ATAN2 ("fld1", "fld2") FROM "table1";

CEILING

This command obtains the smallest integer greater than or equal to the specified numeric expression.

Syntax
CEILING ([numeric expression])

■Application example

- To obtain the smallest integer greater than or equal to the fld1 value from table1

SELECT CEILING ("fld1") FROM "table1";

COS

This command obtains a cosine.

Syntax
COS ([numeric expression])

■Application example

- To obtain the cosine of fld1 from table1

SELECT COS ("fld1") FROM "table1";

COT

This command obtains a cotangent.

Syntax
COT ([numeric expression])

■Application example

- To obtain the cotangent of fld1 from table1

SELECT COT ("fld1") FROM "table1";

DEGREES

This command obtains degrees from radians.

Syntax
DEGREES ([numeric expression])

■Application example

- To obtain degrees of fld1 from table1

SELECT DEGREES ("fld1") FROM "table1";

EXP

This command obtains an exponent value. (It calculates e raised to the power of [numeric expression].)

Syntax
EXP ([numeric expression])

■Application example

- To obtain e raised to the power of fld1 from table1

SELECT EXP ("fld1") FROM "table1";

FLOOR

This command obtains the largest integer less than or equal to the specified numeric expression.

Syntax
FLOOR ([numeric expression])

■Application example

- To obtain the largest integer less than or equal to the fld1 value from table1

SELECT FLOOR ("fld1") FROM "table1";

LOG

This command obtains a natural logarithm.

Syntax
• To obtain a natural logarithm LOG ([numeric expression]) • To obtain a decadic logarithm LOG10 ([numeric expression])

■Application example

- To obtain the natural logarithm of fld1 from table1

SELECT LOG ("fld1") FROM "table1";

- To obtain the decadic logarithm of fld1 from table1

SELECT LOG10 ("fld1") FROM "table1";

POW

This command raises a value. (It raises [numeric expression 1] to the power of [numeric expression 2].)

Syntax
POW ([numeric expression 1], [numeric expression 2]) When [numeric expression 1] = 0, [numeric expression 2] must be greater than 0. When [numeric expression 1] is less than 0, [numeric expression 2] must be an integer.

■Application example

- To obtain fld1 raised to the power of fld2 from table1

SELECT POW ("fld1", "fld2") FROM "table1";

RADIANS

This command obtains radians from degrees.

Syntax
RADIANS ([numeric expression])

■Application example

- To obtain radians of fld1 from table1

SELECT RADIANS ("fld1") FROM "table1";

ROUND

This command rounds a value. (The value of [numeric expression 1] is rounded off to the nth decimal place.)*1

Syntax
ROUND ([numeric expression 1], [numeric expression 2])

*1 Add one to the value specified in n = [numeric expression 2].

■Application example

- To obtain the value of fld1 rounded off to the third decimal place from table1.

SELECT ROUND ("fld1", 2) FROM "table1";

SIGN

This command obtains a sign.

Syntax
SIGN ([numeric expression])

■Application example

- To obtain the sign of fld1 from table1

SELECT SIGN ("fld1") FROM "table1";

SIN

This command obtains a sine.

Syntax
SIN ([numeric expression])

■Application example

- To obtain the sine of fld1 from table1

SELECT SIN ("fld1") FROM "table1";

SQRT

This command obtains a square root.

Syntax
SQRT ([numeric expression])

■Application example

- To obtain the square root of fld1 from table1

SELECT SQRT ("fld1") FROM "table1";

A

TAN

This command obtains a tangent.

Syntax
TAN ([numeric expression])

■Application example

- To obtain the tangent of fld1 from table1

SELECT TAN ("fld1") FROM "table1";

Appendix 15 Precautions for Communications with CPU Module in Redundant System via Module on Extension Base Unit

In the case where the engineering tool accesses the CPU module in a redundant system via a module on an extension base unit, executable functions of the engineering tool varies depending on the setting of "Specify Redundant CPU" in the "Specify Connection Destination" window. The following table lists these functions and shows whether each function can be executed or not.

For the functions that cannot be executed regardless of the setting or those that are not listed below, execute them either connecting the computer directly to the CPU module or accessing the CPU module via a module on the main base unit.

○: Yes, ×: No

Function		Specify Redundant CPU	
		System A/System B	Not Specified/Control System/ Standby System
Read from PLC		○	×
Write to PLC (including the file batch online change)		×	×
Online program change		×	×
Verify with PLC		×	×
Delete PLC Data		×	×
User data operation	Read	○	×
	Write	×	×
	Delete	×	×
Initialization of the CPU built-in memory/SD memory card		×	×
Device/label data test (changing values)		○	○
Clear value (devices, labels, file registers, latches)		×	×
System switching		○	×
Operation mode change		○	×
Monitoring a program (circuit monitor)	Monitor mode	○	○
	Monitor (Write mode)	○*1	○*1
Program list monitor/interrupt program list monitor		○	○
SFC all blocks batch monitor		○	○
SFC auto-scroll		○	○
Device/buffer memory batch/registration monitor		○	○
System monitor		○	○
Module diagnostics		○	○*2

*1 The program consistency check ("Check the consistency between the editing target program file in GX Works3 and the one in PLC.") cannot be executed. (GX Works3 Operating Manual)

*2 When "Specify Redundant CPU" is set to "Not Specified", "Control System", or "Standby System", the event history cannot be displayed or cleared.

Appendix 16 Added and Enhanced Functions

This section describes added and enhanced functions of the CPU module and the engineering tool, as well as the firmware versions of the CPU module and software versions of the engineering tool corresponding to the functions.

Point

The firmware update function enables users to update the firmware versions of the CPU module. For the target CPU modules and how to update the firmware version, refer to the following.
MELSEC iQ-R Module Configuration Manual

○: Supported from the first released product, ×: Not supported, —: Function usable regardless of the software version

Added or enhanced function	F: Firmware version of the CPU module S: Software version of the engineering tool O: Version of the other tools					Reference
	R00/R01/R02CPU	Rn(EN)CPU	RnPCPU	RnPSFCPU and R6PSFM	RnSFCPU and R6SFM	
LED specifications (enhanced use of the FUNCTION LED)	F: ○ S: 1.040S	F: 06 S: 1.005F	×	×	F: 21 S: 1.065T	Page 143 LED display setting
Memory dump function	F: ○*1 S: 1.040S	F: 06 S: 1.005F	×	×	F: 21*10 S: 1.065T	Page 255 DEBUG FUNCTION
Real-time monitor function	F: ○ S: — O: 1.76E*2	F: 06 S: — O: 1.40S*2	×	×	F: 19*10 S: — O: 1.100E*2	• Page 170 MONITOR FUNCTION • GX LogViewer Version 1 Operating Manual
CC-Link IE Controller Network function of the RJ71EN71	F: ○ S: 1.040S	F: 06 S: 1.005F	×	×	F: 19 S: 1.060N	• MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup) • MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)
Mounting of MELSEC iQ-R series modules occupying two slots	F: ○ S: 1.040S	F: 08 S: —	F: ○ S: 1.007H	F: ○ S: 1.040S	F: ○ S: 1.015R	For the applicable modules, refer to the following. MELSEC iQ-R Module Configuration Manual
MELSEC-Q series MELSECNET/H network module	F: ○ S: 1.040S	F: 10 S: 1.010L	×	×	F: 18 S: 1.057K	For the applicable modules, refer to the following. MELSEC iQ-R Module Configuration Manual
SFC support	F: ○ S: 1.040S O: 1.49B*2*3*4	F: 12 S: 1.015R O: 1.49B*2*3*4	■In process mode F: 03 S: 1.020W O: 1.49B*2*3*4 ■In redundant mode F: 18 S: 1.050C	×	F: 26*10 S: 1.080J O: 1.49B*2*3*4	• MELSEC iQ-R CPU Module User's Manual (Startup) • MELSEC iQ-R Programming Manual (Program Design)
Extended SRAM cassette (NZ2MC-16MBS)	×	F: 12 S: 1.015R	×	×	×	• Page 103 Device/label memory area setting • MELSEC iQ-R CPU Module User's Manual (Startup)
iQ Sensor Solution functions supporting Ethernet • Automatic detection of connected device • Communication setting reflection • Sensor parameter read/write	F: ○ S: 1.040S	F: 12 S: 1.015R	×	×	×	iQ Sensor Solution Reference Manual
iQ Sensor Solution functions supporting CC-Link IE Field Network • Automatic detection of connected device • Communication setting reflection • Sensor parameter read/write	F: ○ S: 1.040S	F: 12 S: —	×	×	×	iQ Sensor Solution Reference Manual
Daylight saving time function	F: ○ S: 1.040S	F: 17 S: 1.020W	×	×	F: 34 S: 1.110Q	Page 121 Daylight Saving Time Function

Added or enhanced function	F: Firmware version of the CPU module S: Software version of the engineering tool O: Version of the other tools					Reference
	R00/R01/ R02CPU	Rn(EN)CPU	RnPCPU	RnPSFCPU and R6PSFM	RnSFCPU and R6SFM	
Sequence scan synchronization sampling function	F: ○ S: 1.040S	F: 17 S: 1.020W	×	×	F: 18 S: —	Page 362 SEQUENCE SCAN SYNCHRONIZATION SAMPLING FUNCTION
Duplication check for POU	F: ○ S: 1.040S	F: 19 S: 1.022Y	F: 04 S: 1.025B	F: ○ S: 1.040S	F: 04 S: 1.022Y	GX Works3 Operating Manual
Background processing of program transfer (writing the program restoration information) during changing a program online	F: ○ S: 1.040S	F: 20 S: 1.022Y	■In process mode F: 10 S: 1.022Y ■In redundant mode F: 15 S: 1.045X	×	F: 04 S: 1.022Y	GX Works3 Operating Manual
CPU module data backup/ restoration function	×	F: 22 S: —	F: 20 S: —	×	F: 18 S: 1.057K	Page 299 CPU MODULE DATA BACKUP/RESTORATION FUNCTION
iQ Sensor Solution functions supporting CC-Link IE Field Network • Data backup/restoration (128 modules maximum)*5	F: ○ S: 1.040S	F: 22 S: —	×	×	F: 18 S: —	iQ Sensor Solution Reference Manual
iQ Sensor Solution functions supporting CC-Link IE Field Network • Data backup/restoration (256 modules maximum)*5	F: 14 S: 1.060N	F: 46 S: 1.060N	×	×	×	iQ Sensor Solution Reference Manual
iQ Sensor Solution functions supporting CC-Link-AnyWireASLINK • Data backup/restoration*5	F: ○ S: 1.040S	F: 22 S: —	×	×	×	iQ Sensor Solution Reference Manual
Redundant function	×	×	F: 04 S: 1.025B	F: ○ S: 1.040S	×	• Process CPU: Page 492 FUNCTIONS • SIL2 Process CPU: Page 704 FUNCTIONS
Redundant power supply system	F: ○ S: 1.040S	F: 22 S: 1.025B	F: 04 S: 1.025B	F: ○ S: 1.040S	×	MELSEC iQ-R Module Configuration Manual
Label initialization function	×	×	F: 04 S: 1.025B	F: ○ S: 1.040S	×	Page 364 LABEL INITIALIZATION FUNCTION
Firmware update function (Firmware update using the engineering tool)	F: 14 S: 1.060N	F: 46 S: 1.060N	F: 24 S: 1.065T	×	×	MELSEC iQ-R Module Configuration Manual
Firmware update function (Firmware update using an SD memory card)	F: ○*1 S: —	■For RnCPU F: 23 S: — ■For RnENCPU F: 38 S: —	F: 14 S: —	×	×	MELSEC iQ-R Module Configuration Manual
External input/output forced on/off function	F: ○ S: 1.040S	F: 25 S: 1.030G	F: 15 S: 1.045X	×	F: 16 S: 1.050C	Page 178 External Input/Output Forced On/Off Function
iQ Sensor Solution functions supporting AnyWireASLINK • Automatic detection of connected device • Communication setting reflection • Sensor parameter read/write	F: ○ S: 1.040S	F: 25 S: 1.030G	×	×	×	iQ Sensor Solution Reference Manual
iQ Sensor Solution functions supporting Ethernet • Data backup/restoration*5	F: ○ S: —	F: 25 S: —	×	×	×	
CC-Link IE Field Network Basic	F: ○ S: 1.040S	F: 25 S: 1.030G	×	×	×	CC-Link IE Field Network Basic Reference Manual

Added or enhanced function	F: Firmware version of the CPU module S: Software version of the engineering tool O: Version of the other tools					Reference
	R00/R01/ R02CPU	Rn(EN)CPU	RnPCPU	RnPSFCPU and R6PSFM	RnSFCPU and R6SFM	
Multiple CPU system	F: ○ S: 1.040S	■For RnCPU F: ○ S: 1.000A ■For RnENCPU F: 25 S: 1.032J	F: ○[*6] S: 1.007H[*6]	×	F: ○ S: 1.015R ■When the Safety CPU is used as CPU No.1 F: 14 S: 1.045X	MELSEC iQ-R Module Configuration Manual
CPU module database access function	×	F: 28 S: 1.035M	×	×	×	Page 266 CPU module database access (from external device) function
"Use MC/MCR to Control EN" of subroutine-type FB	F: ○ S: 1.040S	F: 28 S: 1.035M	×	×	×	• GX Works3 Operating Manual • MELSEC iQ-R Programming Manual (Program Design)
Program restoration information write selection	F: ○ S: 1.040S	F: 31 S: 1.040S	■In process mode F: 13 S: 1.040S ■In redundant mode F: 15[*8] S: 1.045X[*8]	×	F: 13[*10] S: 1.045X[*10]	Page 1101 Program Restoration Information Write Selection
File batch online change of FB files and the global label setting file	F: ○ S: 1.040S	F: 31 S: 1.040S	F: 13 S: 1.040S	×	F: 17[*14] S: 1.055H[*14]	Page 129 File batch online change
File batch online change when the program memory does not have enough free space	F: ○ S: 1.040S	F: 31 S: 1.040S	F: 13 S: 1.040S	×	F: 17[*14] S: 1.055H[*14]	Page 129 File batch online change
Storage of logging data in the function memory	×	F: 31 S: — O: 1.76E[*4]	×	×	×	Page 212 Storage location of data logging files
Data logging file transfer function	F: 06[*1] S: — O: 1.76E[*4]	F: 31 S: — O: 1.76E[*4]	×	×	×	Page 226 Data Logging File Transfer (Auto Transfer to FTP Server)
Extension of points for safety input device and safety output device	×	×	×	F: ○ S: 1.040S	F: 04 S: 1.019V	• Page 636 Safety device area setting range • Page 654 List of safety devices • MELSEC iQ-R CPU Module User's Manual (Startup)
Auto logout time change of the user authentication function (5 minutes → 60 minutes)	×	×	×	F: ○ S: —	F: 04 S: —	—
Safety communication function using the simple motion module[*9]	×	×	×	×	F: 07 S: 1.030G	MELSEC iQ-R Simple Motion Module User's Manual (Application)
Error detection invalidation setting	F: 03 S: —	F: 33 S: —	F: 14 S: —	×	×	Page 144 Error detection invalidation setting
Local device and label specification for target data/condition specification of the data logging function	F: 05[*1] S: 1.045X O: 1.82L[*4]	F: 35 S: 1.045X O: 1.82L[*4]	×	×	F: 34 S: 1.110Q O: 1.160S[*4]	Page 199 DATA LOGGING FUNCTION
Local device and label available for monitor target and monitor condition of the real-time monitor function	F: 05[*1] S: 1.045X O: 1.82L[*2]	F: 35 S: 1.045X O: 1.82L[*2]	×	×	F: 34 S: 1.110Q O: 1.160S[*2]	GX LogViewer Version 1 Operating Manual
'Communication load status' (Un\G100 to Un\G103)	F: 05 S: —	F: 35 S: —	F: 22 S: —	×	×	MELSEC iQ-R Ethernet User's Manual (Application)
Battery-less option cassette	×	F: 35 S: 1.045X	×	×	×	Page 443 Latch with Battery-less Option Cassette
iQ Sensor Solution functions for CC-Link • Data backup/restoration[*5]	F: 05[*1] S: —	F: 35 S: —	×	×	×	iQ Sensor Solution Reference Manual

Added or enhanced function	F: Firmware version of the CPU module S: Software version of the engineering tool O: Version of the other tools					Reference
	R00/R01/R02CPU	Rn(EN)CPU	RnPCPU	RnPSFCPU and R6PSFM	RnSFCPU and R6SFM	
Online change (SFC block)	F: 05*11 S: 1.045X*11	F: 35*12 S: 1.045X*12	F: 20 S: 1.055H	×	F: 26*10 S: 1.080J	MELSEC iQ-R Programming Manual (Program Design)
Access level "Assistant Developers" of the user authentication function	×	×	×	F: 08 S: 1.080J	F: 13 S: 1.045X	GX Works3 Operating Manual
Omission of writing files that are not changed	F: 05*15 S: 1.045X*15	F: 35*16 S: 1.045X*16	F: 18 S: 1.050C	×	F: 13*17 S: 1.045X*17	GX Works3 Operating Manual
Device test with execution conditions	F: 06 S: 1.047Z	F: 38 S: 1.047Z	F: 20 S: 1.055H	×	F: 19*10 S: 1.060N	Page 185 Device Test with Execution Conditions
Setting to Wait Receiving Cyclic Data after Switching System*13	×	×	F: 18 S: 1.050C	F: 04 S: 1.060N	×	• Page 554 Setting to wait cyclic data receive after system switching • Page 754 Setting to wait cyclic data receive after system switching
SFC-activated step monitor	F: 08 S: 1.050C	F: 40 S: 1.050C	×	×	F: 26*10 S: 1.080J	GX Works3 Operating Manual
Event history logging restriction	F: 08 S: —	F: 40 S: —	F: 20 S: —	×	F: 16 S: —	Page 153 Event history logging restriction
Time specification of a timing type in the realtime monitor function	F: 08 S: — O: 1.88S*2	F: 40 S: — O: 1.88S*2	×	×	F: 34 S: — O: 1.160S*2	GX LogViewer Version 1 Operating Manual
RJ71GN11-T2	F: 11 S: 1.055H	F: 43 S: 1.055H	×	×	F: 20 S: 1.065T	• Page 374 Device List • Page 396 Link Direct Device • MELSEC iQ-R CC-Link IE TSN User's Manual (Application)
Motion module (RD78G4, RD78G8, RD78G16, RD78G32, RD78G64)	F: 12 S: 1.056J	F: 44 S: 1.056J	×	×	F: 21 S: 1.065T	• Page 396 Link Direct Device • MELSEC iQ-R Motion Module User's Manual (Application)
Motion module (RD78GHV, RD78GHW)	F: 14 S: 1.060N	F: 46 S: 1.060N	×	×	F: 21 S: 1.065T	• Page 396 Link Direct Device • MELSEC iQ-R Motion Module User's Manual (Application)
Extension of points for CC-Link IE Controller Network	F: 11 S: 1.055H	F: 43 S: 1.055H	F: 27 S: 1.075D	×	F: 21 S: 1.065T	• Page 374 Device List • Page 396 Link Direct Device • MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)
Online Change	F: ○ S: 1.040S	F: ○ S: 1.000A	■In process mode F: ○ S: 1.007H ■In redundant mode F: 04 S: 1.025B	F: ○*10 S: 1.040S*10	F: ○*14 S: 1.015R*14	• Page 125 Online Change • MELSEC iQ-R CPU Module User's Manual (Startup)
Data communications between CPU modules	F: ○ S: 1.040S	F: ○ S: 1.000A	F: ○*6 S: 1.007H*6	×	F: 17*10 S: 1.055H*10	• Page 326 MULTIPLE CPU SYSTEM FUNCTION • Page 335 Data Communication Between CPU Modules
Multiple CPU synchronous interrupt (I45)	F: ○ S: 1.040S	F: ○ S: 1.000A	F: ○*6 S: 1.007H*6	×	F: 17*10 S: 1.055H*10	• Page 326 MULTIPLE CPU SYSTEM FUNCTION • MELSEC iQ-R CPU Module User's Manual (Startup)
Inter-module synchronous interrupt (I44)	F: ○ S: 1.040S	F: ○ S: 1.000A	F: ○*6 S: 1.007H*6	×	F: 17*10 S: 1.055H*10	• MELSEC iQ-R CPU Module User's Manual (Startup) • MELSEC iQ-R Inter-Module Synchronization Function Reference Manual
Writing of program restoration information	×	×	F: 20 S: 1.055H	×	×	Page 1101 Program Restoration Information Write Selection

Added or enhanced function	F: Firmware version of the CPU module S: Software version of the engineering tool O: Version of the other tools					Reference
	R00/R01/ R02CPU	Rn(EN)CPU	RnPCPU	RnPSFCPU and R6PSFM	RnSFCPU and R6SFM	
FB hierarchy information	F: 14 S: 1.060N	F: 46 S: 1.060N	F: 24 S: 1.060N	×	×	GX Works3 Operating Manual
RJ71LP21-25	F: 16 S: 1.063R	F: 48 S: 1.063R	F: 23 S: 1.063R	×	F: 30 S: 1.095Z	• MELSEC iQ-R Module Configuration Manual • MELSEC iQ-R MELSECNET/H Network Module User's Manual (Application)
Extended SRAM cassette (NZ2MC-2MBSE)	×	×	F: — S: 1.007H	F: — S: 1.065T	F: — S: 1.015R	• Page 103 Device/label memory area setting • Page 632 Device/Label Memory Area Setting • Page 767 Device/Label Memory Area Setting • MELSEC iQ-R CPU Module User's Manual (Startup)
CC-Link IE TSN network synchronous communication	F: 18 S: 1.065T	F: 50 S: 1.065T	×	×	F: 30 S: 1.065T	MELSEC iQ-R CC-Link IE TSN User's Manual (Application)
Function memory capacity increased to 20480K bytes, file size unit (cluster size) increased to 8192 bytes.	×	F: 50*18 S: —	×	×	×	• MELSEC iQ-R CPU Module User's Manual (Startup) • Page 110 File size unit based on memory area
Saving of device/label data write operation histories of the event history function	×	F: 50*18 S: 1.065T	×	×	F: 23 S: 1.070Y	Page 154 Device/label data write operation saving
Simultaneous execution of up to 10 data loggings when the storage destination of data logging files is the function memory	×	F: 50*18 S: 1.065T	×	×	×	• Page 212 Storage location of data logging files • Page 1107 Setting items
MELSEC iQ-R series recorder module	×	F: 50*18 S: 1.065T	×	×	F: 23 S: 1.070Y	• MELSEC iQ-R Module Configuration Manual • MELSEC iQ-R System Recorder User's Manual (Startup) • MELSEC iQ-R System Recorder User's Manual (Application)
Recording function	×	F: 50*18 S: 1.065T	×	×	F: 23 S: 1.070Y	• MELSEC iQ-R System Recorder User's Manual (Startup) • MELSEC iQ-R System Recorder User's Manual (Application)
Data collection instruction (DATATRG)	×	F: 50*18 S: 1.065T	×	×	F: 23 S: 1.070Y	MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)
CSV file format for the storage format of data logging files	×	F: 53 S: — O: 1.112R*4	×	×	×	Page 212 Storage format of data logging files
"Condition specification" for the file switching timing for data logging setting	×	F: 53 S: — O: 1.112R*4	×	×	×	Page 220 File switching condition
Addition of device value to the data logging storage file name	×	F: 53 S: — O: 1.112R*4	×	×	×	Page 222 Storage file
Redundant system with redundant extension base unit	×	×	■For R68WRB F: 25*7 S: 1.070Y ■For R66WRB-HT F: 25*7 S: 1.072A	×	×	• Page 492 FUNCTIONS • MELSEC iQ-R Module Configuration Manual

A

Added or enhanced function	F: Firmware version of the CPU module S: Software version of the engineering tool O: Version of the other tools					Reference
	R00/R01/ R02CPU	Rn(EN)CPU	RnPCPU	RnPSFCPU and R6PSFM	RnSFCPU and R6SFM	
Laser displacement sensor control module supporting the redundant system with redundant extension base unit	×	×	F: 26[*7] S: —	×	×	MELSEC iQ-R Module Configuration Manual
MELSEC iQ-R series camera recorder module	×	F: 55[*18] S: 1.072A	×	×	F: 24 S: 1.072A	• MELSEC iQ-R Module Configuration Manual • MELSEC iQ-R System Recorder User's Manual (Startup) • MELSEC iQ-R System Recorder User's Manual (Application)
Program start/stop	F: 24 S: 1.075D	F: 57 S: 1.075D	F: 27 S: 1.075D	×	×	GX Works3 Operating Manual
Specified program monitor	F: 24 S: 1.075D	F: 57 S: 1.075D	×	×	F: 30 S: 1.095Z	• Page 176 Specified Program Monitor • GX Works3 Operating Manual
Phase processing instructions	F: 24 S: 1.075D	F: 57 S: 1.075D	×	×	×	MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)
Label memory read/write	F: 24 S: 1.075D	F: 57 S: 1.075D	F: 27 S: 1.075D	F: ○ S: 1.075D	F: ○ S: 1.075D	GX Works3 Operating Manual
Device/label access service processing constant wait function	×	×	F: 27 S: 1.075D	×	×	Page 50 Device/label access service processing constant wait function
Copying/cutting across data logging settings for a data setting item	F: — S: — O: 1.118X[*4]	F: — S: — O: 1.118X[*4]	F: — S: — O: 1.118X[*4]	F: — S: — O: 1.118X[*4]	F: — S: — O: 1.118X[*4]	—
System monitoring information (SD2000 to SD2036)	F: 24 S: —	F: 57 S: —	×	×	×	Page 1020 System monitoring information
CC-Link TSN motion module inter-module synchronization function	F: 24 S: 1.075D	F: 57 S: 1.075D	×	×	F: 30 S: 1.095Z	MELSEC iQ-R Motion Module User's Manual (Network)
Scan time clear	F: 26 S: 1.080J	F: 59 S: 1.080J	×	×	F: 26 S: 1.080J	• Page 942 Fixed scan function information • GX Works3 Operating Manual
RJ71GN11-EIP	F: 29 S: 1.082L	F: 62 S: 1.082L	×	×	F: 33[*19] S: 1.082L	• Page 396 Link Direct Device • MELSEC iQ-R CC-Link IE TSN Plus Master/Local Module User's Manual
Scan time measurement	F: 29 S: 1.082L	F: 62 S: 1.082L	×	×	F: 30 S: 1.095Z	• Page 171 Scan Time Measurement • GX Works3 Operating Manual
Write-protect function for device data (from outside the CPU module)	F: 30 S: 1.085P	F: 63 S: 1.085P	×	×	F: 32 S: 1.100E	Page 358 Write-Protect Function for Device Data (from Outside the CPU Module)
System memory supports the memory copy from control system to standby system	×	×	F: 33[*7] S: —	×	×	Page 530 Memory Copy from Control System to Standby System
User authentication function (enhanced vulnerability-measures taken)	×	×	×	F: 12 S: 1.105K	F: 27 S: 1.087R	• Page 653 User Authentication Function of CPU Modules • Page 763 User Authentication Function of CPU Modules • GX Works3 Operating Manual
RJ71GN11-SX	F: 11 S: 1.095Z	F: 43 S: 1.095Z	×	×	F: 20[*19] S: 1.095Z	• Page 374 Device List • Page 396 Link Direct Device • MELSEC iQ-R CC-Link IE TSN User's Manual (Application)

Added or enhanced function	F: Firmware version of the CPU module S: Software version of the engineering tool O: Version of the other tools					Reference
	R00/R01/ R02CPU	Rn(EN)CPU	RnPCPU	RnPSFCPU and R6PSFM	RnSFCPU and R6SFM	
RJ71BR11	F: 35 S: 1.095Z	F: 68 S: 1.095Z	×	×	F: 30 S: 1.095Z	• MELSEC iQ-R Module Configuration Manual • MELSEC iQ-R MELSECNET/H Network Module User's Manual (Application)
I/O module with safety functions (RX40NC6S-TS, RY48PT20S-TS)	×	×	×	×	F: 31 S: 1.100E	• Page 631 Safety I/O Refresh Function • MELSEC iQ-R Module Configuration Manual • MELSEC iQ-R I/O Module (With Safety Functions) User's Manual
Devices and labels specification together in a single setting for sampling target data and trigger condition in the data logging function	F: 38 S: 1.045X O: 1.154L*4	F: 71 S: 1.045X O: 1.154L*4	×	×	F: 34*10 S: 1.110Q*10 O: 1.160S*4	• Page 200 Data to Be Collected • Page 639 Data to be collected • Page 639 Trigger conditions
Time specification and interrupt occurrence for the data collecting condition in the data logging function	F: ○ S: — O: ○	F: ○ S: — O: ○	F: ○ S: —	×	F: 34*10 S: —*10 O: ○*4	• Page 200 Data to Be Collected • Page 639 Data collection conditions
Monitoring of up to 32 points of data in the real-time monitor function	×	F: 71 S: — O: 1.154L*2	×	×	F: 34 S: — O: 1.154L*2	GX LogViewer Version 1 Operating Manual

*1 The R00CPU is excluded.
*2 This indicates the compatible software version of GX LogViewer.
*3 Step relays (BLn\Sn) can be specified with CPU Module Logging Configuration Tool or GX LogViewer.
*4 This indicates the compatible software version of CPU Module Logging Configuration Tool.
*5 Data is backed up or restored by using programs (special relay and special register).
*6 This function is not supported in redundant mode.
*7 This function is not supported in process mode.
*8 Support of the program restoration information write selection of writing to the programmable controller while the Process CPU is stopped and of the file batch online change is as follows: F: 13, S: 1.040S.
*9 The firmware version of the Simple Motion Module is "Ver.05" or later.
*10 Only the standard program is supported.
*11 Online change (inactive SFC block) is supported in F: ○, S: 1.040S.
*12 Online change (inactive SFC block) is supported in F: 28, S: 1.035M.
*13 The firmware version of the usable RJ71GF11-T2 is "35" or later.
*14 The safety program is supported in F: 17, S: 1.055H. It is available only when the safety operation mode is TEST MODE. (☞ Page 621 FUNCTIONS, ☞ Page 627 Operations restricted in SAFETY MODE)
*15 When writing to the SD memory card, it is supported in F: 11, S: 1.055H.
*16 When writing to the SD memory card, it is supported in F: 43, S: 1.055H.
*17 When writing to the SD memory card, it is supported in F: 20, S: 1.065T.
*18 Supported when the 3rd and 4th digits from the left of CPU module production information are as follows.

CPU module	3rd and 4th digits from left of production information
R04CPU	"19" or later
R08CPU	"20" or later
R16CPU	"20" or later
R32CPU	"17" or later
R120CPU	"17" or later
R04ENCPU	"32" or later
R08ENCPU	"30" or later
R16ENCPU	"27" or later
R32ENCPU	"30" or later
R120ENCPU	"22" or later

*19 Only the standard communication is supported.

A

INDEX

Symbols

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

I

R

S

T

U

W

I

REVISIONS

*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
June 2014	SH(NA)-081264ENG-A	First edition
July 2014	SH(NA)-081264ENG-B	Error correction
October 2014	SH(NA)-081264ENG-C	■Added functions Memory dump function, real-time monitor function, LED specifications (enhanced) ■Added or modified parts SAFETY PRECAUTIONS, RELEVANT MANUALS, Section 1.4, 1.5, 4.4, 6.2, 8.1, 12.1, 12.2, 12.3, 12.4, Chapter 14, Section 17.1, 17.2, 20.2, Chapter 22, Section 22.1, 22.2, 22.4, 22.5, 22.12, 22.13, 23.1, 23.2, 23.3, 23.4, 23.5, 23.6, 23.7, 23.8, Chapter 24, Section 25.1, 25.3, 26.4, 30.2, Appendix 1, 2, 4, 5, 6, 10
November 2014	SH(NA)-081264ENG-D	■Added or modified parts Section 1.4, 23.2, 28.2, Appendix 6
January 2015	SH(NA)-081264ENG-E	■Added models R08PCPU, R16PCPU, R32PCPU, R120PCPU ■Added function Process control function ■Added or modified parts SAFETY PRECAUTIONS, INTRODUCTION, TERMS, Section 1.4, 4.2, 4.4, Chapter 9, 10, Section 14.1, 14.2, 14.4, Chapter 15, 18, Section 23.12, Chapter 24, 25, Section 27.1, 27.2, 29.1, 29.2, 29.11, 30.3, Appendix 1, 2, 4, 5, 6, 7, 10
May 2015	SH(NA)-081264ENG-F	■Added or modified parts TERMS, Section 6.3, 19.2, Appendix 1, 5, 6, 9, 10
August 2015	SH(NA)-081264ENG-G	■Added models R04ENCPU, R08ENCPU, R08SFCPU, R16ENCPU, R16SFCPU, R32ENCPU, R32SFCPU, R120ENCPU, R120SFCPU, R6SFM ■Added functions User authentication function, safety function ■Added or modified parts SAFETY PRECAUTIONS, CONDITIONS OF USE FOR THE PRODUCT, INTRODUCTION, TERMS, Section 14.4, Chapter 28, Section 29.1, 29.2, PART 4, Chapter32, 33, 34, 35, Appendix 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, WARRANTY
August 2015	SH(NA)-081264ENG-H	■Added or modified parts TERMS, Section 1.5, 1.6, 4.3, 7.1, 14.2, 14.3, 14.4, 21.2, 24.1, 26.2, 27.4, Chapter 28, Section 29.1, 29.2, 31.3, 32.2, 32.4, 33.4, Chapter 34, Section 34.2, 34.6, 34.9, 35.1, Appendix 1, 4, 5, 8, 10, 13, WARRANTY
January 2016	SH(NA)-081264ENG-I	■Added functions Daylight saving time function, sequence scan synchronization sampling function ■Added or modified parts Section 6.3, 14.4, 23.1, 23.2, Chapter 29, Section 30.1, 32.3, 35.6, 36.1, Appendix 1, 2, 4, 9, 13
May 2016	SH(NA)-081264ENG-J	■Added model R6RFM ■Added functions CPU module data backup/restoration function, iQ Sensor Solution data backup/restoration function, redundant function, and Label initialization function ■Added or modified parts SAFETY PRECAUTIONS, INTRODUCTION, TERMS, Section 1.1, 2.2, 4.4, 16.3, 20.2, 20.3, 26.2, Chapter 27, Section 27.1, 27.2, Chapter 31, Section 31.1, Chapter 31.2, Chapter 32, PART 4, Chapter 36, 37, 38, 39, 40, Appendix 1, 2, 3, 4, 5, 9, 10, 12, 13
June 2016	SH(NA)-081264ENG-K	■Added or modified parts SAFETY PRECAUTIONS, TERMS, Section 21.1, 21.2, 26.1, Chapter 27, 31, PART 4, Appendix 1, 2, 4, 5, 9, 13
October 2016	SH(NA)-081264ENG-L	■Added functions External input/output forced on/off function, CC-Link IE Field Network Basic function, firmware update function ■Added or modified parts SAFETY PRECAUTIONS, Section 1.1, Chapter 3, Section 4.4, 6.4, 7.1, 23.5, 23.14, 24.6, 26.1, 26.2, 26.3, 27.1, 27.2, Chapter 28, Section 29.2, 29.4, 37.1, Chapter 38, Section 38.2, 38.3, 38.5, Appendix 1, 2, 3, 4, 5, 8, 9, 10, 12, 13 ■Deleted parts Chapter 32

Revision date	*Manual number	Description
January 2017	SH(NA)-081264ENG-M	■Added or modified parts TERMS, Section 10.1, 10.2, Chapter 31, Section 38.3, Appendix 1, 13
May 2017	SH(NA)-081264ENG-N	■Added functions CPU module database access function, online change (inactive SFC block only) ■Added or modified parts Section 4.4, Chapter 7, Section 7.1, Chapter 12, Section 12.1, 12.3, Chapter 27, Section 27.2, Appendix 1, 4, 5, 8, 9, 12, 13, 14
November 2017	SH(NA)-081264ENG-O	■Added models R00CPU, R01CPU, R02CPU, R08PSFCPU, R16PSFCPU, R32PSFCPU, R120PSFCPU, R6PSFM ■Added functions Data logging file transfer function ■Added or modified parts SAFETY PRECAUTIONS, CONDITIONS OF USE FOR THE PRODUCT, INTRODUCTION, TERMS, Chapter 3, 5, 11, 22, PART 6, Chapter 35, 36, 37, 38, 39, 40, Appendix 12, 13, WARRANTY
April 2018	SH(NA)-081264ENG-P	■Added functions Data logging function (for local devices/labels), error detection invalidation setting, battery-less option cassette, online change (SFC block) ■Added or modified parts Section 1.3, 3.1, 3.3, 6.2, Chapter 9, Section 9.1, Chapter 11, Section 11.1, 11.2, 11.3, 11.16, 21.19, Chapter 23, Section 28.14, Chapter 33, Section 33.1, 34.2, Appendix 1, 2, 4, 5, 8, 9, 10, 12, 13, 15
July 2018	SH(NA)-081264ENG-Q	■Added functions Device test with execution condition ■Added or modified parts Section 6.2, 10.2, 11.8, 37.2, Appendix 1, 2, 5, 12, 15
October 2018	SH(NA)-081264ENG-R	■Added functions Setting to wait cyclic data receive after system switching ■Added or modified parts Section 6.4, 22.7, 28.3, 28.7, Appendix 1, 2, 4, 5, 9, 15
April 2019	SH(NA)-081264ENG-S	■Added or modified parts TERMS, GENERIC TERMS AND ABBREVIATIONS, Section 3.1, 3.4, 5.2, 7.4, 10.1, 10.2, Chapter 15, Section 15.1, 15.2, 16.5, 19.2, 21.1, 21.5, 28.15, 28.17, Chapter 33, Section 33.1, 37.15, Appendix 1, 4, 5, 9, 10, 12, 13, 15
May 2019	SH(NA)-081264ENG-T	■Added or modified part SAFETY PRECAUTIONS
June 2019	SH(NA)-081264ENG-U	■Added or modified parts TERMS, GENERIC TERMS AND ABBREVIATIONS, Section 21.5
August 2019	SH(NA)-081264ENG-V	■Added or modified parts Section 3.4, Chapter 15, Section 15.2, Chapter 18, Section 28.3, 32.4, Chapter 33, Section 33.8, Appendix 1, 4, 5, 15
October 2019	SH(NA)-081264ENG-W	■Added or modified parts RELEVANT MANUALS, Section 4.4, 6.4, 9.1, 10.2, Chapter 16, 21, Section 27.7, Chapter 34, Section 34.7, 38.6, Appendix 1, 2, 3, 4, 5, 13, 15
February 2020	SH(NA)-081264ENG-Y	■Added or modified parts RELEVANT MANUALS, Chapter 21, Section 29.2, 29.14, Appendix 1, 9, 15
May 2020	SH(NA)-081264ENG-Z	■Added or modified parts Section 3.1, 3.4, 6.2, 6.4, 12.1, Chapter 15, Section 32.3, 33.4, Chapter 34, Section 34.5, 38.15, Appendix 1, 2, 4, 5, 6, 7, 9, 10, 13, 15
June 2020	SH(NA)-081264ENG-AA	■Added or modified part Appendix 15
July 2020	SH(NA)-081264ENG-AB	■Added or modified parts SAFETY PRECAUTIONS, CONDITIONS OF USE FOR THE PRODUCT
October 2020	SH(NA)-081264ENG-AC	■Added or modified parts SAFETY PRECAUTIONS, TERMS, GENERIC TERMS AND ABBREVIATIONS, Section 2.3, 6.2, 6.3, 6.4, 27.6, 27.7, 27.8, 27.10, 28.1, 28.5, Chapter 29, Section 29.1, 29.2, 29.3, 29.5, 29.6, 29.7, 29.9, 29.10, 29.11, 29.13, 29.14, 29.15, 29.16, 29.17, 29.18, 29.20, 31.1, Appendix 1, 2, 3, 4, 5, 9, 10, 13, 15
December 2020	SH(NA)-081264ENG-AD	■Added or modified part Chapter 21
January 2021	SH(NA)-081264ENG-AE	■Added or modified parts Appendix 1, 16

Revision date	*Manual number	Description
April 2021	SH(NA)-081264ENG-AF	■Added functions Specified program monitor, device/label access service processing constant wait function ■Added or modified parts Section 1.1, 1.2, 8.4, Chapter 9, Section 9.2, Chapter 18, 21, Section 22.3, Chapter 29, Section 29.2, 29.3, 29.5, 29.19, Chapter 31, Section 31.1, Chapter 34, Appendix 1, 2, 4, 5, 11, 13, 16
June 2021	SH(NA)-081264ENG-AG	■Added or modified parts Section 23.7, 29.4, Appendix 1
October 2021	SH(NA)-081264ENG-AH	■Added function Scan time clear ■Added or modified parts Section 1.7, 8.5, Chapter 9, Section 9.2, 11.6, Chapter 17, Section 27.5, 27.8, 29.3, 36.8, 37.3, Appendix 1, 2, 4, 5, 9, 16
January 2022	SH(NA)-081264ENG-AI	■Added function Scan time measurement ■Added or modified parts GENERIC TERMS AND ABBREVIATIONS, Section 6.2, Chapter 9, Section 9.2, Chapter 21, Appendix 2, 6, 7, 9, 16
April 2022	SH(NA)-081264ENG-AJ	■Added function Write-protect function for device data (from outside the CPU module) ■Added or modified parts SAFETY PRECAUTIONS, Section 3.4, Chapter 17, Section 17.1, 22.3, 29.4, Appendix 1, 10, 16
July 2022	SH(NA)-081264ENG-AK	■Added or modified parts INTRODUCTION, RELEVANT MANUALS
August 2022	SH(NA)-081264ENG-AL	■Added or modified parts INTRODUCTION, Chapter 17, 34, Appendix 1, 16
November 2022	SH(NA)-081264ENG-AM	■Added or modified parts CONDITIONS OF USE FOR THE PRODUCT, Chapter 36, Appendix 9, WARRANTY
April 2023	SH(NA)-081264ENG-AN	■Added or modified parts Section 9.2, 9.3, Chapter 34, Section 34.9, Appendix 1, 9, 16
January 2024	SH(NA)-081264ENG-AO	■Added or modified parts Section 15.2, 17.1, 25.1, 25.3, 32.2, 32.3, 32.4, Chapter 34, Section 34.6, 34.11, Appendix 1, 4, 5, 7, 9, 10, 16
April 2024	SH(NA)-081264ENG-AP	■Added or modified parts Section 15.2, Chapter 17, Section 17.1, 25.1, 25.3, 32.2, 32.3, 32.4, Chapter 34, Section 34.6, 34.11, Chapter 38, Section 38.12, Appendix 1, 4, 5, 7, 9, 10, 16
October 2024	SH(NA)-081264ENG-AQ	■Added or modified parts Section 3.4, Chapter 4, Section 4.1, 4.2, 4.3, 4.4, Chapter 11, Section 11.1, 11.16, Chapter 15, Chapter 18, Section 33.4, Chapter 34, Section 34.9, 34.10, Appendix 1, 4, 5, 9, 13, 16

Japanese manual number: SH-081224-AT

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

(1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.

(2) Even within the gratis warranty term, repairs shall be charged for in the following cases.

1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
2. Failure caused by unapproved modifications, etc., to the product by the user.
3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

(1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.

(2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to:

(1) Damages caused by any cause found not to be the responsibility of Mitsubishi.

(2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.

(3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.

(4) Replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

· For SIL2 Process CPUs

Please confirm the following product warranty details before using this product.

1. Limited Warranty and Product Support.

a. Mitsubishi Electric Company ("MELCO") warrants that for a period of eighteen (18) months after date of delivery from the point of manufacture or one year from date of Customer's purchase, whichever is less, Mitsubishi programmable logic controllers (the "Products") will be free from defects in material and workmanship.

b. At MELCO's option, for those Products MELCO determines are not as warranted, MELCO shall either repair or replace them or issue a credit or return the purchase price paid for them.

c. For this warranty to apply:

(1) Customer shall give MELCO (i) notice of a warranty claim to MELCO and the authorized dealer or distributor from whom the Products were purchased, (ii) the notice shall describe in reasonable details the warranty problem, (iii) the notice shall be provided promptly and in no event later than thirty (30) days after the Customer knows or has reason to believe that Products are not as warranted, and (iv) in any event, the notice must given within the warranty period;

(2) Customer shall cooperate with MELCO and MELCO's representatives in MELCO's investigation of the warranty claim, including preserving evidence of the claim and its causes, meaningfully responding to MELCO's questions and investigation of the problem, grant MELCO access to witnesses, personnel, documents, physical evidence and records concerning the warranty problem, and allow MELCO to examine and test the Products in question offsite or at the premises where they are installed or used; and

(3) If MELCO requests, Customer shall remove Products it claims are defective and ship them to MELCO or MELCO's authorized representative for examination and, if found defective, for repair or replacement. The costs of removal, shipment to and from MELCO's designated examination point, and reinstallation of repaired or replaced Products shall be at Customer's expense.

(4) If Customer requests and MELCO agrees to effect repairs onsite at any domestic or overseas location, the Customer will pay for the costs of sending repair personnel and shipping parts. MELCO is not responsible for any re-commissioning, maintenance, or testing on-site that involves repairs or replacing of the Products.

d. Repairs of Products located outside of Japan are accepted by MELCO's local authorized service facility centers ("FA Centers"). Terms and conditions on which each FA Center offers repair services for Products that are out of warranty or not covered by MELCO's limited warranty may vary.

e. Subject to availability of spare parts, MELCO will offer Product repair services for (7) years after each Product model or line is discontinued, at MELCO's or its FA Centers' rates and charges and standard terms in effect at the time of repair. MELCO usually produces and retains sufficient spare parts for repairs of its Products for a period of seven (7) years after production is discontinued.

f. MELCO generally announces discontinuation of Products through MELCO's Technical Bulletins. Products discontinued and repair parts for them may not be available after their production is discontinued.

2. Limits of Warranties.

a. MELCO does not warrant or guarantee the design, specify, manufacture, construction or installation of the materials, construction criteria, functionality, use, properties or other characteristics of the equipment, systems, or production lines into which the Products may be incorporated, including any safety, fail-safe and shut down systems using the Products.

b. MELCO is not responsible for determining the suitability of the Products for their intended purpose and use, including determining if the Products provide appropriate safety margins and redundancies for the applications, equipment or systems into which they are incorporated.

c. Customer acknowledges that qualified and experienced personnel are required to determine the suitability, application, design, construction and proper installation and integration of the Products. MELCO does not supply such personnel.

d. MELCO is not responsible for designing and conducting tests to determine that the Product functions appropriately and meets application standards and requirements as installed or incorporated into the end-user's equipment, production lines or systems.

e. MELCO does not warrant any Product:

(1) repaired or altered by persons other than MELCO or its authorized engineers or FA Centers;

(2) subjected to negligence, carelessness, accident, misuse, or damage;

(3) improperly stored, handled, installed or maintained;

(4) integrated or used in connection with improperly designed, incompatible or defective hardware or software;

(5) that fails because consumable parts such as batteries, backlights, or fuses were not tested, serviced or replaced;

(6) operated or used with equipment, production lines or systems that do not meet applicable and commensurate legal, safety and industry-accepted standards;

(7) operated or used in abnormal applications;

(8) installed, operated or used in contravention of instructions, precautions or warnings contained in MELCO's user, instruction and/or safety manuals, technical bulletins and guidelines for the Products;

(9) used with obsolete technologies or technologies not fully tested and widely accepted and in use at the time of the Product's manufacture;

(10) subjected to excessive heat or moisture, abnormal voltages, shock, excessive vibration, physical damage or other improper environment; or

(11) damaged or malfunctioning due to Acts of God, fires, acts of vandals, criminals or terrorists, communication or power failures, or any other cause or failure that results from circumstances beyond MELCO's control.

f. All Product information and specifications contained on MELCO's website and in catalogs, manuals, or technical information materials provided by MELCO are subject to change without prior notice.

g. The Product information and statements contained on MELCO's website and in catalogs, manuals, technical bulletins or other materials provided by MELCO are provided as a guide for Customer's use. They do not constitute warranties and are not incorporated in the contract of sale for the Products.
h. These terms and conditions constitute the entire agreement between Customer and MELCO with respect to warranties, remedies and damages and supersede any other understandings, whether written or oral, between the parties. Customer expressly acknowledges that any representations or statements made by MELCO or others concerning the Products outside these terms are not part of the basis of the bargain between the parties and are not factored into the pricing of the Products.
i. THE WARRANTIES AND REMEDIES SET FORTH IN THESE TERMS ARE THE EXCLUSIVE AND ONLY WARRANTIES AND REMEDIES THAT APPLY TO THE PRODUCTS.
j. MELCO DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

3. Limits on Damages.

a. MELCO'S MAXIMUM CUMULATIVE LIABILITY BASED ON ANY CLAIMS FOR BREACH OF WARRANTY OR CONTRACT, NEGLIGENCE, STRICT TORT LIABILITY OR OTHER THEORIES OF RECOVERY REGARDING THE SALE, REPAIR, REPLACEMENT, DELIVERY, PERFORMANCE, CONDITION, SUITABILITY, COMPLIANCE, OR OTHER ASPECTS OF THE PRODUCTS OR THEIR SALE, INSTALLATION OR USE SHALL BE LIMITED TO THE PRICE PAID FOR PRODUCTS NOT AS WARRANTED.
b. Although MELCO has declared Product's compliance with the international safety standards IEC61508, IEC61511, this fact does not guarantee that Product will be free from any malfunction or failure. The user of this Product shall comply with any and all applicable safety standard, regulation or law and take appropriate safety measures for the system in which the Product is installed or used and shall take the second or third safety measures other than the Product. MELCO is not liable for damages that could have been prevented by compliance with any applicable safety standard, regulation or law.
c. MELCO prohibits the use of Products with or in any application involving power plants, trains, railway systems, airplanes, airline operations, other transportation systems, amusement equipments, hospitals, medical care, dialysis and life support facilities or equipment, incineration and fuel devices, handling of nuclear or hazardous materials or chemicals, mining and drilling, and other applications where the level of risk to human life, health or property are elevated.
d. MELCO SHALL NOT BE LIABLE FOR SPECIAL, INCIDENTAL, CONSEQUENTIAL, INDIRECT OR PUNITIVE DAMAGES, FOR LOSS OF PROFITS, SALES, OR REVENUE, FOR INCREASED LABOR OR OVERHEAD COSTS, FOR DOWNTIME OR LOSS OF PRODUCTION, FOR COST OVERRUNS, OR FOR ENVIRONMENTAL OR POLLUTION DAMAGES OR CLEAN-UP COSTS, WHETHER THE LOSS IS BASED ON CLAIMS FOR BREACH OF CONTRACT OR WARRANTY, VIOLATION OF STATUTE, NEGLIGENCE OR OTHER TORT, STRICT LIABILITY OR OTHERWISE.
e. In the event that any damages which are asserted against MELCO arising out of or relating to the Products or defects in them, consist of personal injury, wrongful death and/or physical property damages as well as damages of a pecuniary nature, the disclaimers and limitations contained in these terms shall apply to all three types of damages to the fullest extent permitted by law. If, however, the personal injury, wrongful death and/or physical property damages cannot be disclaimed or limited by law or public policy to the extent provided by these terms, then in any such event the disclaimer of and limitations on pecuniary or economic consequential and incidental damages shall nevertheless be enforceable to the fullest extent allowed by law.
f. In no event shall any cause of action arising out of breach of warranty or otherwise concerning the Products be brought by Customer more than one year after the cause of action accrues.
g. Each of the limitations on remedies and damages set forth in these terms is separate and independently enforceable, notwithstanding the unenforceability or failure of essential purpose of any warranty, undertaking, damage limitation, other provision of these terms or other terms comprising the contract of sale between Customer and MELCO.

4. Delivery/Force Majeure.

a. Any delivery date for the Products acknowledged by MELCO is an estimated and not a promised date. MELCO will make all reasonable efforts to meet the delivery schedule set forth in Customer's order or the purchase contract but shall not be liable for failure to do so.
b. Products stored at the request of Customer or because Customer refuses or delays shipment shall be at the risk and expense of Customer.
c. MELCO shall not be liable for any damage to or loss of the Products or any delay in or failure to deliver, service, repair or replace the Products arising from shortage of raw materials, failure of suppliers to make timely delivery, labor difficulties of any kind, earthquake, fire, windstorm, flood, theft, criminal or terrorist acts, war, embargoes, governmental acts or rulings, loss or damage or delays in carriage, acts of God, vandals or any other circumstances reasonably beyond MELCO's control.

5. Choice of Law/Jurisdiction.

These terms and any agreement or contract between Customer and MELCO shall be governed by the laws of the State of New York without regard to conflicts of laws. To the extent any action or dispute is not arbitrated, the parties consent to the exclusive jurisdiction and venue of the federal and state courts located in the Southern District of the State of New York. Any judgment there obtained may be enforced in any court of competent jurisdiction.

6. Arbitration.

Any controversy or claim arising out of, or relating to or in connection with the Products, their sale or use or these terms, shall be settled by arbitration conducted in accordance with the Center for Public Resources (CPR) Rules for Non-Administered Arbitration of International Disputes, by a sole arbitrator chosen from the CPR's panels of distinguished neutrals. Judgment upon the award rendered by the Arbitrator shall be final and binding and may be entered by any court having jurisdiction thereof. The place of the arbitration shall be New York City, New York. The language of the arbitration shall be English. The neutral organization designated to perform the functions specified in Rule 6 and Rules 7.7(b), 7.8 and 7.9 shall be the CPR.

· For Safety CPUs

Please confirm the following product warranty details before using this product.

1. Limited Warranty and Product Support.

a. Mitsubishi Electric Company ("MELCO") warrants that for a period of eighteen (18) months after date of delivery from the point of manufacture or one year from date of Customer's purchase, whichever is less, Mitsubishi MELSEC Safety programmable logic controllers (the "Products") will be free from defects in material and workmanship.

b. At MELCO's option, for those Products MELCO determines are not as warranted, MELCO shall either repair or replace them or issue a credit or return the purchase price paid for them.

c. For this warranty to apply:

(1) Customer shall give MELCO (i) notice of a warranty claim to MELCO and the authorized dealer or distributor from whom the Products were purchased, (ii) the notice shall describe in reasonable details the warranty problem, (iii) the notice shall be provided promptly and in no event later than thirty (30) days after the Customer knows or has reason to believe that Products are not as warranted, and (iv) in any event, the notice must given within the warranty period;

(2) Customer shall cooperate with MELCO and MELCO's representatives in MELCO's investigation of the warranty claim, including preserving evidence of the claim and its causes, meaningfully responding to MELCO's questions and investigation of the problem, grant MELCO access to witnesses, personnel, documents, physical evidence and records concerning the warranty problem, and allow MELCO to examine and test the Products in question offsite or at the premises where they are installed or used; and

(3) If MELCO requests, Customer shall remove Products it claims are defective and ship them to MELCO or MELCO's authorized representative for examination and, if found defective, for repair or replacement. The costs of removal, shipment to and from MELCO's designated examination point, and reinstallation of repaired or replaced Products shall be at Customer's expense.

(4) If Customer requests and MELCO agrees to effect repairs onsite at any domestic or overseas location, the Customer will pay for the costs of sending repair personnel and shipping parts. MELCO is not responsible for any re-commissioning, maintenance, or testing on-site that involves repairs or replacing of the Products.

d. Repairs of Products located outside of Japan are accepted by MELCO's local authorized service facility centers ("FA Centers"). Terms and conditions on which each FA Center offers repair services for Products that are out of warranty or not covered by MELCO's limited warranty may vary.

e. Subject to availability of spare parts, MELCO will offer Product repair services for (7) years after each Product model or line is discontinued, at MELCO's or its FA Centers' rates and charges and standard terms in effect at the time of repair. MELCO usually produces and retains sufficient spare parts for repairs of its Products for a period of seven (7) years after production is discontinued.

f. MELCO generally announces discontinuation of Products through MELCO's Technical Bulletins. Products discontinued and repair parts for them may not be available after their production is discontinued.

2. Limits of Warranties.

a. MELCO does not warrant or guarantee the design, specify, manufacture, construction or installation of the materials, construction criteria, functionality, use, properties or other characteristics of the equipment, systems, or production lines into which the Products may be incorporated, including any safety, fail-safe and shut down systems using the Products.

b. MELCO is not responsible for determining the suitability of the Products for their intended purpose and use, including determining if the Products provide appropriate safety margins and redundancies for the applications, equipment or systems into which they are incorporated.

c. Customer acknowledges that qualified and experienced personnel are required to determine the suitability, application, design, construction and proper installation and integration of the Products. MELCO does not supply such personnel.

d. MELCO is not responsible for designing and conducting tests to determine that the Product functions appropriately and meets application standards and requirements as installed or incorporated into the end-user's equipment, production lines or systems.

e. MELCO does not warrant any Product:

(1) repaired or altered by persons other than MELCO or its authorized engineers or FA Centers;

(2) subjected to negligence, carelessness, accident, misuse, or damage;

(3) improperly stored, handled, installed or maintained;

(4) integrated or used in connection with improperly designed, incompatible or defective hardware or software;

(5) that fails because consumable parts such as batteries, backlights, or fuses were not tested, serviced or replaced;

(6) operated or used with equipment, production lines or systems that do not meet applicable and commensurate legal, safety and industry-accepted standards;

(7) operated or used in abnormal applications;

(8) installed, operated or used in contravention of instructions, precautions or warnings contained in MELCO's user, instruction and/or safety manuals, technical bulletins and guidelines for the Products;

(9) used with obsolete technologies or technologies not fully tested and widely accepted and in use at the time of the Product's manufacture;

(10) subjected to excessive heat or moisture, abnormal voltages, shock, excessive vibration, physical damage or other improper environment; or

(11) damaged or malfunctioning due to Acts of God, fires, acts of vandals, criminals or terrorists, communication or power failures, or any other cause or failure that results from circumstances beyond MELCO's control.

f. All Product information and specifications contained on MELCO's website and in catalogs, manuals, or technical information materials provided by MELCO are subject to change without prior notice.

g. The Product information and statements contained on MELCO's website and in catalogs, manuals, technical bulletins or other materials provided by MELCO are provided as a guide for Customer's use. They do not constitute warranties and are not incorporated in the contract of sale for the Products.
h. These terms and conditions constitute the entire agreement between Customer and MELCO with respect to warranties, remedies and damages and supersede any other understandings, whether written or oral, between the parties. Customer expressly acknowledges that any representations or statements made by MELCO or others concerning the Products outside these terms are not part of the basis of the bargain between the parties and are not factored into the pricing of the Products.
i. THE WARRANTIES AND REMEDIES SET FORTH IN THESE TERMS ARE THE EXCLUSIVE AND ONLY WARRANTIES AND REMEDIES THAT APPLY TO THE PRODUCTS.
j. MELCO DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

3. Limits on Damages.

a. MELCO'S MAXIMUM CUMULATIVE LIABILITY BASED ON ANY CLAIMS FOR BREACH OF WARRANTY OR CONTRACT, NEGLIGENCE, STRICT TORT LIABILITY OR OTHER THEORIES OF RECOVERY REGARDING THE SALE, REPAIR, REPLACEMENT, DELIVERY, PERFORMANCE, CONDITION, SUITABILITY, COMPLIANCE, OR OTHER ASPECTS OF THE PRODUCTS OR THEIR SALE, INSTALLATION OR USE SHALL BE LIMITED TO THE PRICE PAID FOR PRODUCTS NOT AS WARRANTED.
b. Although MELCO has obtained the certification for Product's compliance to the international safety standards IEC61508 and ISO13849-1 from TUV Rheinland, this fact does not guarantee that Product will be free from any malfunction or failure. The user of this Product shall comply with any and all applicable safety standard, regulation or law and take appropriate safety measures for the system in which the Product is installed or used and shall take the second or third safety measures other than the Product. MELCO is not liable for damages that could have been prevented by compliance with any applicable safety standard, regulation or law.
c. MELCO prohibits the use of Products with or in any application involving power plants, trains, railway systems, airplanes, airline operations, other transportation systems, amusement equipments, hospitals, medical care, dialysis and life support facilities or equipment, incineration and fuel devices, handling of nuclear or hazardous materials or chemicals, mining and drilling, and other applications where the level of risk to human life, health or property are elevated.
d. MELCO SHALL NOT BE LIABLE FOR SPECIAL, INCIDENTAL, CONSEQUENTIAL, INDIRECT OR PUNITIVE DAMAGES, FOR LOSS OF PROFITS, SALES, OR REVENUE, FOR INCREASED LABOR OR OVERHEAD COSTS, FOR DOWNTIME OR LOSS OF PRODUCTION, FOR COST OVERRUNS, OR FOR ENVIRONMENTAL OR POLLUTION DAMAGES OR CLEAN-UP COSTS, WHETHER THE LOSS IS BASED ON CLAIMS FOR BREACH OF CONTRACT OR WARRANTY, VIOLATION OF STATUTE, NEGLIGENCE OR OTHER TORT, STRICT LIABILITY OR OTHERWISE.
e. In the event that any damages which are asserted against MELCO arising out of or relating to the Products or defects in them, consist of personal injury, wrongful death and/or physical property damages as well as damages of a pecuniary nature, the disclaimers and limitations contained in these terms shall apply to all three types of damages to the fullest extent permitted by law. If, however, the personal injury, wrongful death and/or physical property damages cannot be disclaimed or limited by law or public policy to the extent provided by these terms, then in any such event the disclaimer of and limitations on pecuniary or economic consequential and incidental damages shall nevertheless be enforceable to the fullest extent allowed by law.
f. In no event shall any cause of action arising out of breach of warranty or otherwise concerning the Products be brought by Customer more than one year after the cause of action accrues.
g. Each of the limitations on remedies and damages set forth in these terms is separate and independently enforceable, notwithstanding the unenforceability or failure of essential purpose of any warranty, undertaking, damage limitation, other provision of these terms or other terms comprising the contract of sale between Customer and MELCO.

4. Delivery/Force Majeure.

a. Any delivery date for the Products acknowledged by MELCO is an estimated and not a promised date. MELCO will make all reasonable efforts to meet the delivery schedule set forth in Customer's order or the purchase contract but shall not be liable for failure to do so.
b. Products stored at the request of Customer or because Customer refuses or delays shipment shall be at the risk and expense of Customer.
c. MELCO shall not be liable for any damage to or loss of the Products or any delay in or failure to deliver, service, repair or replace the Products arising from shortage of raw materials, failure of suppliers to make timely delivery, labor difficulties of any kind, earthquake, fire, windstorm, flood, theft, criminal or terrorist acts, war, embargoes, governmental acts or rulings, loss or damage or delays in carriage, acts of God, vandals or any other circumstances reasonably beyond MELCO's control.

5. Choice of Law/Jurisdiction.

These terms and any agreement or contract between Customer and MELCO shall be governed by the laws of the State of New York without regard to conflicts of laws. To the extent any action or dispute is not arbitrated, the parties consent to the exclusive jurisdiction and venue of the federal and state courts located in the Southern District of the State of New York. Any judgment there obtained may be enforced in any court of competent jurisdiction.

6. Arbitration.

Any controversy or claim arising out of, or relating to or in connection with the Products, their sale or use or these terms, shall be settled by arbitration conducted in accordance with the Center for Public Resources (CPR) Rules for Non-Administered Arbitration of International Disputes, by a sole arbitrator chosen from the CPR's panels of distinguished neutrals. Judgment upon the award rendered by the Arbitrator shall be final and binding and may be entered by any court having jurisdiction thereof. The place of the arbitration shall be New York City, New York. The language of the arbitration shall be English. The neutral organization designated to perform the functions specified in Rule 6 and Rules 7.7(b), 7.8 and 7.9 shall be the CPR.

TRADEMARKS

Microsoft, Microsoft Access, Excel, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, Windows NT, Windows Server, Windows Vista, and Windows XP are trademarks of the Microsoft group of companies.
The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.
In some cases, trademark symbols such as '™' or '®' are not specified in this manual.

SH(NA)-081264ENG-AQ(2410)MEE
MODEL: RCPU-U-OU-E

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA 461-8670, JAPAN