MITSUBISHI ELECTRIC

Programmable Controller

MELSEC iQ-R series

MELSEC iQ-R Process CPU Module User's Manual

-R08PCPU
-R16PCPU
-R32PCPU
-R120PCPU
-R6RFM

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.
- 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.
- 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 redundant function modules]

- The optical transmitter and receiver of the redundant function module use laser diodes (class 1 in accordance with IEC 60825-1: 2014/JIS C6802: 2014). Do not look directly at a laser beam. Doing so may harm your eyes. Also, do not handle the product in ways other than those described in this manual, or do not disassemble or modify the product. Doing so may result in exposure to laser radiation.

[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.

[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 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, or connector. Doing so can cause malfunction or failure of the module.

[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 before powering on the system for operation. Also, attach an extension connector protective cover[*1] to each unused extension cable connector as necessary. Directly touching any conductive parts of the connectors while power is on may result in electric shock.
  *1 For details, please consult your local Mitsubishi Electric representative.

[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 Page 78 Redundant Function Module. 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) more than 25cm away in all directions from the programmable controller. Failure to do so may cause malfunction.
- 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 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. 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 do so may cause the module to fail or malfunction.

[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.

INTRODUCTION

Thank you for purchasing the Mitsubishi Electric MELSEC iQ-R series programmable controllers.
This manual describes the procedures before operation, specifications, functions, devices, parameters, and troubleshooting 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.
Please make sure that the end users read this manual.

Relevant products

Item	Model
CPU module	R08PCPU, R16PCPU, R32PCPU, R120PCPU
Redundant function module	R6RFM

CONTENTS

SAFETY PRECAUTIONS 1
INTRODUCTION 11
RELEVANT MANUALS 24
TERMS 25
GENERIC TERMS AND ABBREVIATIONS 26

PART 1 PART NAMES

CHAPTER 1 CPU MODULE 28

CHAPTER 2 EXTENDED SRAM CASSETTE 31

CHAPTER 3 REDUNDANT FUNCTION MODULE 33

PART 2 PROCEDURES BEFORE OPERATION

CHAPTER 4 START-UP PROCEDURE 36

4.1 Overview. 36
Procedure for process mode 36
4.2 Installing a Battery 38
Installation procedure 38
4.3 Inserting or Removing an Extended SRAM Cassette 39
Insertion procedure 39
Removal procedure 39
4.4 Inserting and Removing an SD Memory Card 40
Insertion procedure 40
Removal procedure 40
4.5 Creating a Project 41
4.6 Connecting a Personal Computer 42
4.7 Initializing the CPU Module 43
4.8 Setting Parameters 43
4.9 Programming 46
Registering labels 46
Inserting program elements 47
Inserting POUs by key input 49
4.10 Converting the Program 50
4.11 Saving the Project 50
4.12 Writing Data to the Programmable Controller 50
4.13 Resetting the CPU Module 51
4.14 Executing the Program 52
4.15 Monitoring the Program 53

CHAPTER 5 PROCEDURE FOR STARTING UP A REDUNDANT SYSTEM 54

5.1 Overview. 54
Starting up both systems simultaneously 54
Starting up the systems one by one 57

Precautions when the data logging function is used . . . 61
5.2 Wiring . . . 62
Redundant function modules . . . 62
Redundant extension base units . . . 63
Power supply modules in a redundant system . . . 63
5.3 Creating a Project . . . 64
5.4 Connecting a Personal Computer and the CPU Module . . . 64
5.5 Setting the System (System A or System B) . . . 65
5.6 Writing Data to the Programmable Controller . . . 67
5.7 Monitoring the Program . . . 68

PART 3 SYSTEM CONFIGURATION

CHAPTER 6 SYSTEM CONFIGURATION 70

PART 4 SPECIFICATIONS

CHAPTER 7 PERFORMANCE SPECIFICATIONS 74

7.1 CPU Module . . . 74
Hardware specifications . . . 74
Programming specifications . . . 76
7.2 Extended SRAM Cassette . . . 78
7.3 Redundant Function Module . . . 78

PART 5 CPU MODULE OPERATION

CHAPTER 8 RUNNING A PROGRAM 80

8.1 Scan Configuration . . . 80
Initial processing (when powered on or switched to RUN) . . . 82
I/O refresh . . . 82
Program operation . . . 82
END processing . . . 83
8.2 Scan Time . . . 84
Initial scan time . . . 84
Constant scan . . . 85
Device/label access service processing setting . . . 87
Device/label access service processing constant wait function . . . 91
8.3 Data Communication and I/O Processing . . . 94
Data communication . . . 94
I/O processing and response delay . . . 94
Refresh mode . . . 95
Direct mode . . . 98
8.4 Program Flow . . . 100
8.5 Program Execution Type . . . 101
Initial execution type program . . . 101
Scan execution type program . . . 102
Fixed scan execution type program . . . 102
Event execution type program . . . 107

Standby type program . . . 111
Execution type change . . . 112
Group setting for refresh . . . 113
8.6 Subroutine Program . . . 114
8.7 Interrupt Program . . . 115
Interrupt period setting . . . 125
Interrupt enabled during instruction execution . . . 125
Processing at interrupt program startup . . . 126
Multiple interrupt function . . . 129
Precautions . . . 133

CHAPTER 9 CPU MODULE OPERATION PROCESSING 135

9.1 Operation Processing by Operating Status . . . 135
9.2 Operation Processing When Operating Status Is Changed . . . 136
Output mode at operating status change (STOP to RUN) . . . 137
9.3 Operation Processing at Momentary Power Failure . . . 138

CHAPTER 10 MEMORY CONFIGURATION OF THE CPU MODULE 139

10.1 Memory Configuration . . . 139
Program memory/program cache memory . . . 140
Device/label memory . . . 141
Data memory . . . 145
Refresh memory . . . 145
CPU buffer memory . . . 145
Signal flow memory . . . 146
SD memory card . . . 147
10.2 File Size Unit in Memory . . . 149
10.3 Memory Operation . . . 150
Initialization and value clear . . . 150
10.4 Files . . . 151
File types and storage memory . . . 151
File operation available . . . 153
File size . . . 154

CHAPTER 11 BASIC CONCEPT OF REDUNDANT SYSTEM 157

11.1 Concept of "Systems" in a Redundant System . . . 157
11.2 System Switching Between Control System and Standby System . . . 158
11.3 System Consistency Check . . . 158
11.4 Operation Modes of the Process CPU . . . 158
11.5 Operation Modes of a Redundant System . . . 159
11.6 System Determination . . . 161
When both systems are started up . . . 161
When only one system is started up . . . 163
When one system is started automatically even though a tracking communication error has occurred . . . 165
When the previous control system is started up as the control system . . . 169
11.7 State Transition of a Redundant System . . . 175
11.8 Access in a Redundant System with Redundant Extension Base Unit . . . 176
Access to modules on the extension base unit . . . 178

PART 6 FUNCTIONS

CHAPTER 12 FUNCTION LIST 180

CHAPTER 13 CLOCK FUNCTION 185

13.1 Time Setting . . . 185
Clock data . . . 185
Changing the clock data . . . 185
Reading the clock data . . . 186
Precautions on the clock data . . . 186
13.2 Setting Time Zone . . . 187
13.3 System Clock . . . 188
Special relay used for system clock . . . 188
Special register used for system clock . . . 188

CHAPTER 14 WRITING DATA TO THE CPU MODULE 189

14.1 Writing Data to the Programmable Controller . . . 189
14.2 Online Change . . . 189
Online change (ladder block) . . . 189
File batch online change . . . 193
14.3 Precautions . . . 194

CHAPTER 15 RAS FUNCTIONS 200

15.1 Scan Monitoring Function . . . 200
Scan time monitoring time setting . . . 200
Watchdog timer reset . . . 200
Precautions . . . 201
15.2 Self-Diagnostics Function . . . 202
Self-diagnostics timing . . . 202
Check method of error . . . 202
CPU module operation upon error detection setting . . . 203
Error detection invalidation setting . . . 207
15.3 Error Clear . . . 208
Clearing errors in the CPU module of the standby system from the CPU module of the control system . . . 210
15.4 Event History Function . . . 211
Event history setting . . . 212
Logging of the event history . . . 213
Viewing the event history . . . 216
Clearing the event history . . . 216
Precautions . . . 217
15.5 Program Cache Memory Auto Recovery Function . . . 218

CHAPTER 16 REMOTE OPERATION 219

16.1 Remote RUN/STOP . . . 219
Executing method . . . 219
Precautions . . . 220
16.2 Remote PAUSE . . . 221
Executing method . . . 221
16.3 Setting RUN-PAUSE Contacts . . . 222

16.4 Remote RESET . . . 223
Enabling remote RESET . . . 223
Executing method . . . 224
16.5 Precautions for Redundant System . . . 224

CHAPTER 17 BOOT OPERATION 225

17.1 Boot Operation Procedure . . . 225
17.2 Specifiable File Types . . . 226
17.3 Allowed Maximum Number of Boot Files . . . 226
17.4 Configuring the Boot Setting . . . 227
17.5 Writing Boot Settings and Boot Files . . . 229
17.6 Operation When Security Functions Are Enabled . . . 229
When a security key is set . . . 229
When a file password is set . . . 229
17.7 Precautions . . . 230

CHAPTER 18 MONITOR FUNCTION 231

18.1 Module Error Information Collection Function . . . 232

CHAPTER 19 TEST FUNCTION 235

19.1 External Input/Output Forced On/Off Function . . . 235
19.2 Device Test with Execution Conditions . . . 244

CHAPTER 20 DATA LOGGING FUNCTION 258

20.1 Data Logging Procedure . . . 259
20.2 Specifications . . . 260
20.3 Data to Be Collected . . . 261
Number of data points . . . 261
Data type . . . 261
Data to be collected . . . 261
20.4 Data Collection Conditions . . . 262
Each scan . . . 262
Time specification . . . 262
Interrupt occurrence . . . 263
Condition specification . . . 264
20.5 Logging Type . . . 267
Operating procedure for continuous logging . . . 267
Operating procedure for trigger logging . . . 268
20.6 Data Logging File . . . 272
Storage format of data logging files . . . 272
Storage location of data logging files . . . 281
20.7 States of the Data Logging Function . . . 282
Data logging states . . . 282
LED status . . . 284
20.8 Steps Until the Collected Data Is Saved . . . 285
Internal buffer . . . 286
Switching to a storage file . . . 287
20.9 Missing Data . . . 291
Conditions under which missing data occurs . . . 291
20.10 Setting Behavior at the Time of Transition to RUN . . . 292

20.11 Auto Logging . . . 293
20.12 SD Memory Card Replacement . . . 297
20.13 SD Memory Card Life When the Data Logging Function Is Used . . . 299
20.14 Errors Generated During Data Logging . . . 300
20.15 Special Relay and Special Register Used by the Data Logging Function . . . 300
20.16 Precautions to Take When Using the Data Logging Function . . . 301

CHAPTER 21 PID CONTROL/PROCESS CONTROL FUNCTION 308

21.1 PID Control Function . . . 308
21.2 Process Control Function . . . 309
Process control by using process control function blocks . . . 309
Process control by using process control instructions . . . 310

CHAPTER 22 CPU MODULE DATA BACKUP/RESTORATION FUNCTION 311

22.1 Backup Function . . . 317
Backup processing triggered by turning on SM1351 . . . 319
Automatic backup using SD944 . . . 320
Settings for automatic restoration with the SD CARD OFF button . . . 322
Checking backup errors . . . 322
Precautions . . . 323
22.2 Restoration Function . . . 327
Automatic restoration using SD955 . . . 329
Automatic restoration with the SD CARD OFF button . . . 331
Checking restoration errors . . . 333
Precautions . . . 333

CHAPTER 23 MULTIPLE CPU SYSTEM FUNCTION 337

23.1 Out-of-group I/O Fetch . . . 338
Accessing controlled module . . . 338
Accessing non-controlled module . . . 338
23.2 Operation Settings . . . 341
Stop setting . . . 341
Synchronous startup setting . . . 342
Clock data . . . 345
23.3 Multiple CPU Parameter Checking . . . 346
23.4 Data Communication Between CPU Modules . . . 347
Memory to be used . . . 347
Fixed scan communication setting . . . 352
Error detection setting . . . 352
Module-by-module data guarantee . . . 353
Communication through refresh . . . 355
Communication through direct access . . . 359
Data assurance by program . . . 362
Communication between CPU modules in error state . . . 365
23.5 Multiple CPU Synchronous Interrupt . . . 366
Execution timing . . . 367
Multiple interrupt . . . 367
Precautions . . . 367

CHAPTER 24 SECURITY FUNCTION 368

CHAPTER 25 ROUTING SETTING 370

25.1 Setting Method. 370
25.2 Setting Example. 371
25.3 Precautions 372

CHAPTER 26 REDUNDANT FUNCTION 373

26.1 Operation Mode Change 374
Procedures 374
Precautions 376
26.2 System Switching 377
System switching method 377
Operation at system switching. 383
Execution availability of system switching 384
Check method of system switching information 386
Precautions 388
26.3 Tracking Transfer. 391
Tracking data 392
Tracking block and tracking trigger 396
Setting procedure for tracking transfer 397
Tracking transfer setting 398
Tracking mode 403
Precautions 406
26.4 Memory Copy from Control System to Standby System. 409
Auto memory copy. 411
Memory copy using the engineering tool 412
Memory copy using the special relay and special register 413
Precautions 415
26.5 System Consistency Check 417
Files 419
Operating status 420
Modules on the main base unit 420
Checking the system configuration in a redundant system with redundant extension base unit. 421
SD memory card 421
26.6 Program Execution in Both Systems 422
Operation of a program executed in both systems 424
Precautions 426
26.7 Redundant System Operation Setting. 429
Standby system output setting 431
Setting to wait cyclic data receive after system switching 432
26.8 Redundant Function Module Communication Test 434
Execution procedure of the module communication test. 434
26.9 Settings for Redundant System with Redundant Extension Base Unit 435
Automatic recovery of the CPU module of the standby system 436
26.10 Control System Start-up at One System Error 438
Setting method 438
Starting a control system 438
Precautions 441
26.11 SLMP Communications 443
26.12 Precautions on Programming 444
Instructions not available in redundant system 444

Interrupt from modules . . . 451
Precautions for using the annunciator (F) . . . 452
Precautions on timers and timer function blocks . . . 453
Precautions on access to intelligent function module or external devices . . . 454
Precautions on writing data from GOT or external devices . . . 456
Precautions on outputting in the middle of the scan . . . 456
Precautions for the redundant extension base unit configuration . . . 457

PART 7 DEVICES, LABELS, AND CONSTANTS

CHAPTER 27 DEVICES 460

27.1 Device List . . . 460
27.2 Device Setting . . . 462
Range of use of device points . . . 463
27.3 User Device . . . 464
Input (X) . . . 464
Output (Y) . . . 464
Internal relay (M) . . . 465
Latch relay (L) . . . 465
Link relay (B) . . . 465
Annunciator (F) . . . 465
Link special relay (SB) . . . 467
Edge relay (V) . . . 468
Step relay (S) . . . 468
Timer . . . 469
Counter . . . 477
Data register (D) . . . 480
Link register (W) . . . 480
Link special register (SW) . . . 480
27.4 System Device . . . 481
Function device (FX/FY/FD) . . . 481
Special relay (SM) . . . 482
Special register (SD) . . . 482
27.5 Link Direct Device . . . 483
Specification method . . . 483
Specification range . . . 484
Difference from link refresh . . . 485
27.6 Module Access Device . . . 486
Precautions . . . 486
27.7 CPU Buffer Memory Access Device . . . 487
Specification method . . . 487
27.8 Index Register (Z/LZ) . . . 488
16-bit index modification . . . 488
32-bit index modification . . . 488
Device for which index modification can be performed . . . 489
Index register setting . . . 489
Combination of index modification . . . 490
Precautions . . . 490
27.9 File Register (R/ZR) . . . 492
Specification method . . . 492

Setting file registers . . . 493
Clearing file registers . . . 493
27.10 Refresh Data Register (RD) . . . 494
Refresh memory setting . . . 494
27.11 Nesting (N) . . . 495
27.12 Pointer (P) . . . 496
Global pointer . . . 496
Local pointer . . . 497
Pointer setting . . . 498
27.13 Interrupt Pointer (I) . . . 499
Interrupt factors of the interrupt pointer numbers . . . 499
The priority for the interrupt pointer numbers and interrupt factors . . . 500
27.14 Network No. Specification Device (J) . . . 501
27.15 I/O No. Specification Device (U) . . . 501
27.16 SFC Block Device (BL) . . . 501
27.17 SFC Transition Device (TR) . . . 501
27.18 Global Device . . . 501
27.19 Local Device . . . 502
27.20 Indirect Specification . . . 506

CHAPTER 28 LABELS 507

28.1 Global Labels . . . 507
28.2 Local Labels . . . 508
28.3 Classes . . . 508
28.4 Data Types . . . 509
28.5 Arrays . . . 512
28.6 Structures . . . 516
28.7 Label Access Setting from External Device . . . 518
Configuration procedure . . . 520
Label communication data storage location and operation at the time of each setting operation . . . 520
File operation . . . 520
28.8 Precautions . . . 521

CHAPTER 29 LATCH FUNCTION 523

29.1 Latch with Battery . . . 523
Types of latch . . . 523
Applicable devices and labels . . . 524
Setting latch on devices . . . 525
Setting latch on labels . . . 529
Clearing latch range data . . . 529
Precautions . . . 529

CHAPTER 30 DEVICE/LABEL INITIAL VALUE SETTINGS 530

30.1 Setting Initial Device/Label Values . . . 531
Setting initial device values . . . 531
Setting initial label values . . . 532
30.2 Applicable Devices/Labels . . . 533
30.3 Precautions . . . 533

CHAPTER 31 LABEL INITIALIZATION FUNCTION 534
31.1 Initialization of Labels After Rebuilt All (Reassignment) 534
31.2 Label Initial Value Reflection Setting 536
CHAPTER 32 CONSTANTS 538
32.1 Decimal Constant (K) 538
32.2 Hexadecimal Constant (H) 538
32.3 Real Constant (E) 539
Setting range for real numbers 539
Processing when operation is in progress 539
32.4 Character String Constant 539
32.5 Notation of Constants 540
PART 8 TROUBLESHOOTING
CHAPTER 33 TROUBLESHOOTING PROCEDURE 544
33.1 Troubleshooting with LED Indicators 545
LED status of the CPU module 545
33.2 Troubleshooting Using the Engineering Tool 546
System monitor 546
Module diagnostics 546
Event history 547
CHAPTER 34 TROUBLESHOOTING BY SYMPTOM 548
34.1 When the POWER LED of the Power Supply Module Turns Off 548
34.2 When the READY LED of the CPU Module Turns Off 548
34.3 When an Error Has Occurred in a Redundant Function Module 549
When the RUN LED turns off 549
When the ERR LED turns on or flashes 549
When the L ERR LED turns on 549
34.4 When the Specific Extension Base Unit Cannot Be Recognized 550
34.5 When the Specific Q Series Extension Base Unit Cannot Be Recognized 550
34.6 When an Error Has Occurred in an Extension Base Unit for a Redundant System 550
34.7 When Data Cannot Be Written to the Programmable Controller 550
34.8 When Data Cannot Be Read from the Programmable Controller 550
34.9 When the Operating Status of the CPU Module Cannot Be Changed 551
34.10 When the Ethernet Function Cannot Be Used 551
34.11 When Cyclic Data Is Turned Off at the Time of System Switching 551
CHAPTER 35 MAINTENANCE AND INSPECTION OF A REDUNDANT SYSTEM 552
35.1 Module Replacement in a Redundant System 552
Replacing the CPU module 554
Replacing the power supply module 556
Replacing the redundant power supply module 556
Replacing the redundant function module 557
Replacing the I/O module 557
Replacing the intelligent function module 558
Procedure for remote head module replacement 558
Replacing the main base unit 559

Replacing the extension cable between the main base unit and extension base unit. 560
Replacing the extension cable between extension base units 560
35.2 Replacement/Addition of Extension Cables (Online). 561

CHAPTER 36 ERROR CODES 563

36.1 Error Code System 563
36.2 Operation When an Error Occurs 565
36.3 How to Clear Errors. 565
36.4 List of Error Codes 565
Codes of errors detected by the self-diagnostic function (1000H to 3FFFH) 565
Codes of errors detected by other than the self-diagnostic function (4000H to 4FFFH) 617
Codes of errors detected by other than the self-diagnostic function (6F00H to 6FFFH). 642
Codes of errors detected by other than the self-diagnostic function (C000H to CFFFH) 643

CHAPTER 37 EVENT LIST 644

37.1 How to Read the Event List 644
37.2 Event List 647

APPENDICES 653

Appendix 1 External Dimensions 653
Appendix 2 Compliance with EMC and Low Voltage Directives 656
Appendix 3 Functional Availability by CPU Module Model 657
Appendix 4 List of Special Relay Areas 659
Diagnostic information 660
System information 663
SFC information. 665
System clock 666
Fixed scan function information. 668
Drive information 669
Instruction related 670
Latch area 672
Data logging function. 673
CPU module data backup/restoration function 675
File transfer function (FTP client) 675
Event history function 675
Ethernet function 676
Online module change function. 677
Redundant function 679
Module error information collection function 681
Appendix 5 List of Special Register Areas 682
Diagnostic information 683
System information 694
SFC information. 698
System clock 698
Fixed scan function information. 698
Drive information 701
Instruction related 703
Firmware update function 705
Latch area 707
Data logging function. 709

CPU module data backup/restoration function . . . 710
Interrupt pointer mask pattern . . . 710
Event history function . . . 711
Debug function . . . 711
Ethernet function . . . 712
Online module change function . . . 714
System information . . . 715
Redundant function . . . 715
Module error information collection function . . . 721
Appendix 6 Buffer Memory . . . 722
Appendix 7 Processing Time . . . 723
Instruction execution time . . . 723
Program execution time . . . 723
END processing time . . . 725
Data logging function processing time . . . 735
Process control function processing time . . . 736
SFC program processing time . . . 737
SFC program switching . . . 738
Redundant function processing time . . . 739
Appendix 8 Parameter List . . . 747
System parameters . . . 747
CPU parameters . . . 748
Module parameters . . . 749
Memory card parameters . . . 749
Appendix 9 Target List and Operation Details of the Device/Label Access Service Processing Setting . . . 750
Target list . . . 750
Operation details . . . 752
Appendix 10 Program Restoration Information Write Selection . . . 756
Checking the program restoration information write status . . . 756
Precautions . . . 758
Appendix 11 Precautions for Communications with CPU Module in Redundant System via Module on Extension Base Unit . . . 759
Appendix 12 Program for System Switching at Built-in Ethernet Communication Error . . . 760
Appendix 13 Added and Enhanced Functions . . . 762

INDEX 765

REVISIONS . . . 769
CONDITIONS OF USE FOR THE PRODUCT . . . 770
WARRANTY . . . 771
INFORMATION AND SERVICES . . . 772
TRADEMARKS . . . 772
COPYRIGHTS . . . 772

RELEVANT MANUALS

The following manuals are relevant to this product.

Manual name [manual number]	Description
MELSEC iQ-R Process CPU Module User's Manual [SH-082493ENG] (this manual)	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 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]	General process FBs, tag access FBs, tag FBs, and process control instructions designed for 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

This manual does not include detailed information on the following:

- General specifications
- Applicable combinations of CPU modules and the other modules, and the number of mountable modules
- Installation

For details, refer to the following.

MELSEC iQ-R Module Configuration Manual

This manual does not include information on the module function blocks.

For details, refer to the Function Block Reference for the module used.

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.
FB instance	A function block that is inserted to a sequence program
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 where the I/O signals and buffer memory areas of a module have already been defined. For the module used, GX Works3 automatically generates this label, which can be used as a global label. The module labels allow the user to create programs in a simple way, without considering module internal addresses.
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.
Process mode	Mode where a redundant system is not configured. Process control function blocks and the online module change function can be executed.
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 mode	Mode for configuring a redundant system. Process control function blocks and the online module change function can be used even in this mode.
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)
Redundant system with redundant extension base unit	A redundant system that is configured using extension base unit(s)
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.
Standby system	A backup system in a redundant system

Term	Description
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.
Tracking cable	An optical fiber cable used to connect two redundant function modules in a redundant system

GENERIC TERMS AND ABBREVIATIONS

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

Generic term and abbreviation	Description
Base unit	A main base unit, an extension base unit, an RQ extension base unit
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
CPU module	R08PCPU, R16PCPU, R32PCPU, R120PCPU
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
Network module	Includes the following: • Ethernet interface module • CC-Link IE Controller Network module • CC-Link IE Field Network master/local module • MELSECNET/H network module • MELSECNET/10 network module
Power supply module	A MELSEC iQ-R series power supply module
Process CPU	R08PCPU, R16PCPU, R32PCPU, R120PCPU
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
Station sub ID number	An ID number of a sensor connected to a CC-Link-compatible communication module

PART 1 PART NAMES

This part consists of the following chapters.

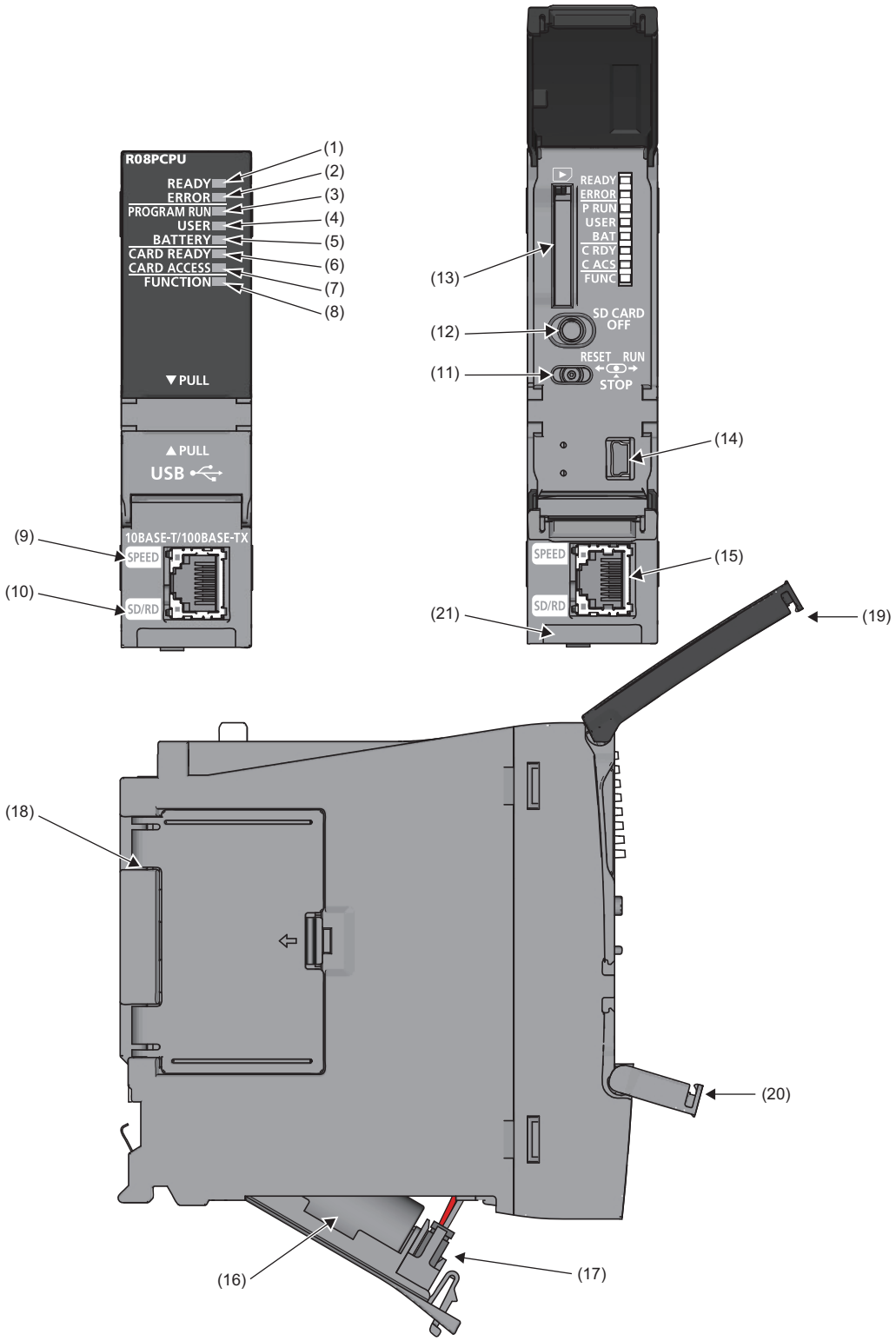
1 CPU MODULE

2 EXTENDED SRAM CASSETTE

3 REDUNDANT FUNCTION MODULE

1 CPU MODULE

This chapter describes the part names of the CPU module.
The R08PCPU is used as an example.



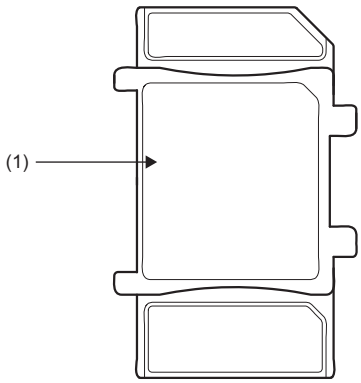
No.	Name	Description
(1)	READY LED	Indicates the operating status of the CPU module and the error level. (☞ Page 545 LED status of the CPU module) ● READY LED-ERROR LED status On-off: Normal operation On-on: Minor error On-flashing: Moderate error Flashing-on: Minor error (Changing module online) Flashing-flashing: Preparing for reset Flashing (every 2s)-off: Initial processing Flashing (every 400ms)-off: Changing module online Off-on/flashing: Major error
(2)	ERROR LED	
(3)	PROGRAM RUN LED	Indicates the operating status of the program. On: Being executed (RUN state) Flashing: Being suspended (PAUSE state) Off: Stopped (STOP state) or stop error
	PROGRAM RUN LED (WHEN USING REDUNDANT MODE)	Indicates the operating status of the program. ■Control system (CTRL LED of the redundant function module: On) On: Being executed (RUN state) Flashing: Being suspended (PAUSE state) Off: Stopped (STOP state) or stop error ■Standby system (SBY LED of the redundant function module: On) [Backup mode] On: Being executed (RUN state) (programs being executed in both systems) Flashing: Being suspended (PAUSE state) (programs being executed in both systems) Off: Stopped (STOP state/RUN state/PAUSE state) (no program being executed in both systems) or stop error [Separate mode] On: Being executed (RUN state) Flashing: Being suspended (PAUSE state) or waiting for state transition to RUN (same as STOP state) Off: Stopped (STOP state) or stop error ■Control or standby system has not been determined yet Flashing: Waiting for state transition to RUN by switch operation (same as STOP state) Off: Normal operation
(4)	USER LED	Indicates the status of the annunciator (F). (☞ Page 465 Annunciator (F)) On: Annunciator (F) ON Off: Normal operation
(5)	BATTERY LED	Indicates the battery status. Flashing: Battery low Off: Normal operation
(6)	CARD READY LED	Indicates the availability of the SD memory card. On: Available Flashing: Ready Off: Not available or not inserted
(7)	CARD ACCESS LED	Indicates the access status of the SD memory card. On: Being accessed Off: Not accessed
(8)	FUNCTION LED	Turns on or flashes when the following functions are used: • LED display setting • External input/output forced on/off function • Program restoration information write • Device test with execution condition For the LED status, refer to the description of each function. ☞ Page 206 LED display setting
(9)	SPEED LED	Refer to the following. 📖 MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)
(10)	SD/RD LED	
(11)	RUN/STOP/RESET switch	A switch for controlling the operating status of the CPU module. (☞ Page 52 Executing the Program) RUN: Executes the program. STOP: Stops the program. RESET: Resets the CPU module. (Keep the switch in the RESET position for approximately one second.) Operate the RUN/STOP/RESET switch with your fingers. To prevent the switch from being damaged, do not use any tool such as a screwdriver.
(12)	SD CARD OFF button	A switch for disabling access to the SD memory card to remove it from the CPU module (☞ Page 40 Inserting and Removing an SD Memory Card)
(13)	SD memory card slot	A slot where an SD memory card is inserted
(14)	USB port*1	A connector for a USB-compatible peripheral (connector type: miniB)

No.	Name	Description
(15)	Ethernet port	Refer to the following. 📖 MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)
(16)	Battery	A backup battery to hold clock data and to use the backup power function for the device/label memory
(17)	Battery connector pin	A pin for connecting a lead wire of the battery (To save the battery, the lead wire is disconnected from the connector before shipment.)
(18)	Cassette cover	A cover for the connector where an extended SRAM cassette is inserted (☞ Page 31 EXTENDED SRAM CASSETTE, ☞ Page 39 Inserting or Removing an Extended SRAM Cassette)
(19)	LED cover	A cover for the LED indicators, SD memory card slot, and switches. Open this cover and insert or remove an SD memory card or set the RUN/STOP/RESET switch. Otherwise, keep the cover closed to prevent entry of foreign matter such as dust.
(20)	USB cover	A cover for the USB port. Open this cover and connect a USB-compatible peripheral. Otherwise, keep the cover closed to prevent entry of foreign matter such as dust.
(21)	Production information marking	Shows the production information (16 digits) of the module.

*1 When a cable is connected to the USB connector at all times, clamp the cable to prevent a poor connection, moving, and disconnection by unintentional pulling.

2 EXTENDED SRAM CASSETTE

This chapter describes the part names of the extended SRAM cassette.

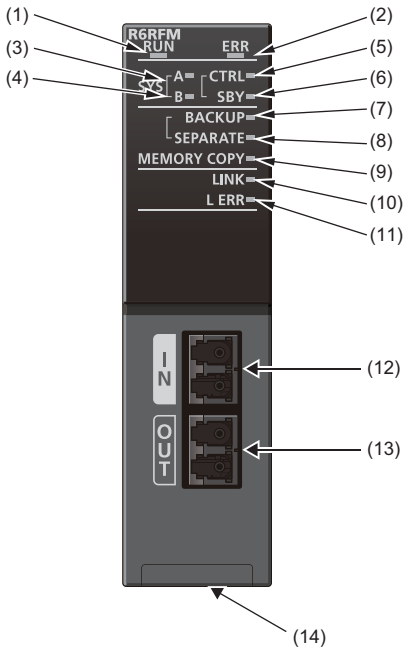


No.	Name	Description
(1)	Tab for cassette insertion/removal	The part that is held when an extended SRAM cassette is inserted or removed (☞ Page 39 Inserting or Removing an Extended SRAM Cassette)

MEMO

3 REDUNDANT FUNCTION MODULE

This chapter describes the part names of the redundant function module.



No.	Name	Description
(1)	RUN LED	Indicates the operating status. On: Normal operation Flashing: Changing module online or executing a module communication test Off: Error (☞ Page 549 When the RUN LED turns off)
(2)	ERR LED	Indicates the error status of the module. On: Error or module communication test completed with an error (☞ Page 549 When the ERR LED turns on or flashes) Flashing: Error (☞ Page 549 When the ERR LED turns on or flashes) Off: Normal operation
(3)	SYS A LED	Indicates the system A/B setting. On: System A Flashing: Parameter (set to system A) invalid Off: System B or not set
(4)	SYS B LED	Indicates the system A/B setting. On: System B Flashing: Parameter (set to system B) invalid Off: System A or not set
(5)	CTRL LED	Indicates the control status of the CPU module. Flashing: Control system Off: Standby system or not determined
(6)	SBY LED	Indicates the control status of the CPU module. On: Standby system Off: Control system or not determined
(7)	BACKUP LED	Indicates the operation mode. On: Backup mode Flashing: Cause of system switching failure being present*1 Off: Separate mode
(8)	SEPARATE LED	Indicates the operation mode. On: Separate mode Flashing: Cause of system switching failure being present*1 Off: Backup mode
(9)	MEMORY COPY LED	Indicates the memory copy status. On: Memory copy completed (standby system) Flashing (every 200ms): Memory copy being executed Flashing (every 1s): Memory copy error (standby system) Off: Memory copy not executed, or memory copy completed (control system)

No.	Name	Description
(10)	LINK LED	Indicates the tracking communication status. On: Tracking communications being performed Off: Tracking communications not performed
(11)	L ERR LED	Indicates the tracking communication error. On: Tracking communication error (one of the following) • A receive data is faulty (receive frame error). • A tracking cable is disconnected. • A tracking cable is incorrectly inserted. Off: Normal operation
(12)	Optical connector (IN)	A connector for one of two tracking cables. The other end of the cable is connected to the OUT connector of the redundant function module in the other system.
(13)	Optical connector (OUT)	A connector for one of two tracking cables. The other end of the cable is connected to the IN connector of the redundant function module in the other system.
(14)	Production information marking	Shows the production information (16 digits) of the module.

*1 The cause of flashing can be checked in SD1642 (BACKUP/SEPARATE LED flashing cause).

PART 2 PROCEDURES BEFORE OPERATION

This part consists of the following chapters.

4 START-UP PROCEDURE

5 PROCEDURE FOR STARTING UP A REDUNDANT SYSTEM

4 START-UP PROCEDURE

This chapter describes the procedures before operation.

4.1 Overview

This section describes an outline of the procedure before operation for each CPU module.

Procedure for process mode

This section describes an outline of the procedure when using process mode.

Point

The procedure for starting up a redundant system is partially different.
For the procedure to start up a redundant system, refer to the following.
☞ Page 54 PROCEDURE FOR STARTING UP A REDUNDANT SYSTEM

1. Installing a Battery

Install a battery to the CPU module. (☞ Page 38 Installing a Battery)

2. Inserting an extended SRAM cassette or an SD memory card

Insert an extended SRAM cassette or an SD memory card to the CPU module as needed. (☞ Page 39 Inserting or Removing an Extended SRAM Cassette, ☞ Page 40 Inserting and Removing an SD Memory Card)

3. Mounting modules and connecting cables

Mount modules on the base unit, and connect cables. (📖 MELSEC iQ-R Module Configuration Manual)

4. Powering on the system

Check the following before powering on the system.

- A cable is correctly connected to the power supply.
- Power supply voltage is within the specified range.
- The CPU module is in the STOP state.
- A personal computer on which an engineering tool has been installed is connected to the CPU module. (☞ Page 42 Connecting a Personal Computer)

5. Powering on the personal computer

Power on the personal computer on which the engineering tool has been installed (the personal computer connected to the CPU module).

6. Starting up the engineering tool

Start up the engineering tool that has been installed on the personal computer connected to the CPU module.

7. Initializing the CPU module

Initialize the CPU module using the engineering tool. (☞ Page 43 Initializing the CPU Module)

8. Setting parameters

The following table shows which parameters are required when changing the number of slots/the number of occupied points of a module, using an SD memory card, or using specific functions.

When	Required parameter	Reference
Changing the number of slots or the number of occupied points of a module	• System parameters • CPU parameters	☞ Page 43 Setting Parameters
Using an SD memory card	• Memory card parameters	☞ Page 225 BOOT OPERATION ☞ Page 518 Label Access Setting from External Device
Using the Ethernet function of the CPU module	• Module parameters Some intelligent function modules require multiple module parameters and module extension parameters.	📖 MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)
Using an intelligent function module		📖 User's manual for the module used

System parameters are automatically set by loading an actual system configuration to the Module Configuration window of the engineering tool.

9. Programming

Create a program using the engineering tool. (☞ Page 46 Programming)

10. Writing data to the programmable controller

Write the parameters set and the program created by using the engineering tool to the CPU module. (☞ Page 50 Writing Data to the Programmable Controller)

11. Resetting the CPU module

Restart the system in either of the following ways.

- Power off and on the system.
- Reset the CPU module. (☞ Page 51 Resetting the CPU Module)

12. Checking for errors

Check the status of the READY LED and ERROR LED of the CPU module. If an error is detected, identify the error using the engineering tool, and eliminate the error cause. (☞ Page 544 TROUBLESHOOTING PROCEDURE)

13. Executing the program

Run the CPU module, and check that the PROGRAM RUN LED (P RUN LED) turns on. (☞ Page 52 Executing the Program)

14. Monitoring the program

Check that the program operates normally using the engineering tool. (☞ Page 53 Monitoring the Program)

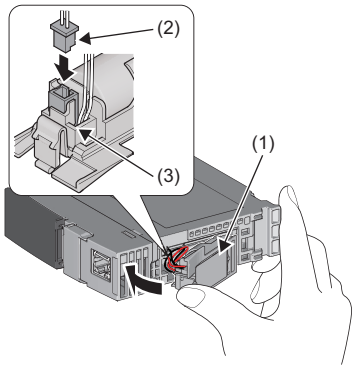
4.2 Installing a Battery

Install a battery to the CPU module.

Installation procedure

Q6BAT

The connector plug of the Q6BAT is disconnected from the jack of the CPU module before shipment. To use the battery, connect the connector plug of the Q6BAT and the jack of the CPU module by following the procedure below.



1. Open the battery cover located on the bottom of the CPU module.
2. Check that the Q6BAT (1) is correctly installed.
3. Check the direction and securely insert the connector plug (2) of the Q6BAT to the jack (3) of the CPU module.
4. Close the battery cover.

■Precautions

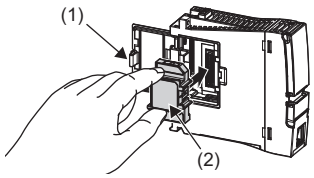
- When using the CPU module with the battery removed, check that the jack (3) of the CPU module is inserted in the socket on the battery cover located on the bottom of the CPU module.
- Do not drop or apply strong shock to the module and the battery.

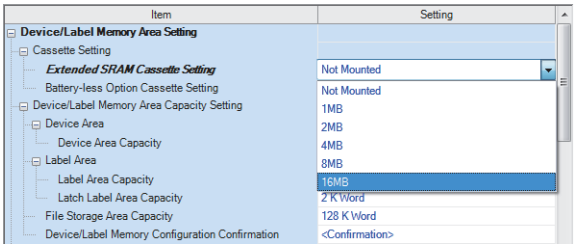
4.3 Inserting or Removing an Extended SRAM Cassette

Insert an extended SRAM cassette to the CPU module as needed.

Insertion procedure

Insert an extended SRAM cassette while the programmable controller is powered off.





1. Open the cassette cover (1) located on the side of the CPU module.
2. Hold the top and bottom of the tab (2) of an extended SRAM cassette (with the notched edge facing to the right), and insert the cassette straight into the connector. After inserting the cassette, check that it is inserted completely.
3. Close the cover, and mount the CPU module on the base unit.
4. Power on the programmable controller.
5. Set the capacity of the inserted cassette to "Extended SRAM Cassette Setting" in "CPU Parameter" using the engineering tool.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Setting] ⇨ [Cassette Setting] ⇨ [Extended SRAM Cassette Setting]

6. Using the engineering tool, check that SM626 (Extended SRAM cassette insertion flag) is on.

Precautions

- The availability of the extended SRAM cassette differs depending on the CPU module used. For the availability, refer to the performance specifications of the extended SRAM cassette. (☞ Page 31 EXTENDED SRAM CASSETTE)
- When the extended SRAM cassette is inserted or removed, all the data such as devices and the file register stored in the device/label memory are erased. Back up the data in the programmable controller before replacing the cassette.
- If the capacity of the extended SRAM cassette differs before and after the replacement, the ERROR LED of the CPU module may flash. But, it is not an error. Change the capacity setting in the CPU parameters. (Refer to step 5 above.)

Restriction

The extended SRAM cassette for the Universal model QCPU (Q4MCA-□MBS) cannot be used.

Removal procedure

Remove the extended SRAM cassette while the programmable controller is powered off.

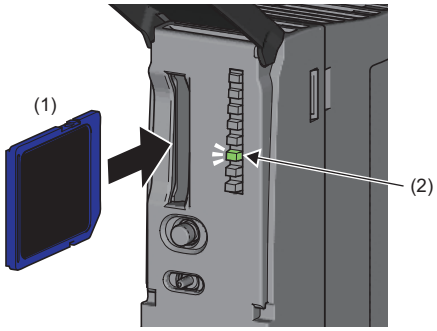
1. Read the data on the device/label memory from the CPU module, and save it in advance using the engineering tool. (When the extended SRAM cassette is removed, all of the data on the device/label memory are erased.)
2. Power off the programmable controller.
3. Remove the CPU module from the base unit, and open the cassette cover located on the side of the CPU module.
4. Hold the top and bottom of the tab of the extended SRAM cassette, and pull the cassette straight out of the connector.
5. Close the cover, and mount the CPU module back on the base unit.
6. Power on the programmable controller.
7. Set the "Extended SRAM Cassette Setting" in "CPU Parameter" to "Not Mounted".

4.4 Inserting and Removing an SD Memory Card

Insert an SD memory card to the CPU module as needed.

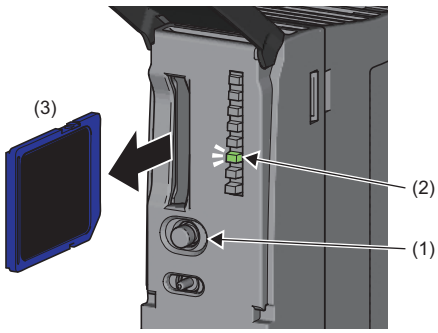
Insertion procedure

Check the direction and insert an SD memory card, following the procedure below.



1. Insert an SD memory card (1) into the card slot until it clicks with the notched edge in the direction as illustrated. After inserting the cassette, check that it is inserted completely. Poor contact may cause malfunction.
2. The CARD READY LED (C RDY LED) (2) starts flashing. When the card is ready to be used, the CARD READY LED (C RDY LED) stops flashing and turns on.
3. If the CARD READY LED (C RDY LED) does not turn on even after the card is inserted, check that SM606 (SD memory card forced disable instruction) and SM607 (SD memory card forced disable status flag) are off.

Removal procedure



1. Press the SD CARD OFF button (1) for one second or longer to disable access to the SD memory card.*1
2. The CARD READY LED (C RDY LED) (2) flashes during the access stop processing, and turns off upon completion of the processing.
3. Push in and release the SD memory card (3), and then pull the card out of the slot.

*1 SM606 (SD memory card forced disable instruction) can also disable access to an SD memory card. For details, refer to the following.
☞ Page 147 How to forcibly disable the SD memory card with a special relay

Precautions

- Do not drop or apply strong shock to the module and SD memory cards.
- Follow the procedure above when inserting or removing the SD memory card while the system is powered on. If not, the data on the SD memory card may corrupt.
- If any function that accesses the SD memory card is being executed when the SD CARD OFF button is pressed to remove the card, the CARD READY LED turns off after the processing of the function is completed. For this reason, the time required until the LED turns off differs depending on the function being executed.
- If SM605 (Memory card remove/insert prohibit flag) is on, the CARD READY LED does not turn off even if the SD CARD OFF button is pressed. If not, turn on SM606 (SD memory card forced disable instruction) to forcibly disable access to the card.

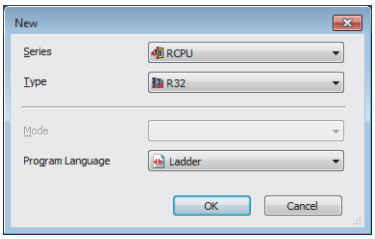
4.5 Creating a Project

Activate the engineering tool and create a project.

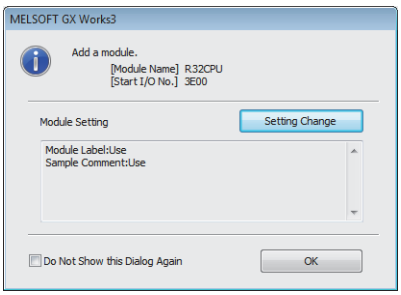
[Project] ⇨ [New]

Procedure

Create a program, following the procedure below. The procedure is for the program described in ladder diagrams.



1. Select "RCPU" in "Series", and the CPU module model to use in "Type". Then, select a programming language to use in the project in "Program". Select "Ladder", and click the [OK] button.



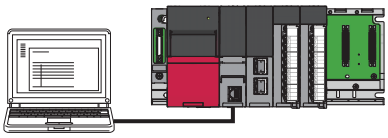
2. Click the [OK] button when the window for adding the module appears.

4.6 Connecting a Personal Computer

Connect a personal computer where an engineering tool has been installed to the CPU module.

Procedure

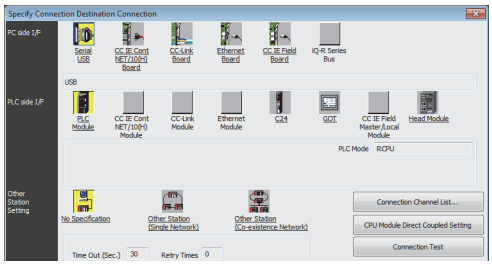
Connect a personal computer directly to the CPU module, following the procedure below.



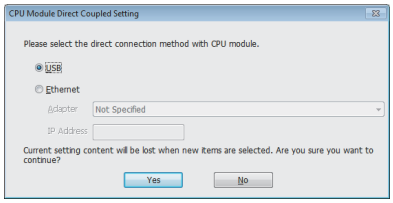
1. Connect a personal computer to the CPU module using a USB cable or Ethernet cable.

2. Select [Online] ⇨ [Current Connection Destination] on the menu bar of the engineering tool.

3. Click the [CPU Module Direct Coupled Setting] button on the "Specify Connection Destination Connection" window.



4. Select the connection method, and click the [Yes] button.



5. Click the [Connection Test] button, and check if the personal computer is connected to the CPU module.

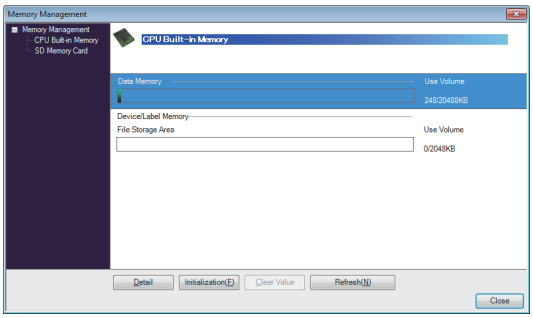
Install a USB driver to use a USB cable for the first time. (GX Works3 Operating Manual)

4.7 Initializing the CPU Module

Initialize the CPU module.

[Online] ⇨ [CPU Memory Operation]

Procedure



1. Select "Data Memory" on the "Memory Management" window, and click the [Initialization] button.
2. Select "File Storage Area", and click the [Initialization] button.
3. After the initialization processing completes, click the [Close] button.

4.8 Setting Parameters

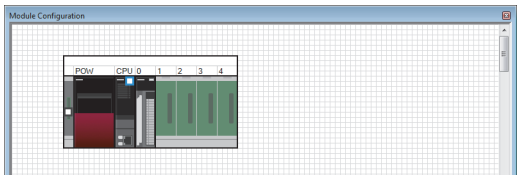
Set system parameters and parameters for each module.

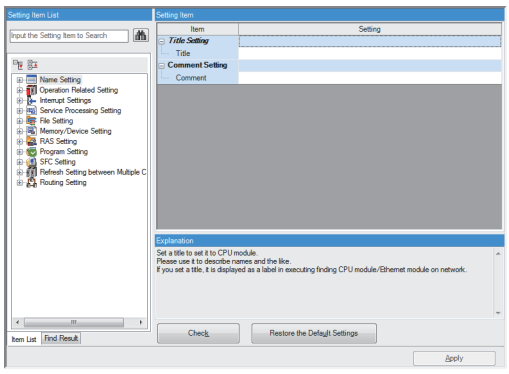
When the engineering tool is connected to the actual system

Set parameters by loading an actual system configuration to the Module Configuration window of the engineering tool.

[Navigation window] ⇨ [Module Configuration]

1. Open the Module Configuration window, and select [Online] ⇨ [Read Module Configuration from PLC] on the menu bar.
2. Click the [Yes] button when the window for adding module labels of the mounted modules appears.
3. The system parameters are automatically set, and the actual system configuration is displayed on the Module Configuration window.





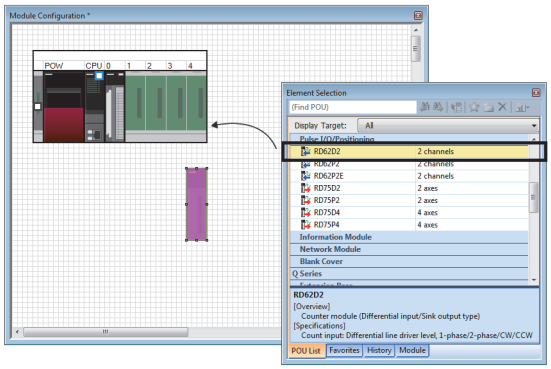
4. Double-click each module (CPU module, I/O module, or intelligent function module) to display the corresponding module parameter editor.
5. Set parameters, and click the [Apply] button to close the window.

■Precautions

The RQ extension base units, MELSEC-Q series extension base units, and modules mounted on those base units cannot be loaded. For those cannot be loaded, select the units and modules on the Element Selection window, and drag and drop them to the system configured on the Module Configuration window.

When the engineering tool is not connected to the actual system

Set parameters by configuring a system manually on the Module Configuration window of the engineering tool.



1. Select a base unit on the Element Selection window, and drag and drop it to the Module Configuration window.
2. Drag and drop modules to be used on the base unit placed.
3. Select [Edit] ⇨ [Parameter] ⇨ [Fix] on the menu bar.
4. Click the [Yes] button when the window for adding module labels of the configured modules appears.
5. Open the parameter editor of each module by double-clicking the module.
6. Set parameters, and click the [Apply] button to close the window.

Setting parameters from the Navigation window

Set the following parameters from the Navigation window.

■System parameters

These parameters need to be set from the Navigation window in the following cases: to change the number of slots on the base unit or the number of occupied points of the module; for a multiple CPU system; and for module synchronization operations.

[Navigation window] ⇨ [Parameter] ⇨ [System Parameter]

■Module parameters of the CPU module

These parameters are required to execute the Ethernet function of the CPU module.

[Navigation window] ⇨ [Parameter] ⇨ [Module model name] ⇨ [Module Parameter]

■Memory card parameters

These parameters are required to execute functions that access the SD memory card.

[Navigation window] ⇨ [Parameter] ⇨ [Module model name] ⇨ [Memory Card Parameter]

■Multiple module parameters and module extension parameters

Some intelligent function modules require multiple module parameters and module extension parameters.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [Intelligent function module] ⇨ [Module Parameter] or [Module Extended Parameter]

Point

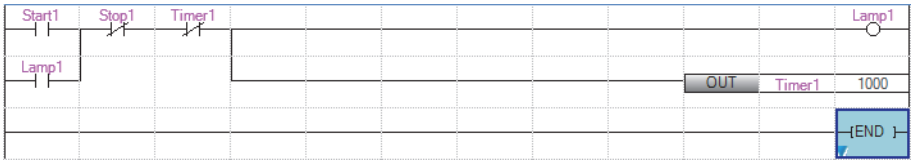
System parameters can be set on the following windows. Select the window depending on their application purposes.

- Module Configuration window: Use this window when using the module specific data (such as number of occupied points) as is.
- System Parameter window: Use this window when changing the number of slots or the number of occupied points.

4.9 Programming

Create a program. This section describes how to create a program using the following program example.

Program example



- When Start1 turns on, Timer1 starts counting, and Lamp1 turns on.
- When the current value of Timer1 reaches 1000, Lamp1 turns off.
- When Stop1 turns on, Lamp1 turns off.

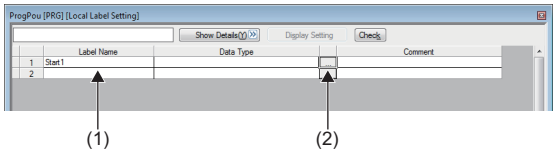
Registering labels

Label is a variable whose name and data type can be declared by a user.
Use of labels allows programming without being aware of devices and buffer memory addresses. For this reason, programs using labels can be used in other systems where the module configuration is different.
Labels can be registered on the label editor.

[Navigation window] ⇨ [Program] ⇨ [Scan] ⇨ [MAIN] ⇨ [ProgPou] ⇨ [Local Label]

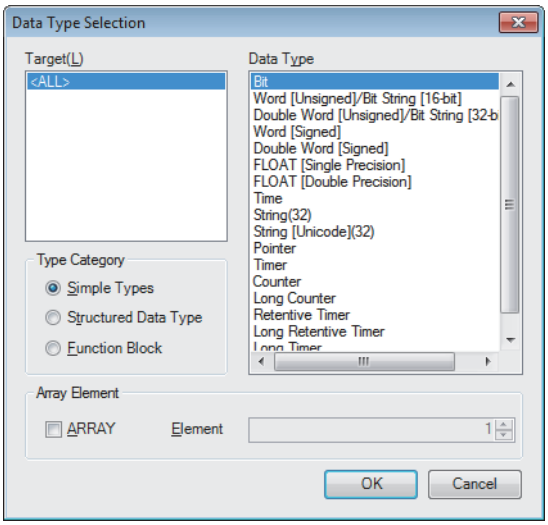
Procedure

Register the label "Start1" in the program example, following the procedure below.



1. Enter the name, "Start1", in the "Label Name" field (1).

2. Click the button (2) on the right of the "Data Type" field to open the "Data Type Selection" window.



3. Specify the data type. Select "Bit", and click the [OK] button.

Register other labels in the program example in the same way.

Point

- The class, initial value, and constant of labels can be set as needed by clicking the [Show Details] button on the label editor.
- Labels can also be registered while programming without opening the label editor. (☞ Page 49 Inserting POUs by key input)
- Devices can be assigned to global labels. Open the global label editor, and enter a device in the "Assign (Device/Label)" field.

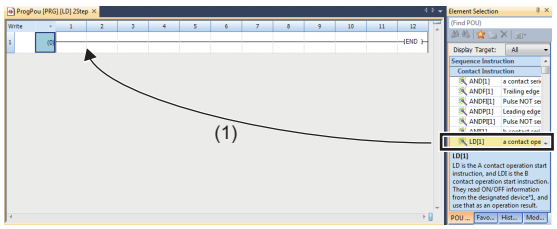
Inserting program elements

Drag and drop required program elements to the ladder editor.

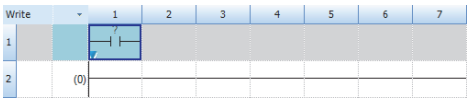
[Navigation window] ⇨ [Program] ⇨ [Scan] ⇨ [MAIN] ⇨ [ProgPou] ⇨ [ProgramBody]

Procedure

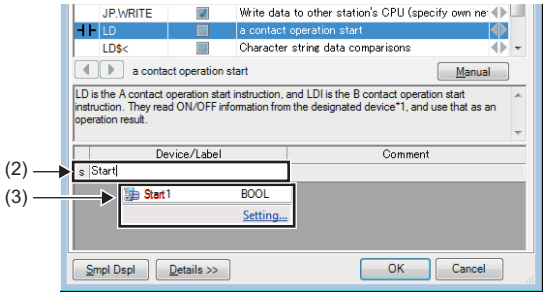
Insert a normally open contact of "Start1" in the program example, following the procedure below.



1. Select a program element from the Element Selection window, and drag and drop (1) it to the desired position on the ladder editor. In this example, drag and drop "LD[1]".

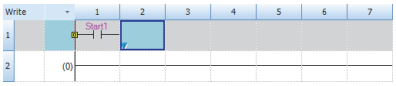


2. Double-click the inserted program element, and click the [Extd Dspl] button to open the ladder entry window.



3. Specify the operand. In this example, enter "Start" for "s" (2) in the "Device/Label" field.

4. Select an item from the displayed list (3). In this example, select "Start1".

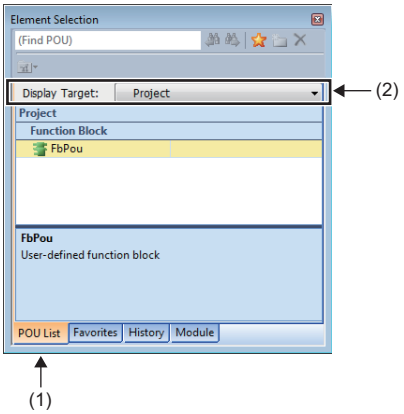


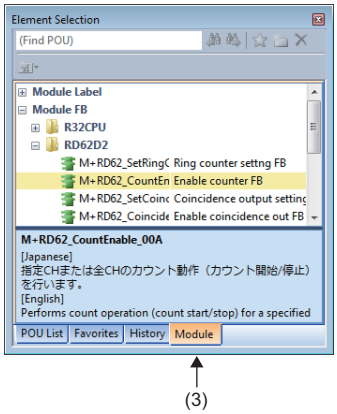
5. The normally open contact of "Start1" is inserted to the program.

Insert other program elements in the program example in the same way.

Point

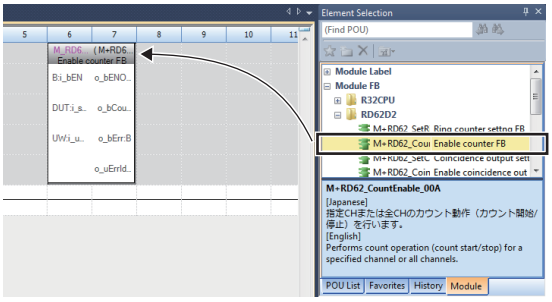
- Common instructions, such as contacts and coils, standard functions/function blocks, and user-defined function blocks/functions can be found in the [POU List] (1) in the Element Selection window. Narrow the list using the drop-down menu (2) for "Display Target", and select program elements.
- Module labels and module function blocks can be found in the [Module] list (3). Programs are efficiently created only by dragging and dropping the module labels and module function blocks on the editor.





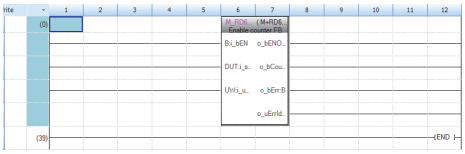
■Inserting function blocks

Insert function blocks, following the procedure below.

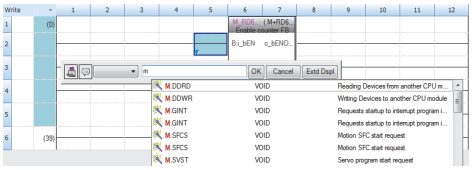


1. Select a function block from the Element Selection window, and drag and drop it to the desired position on the ladder editor.

2. The "FB Instance Name" window opens. Select the target label (global label or local label), and enter an instance name.



3. Select [Convert] ⇨ [Convert] on the menu bar. The ladder is converted, and the rungs are connected to the input and output labels of the FB instance.

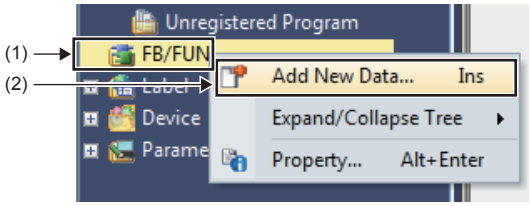


4. Add the input and output parts of the inserted function block to complete the program. For details on the input and output parts, refer to the following.

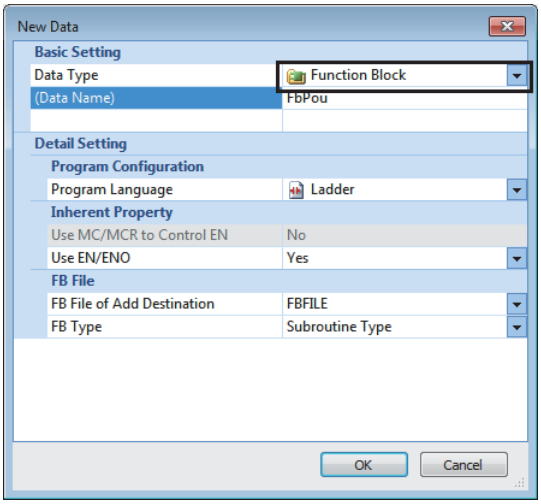
📖 Function Block Reference for the module used

■Creating function blocks/functions

User-defined function blocks/functions must be created before inserted to the program.



1. Right-click "FB/FUN" (1) on the Navigation window, and select "Add New Data" (2) to open the setting window.



2. To create a function block, set the data type to "Function Block" using the drop-down list. To create a function, set the data type to "Function". Then, click the [OK] button.

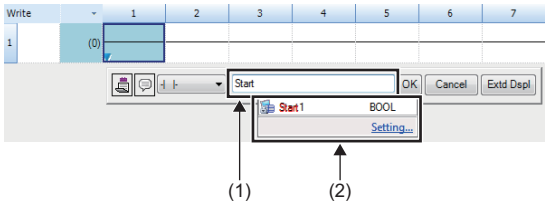
3. Program the operation of the function block or function in "ProgramBody".

Inserting POUs by key input

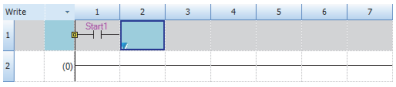
POUs can be inserted by key input.

Procedure

Inserting a normally open contact of "Start1" in the program example, following the procedure below.



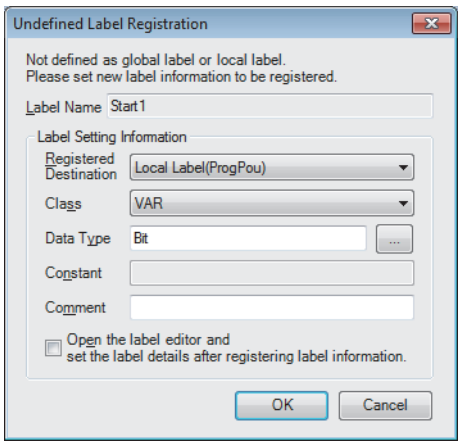
1. Click the insertion position on the ladder editor, and press F5.
2. Enter the name, "Start", in the entry field (1). Select "Start1" from the displayed list (2).



3. The normally open contact of "Start1" is inserted to the program.

Point

A new label can be registered during insertion. Enter the name of a new label on the ladder entry window, and click the [OK] button. Then, specify the registered destination, class, and data type of the label on the "Undefined Label Registration" window, and click the [OK] button.



4.10 Converting the Program

Determine the input ladder blocks.

Procedure

1. Select [Convert] ⇨ [Convert] on the menu bar.
2. When the conversion processing completes and the input ladder blocks are determined, the color of those ladder blocks changes from gray to white.

4.11 Saving the Project

Save the created project.

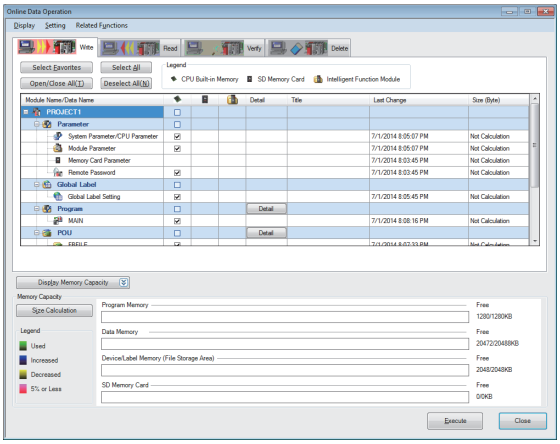
[Project] ⇨ [Save as]

4.12 Writing Data to the Programmable Controller

Write the set parameters and the created program to the CPU module.

[Online] ⇨ [Write to PLC]

Procedure



1. Select the system parameter file, CPU parameter file, module parameter file, and program file on the "Online Data Operation" window. If any function block is used, select the corresponding FB/FUN file as well.
2. Click the [Execute] button.
3. After the write processing completes, click the [Close] button.

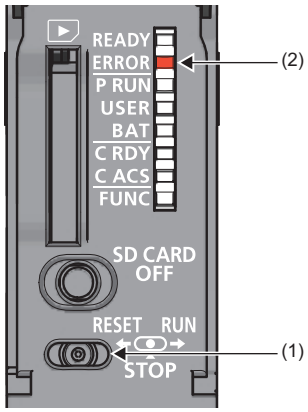
Point

- For the operation of the CPU module, the system parameter file, CPU parameter file, and program file must be written. For the operation of the I/O modules and intelligent function modules, the module parameter files and module extension parameter file must be written as well.
- When a parameter setting is changed, reset the CPU module. (☞ Page 51 Resetting the CPU Module)
- Use of the [Select Favorites] button enables users to easily select frequently-used files, such as the system parameter file, CPU parameter file, and program file. Register items as favorites on the window opened by selecting [Setting] ⇨ [Register Favorites Selection] on the menu bar.

4.13 Resetting the CPU Module

Reset the CPU module using the RUN/STOP/RESET switch located on the front of the CPU module.

Procedure



1. Set the RUN/STOP/RESET switch (1) to the RESET position for a second or longer.
2. Check that the ERROR LED (2) flashes for several times and turns off.
3. Set the switch back to the STOP position.

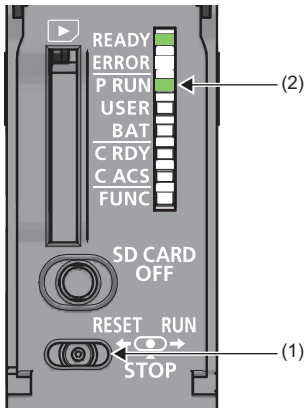
Point

Operate the RUN/STOP/RESET switch with your fingers. Use of a tool such as a screwdriver may damage the switch.

4.14 Executing the Program

Execute the program written to the programmable controller by using the RUN/STOP/RESET switch.

Procedure



1. Set the RUN/STOP/RESET switch (1) to the RUN position.
2. Check that the PROGRAM RUN LED (P RUN) (2) turns on.

4.15 Monitoring the Program

Monitor the program operation using the engineering tool.

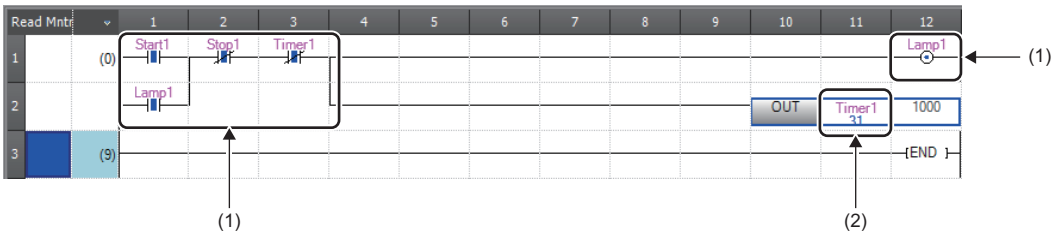
Monitoring on the monitor status bar

For the monitor status bar, refer to the following.

GX Works3 Operating Manual

Monitoring on the ladder editor

The on/off states of contacts and coils and the current values of word devices and labels can be monitored on the ladder editor.



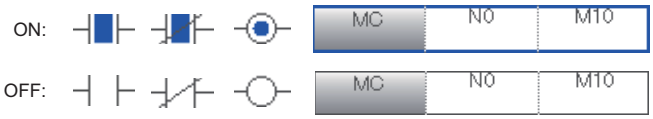
(1) The on/off states of the contacts and coils are displayed.
(2) The current value of the word/double word type data is displayed.

1. Select [Online] ⇨ [Monitor] ⇨ [Start Monitoring] on the menu bar.

2. Monitor the on/off states of contacts and coils and the current values of word devices and labels.

■On/off state display

The on/off states are displayed on the editor as follows:



■Changing the current value

To change the current value, select the cell on the ladder editor, and press Shift + double-click the cell or press Shift + Enter while the program is being monitored.

Point

The program can also be monitored on the Device/Buffer Memory Batch window or the Watch window. (GX Works3 Operating Manual)

5 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.

5.1 Overview

There are two ways to start up a redundant system.

Start-up procedure	Description
Starting up both systems simultaneously	Configure two 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 a redundant system without extension base units, follow this procedure to start up a control system first to perform control, and then start up a standby system to configure a redundant system. • Debugging before operation can be performed with only one system. • One system has failed and the redundant system is powered off before the failed module is replaced. In this case, only one system can be started up when the redundant system is powered on next time. In a redundant system with redundant extension base unit, the system that is started up first will be the control system.

Starting up both systems simultaneously

1. Installing a battery

Install a battery to the CPU module of each system. (☞ Page 38 Installing a Battery)

2. Inserting an extended SRAM cassette or an SD memory card

Insert an extended SRAM cassette or an SD memory card into the CPU module of each system as needed. (☞ Page 39 Inserting or Removing an Extended SRAM Cassette, Page 40 Inserting and Removing 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. (☞ Page 162 When an SD memory card is used)

3. Mounting modules and connecting cables

Mount modules on the base unit of each system, and connect cables. Use modules of the same model, and mount them on the same slots on each base unit. (📖 MELSEC iQ-R Module Configuration Manual, ☞ Page 62 Wiring)

4. Powering on the redundant system

Check the following on each system, and supply power to the main base unit and extension base units of both systems.

- A cable is correctly connected to the power supply.
- The power supply voltage is within the range of the specifications.
- The CPU module is in the STOP state.

Check that the following LEDs turn on after the system is powered on.

- Power supply module on the main base unit and extension base units: 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 to the CPU module

Start up the engineering tool installed on a personal computer. (☞ Page 64 Creating a Project)

Connect the personal computer to the CPU module of one system. (☞ Page 64 Connecting a Personal Computer and the CPU Module)

6. Initializing the CPU module

Initialize the CPU module using the engineering tool. (☞ Page 43 Initializing the CPU Module)

After initialing the CPU module of one system, connect the engineering tool to the CPU module of the other system. (☞ Page 64 Connecting a Personal Computer and the CPU Module)

Initialize the CPU module in the same way.

7. Setting parameters

Set system parameters, CPU parameters, and module parameters. (☞ Page 43 Setting Parameters)

- To execute the functions that access the SD memory card, set memory card parameters.
- When an intelligent function module is used in the system, set intelligent function module parameters.

Point

Users can set system parameters by loading the actual system configuration to the "Module Configuration" window 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. (☞ Page 46 Programming)

9. Writing the system A/B setting

Set the system A or B using the engineering tool. (☞ Page 65 Setting the System (System A or System B))

10. Writing data to the programmable controller

Write the set parameters and created program to the CPU modules of both systems using the engineering tool. (☞ Page 67 Writing Data to the Programmable Controller)

11. Resetting the CPU modules

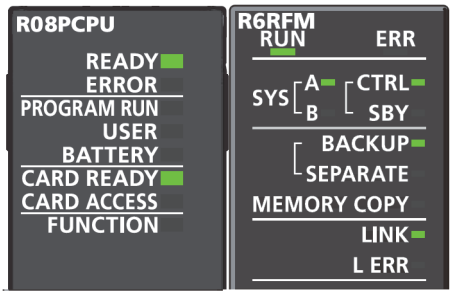
Restart both systems in either of the following ways.

- Powering off and on the system
- Resetting the CPU module

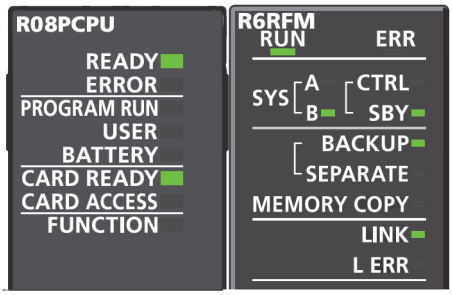
12. Checking the LEDs

Check that the LEDs of each CPU module and redundant function module are in the following states. The CARD READY LED status (on or off) depends on whether an SD memory card is inserted to the CPU module or not.

• LEDs of system A



• LEDs of system B



If an error has been detected, the following LEDs turn on. Identify the error using the engineering tool, and eliminate the error cause.

- CPU module: ERROR LED (☞ Page 545 LED status of the CPU module)
- Redundant function module: ERR LED, L ERR LED (☞ Page 549 When the ERR LED turns on or flashes, ☞ Page 549 When the L ERR LED turns on)

To start up the system when the data logging function is used, refer to the following.

☞ Page 61 Precautions when the data logging function is used

13. Checking the connection of the extension cables

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. Executing the 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 in the control system turns on.

Point

If the RUN/STOP/RESET switch of each CPU module is set to the RUN position while the CPU modules are powered on, a continuation error due to an operating status mismatch is detected in the CPU module of the standby system. Therefore, to start up both systems simultaneously, performing step 14 is recommended.

15. Monitoring the program

Check that the program operates normally on the engineering tool. (☞ Page 68 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. (☞ Page 38 Installing a Battery)

2. Inserting an extended SRAM cassette or an SD memory card

Insert an extended SRAM cassette or an SD memory card to the CPU module as needed. (☞ Page 39 Inserting or Removing an Extended SRAM Cassette, Page 40 Inserting and Removing 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. (☞ Page 162 When an SD memory card is used)

3. Mounting modules and connecting cables

Mount modules on the base unit and connect cables. (📖 MELSEC iQ-R Module Configuration Manual, ☞ Page 63 Power supply modules in a redundant system)

4. Powering on the control system

Check the following and supply power to the main base unit and extension base units.

- A cable is correctly connected to the power supply.
- The power supply voltage is within the range of the specifications.
- The CPU module is in the STOP state.

Check that the following LEDs turn on after the system is powered on.

- Power supply module on the main base unit and extension base units: 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 to the CPU module

Start up the engineering tool installed on a personal computer. (☞ Page 64 Creating a Project)
Connect the personal computer to the CPU module. (☞ Page 64 Connecting a Personal Computer and the CPU Module)

6. Initializing the CPU module

Initialize the CPU module using the engineering tool. (☞ Page 43 Initializing the CPU Module)

7. Setting parameters

Set system parameters, CPU parameters, and module parameters. (☞ Page 43 Setting Parameters)

- To execute the functions that access the SD memory card, set memory card parameters.
- When an intelligent function module is used in the system, set intelligent function module parameters.

Point

Users can set system parameters by loading the actual system configuration to the "Module Configuration" window 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. (☞ Page 46 Programming)

9. Writing the system A/B setting

Set the system A or B using the engineering tool. (☞ Page 65 Setting the System (System A or System B))

10. Writing data to the programmable controller

Write the set parameters and created programs to the CPU module using the engineering tool. (☞ Page 67 Writing Data to the Programmable Controller)

11. Resetting the CPU module

Restart the control system in either of the following ways.

- Powering off and on the system
- Resetting the CPU module

12. Starting up the system (in a redundant system without extension base units)

Perform the following operation using the engineering tool to start up the system as the 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 429 Redundant System Operation Setting)

- Switch operation: Set the RUN/STOP/RESET switch of the CPU module to RUN → STOP → RUN.
- Input (X) operation: Turn on the input (X) set in the parameters.

13. Starting up the system (in a redundant system with redundant extension base unit)

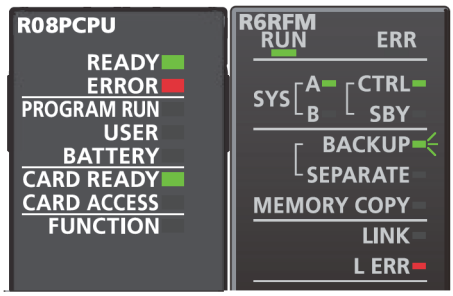
The system starts up as the control system. (The system does not wait for the other system to start and starts up as the control system.)

Point

- If the RUN/STOP/RESET switch is set to the RUN position, the CPU module enters to the RUN state while the standby system does not start. To start control of the redundant system when both systems have started up, start up the control system with the RUN/STOP/RESET switch set to the STOP position.
- The CPU module does not start up if the extension cables between extension base units are incorrectly connected. Ensure that the cables are correctly connected.

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 the own system is set as system A in the system settings. The CARD READY LED status (on or off) depends on whether an SD memory card is inserted 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. In addition, the BACKUP LED of the redundant function module flashes because a system switching disable cause exists.
- 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 cables

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

To start up the systems one by one, complete the start-up process (steps 1 to 15 above) of one system first. Then, start up the other system. During start-up of a system, the system cannot perform tracking communications. For this reason, a stop error may occur in the CPU module of the system that was started up later. In this case, restart the system in which the CPU module where a stop error has occurred. (When automatic recovery is set, the system can be automatically restarted.) (☞ Page 436 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 61 Precautions when the data logging function is used

16. Executing the 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 operates normally on the engineering tool. (☞ Page 68 Monitoring the Program)

Starting up the standby system

Start up the standby system while the control system keeps operating.

Follow the steps 1 to 6 in the start-up procedure for the control system to start up the standby system. (☞ Page 57 Starting up the control system)

Point

Check that the two systems are exactly the same (modules on base units, their model names, and insertion status of the extended SRAM cassette or the SD memory card) before starting up the standby system.

1. Powering off the system

Power off the standby system.

2. Connecting the redundant function modules

Connect the redundant function modules of both systems with tracking cables. (☞ Page 62 Redundant function modules)

3. Starting up the system

Power on the standby system.

Connect the personal computer to the control system, and execute the memory copy function (from the control system to the standby system) using the engineering tool by following the procedure below. (☞ Page 409 Memory Copy from Control System to Standby System)

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

Select "Memory Copy" and click the [Execute] button.

Upon completion of the memory copy function, the MEMORY COPY LED of the redundant function module in the standby system turns on.

After that, reset the CPU module of the standby system and set it to the RUN state.

Point

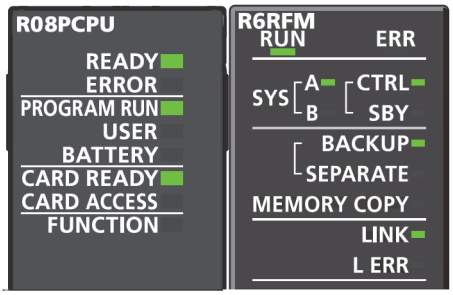
When "Auto Memory Copy Setting" has been set to "Enable" in the CPU parameter in advance, the system can be started up only by the following operation. (☞ Page 429 Redundant System Operation Setting)

- Set the RUN/STOP/RESET switch of the CPU module in the standby system to the RUN position and power on the system.

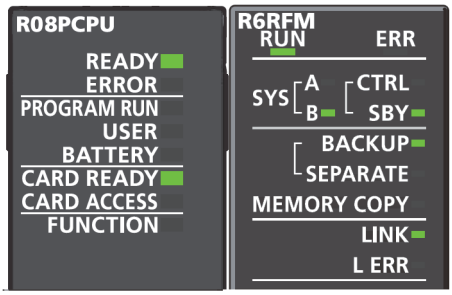
4. Checking the LEDs

Check that the LEDs of each CPU module and redundant function module are in the following states. The following figures show the LED status when the control system has been started up as system A. The CARD READY LED status (on or off) depends on whether an SD memory card is inserted to the CPU module or not.

• LEDs of system A



• LEDs of system B



5. Clearing errors

If the ERROR LED of the CPU module in the control system is on, clear the error using the engineering tool. (☞ Page 208 Error Clear)

Point

When "Watching Standby System Setting" of the CPU parameter was set to "Disable" in the start-up process of the control system, the ERROR LED may be off. (☞ Page 429 Redundant System Operation Setting)

6. Monitoring the program

Check that the program operates normally on the engineering tool. (☞ Page 68 Monitoring the Program)

Precautions when the data logging function is used

If the systems are switched while data logging fails (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 has started, the control system does not fail in data logging collection (processing overflow).*1
- After system switching, the new control system does not fail in data logging collection (processing overflow) and the CPU module operates without errors.*1

*1 Whether a processing overflow has occurred or not can be checked in the special register (Number of processing overflow occurrences) corresponding to the data logging No. (☞ Page 682 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 (📖 CPU Module Logging Configuration Tool Version 1 Operating Manual)
- "Scan Time Monitoring Time (WDT) Setting" of the CPU parameter (☞ Page 200 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.

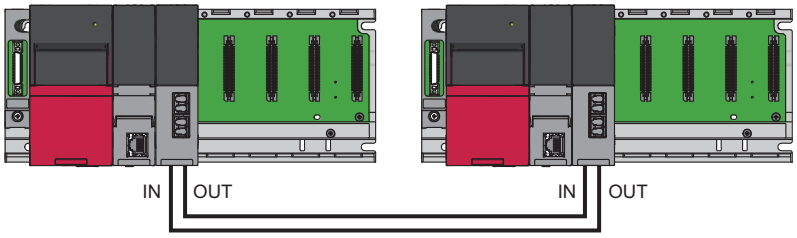
5.2 Wiring

Redundant function modules

This section describes the wiring to 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.

☞ Page 78 Redundant Function Module

Connecting/disconnecting tracking cables

■Connection procedure

1. Pay attention to the orientation of the tracking cable connector and insert the connector 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 system A and the OUT connector of system B, and connect the OUT connector of system A and the IN connector of 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.

Redundant extension base units

For the wiring to redundant extension base units, refer to the following.

MELSEC iQ-R Module Configuration Manual

Power supply modules in a redundant system

This section describes the wiring to power supply modules.

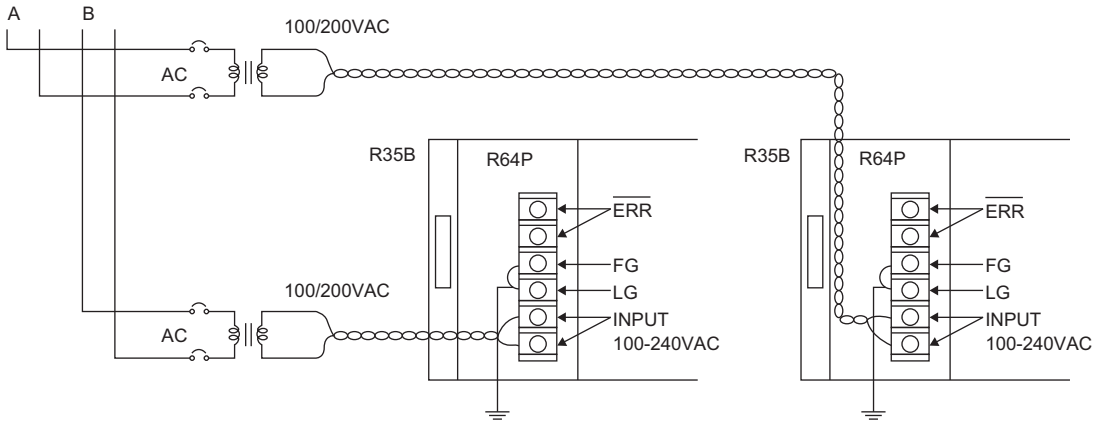
The terminal block of each power supply module has a screw size of M4. Wire cables to the terminal block with the applicable solderless terminal RAV1.25-4 or RAV2-4.

Separately supply power to the system A and B.

Wiring example

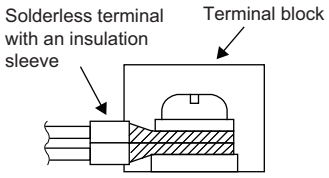
The following figure shows a wiring example of power cables to each main base unit and ground cables.

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.

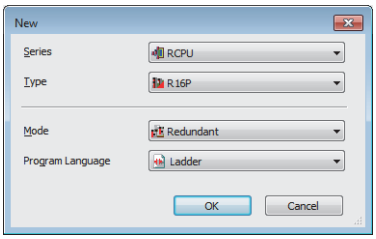


- Ground the LG and FG terminals after short-circuiting them. Failure to do so may be susceptible to the noise. The LG terminal has a half potential of the input voltage.
- When two redundant power supply modules operate in parallel as a redundant power supply system, it is recommended to connect the one redundant power supply module to an AC power supply and the other one to an uninterruptible power supply (UPS).

5.3 Creating a Project

Start up the engineering tool and create a project.

[Project] ⇨ [New]



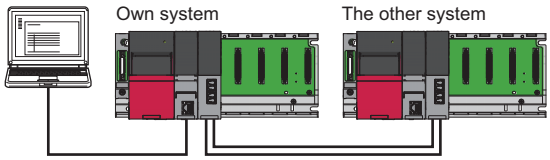
1. Select the CPU module to be used for "Type". Select "Redundant" for "Mode". Select a programming language to be used for "Programming Language" and click the [OK] button.

5.4 Connecting a Personal Computer and the CPU Module

Connect a personal computer on which the engineering tool has been installed to the CPU module of one system.

Connection procedure

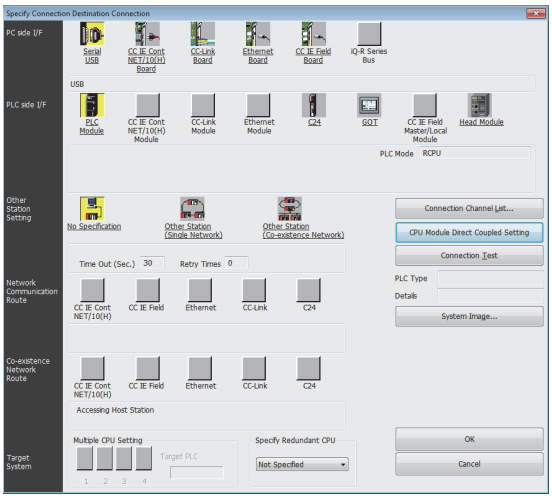
The following describes the procedure for directly connecting the personal computer to the CPU module of one system.

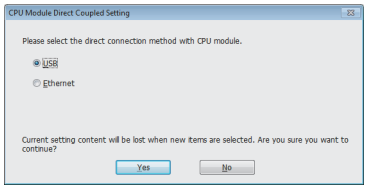


1. Connect the personal computer to the CPU module using a USB cable or an Ethernet cable. This CPU module becomes the own system.

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 to the CPU module.

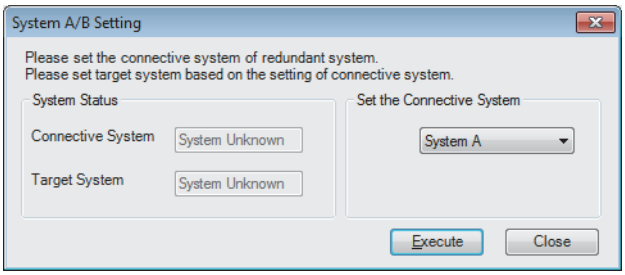
When connecting the personal computer to the CPU module with a USB cable for the first time, install a USB driver. (GX Works3 Operating Manual)

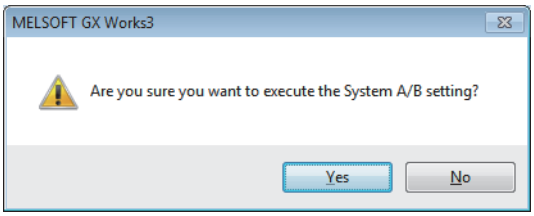
5.5 Setting the System (System A or System B)

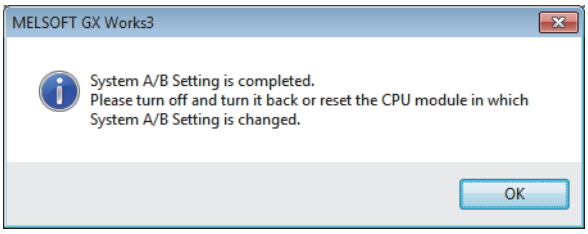
Set the system A or B using the engineering tool and write the system settings to the CPU module.

[Online] ⇨ [Redundant PLC Operation] ⇨ [System A/B Setting]

Setting procedure







1. Set the CPU module to the STOP state.
2. In the "System A/B Setting" window, select system A or system B for the own system.
3. Click the [Execute] button.
4. Click the [Yes] button on the window shown 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 setting, 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 setting is set to the CPU module that is not running in redundant mode, powering off and on the system or resetting the CPU module turns off the SYS A LED or SYS B LED of the redundant function module. Proceed to the next step (Writing Data to the Programmable Controller).

Point

Besides the method that uses the engineering tool, the system A/B setting can be automatically set by the systems themselves.

Setting one system as system A and bringing it into a tracking communicable state can automatically set the other system as system B, and vice versa.

In the system that is automatically set as system A or system B, the original system A/B setting will be overwritten. This can be checked in the event history. (Event code: 00700)

- When a system with no system A/B setting is connected to the system that is set as system A, the system with no system A/B setting is automatically set as system B. (This event is not recorded in the event history.)
- When a system that is set as system A is connected to another system that is also set as system A, the system that is powered off and on or whose CPU module is reset is automatically set as system B. (This event is recorded in the event history.)

Checking method

Check the LEDs of each redundant function module to check the system status.

Setting of the engineering tool	LED of the redundant function module
System A	
System B	
When the setting is switched from "System A" to "System B"	 When the setting is switched to "System B", the SYS B LED flashes. Powering off and on the system or resetting the CPU module turns off the SYS A LED and turns on the SYS B LED.

Point

Users can also check the system status on the engineering tool. (GX Works3 Operating Manual)

- System monitor
- "System A/B Setting" window
- Monitor status bar

Precautions

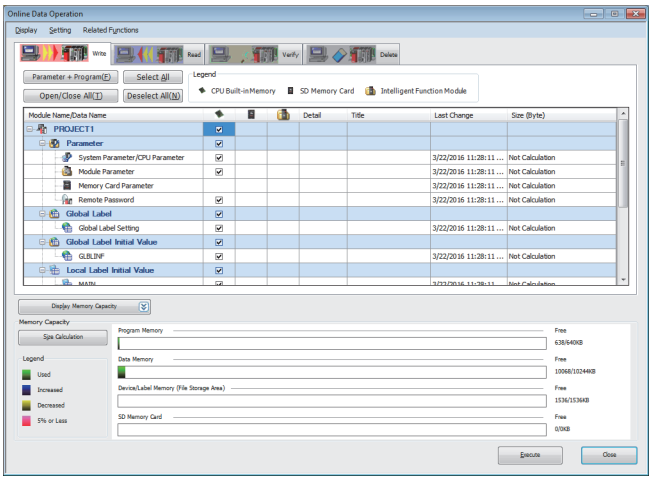
- Do not power off the system or reset the CPU module during the system A/B setting. The system A/B setting may not be reflected properly. If not reflected properly, set the system again.
- For the system A/B setting, use the CPU module that is connected with a USB cable or an Ethernet cable. If the connection destination has been changed on the engineering tool, the system A/B setting cannot be set.
- The system A/B information cannot be deleted once it is set. The information can only be changed.
- If the system A/B setting has not been set or the same system A/B setting has been set when both systems are started up simultaneously, a stop error occurs at the point when tracking communications are established. Set the system A/B setting correctly in both systems.
- If a system with no system A/B setting is started up first, a stop error occurs at the point when tracking communications are established. Set the system A/B setting correctly in both systems. In a redundant system with redundant extension base unit, if the system A/B setting has not been set in both systems, the modules on the extension base units does not start.
- Do not connect system A and system A or system B and system B with tracking cables while running. If connected, a continuation error will occur.

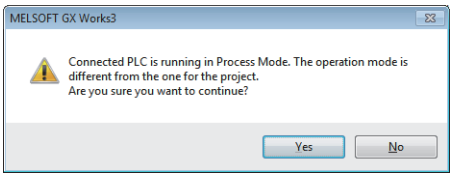
5.6 Writing Data to the Programmable Controller

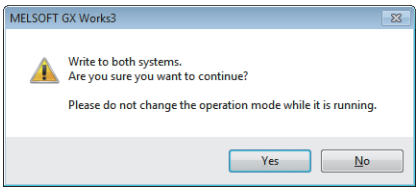
Write the set parameters and created programs to the CPU module.

[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 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 redundant system is in 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 settings 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. (☞ Page 51 Resetting the CPU Module)
- 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

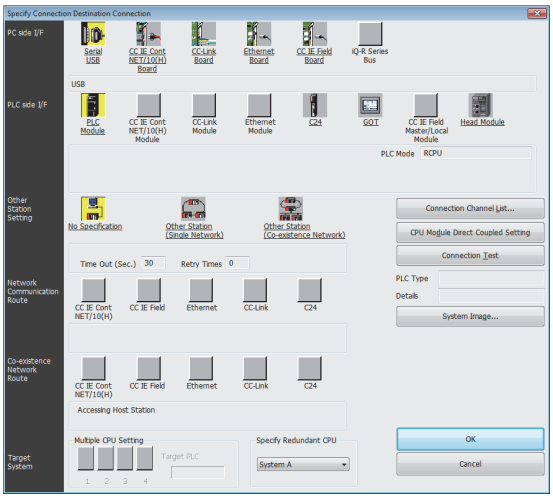
Do not write data to both systems during the initial processing (or while the READY LED is on) after the other system is powered off and on or reset. Writing to the programmable controller may fail due to the operating status mismatch.

5.7 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

PART 3 SYSTEM CONFIGURATION

This part consists of the following chapter.

6 SYSTEM CONFIGURATION

6 SYSTEM CONFIGURATION

For system configurations using the MELSEC iQ-R series modules, applicable combinations of CPU modules and the other modules, the number of mountable modules, installation, and wiring, refer to the following.

MELSEC iQ-R Module Configuration Manual

Handling of an SD memory card in a redundant system

If the data in the SD memory card become corrupted due to power-off, CPU module reset, or card removal during access to the SD memory card, the SD memory card diagnostics may start at the startup of the CPU module.

If the SD memory card diagnostics take a long time due to a large number of files/folders, it may takes more than ten minutes to restore the redundant system (the time for the both systems to start and the system switching ready). Therefore, if the prolonged time may effect the system, do not use an SD memory card.

For the handling of SD memory cards and the necessary precautions, refer to the following.

MELSEC iQ-R Module Configuration Manual

MEMO

MEMO

PART 4 SPECIFICATIONS

This part consists of the following chapter.

7 PERFORMANCE SPECIFICATIONS

7 PERFORMANCE SPECIFICATIONS

7.1 CPU Module

This section describes the specifications of the CPU module.

Hardware specifications

Item			R08PCPU	R16PCPU	R32PCPU	R120PCPU
Operation control method			Stored program cyclic operation			
I/O control mode			Refresh mode (The direct access input/output is available by specifying the direct access input/output (DX, DY).)			
Instruction processing time	LD instruction		0.98ns			
	MOV instruction		1.96ns			
Instruction processing time (structured text language)	IF		1.96ns			
	CASE		1.96ns			
	FOR		1.96ns			
Memory capacity	Program capacity		80K steps (320K bytes)	160K steps (640K bytes)	320K steps (1280K bytes)	1200K steps (4800K bytes)
	Program memory		320K bytes	640K bytes	1280K bytes	4800K bytes
	SD memory card		Differs depending on the SD memory card used. (SD/SDHC memory card: 32G bytes maximum)			
	Device/label memory	Total	1188K bytes	1720K bytes	2316K bytes	3380K bytes
		Device area[*1]	80K bytes			
		Label area[*1]	80K bytes	100K bytes	180K bytes	220K bytes
		Latch label area[*1]	4K bytes		8K bytes	
		File storage area[*1]	1024K bytes	1536K bytes	2048K bytes	3072K bytes
	Signal flow memory	Area for programs	10K bytes	20K bytes	40K bytes	150K bytes
		Area for function blocks	256K bytes[*5]			
	Data memory		5M bytes	10M bytes	20M bytes	40M bytes
	CPU buffer memory		1072K bytes (536K words) (including the fixed scan communication area (24K words))			
	Refresh memory		2048K bytes[*2]			
Number of storable files[*6]	Program memory (P: number of program files, FB: number of FB files)		P: 252, FB: 128 (One FB file can store 64 function blocks.)			
	Device/label memory (file storage area)		324 (with or without an extended SRAM cassette)[*3]			
	Data memory		512[*4]			
	SD memory card		• NZ1MEM-2GBSD: 256[*4] • NZ1MEM-4GBSD, NZ1MEM-8GBSD, NZ1MEM-16GBSD: 32767[*4]			
Number of storable folders[*6]	Data memory		512[*4]			
	SD memory card		• NZ1MEM-2GBSD: 256[*4] • NZ1MEM-4GBSD, NZ1MEM-8GBSD, NZ1MEM-16GBSD: 32767[*4]			
USB port			USB2.0 High Speed (miniB) × 1			
Ethernet port			Refer to the following. MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)			
Clock function			Year, month, date, hour, minute, second, and day of the week (automatic leap year adjustment) -1.00 to +1.00s/d at 0 to 55℃			
Allowable momentary power failure time			The time differs depending on the power supply module used. (MELSEC iQ-R Module Configuration Manual)			
Internal current consumption (5VDC)			0.76A			
External dimensions	Height		106mm (Base unit mounting side: 98mm)			
	Width		27.8mm			
	Depth		110mm			

Item	R08PCPU	R16PCPU	R32PCPU	R120PCPU
Weight	0.20kg			

*1 The capacity of device area, label area, latch label area, and file storage area can be changed in parameter. The capacity of the device/label memory can be increased by inserting an extended SRAM cassette. (☞ Page 142 Device/label memory area setting)

*2 This is the total capacity of the device area and module label area.

*3 System files are included.

*4 The number indicates the number of files and folders (including system files and system folders) can be created in the root directory on the condition that the number of characters in the file or folder name is 13 or less. In a subdirectory, up to 32767 folders can be created. Note that the number of storable files and folders will decrease if many folders with a long name, more than 13 characters (including an extension), are created.

*5 For the CPU module with the firmware version "06" or earlier, the memory capacity is 20K bytes.

6 The following characters cannot be used for file or folder names: A space, "%+,/:;<=>?[\]|'{}&~@^.

Programming specifications

Item			R08PCPU	R16PCPU	R32PCPU	R120PCPU
Programming language			• Ladder diagram (LD) • Sequential function chart (SFC)[*5*7*8] • Structured text (ST) • Function block diagram (FBD/LD)			
Programming supporting function			• Function block (FB) • Label programming (system/local/global)			
Program operation	Execution type		• Initial execution type • Scan execution type • Fixed scan execution type • Event execution type • Standby type			
	Type of interrupt		• Interrupt using the internal timer (I28 to I31) • High-speed internal timer interrupt 1 (I49) • High-speed internal timer interrupt 2 (I48) • Interrupt by a module • Inter-module synchronous interrupt (I44)[*6] • Multiple CPU synchronous interrupt (I45)[*6]			
Number of executable programs			252			
Number of FB files			128			
Tact performance	Constant scan		0.2 to 2000ms (The value can be set in increments of 0.1ms.)			
	Fixed scan interrupt	Interrupt using the internal timer (I28 to I31)	0.5 to 1000ms (The value can be set in increments of 0.5ms.)			
		High-speed internal timer interrupt 1 (I49)	0.05 to 1000ms (The value can be set in increments of 0.05ms.)			
		High-speed internal timer interrupt 2 (I48)	0.05 to 1000ms (The value can be set in increments of 0.05ms.)			
		Inter-module synchronous interrupt (I44)[*6]	0.1 to 10.00ms (The value can be set in increments of 0.05ms.)			
		Multiple CPU synchronous interrupt (I45)[*6]	0.1 to 10.00ms (The value can be set in increments of 0.05ms.)			
Timer performance	Low-speed timer		1 to 1000ms (Default: 100ms)			
	High-speed timer		0.01 to 100ms (Default: 10ms)			
	Long timer		0.001 to 1000ms (Default: 0.001ms)			
Number of I/O points			4096 points			
Number of user device points (default)	Input (X)		12288 points (fixed)			
	Output (Y)		12288 points (fixed)			
	Internal relay (M)		12288 points (user-changeable)[*1]			
	Link relay (B)		8192 points (user-changeable)[*1]			
	Link special relay (SB)		2048 points (user-changeable)[*1]			
	Annunciator (F)		2048 points (user-changeable)[*1]			
	Edge relay (V)		2048 points (user-changeable)[*1]			
	Step relay (S)[*4*5]		0 points (user-changeable)[*1]			
	Timer device	Timer (T)	1024 points (user-changeable)[*1]			
		Long timer (LT)	1024 points (user-changeable)[*1]			
	Retentive timer device	Retentive timer (ST)	0 points (user-changeable)[*1]			
		Long retentive timer (LST)	0 points (user-changeable)[*1]			
	Counter device	Counter (C)	512 points (user-changeable)[*1]			
		Long counter (LC)	512 points (user-changeable)[*1]			
	Data register (D)		18432 points (user-changeable)[*1]			
	Link register (W)		8192 points (user-changeable)[*1]			
	Link special register (SW)		2048 points (user-changeable)[*1]			
	Latch relay (L)		8192 points (user-changeable)[*1]			

Item		R08PCPU	R16PCPU	R32PCPU	R120PCPU
Number of system device points	Special relay (SM)	4096 points (fixed)			
	Special register (SD)	4096 points (fixed)			
	Function input (FX)	16 points (fixed)			
	Function output (FY)	16 points (fixed)			
	Function register (FD)	5 points × 4 words (fixed)			
Number of file register points (default)	File register (R/ZR)	0 points (user-changeable)*1			
Number of index register points (default)	Index register (Z)	20 points (user-changeable, up to 24 points)			
	Long index register (LZ)	2 points (user-changeable, up to 12 points)			
Number of pointer points	Pointer (P) (global/local) (default)	8192 points (user-changeable, up to 16384 points)			16384 points (user-changeable, up to 32768 points)
	Interrupt pointer (I)	1024 points (fixed)			
Number of link direct device points	Link input (J□\X□)	16384 points maximum*2			
	Link output (J□\Y□)	16384 points maximum*2			
	Link relay (J□\B□)	32768 points maximum*2			
	Link register (J□\W□)	131072 points maximum*2			
	Link special relay (J□\SB□)	512 points maximum*2			
	Link special register (J□\SW□)	512 points maximum*2			
Number of module access device points	Intelligent function module device (U□\G□)	268435456 points (Max.)*2			
Number of CPU buffer memory access device points	Buffer memory (U3E□\G□)	524288 points			
	Fixed scan communication area in the buffer memory (U3E□\HG□)*6	12288 points maximum*3			
Number of refresh data register points (default)	Refresh data register (RD)	524288 points (1048576 points maximum)			
Number of nesting points	Nesting (N)	15 points			
Number of other device points	SFC block device (BL)*4*5	320 points			
	SFC transition device (TR)*4*5	0 points (Used only as device comments.)			

*1 For the setting range, refer to the following.
☞ Page 462 Device Setting

*2 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.

*3 The maximum point differs depending on the parameter setting (Multiple CPU Setting).

*4 These devices are used in SFC programs. For details on SFC programs, refer to the following.
MELSEC iQ-R Programming Manual (Program Design)

*5 Before using, check the versions of the CPU module and engineering tool used. (☞ Page 762 Added and Enhanced Functions)

*6 Redundant mode does not support these items.

*7 In redundant mode, the SFC program cannot be set to be executed in both systems.

*8 In redundant mode, the POFF(P) and PSCAN(P) instructions cannot be used for the SFC program.

7.2 Extended SRAM Cassette

This section describes the performance specifications of the extended SRAM cassette.

Item		NZ2MC-2MBSE (ECC-compatible)	NZ2MC-8MBSE (ECC-compatible)
Capacity		2M bytes	8M bytes
Number of insertions and removals		Limited to 50 times	
External dimensions	Height	49mm	
	Width	32mm	
	Depth	18.5mm	
Weight		20g	

7.3 Redundant Function Module

This section describes the specifications of the redundant function module.

Item		R6RFM
Tracking cable	Cable specifications	An optical fiber cable compliant with the following standards (multimode optical fiber (GI)) • IEEE 802.3 (1000BASE-SX) • IEC 60793-2-10 Types A1a.1
	Maximum cable length	550m
Optical fiber specifications		Standard: IEEE802.3, IEC 60793-2-10 (Types A1a.1) Outside diameter of the core/clad: 50μm/125μm Transmission loss: 3.5dB/km or lower [λ=850nm] Transmission band: 500MHz·km or more [λ=850nm]
Connector specifications		Duplex LC connector Standard: IEC 61754-20 (Type LC connector) Connection loss: 0.3dB or lower Polished surface: PC (Physical Contact) polishing
Laser class (IEC 60825-1: 2014/JIS C6802: 2014)		Class 1 laser product
Number of occupied I/O points		32 points*1
Internal current consumption (5VDC)		0.88A
External dimensions	Height	106mm (Base unit mounting side: 98mm)
	Width	27.8mm
	Depth	110mm
Weight		0.18kg

*1 The use of all the I/O signals is prohibited because they are used by the system.

Point

Optical fiber cables with connectors are available from Mitsubishi Electric System & Service Co., Ltd. (Catalogs of the optical fiber cables are also available.)

• Type: Multimode optical fiber (GI)

• Model: QG series

In addition, on-site connector polishing, terminal assembly, and fusion splicing are available. Please consult Mitsubishi Electric System & Service Co., Ltd.

PART 5 CPU MODULE OPERATION

This part consists of the following chapters.

8 RUNNING A PROGRAM

9 CPU MODULE OPERATION PROCESSING

10 MEMORY CONFIGURATION OF THE CPU MODULE

11 BASIC CONCEPT OF REDUNDANT SYSTEM

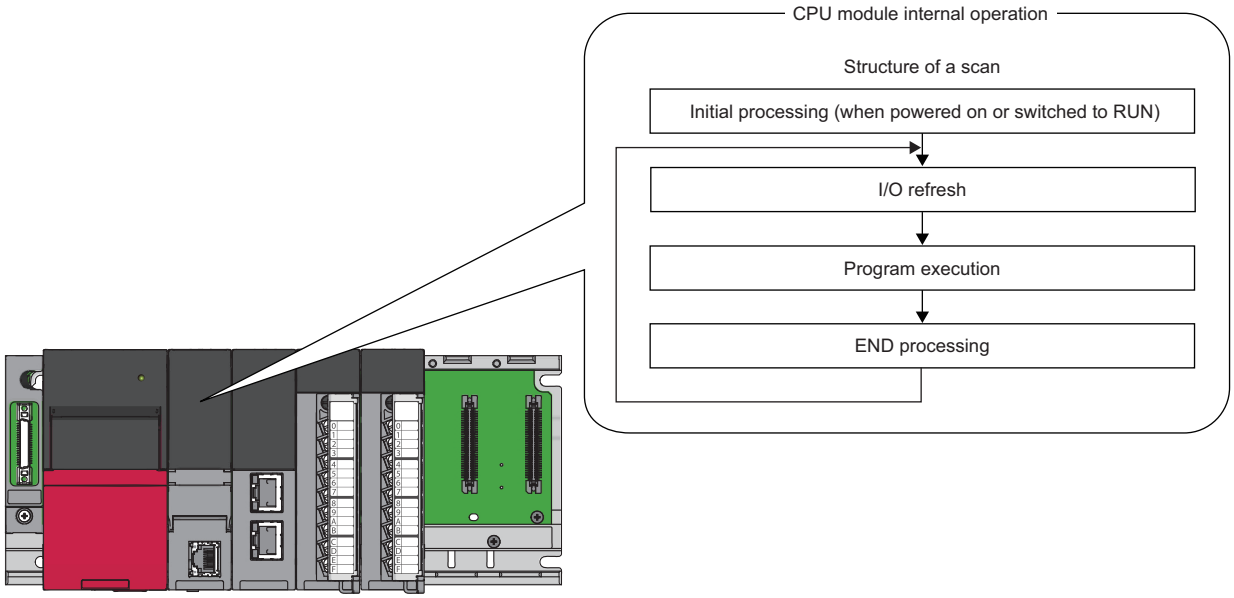
8 RUNNING A PROGRAM

8.1 Scan Configuration

The following shows the scan configuration of the CPU module.

In process mode

The following shows the scan configuration of the CPU module in process mode.



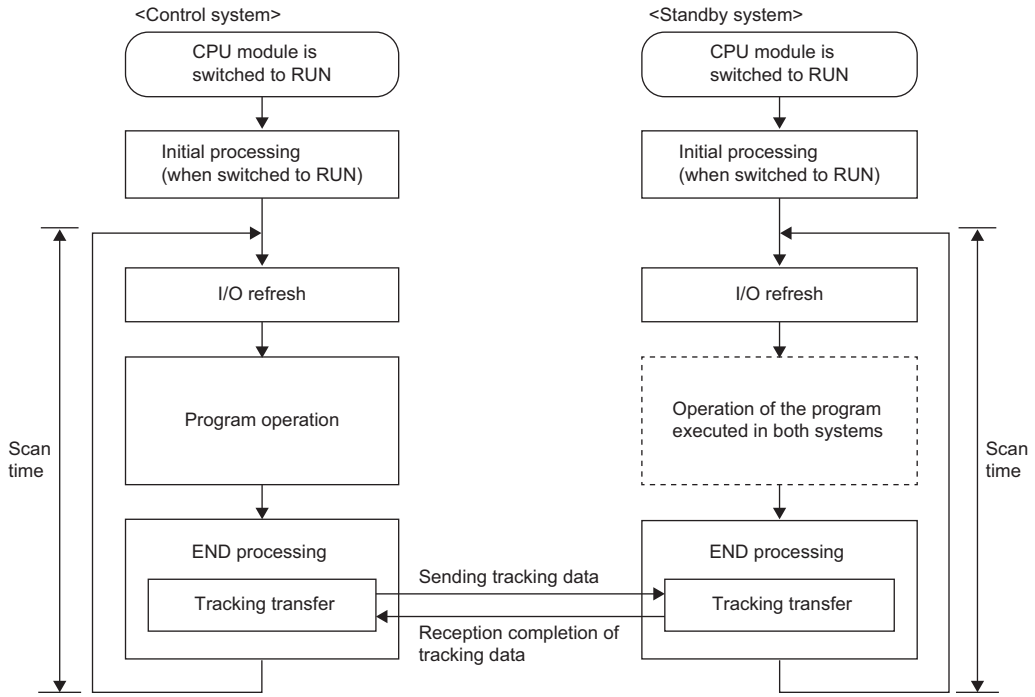
In redundant mode

This section describes the scan configurations of the CPU modules in a redundant system.

In a redundant system, tracking transfer is performed in the END processing. (☞ Page 391 Tracking Transfer)

However, in a redundant system with redundant extension base unit, tracking transfer is performed before the program operation. (The order of processing is reversed.)

The following are the scan configurations of the CPU modules when both systems are simultaneously started up in backup mode.



Processing differs depending on the system (control system or standby system) and the operation mode (backup mode of separate mode).

○: Performed, ×: Not performed

Processing	Backup mode		Separate mode	
	Control system	Standby system	Control system	Standby system
Initial processing (when powered on or 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 429 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 422 Program Execution in Both Systems)

*3 This processing is not performed when the operation mode is switched from the backup mode to the separate mode.

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, refer to the following.
☞ Page 534 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 114 Subroutine Program)

END processing

The CPU module performs the following processing.

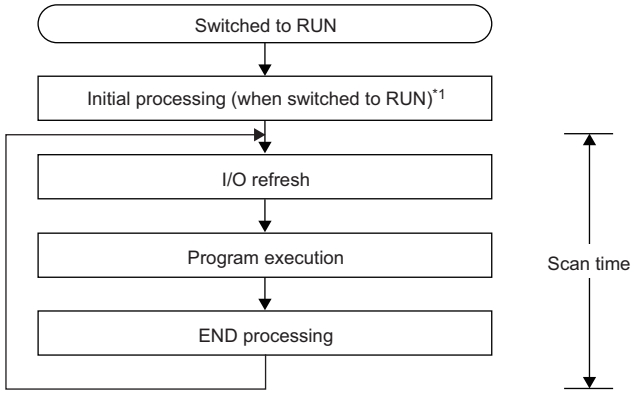
- Network module link refresh
- Intelligent function module refresh
- Instruction end processing (including dedicated instruction for the module)
- Device latch processing
- Service processing such as read and write of devices, labels, and program access files (☞ Page 750 Target List and Operation Details of the Device/Label Access Service Processing Setting)
- Watchdog timer reset (☞ Page 200 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")
- 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.

8.2 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 200 Scan time monitoring time setting)

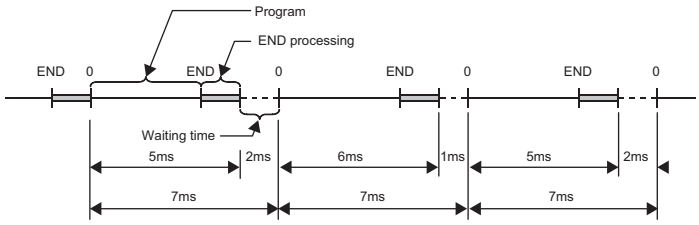
■Precautions on the initial scan time monitoring time

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

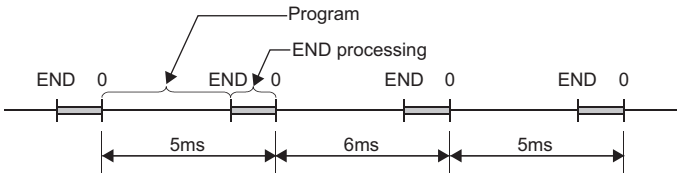
Constant scan

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

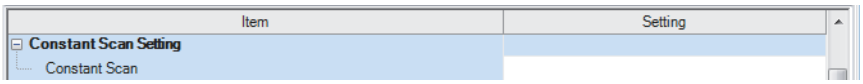


Setting constant scan

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.	0.2 to 2000ms (unit: 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 86 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
- In redundant mode, when the CPU module is powered off, a hardware failure has occurred, or a tracking cable has a failure in the standby system, 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 739 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 208 Error Clear)
- In redundant mode, when the systems are switched in separate mode, the scan time of the control system will increase by the increase of the standby system scan time.
- In redundant mode, in a program executed in both systems, the constant scan function is invalid for the standby system in backup mode.

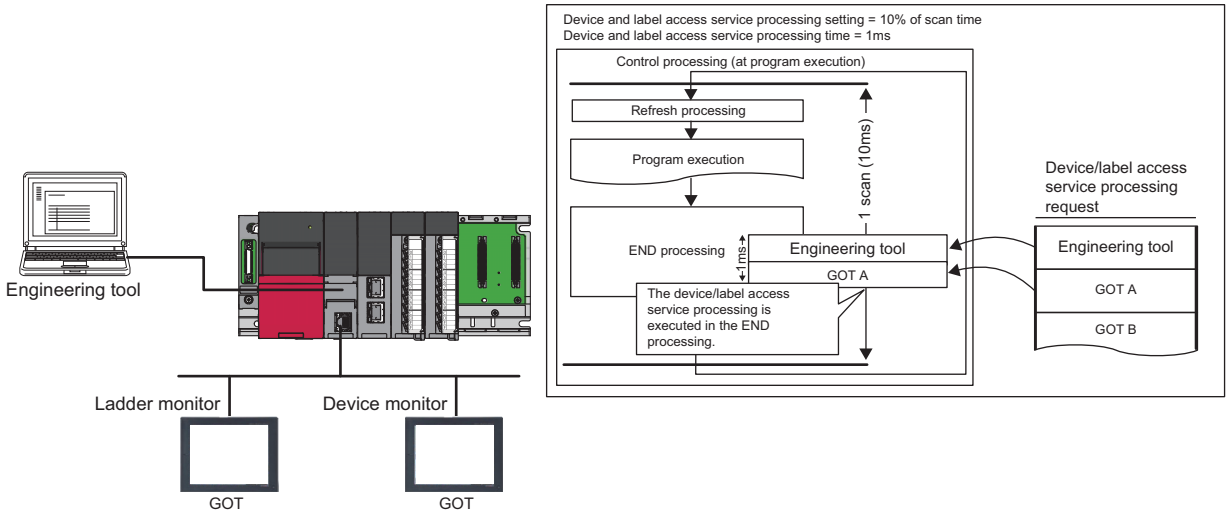
Device/label access service processing setting

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

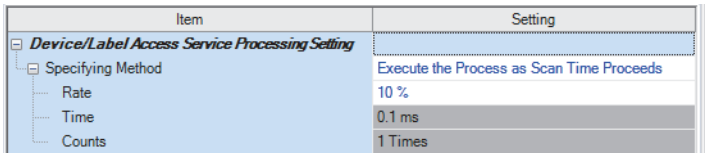
For functions to which the device/label access service processing setting can be applied, refer to the list targeted for device/label access service processing setting. (☞ Page 750 Target list)

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	Select 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	Set 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	Set 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	Set 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 90 When "Execute END Processing between Programs" is enabled)

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

☞ Page 752 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
- Data logging 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]

*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

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

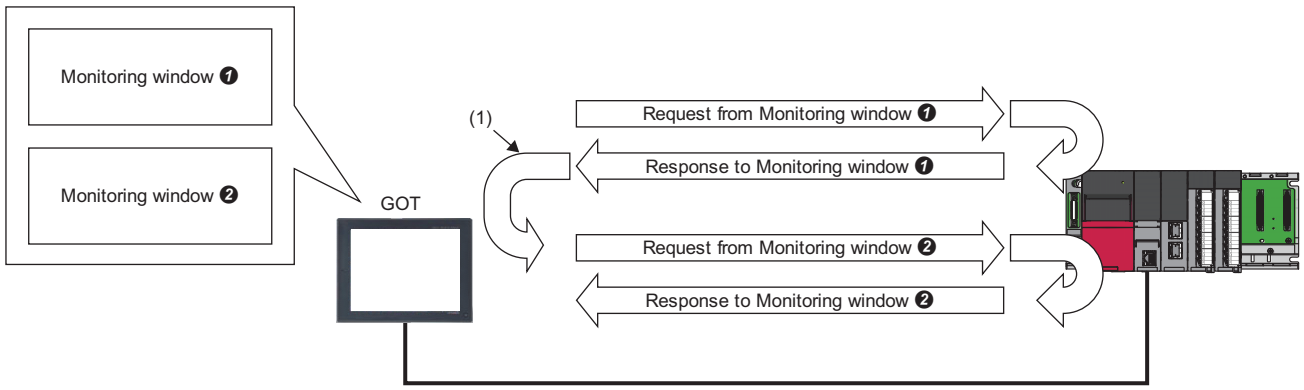
Before using the device/label access service processing constant wait function, check the versions of the CPU module and engineering tool used. (☞ Page 762 Added and Enhanced Functions)

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.

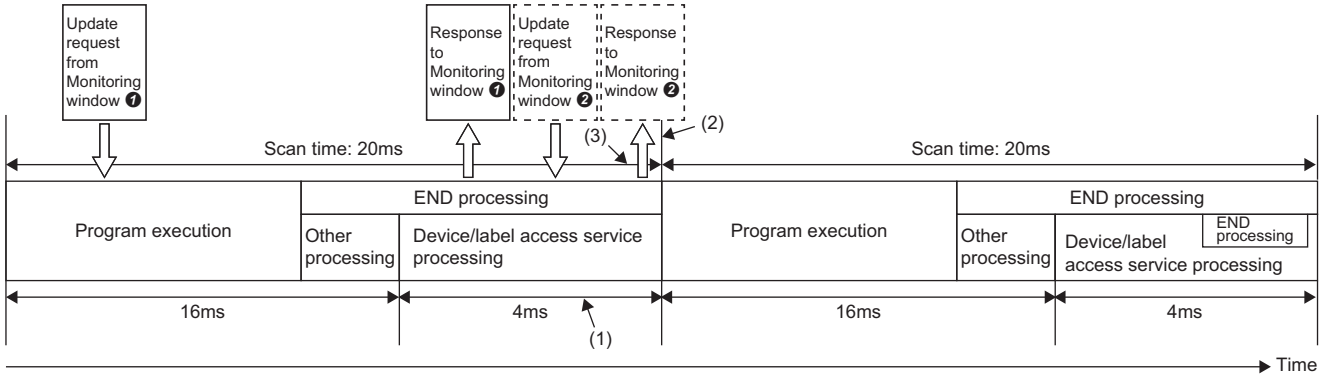
Depending on the GOT, when updating multiple monitor windows, the GOT repeats the action of sending a request (a device/label read request) to the CPU module for each window and waiting for a request (a device/label current value).

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.

In redundant mode, this function is enabled only in the CPU module of the control system in backup mode. In separate mode, this item is enabled for the CPU modules of both systems.

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 88 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.

■In redundant mode

- In 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.
- In separate mode, set SM315 (Service processing constant wait setting flag) and SD315 (Service processing constant wait status setting) of each CPU module.

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.)

8.3 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 95 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 113 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 98 Direct mode)

Differences between refresh mode and direct mode

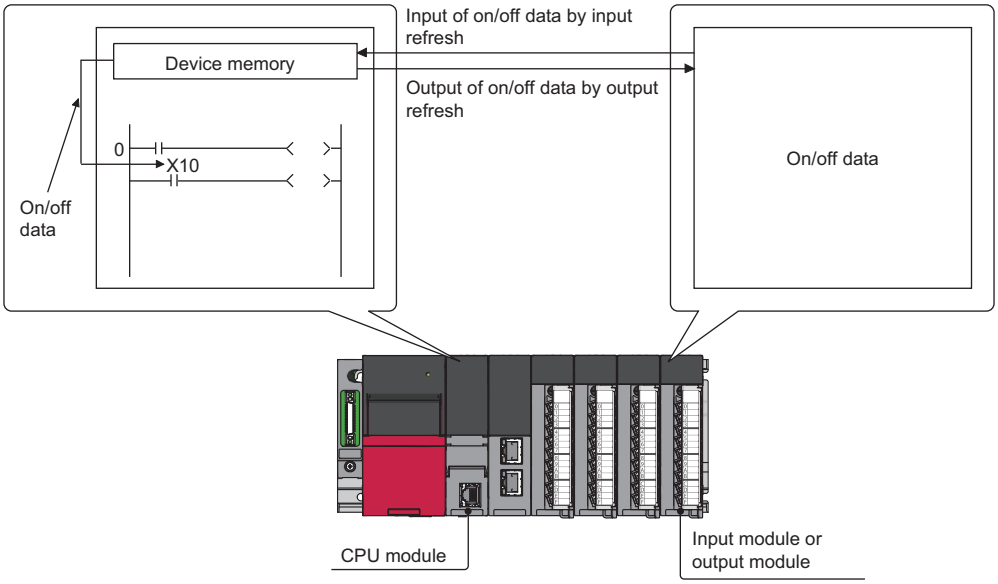
The direct mode directly accesses I/O modules at execution of an instruction. Therefore, data is input faster than when it is input in refresh mode. Processing time required for each instruction, however, takes longer. The following table lists the availability of the refresh mode and the direct mode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

Refresh mode

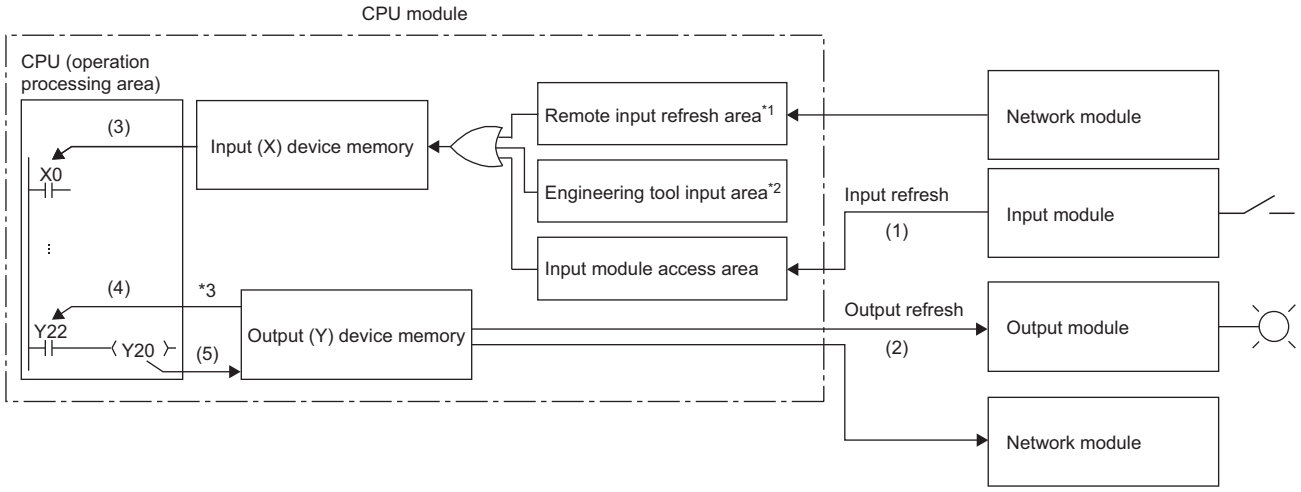
The CPU module performs I/O processing collectively at a specified timing.

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, 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.
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.

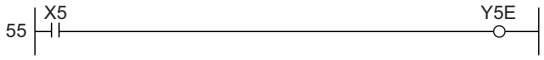
■Output

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).

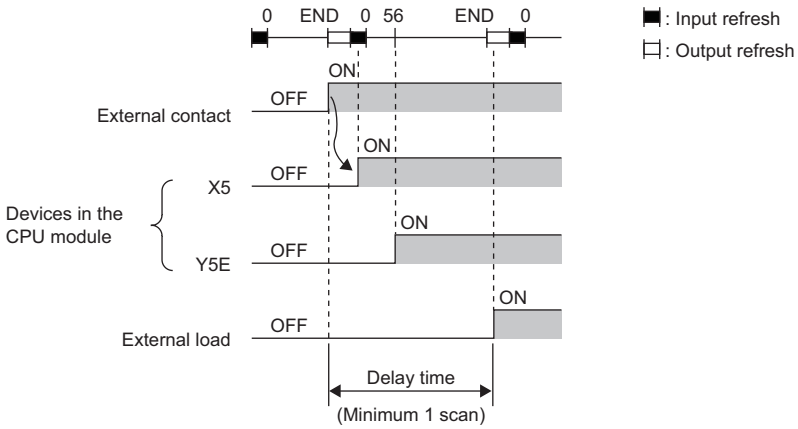
Response delay

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.

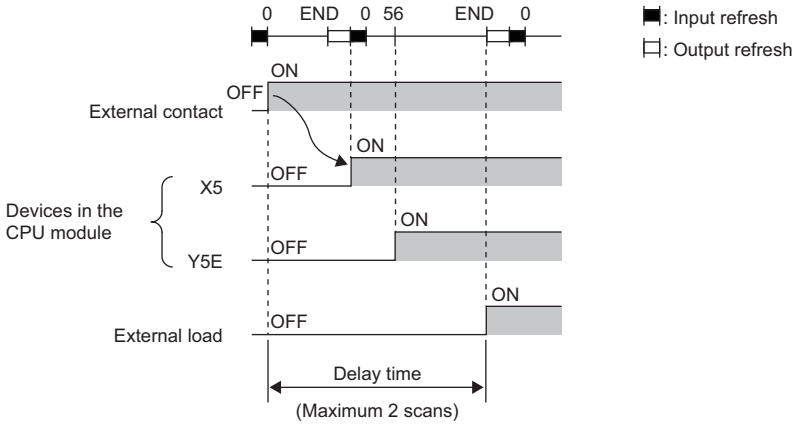
[Example] A program that turns on the output Y5E when the input X5 turns on



• Y5E turns on the earliest



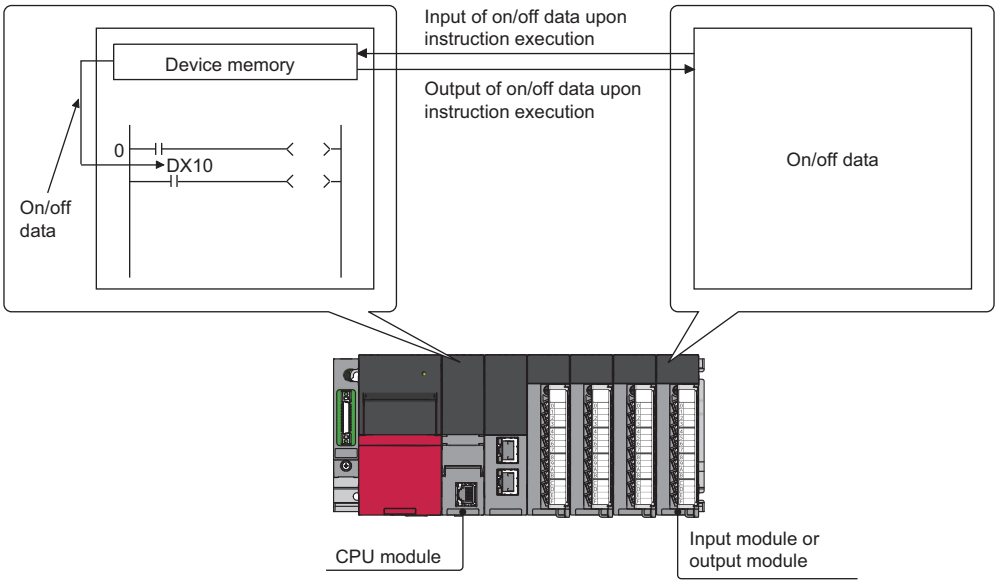
• Y5E turns on the latest



8

Direct mode

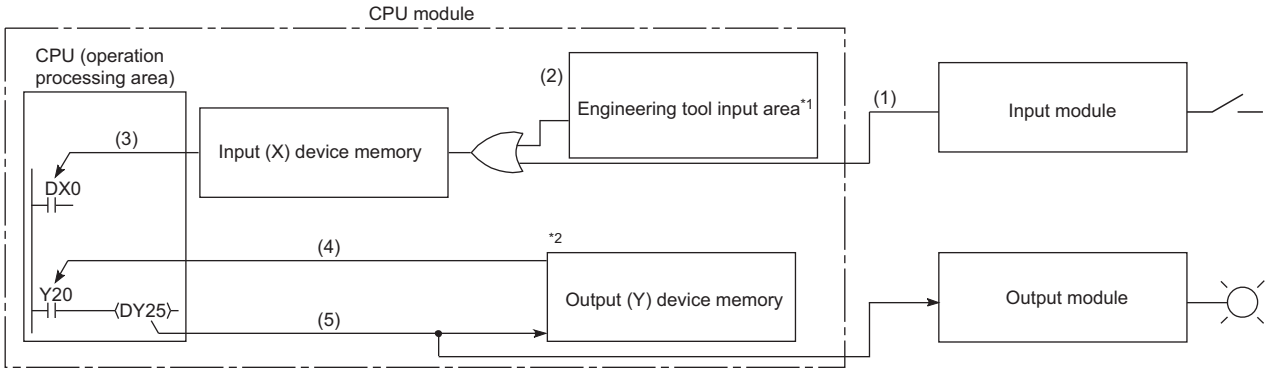
The CPU module performs I/O processing when each instruction is executed in a program.



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

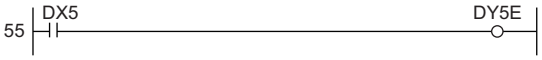
*2 Data in the output (Y) device memory can be turned on or off by the following:
- Test operation of an engineering tool
- Writing data from the network module
- Writing data from an external device using SLMP

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.

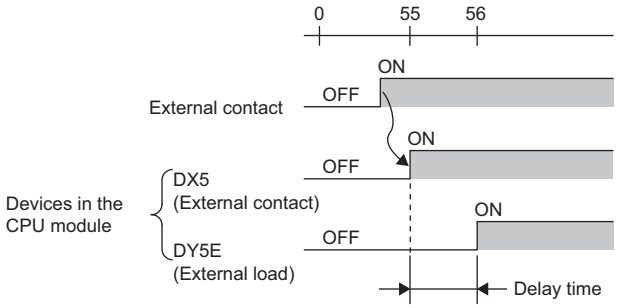
Response delay

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.

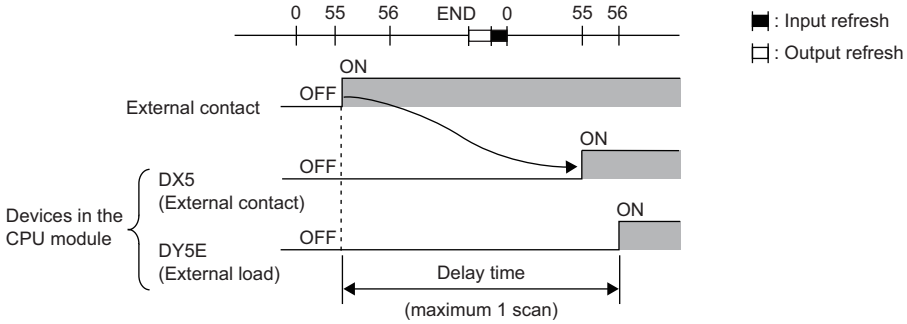
[Example] A program that turns on the output DY5E when the input DX5 turns on



- Y5E turns on the earliest

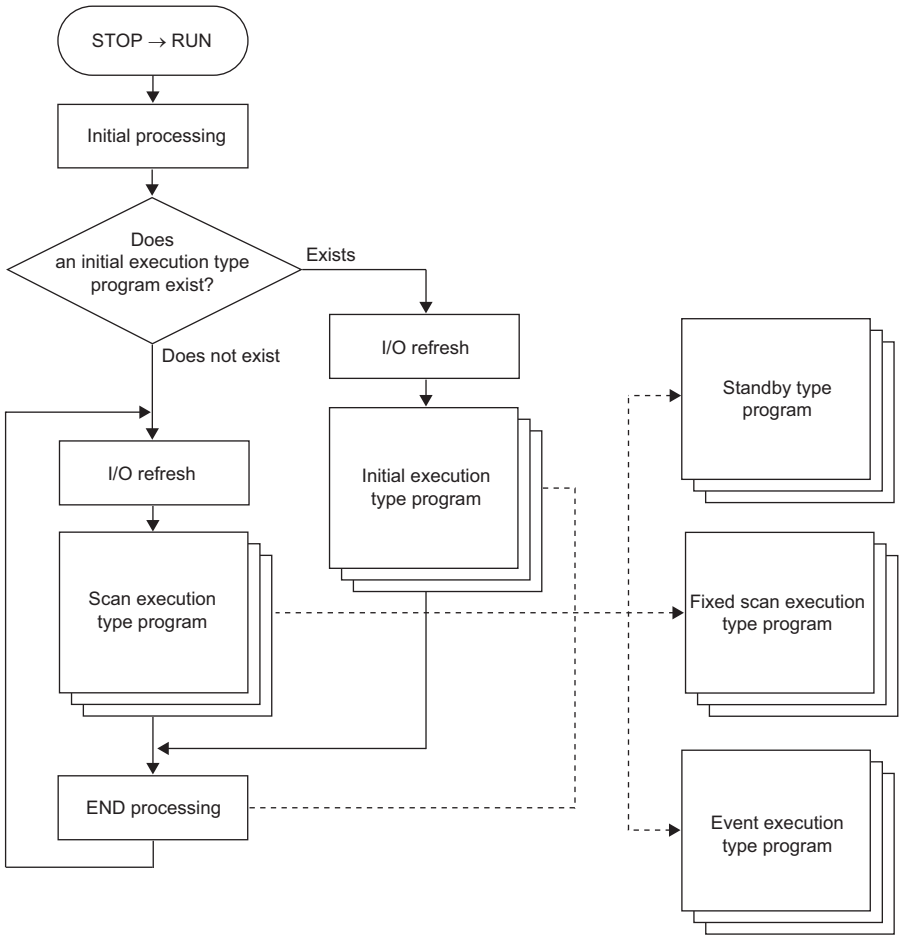


- Y5E turns on the latest



8.4 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 101 Program Execution Type, ☞ Page 112 Execution type change).



Point

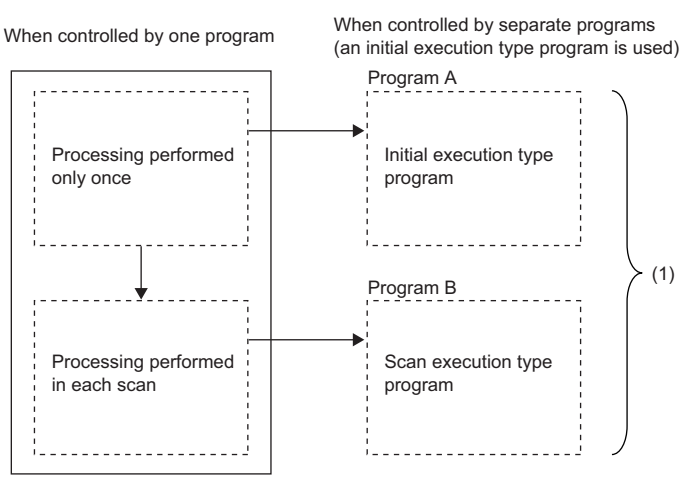
Programs with the same execution type are executed according to the execution sequence settings.

8.5 Program Execution Type

Set the execution condition of the program. (☞ Page 112 Execution type change)

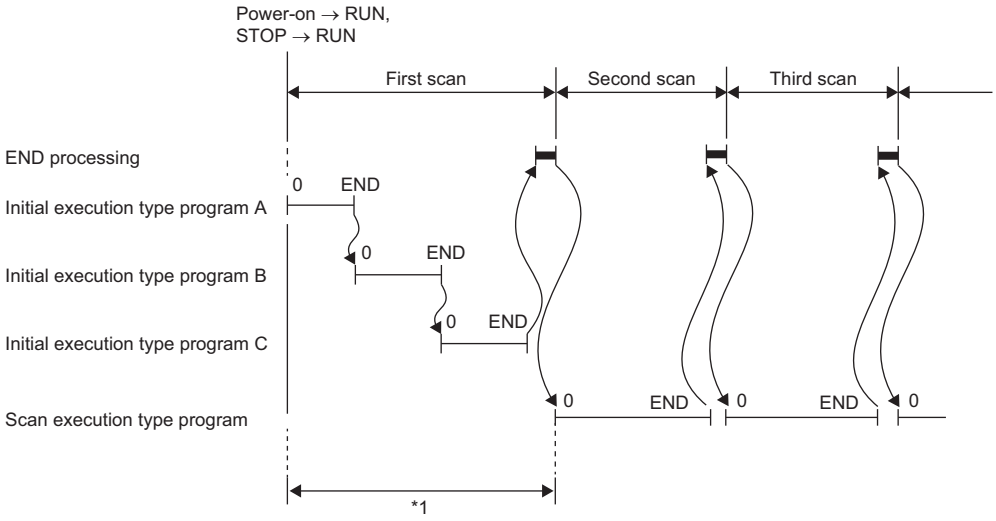
Initial execution type program

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 84 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

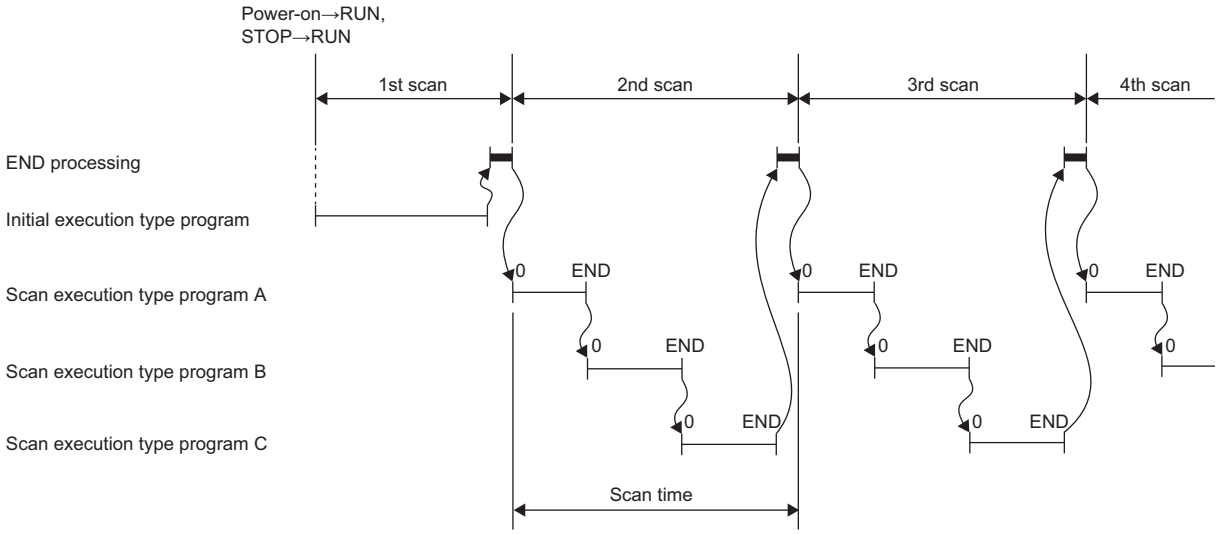
In the initial execution type program, do not use an instruction which requires several scans to complete (i.e., an instruction with a completion device).

Ex.

e.g. SEND and RECV instructions

Scan execution type program

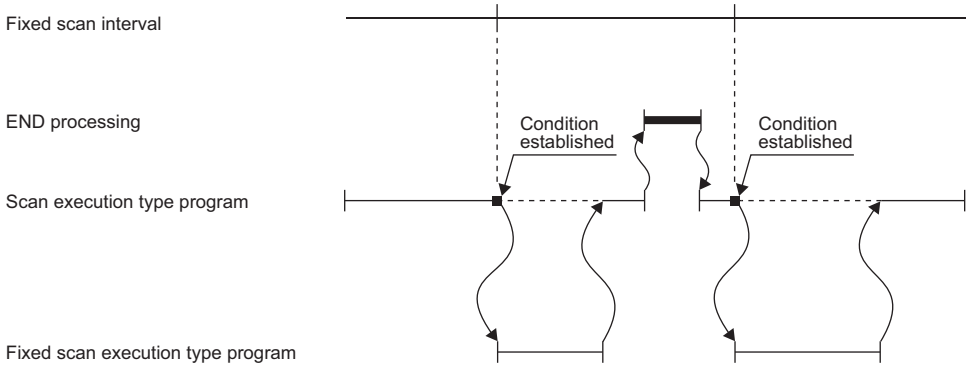
Scan execution type program is executed only once per every scan starting from the scan following the scan in which the initial execution type program was executed.



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.

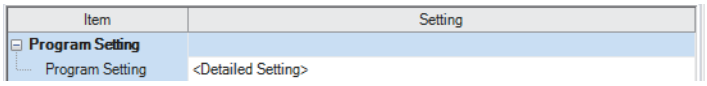
Fixed scan interval setting

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

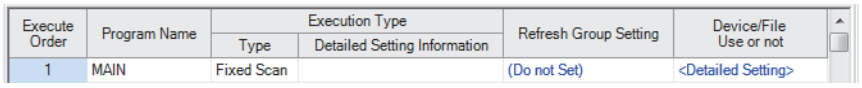
[CPU Parameter] ⇨ [Program Setting]

Operating procedure

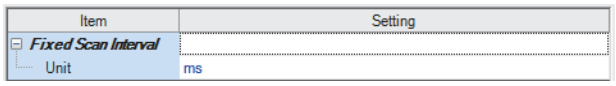
"Program Setting" window



"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

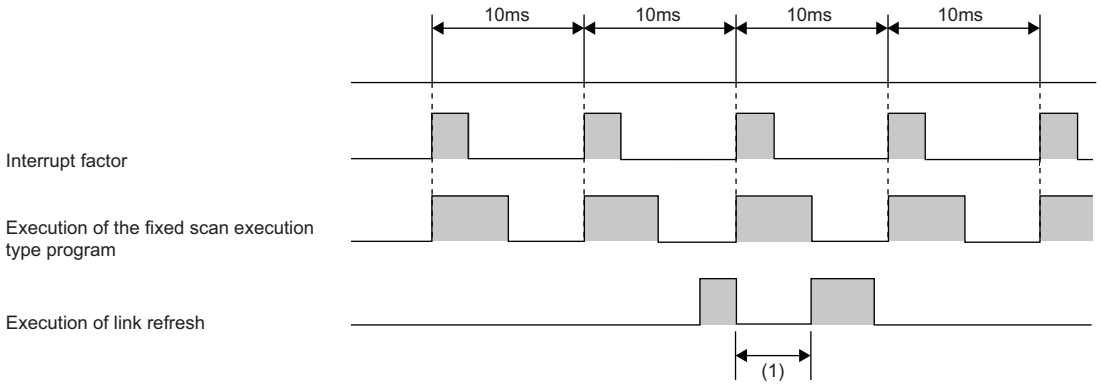
The program is executed according to the fixed scan execution mode setting.

■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

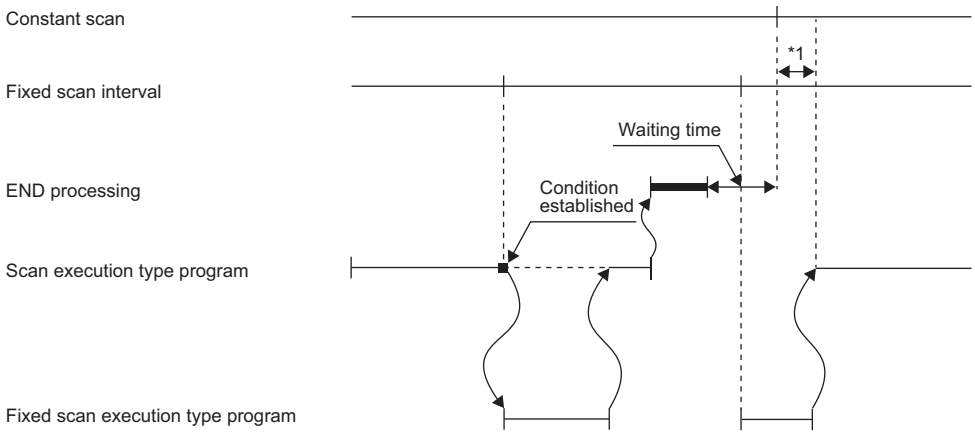
The link refresh is suspended and the fixed scan execution type program is executed. Even while the station-based block data assurance is enabled for cyclic data during refresh of such links as CC-Link IE Field Network, if the fixed scan execution type program uses a device specified as the refresh target, 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 129 Multiple interrupt function)

Processing when the fixed scan execution type program starts

The same processing as when the interrupt program starts. (☞ Page 126 Processing at interrupt program startup)

Fixed scan execution mode

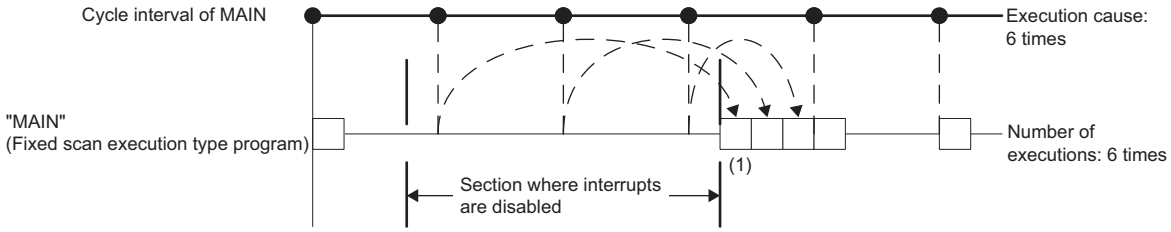
For fixed scan interrupts (I28 to I31, I48, I49) triggered by the fixed scan execution type program or the internal timer of the CPU module, this mode specifies the program execution operation that is performed when more than one interrupt occurs. (☞ Page 116 Operation upon occurrence of an interrupt 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

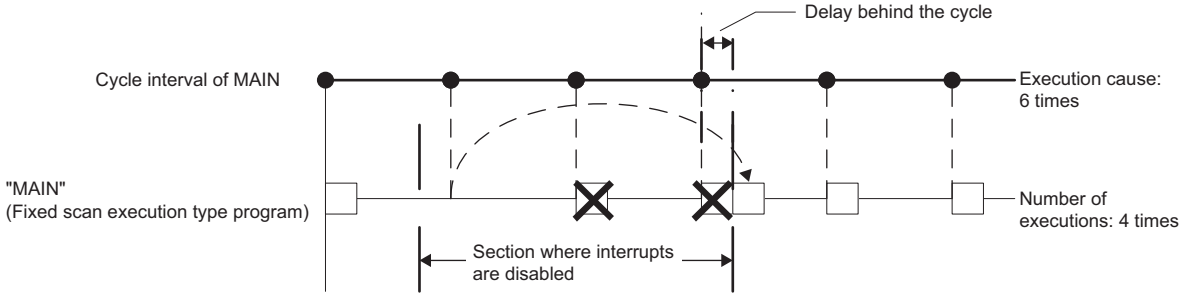
The program is executed for all the pending number of executions so that it can be executed the same number of times as execution causes.



(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 106 Delay limit value behind a cycle) Note that, even if more than one pending execution exists, only one execution is performed.

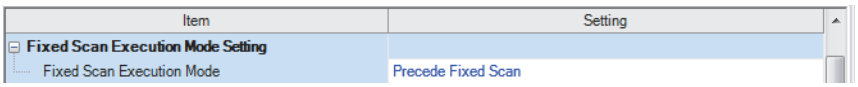


■Fixed scan execution mode setting

Use the fixed scan execution mode setting.

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

Window

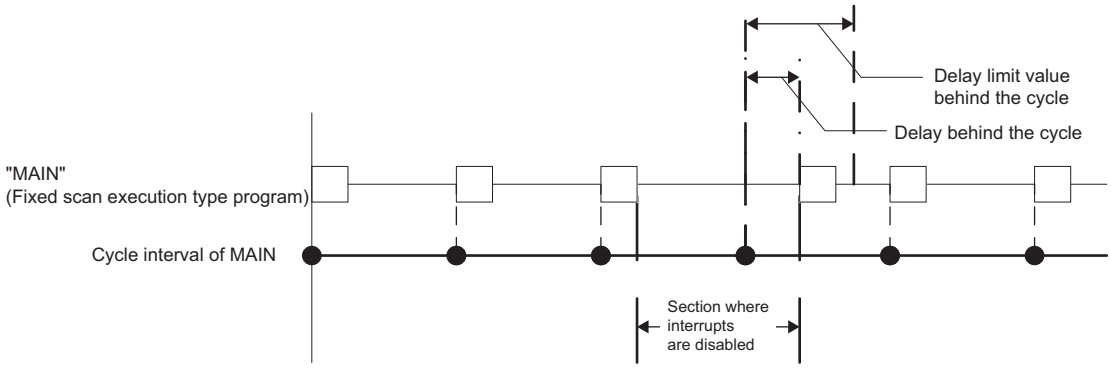


Displayed items

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

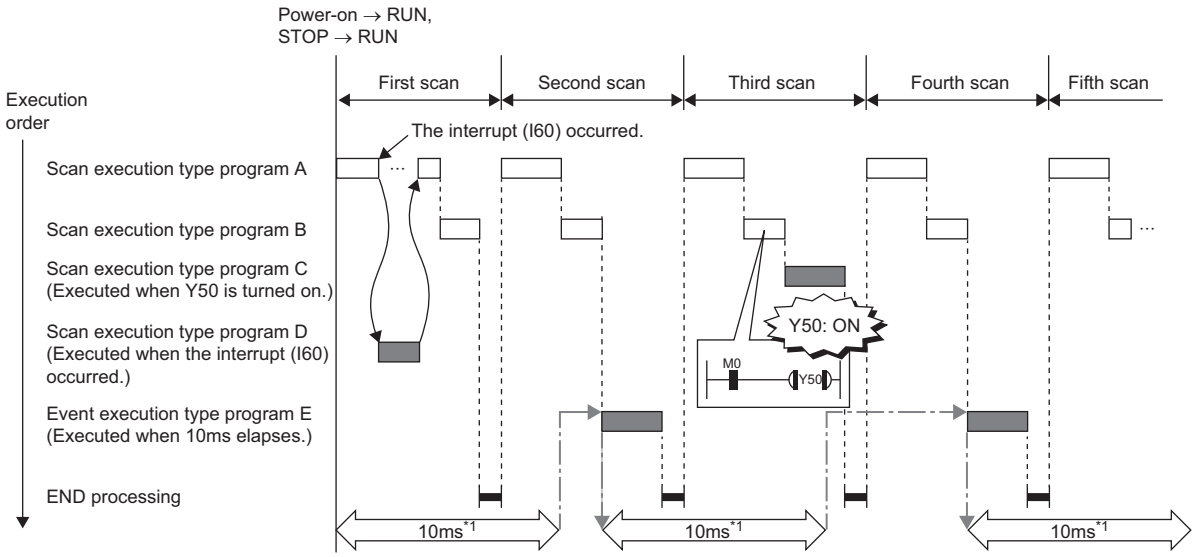
The value is the greatest common divisor of the fixed scan intervals of all the fixed scan execution type programs. For example, if there are fixed scan execution type programs with 2ms, 4ms, and 12ms scan periods, the delay limit value behind a cycle is 2ms.

• 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 107 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.



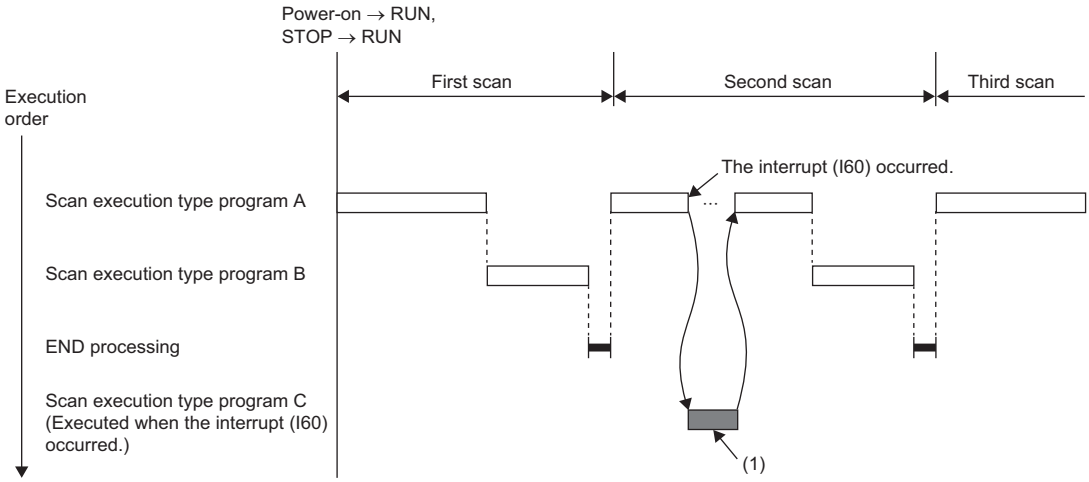
*1 Measurement of elapsed time is 10ms or more because it is determined depending on the scan time.

Trigger type

The following lists the triggers for the event execution type programs. (☞ Page 110 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.

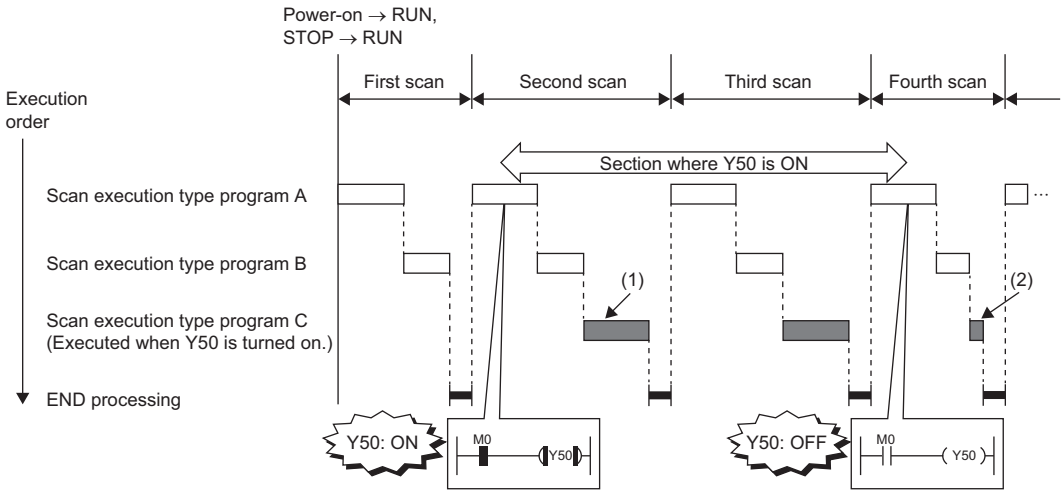
Interrupt pointers (I) which can be specified are I0 to I15, I28 to I31, I44, I45, I48, I49, I50 to I1023.

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 116 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).

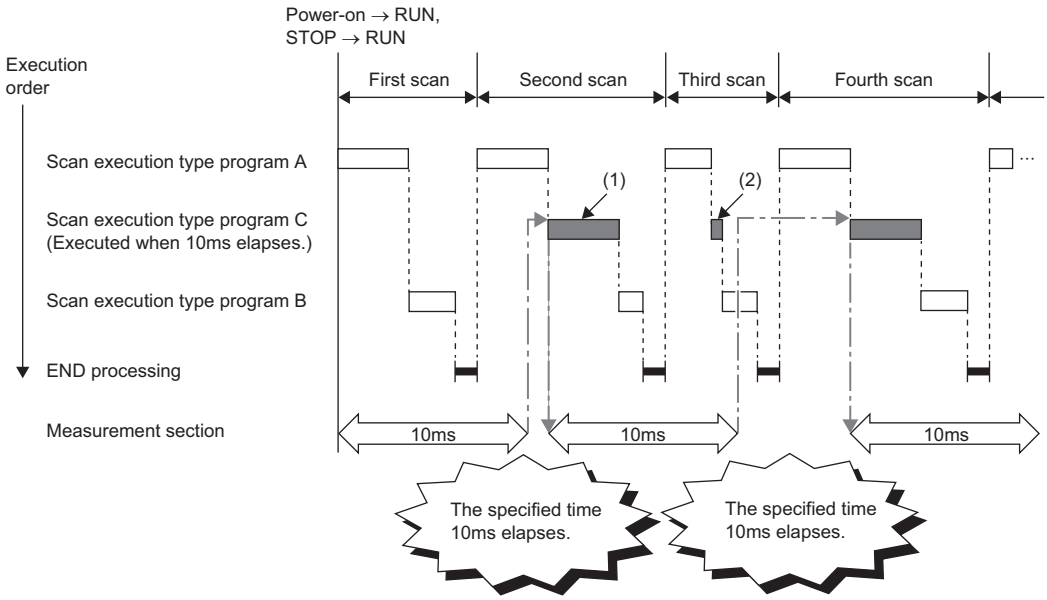
Point

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 110 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).

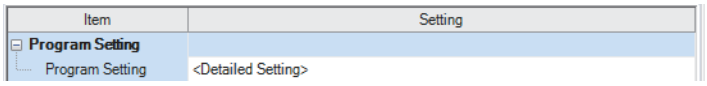
Trigger setting

Use the event execution type detail setting.

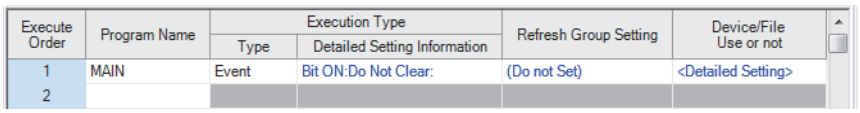
[CPU Parameter] ⇨ [Program Setting]

Operating procedure

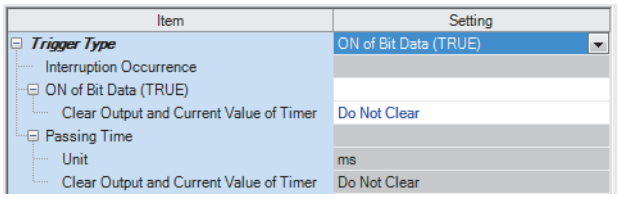
"Program Setting" window



"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

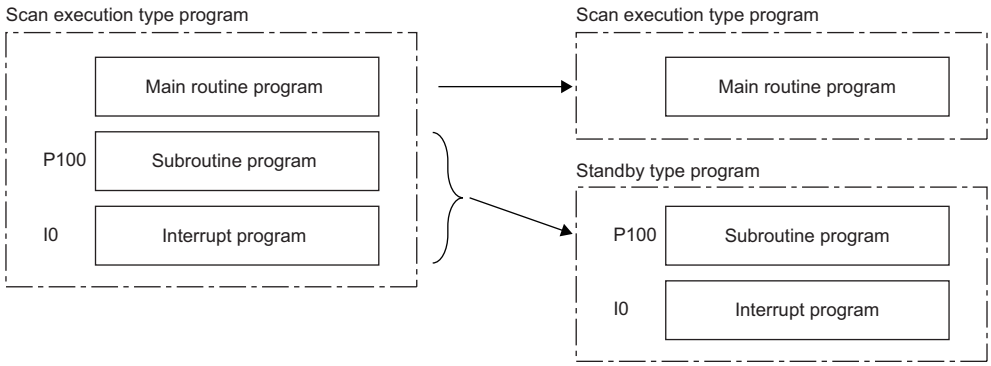
Item		Description	Setting range	Default
Interruption occurrence		Sets the interrupt pointer used as the trigger.	I0 to I15, I28 to I31, I44, I45, I48, I49, I50 to I1023	—
ON of Bit Data (TRUE)		Sets the device used as the trigger.	☞ Page 108 Bit data ON (TRUE)	—
Clear Output and Current Value of Timer		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

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 112 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

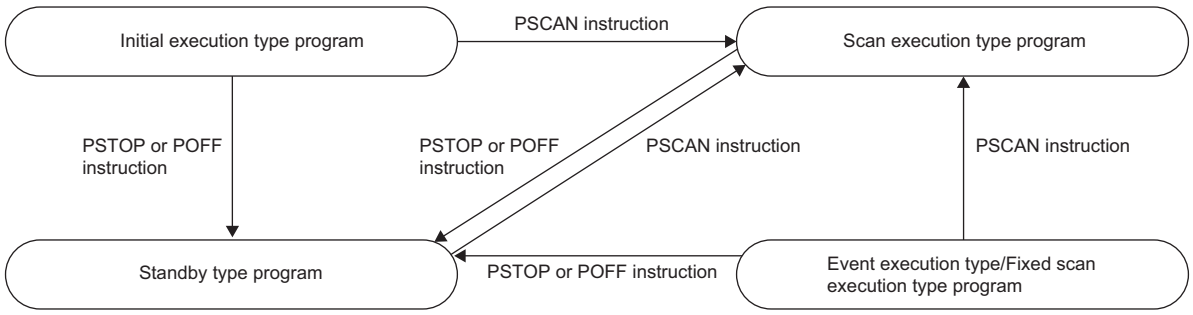
Execute Order	Program Name	Execution Type		Refresh Group Setting	Device/File Use or not
		Type	Detailed Setting Information		
1	MAIN	Scan		(Do not Set)	<Detailed Setting>
2	MAIN1	Fixed Scan		(Do not Set)	<Detailed Setting>
3	MAIN2	Standby		(Do not Set)	<Detailed Setting>

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.

Execution type before change	Execution instruction		
	PSCAN	PSTOP	POFF
Initial	Switches to "Scan" at the next scan.	Switches to "Standby" at the next scan.	Turns off the output at the next scan and switches to "Standby" at the scan after that.
Scan	Non-processing (does not change from "Scan")		
Fixed scan/event	Switches to "Scan" at the next scan.		Turns off the output at the next scan and switches to "Standby" at the scan after that.*1
Standby		Non-processing (does not change from "Standby")	Non-processing (does not change from "Standby")

*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)

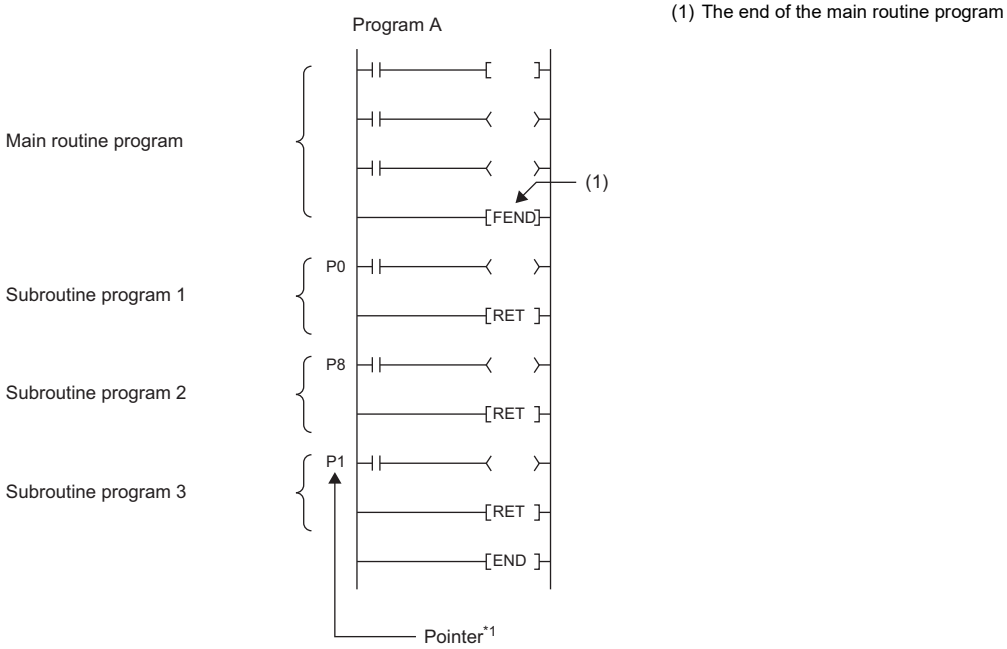
Point

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

8.6 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 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 111 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 495 Nesting (N))

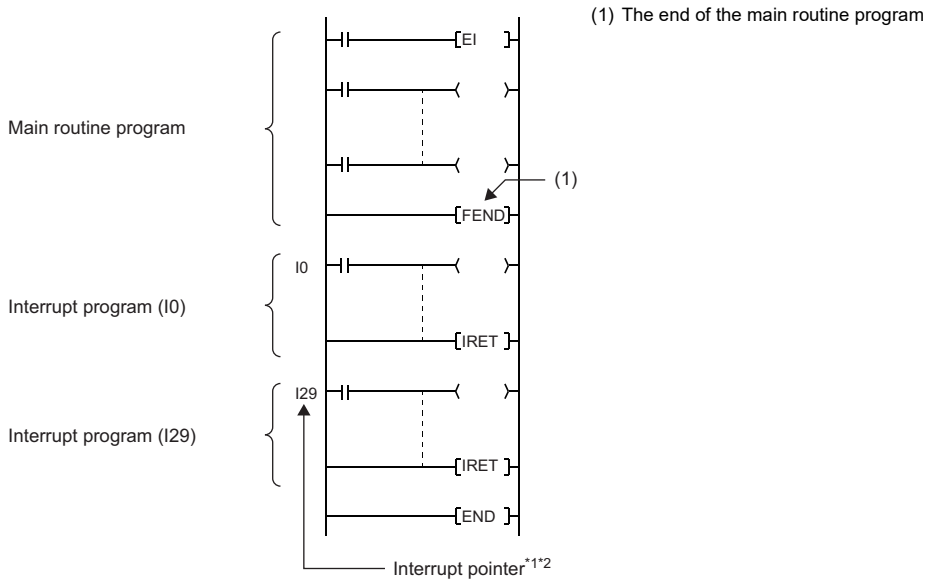
Precautions

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.

8.7 Interrupt Program

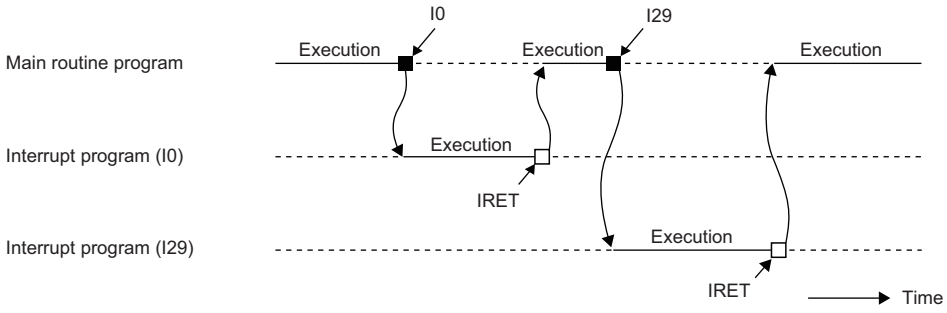
A program from an interrupt pointer (I) through the IRET instruction.



*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 500 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

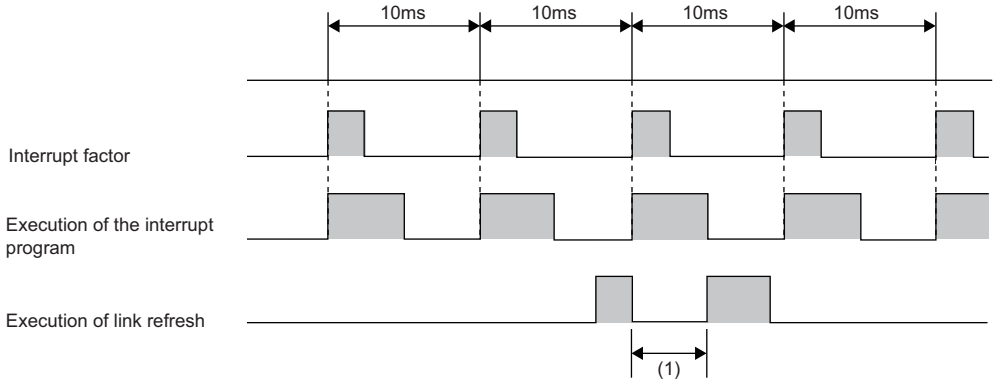
Making a program a standby type allows it to be managed as a separate program. (☞ Page 111 Standby type program)

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

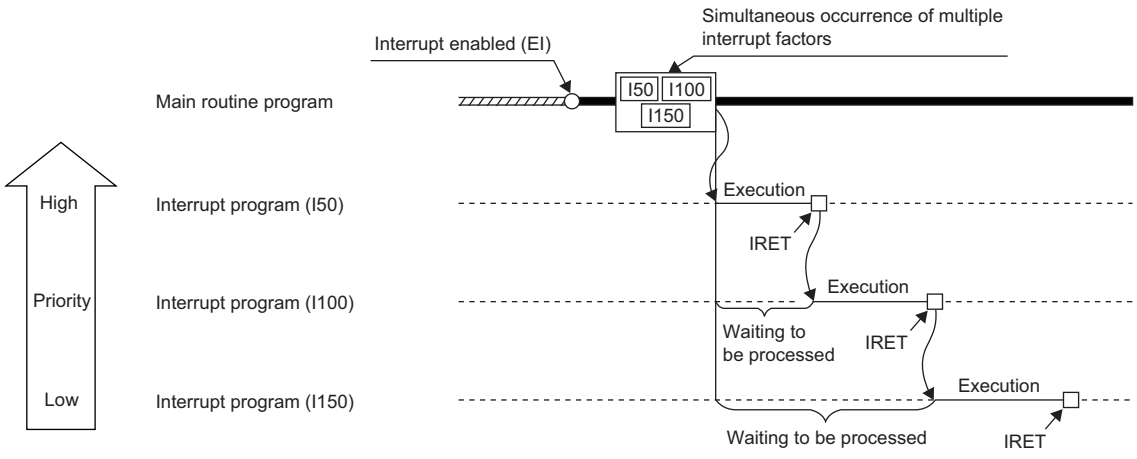
The interrupt program for the interrupt factor is executed.

■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 129 Interrupt priority)

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

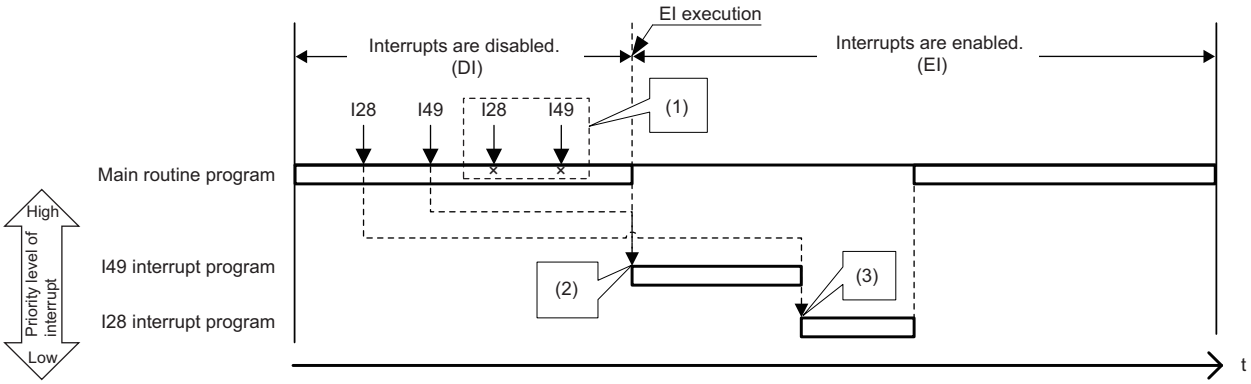
The interrupt programs are executed in the order of priority. If multiple interrupts with the same interrupt priority occur simultaneously, the interrupt programs are executed in the order of interrupt priority. (☞ Page 132 Multiple interrupt execution sequence)



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

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

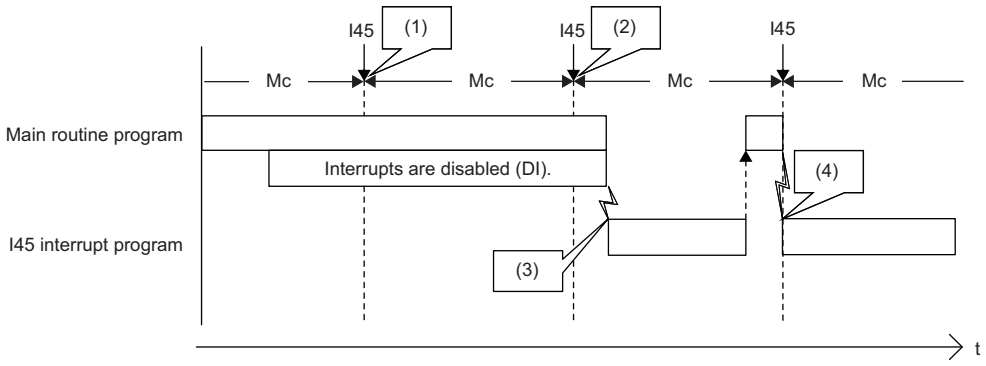
The interrupt factor that has occurred is memorized, and the interrupt program corresponding to the factor will be executed when the interrupt is enabled. Even if the same interrupt factor occurs multiple times, it will be memorized only once. If the IMASK instruction and SIMASK instruction are used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

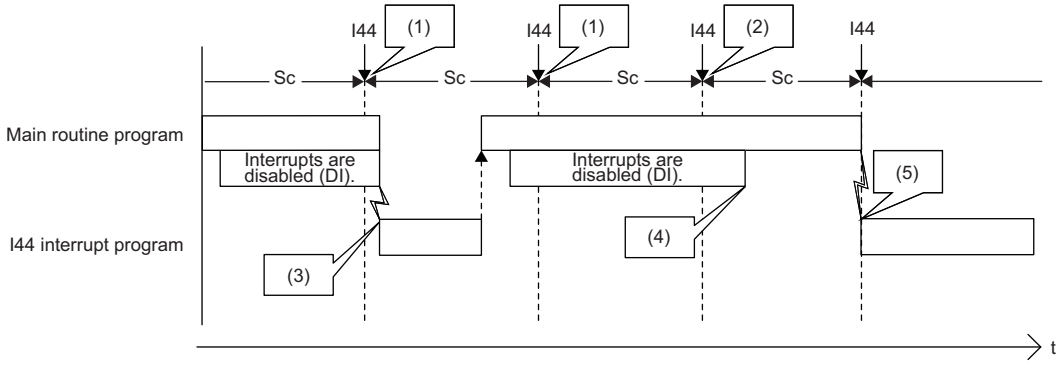
The interrupt factor that has occurred is memorized, and the I45 interrupt program will be executed when the interrupt is enabled. Even if the I45 interrupt occurs multiple times, its interrupt factor will be memorized only once. If the IMASK instruction and SIMASK instruction are used to disable the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

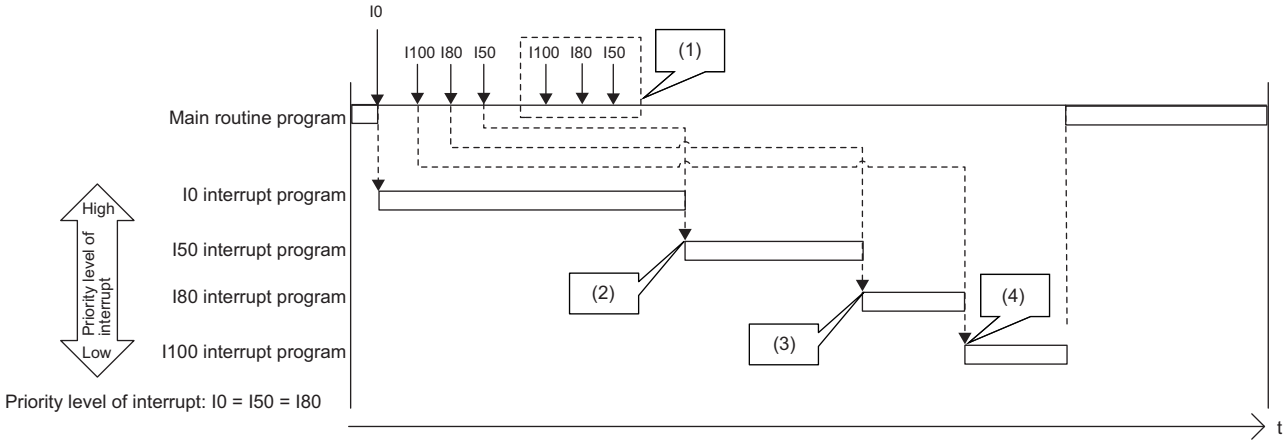
(4) I44 is not executed because interrupts continue to be disabled (DI) beyond the second cycle of I44.

(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

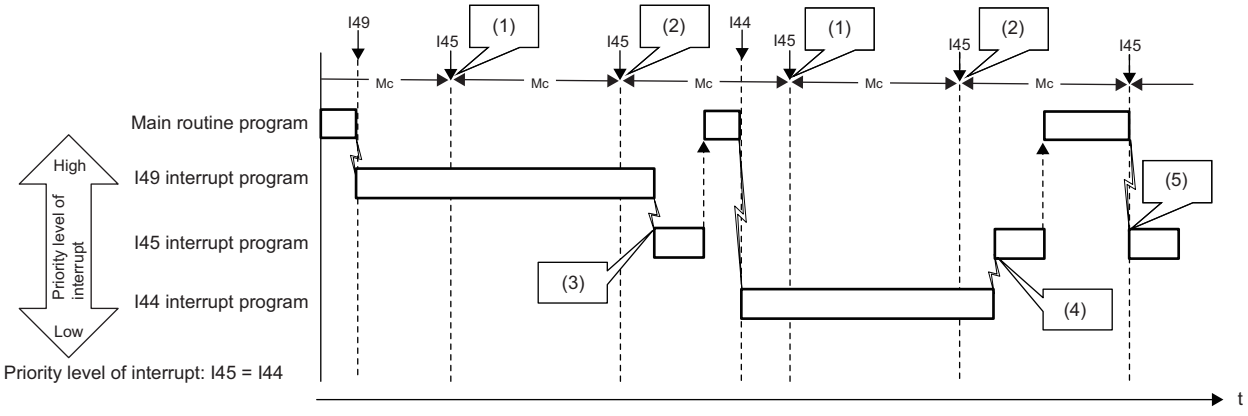
The interrupt factor that has occurred is memorized. After the running interrupt program finishes, the interrupt program corresponding to the factor will be executed. Even if the same interrupt factor occurs multiple times, it will be memorized only once.



(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.

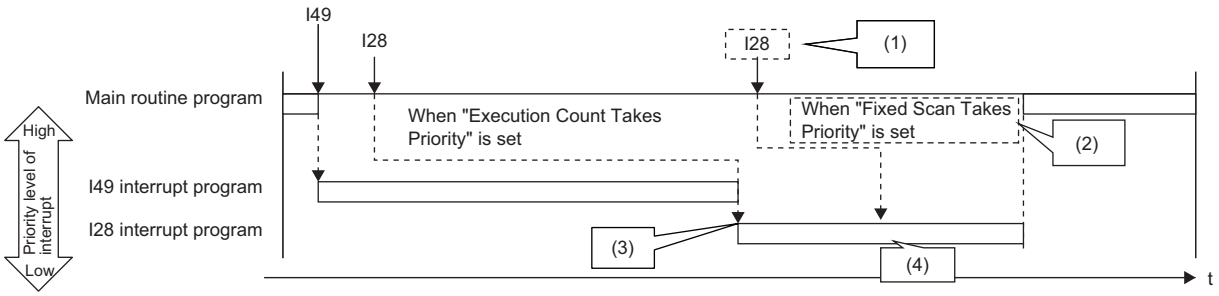


Mc: Multiple CPU synchronization cycle
(1) The interrupt factor is memorized.
(2) The second and following interrupt factors that occur while interrupts have been disabled (DI) are not memorized.
(3) After I49 that has a higher priority level is completed, I45 is executed.
(4) Because both I44 and I45 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

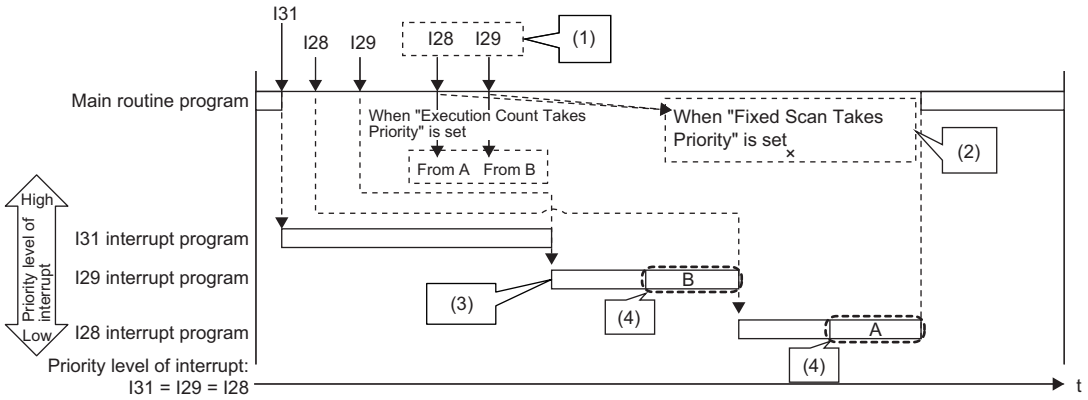
The interrupt factor that has occurred is memorized. After the running interrupt program finishes, the interrupt program corresponding to the factor will be executed. If the same interrupt factor occurs multiple times, it will be memorized once but operation at the second and later occurrences depends on setting of the fixed scan execution mode (☞ Page 105 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.

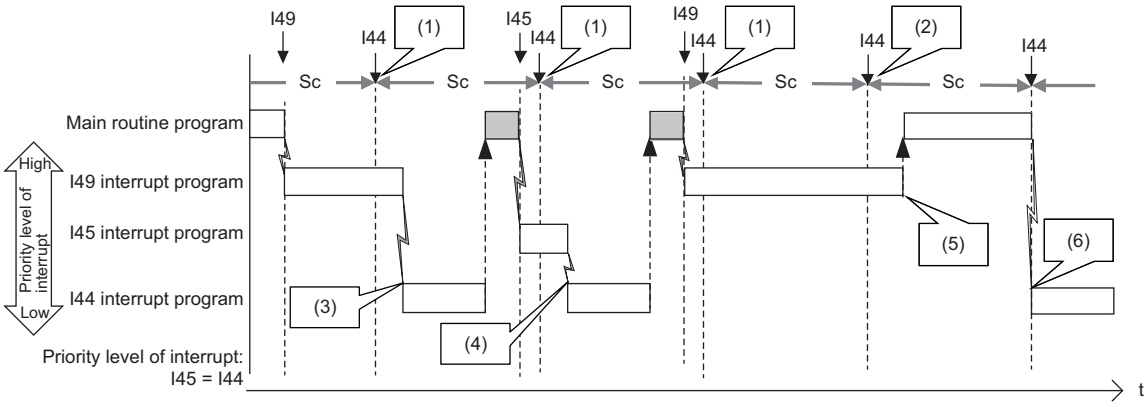
When interrupts with the same priority level as the one being executed occur



(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.

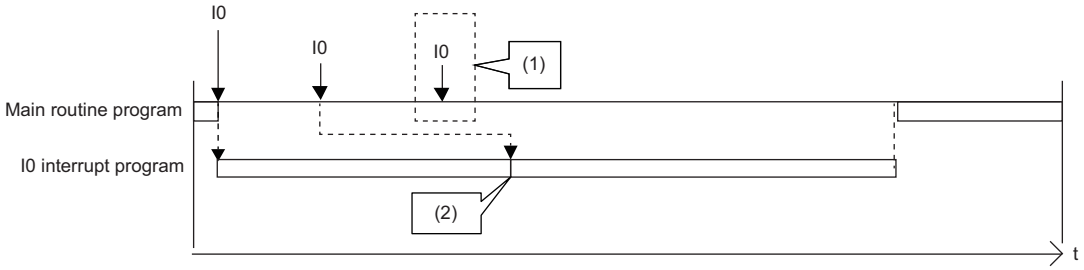


Sc: Inter-module synchronization cycle
(1) The interrupt factor is memorized.
(2) Because an interrupt does not occur, the memorized information is discarded in the second cycle.
(3) After I49 that has a higher priority level is completed, I44 is executed.
(4) After I45 that has the same priority level 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

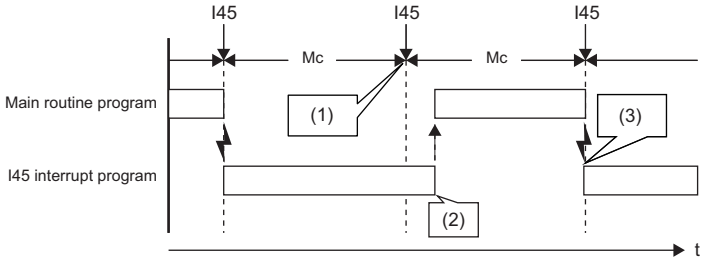
The interrupt factor that has occurred is memorized, and the interrupt program corresponding to the factor will be executed when the interrupt is enabled. Even if the same interrupt factor occurs multiple times, it will be memorized only once.



(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

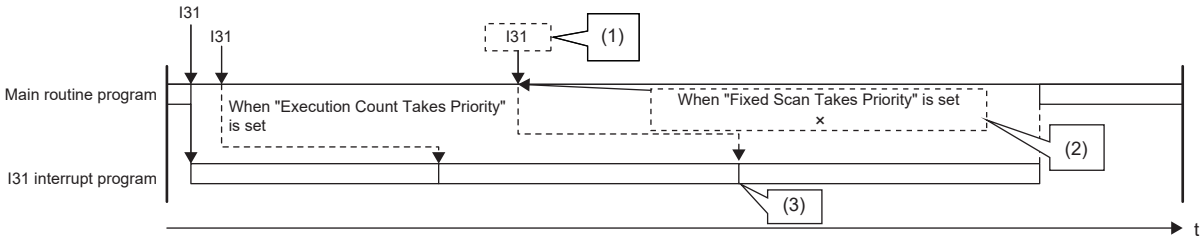
If an interrupt factor which is the same as that for the running interrupt program occurs, the factor is not memorized. Therefore, the corresponding interrupt program will not be executed after the running interrupt program finishes. Also, if the I45 interrupt program for this cause cannot be executed, 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

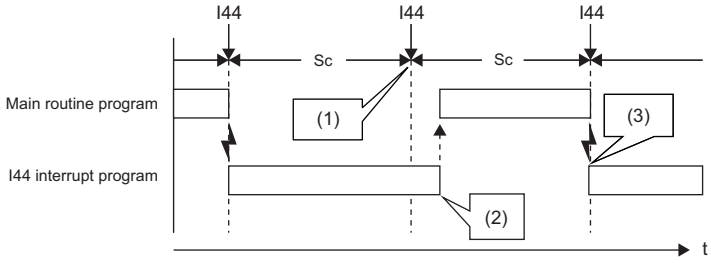
The interrupt factor that has occurred is memorized. After the running interrupt program finishes, the interrupt program corresponding to the factor will be executed. If the same interrupt factor occurs multiple times, it will be memorized once but operation at the second and later occurrences depends on setting of the fixed scan execution mode (☞ Page 105 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.

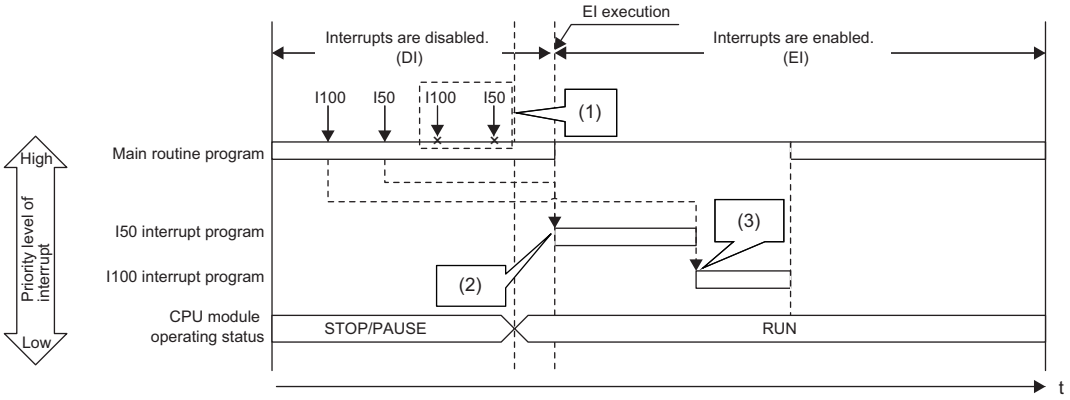


Sc: Inter-module synchronization cycle
(1) The interrupt factor is discarded because the same interrupt (I44) is being executed.
(2) The interrupt is not executed.
(3) The interrupt is executed.

■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.



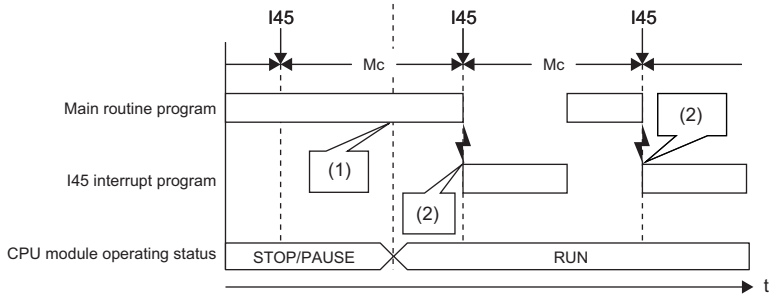
(1) The second and following interrupt factors that occur while the CPU module is in the STOP state are not memorized.

(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.



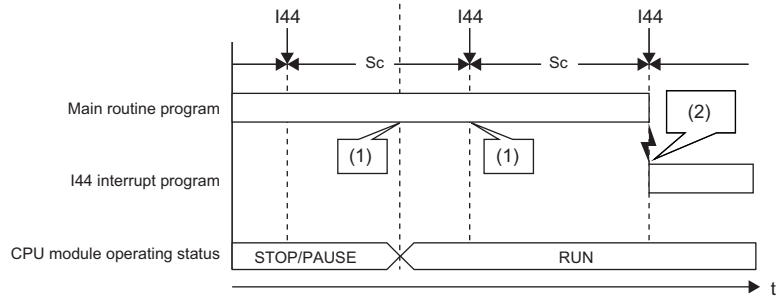
Mc: Multiple CPU synchronization cycle

(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.

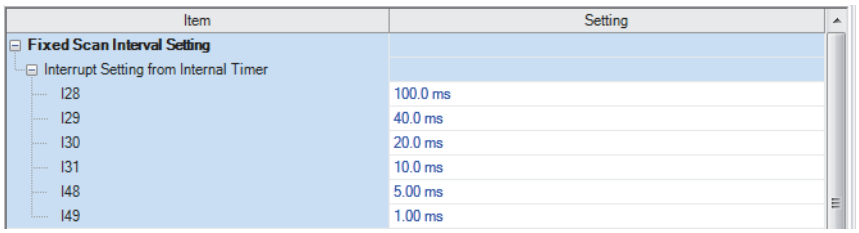
(2) The interrupt is executed.

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

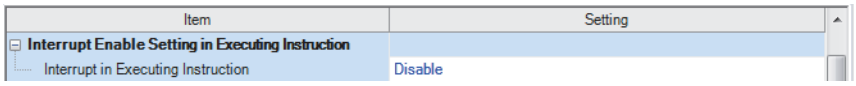
Item	Item	Description	Setting range	Default
Interrupt Setting from Internal Timer	I28	Sets the execution interval of I28.	0.5 to 1000ms (in units of 0.5ms)	100.0ms
	I29	Sets the execution interval of I29.		40.0ms
	I30	Sets the execution interval of I30.		20.0ms
	I31	Sets the execution interval of I31.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

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 133 Precautions)	• Disable • Enable	Disable

Processing at interrupt program startup

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

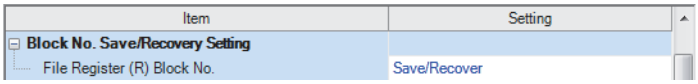
- Saving/restoring of the file register (R) block number
- Saving/restoring of the index register (Z, LZ)

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

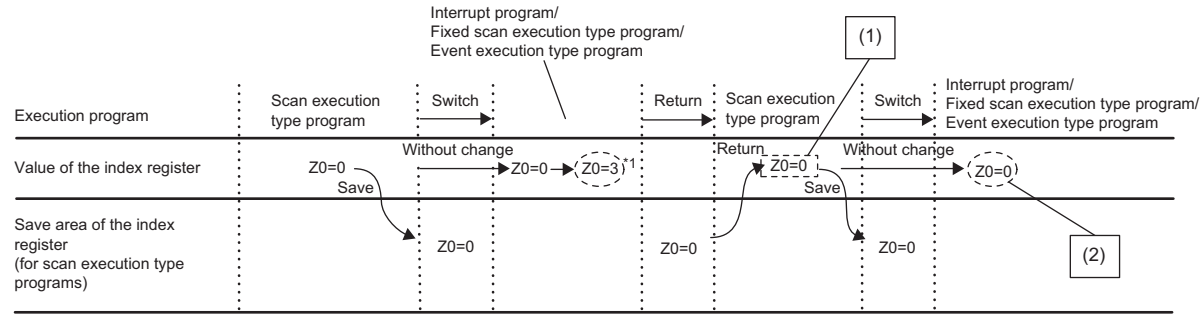
Point

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 724 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)

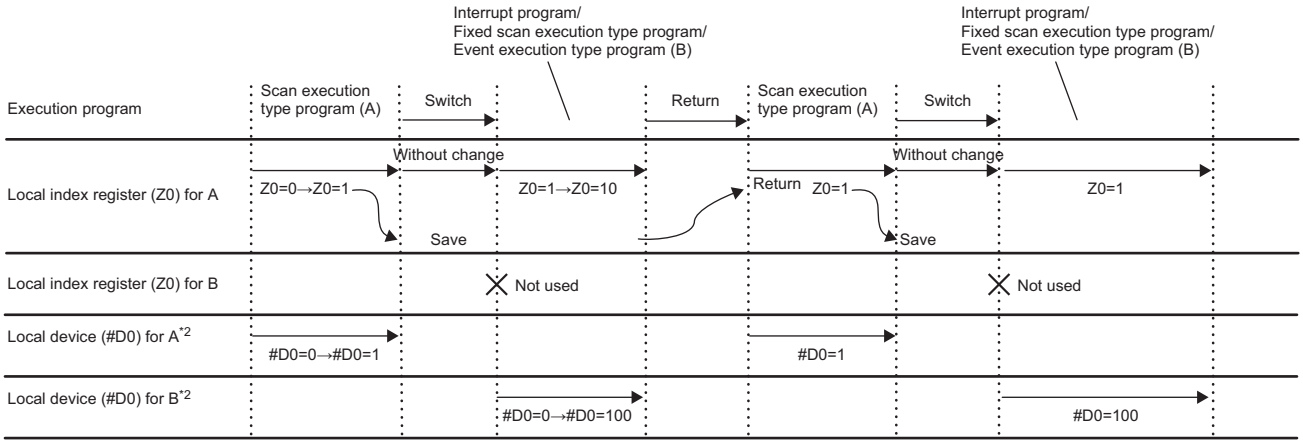
When an interrupt program starts, the value of the index register (Z, LZ) of the running program is saved. When the interrupt program finishes, and the saved value is restored to the running program. Note that when an interrupt program starts, the local index register (Z, LZ) is not switched to the different 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.)

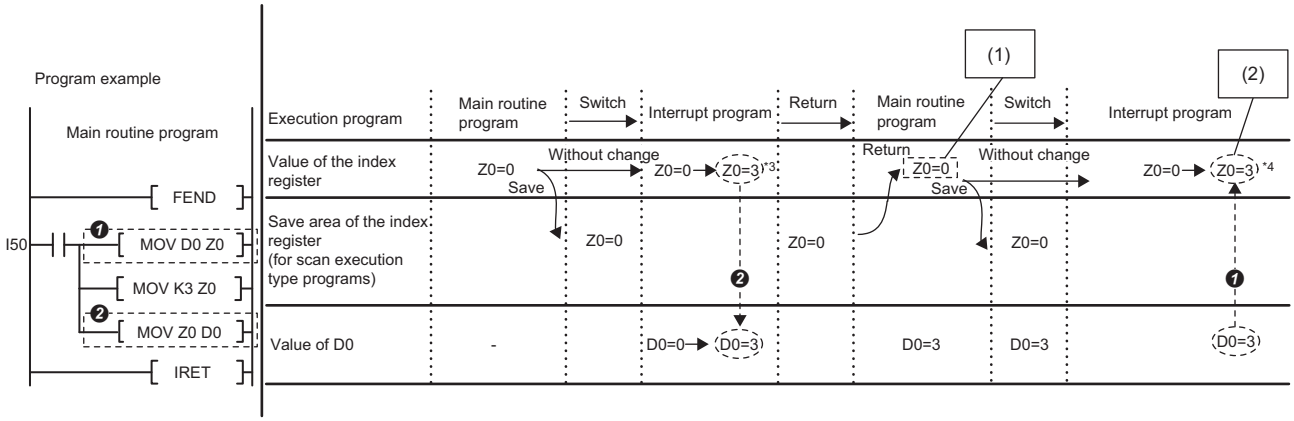
- Operations of the local index register and local device



*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.

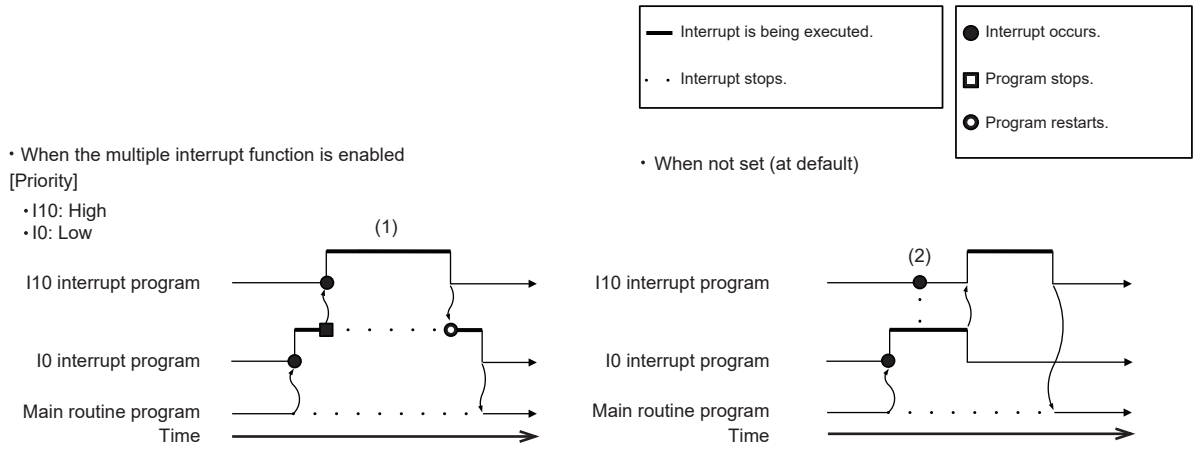


*3 A value in Z0 changes at MOV D0 Z0.
*4 Z is zero when the program ❶ and ❷ are not added.

(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 130 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

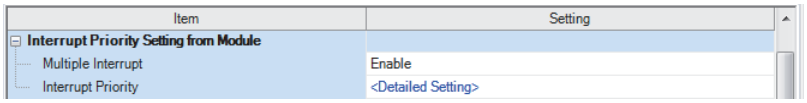
Interrupt priority setting

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

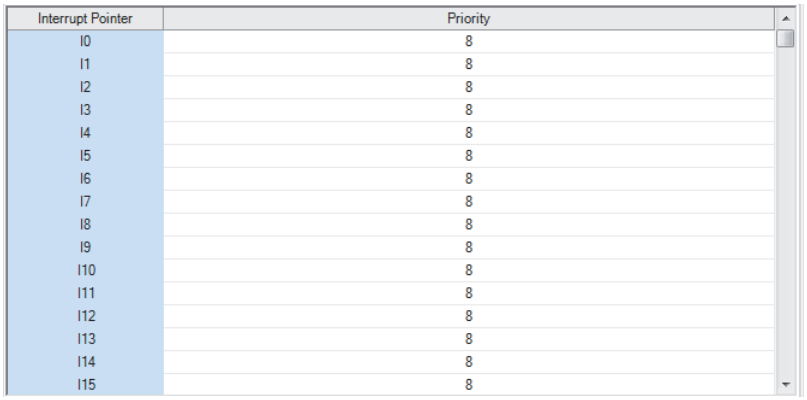
[CPU Parameter] ⇨ [Interrupt Settings] ⇨ [Interrupt Priority Setting from Module]

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.

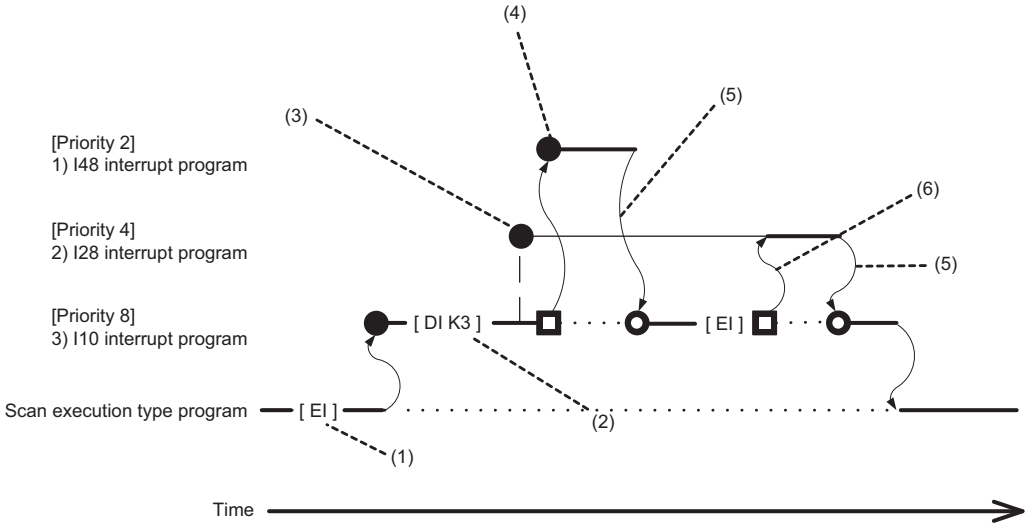
Ex.

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.

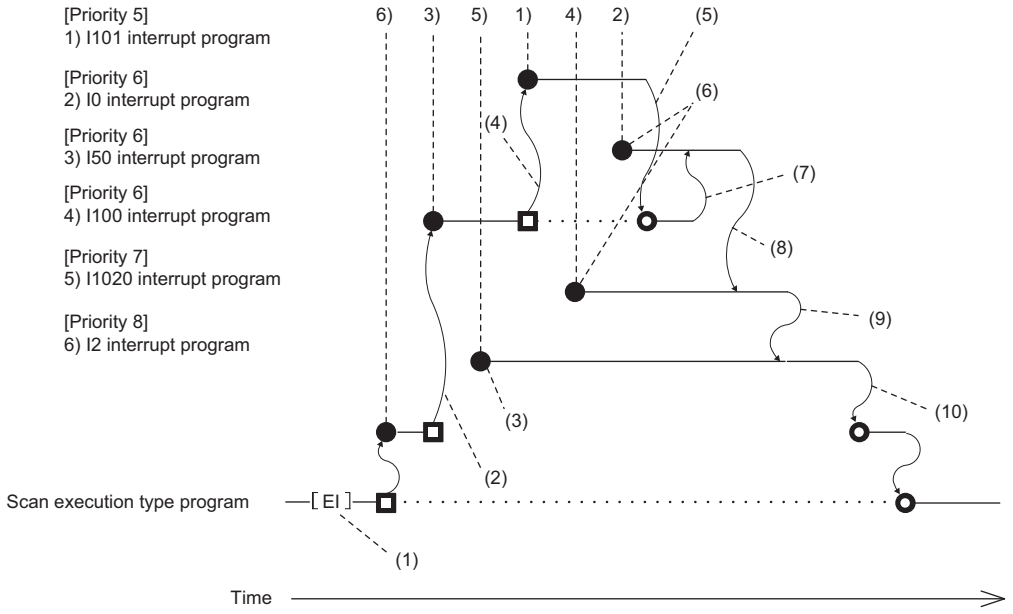
Ex.

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 203 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 203 Error detection setting)

When the data logging function is used

When using occurrence of an interrupt as the data collection condition, set the fixed scan interval of the interrupt program, taking into consideration the processing time for these functions. (☞ Page 307 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 125 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.

In redundant mode

When the systems are switched during time measurement of I28 to I31, I48, or I49, the time measurement is interrupted and the new control system starts the time measurement from 0.

9 CPU MODULE OPERATION PROCESSING

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

- RUN state
- STOP state
- PAUSE state

9.1 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 137 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.

9.2 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 137 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 137 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, refer to the following.
☞ Page 534 LABEL INITIALIZATION FUNCTION

Output mode at operating status change (STOP to RUN)

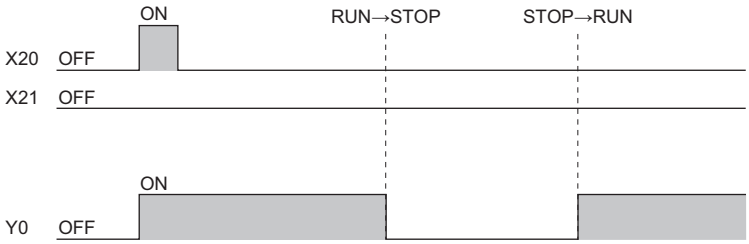
When the operating status changes from RUN to STOP, for example, the CPU module internally stores the status of the outputs (Y) to turn them all off.

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.

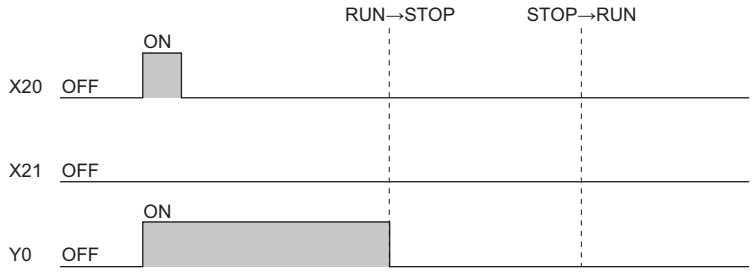
■Outputting the output (Y) status before STOP

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.)

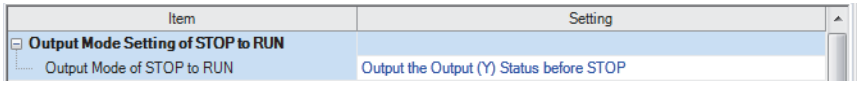


Setting the output mode

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

9.3 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.
In a redundant system with redundant extension base unit, if a momentary power failure occurs in the power supply module mounted on an extension base unit, operation processing in the CPU modules of 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 power supply modules. But if a momentary power failure occurs when power is supplied to either of the power modules, operation processing in the CPU modules of both systems is interrupted.

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 200 Scan time monitoring time setting)

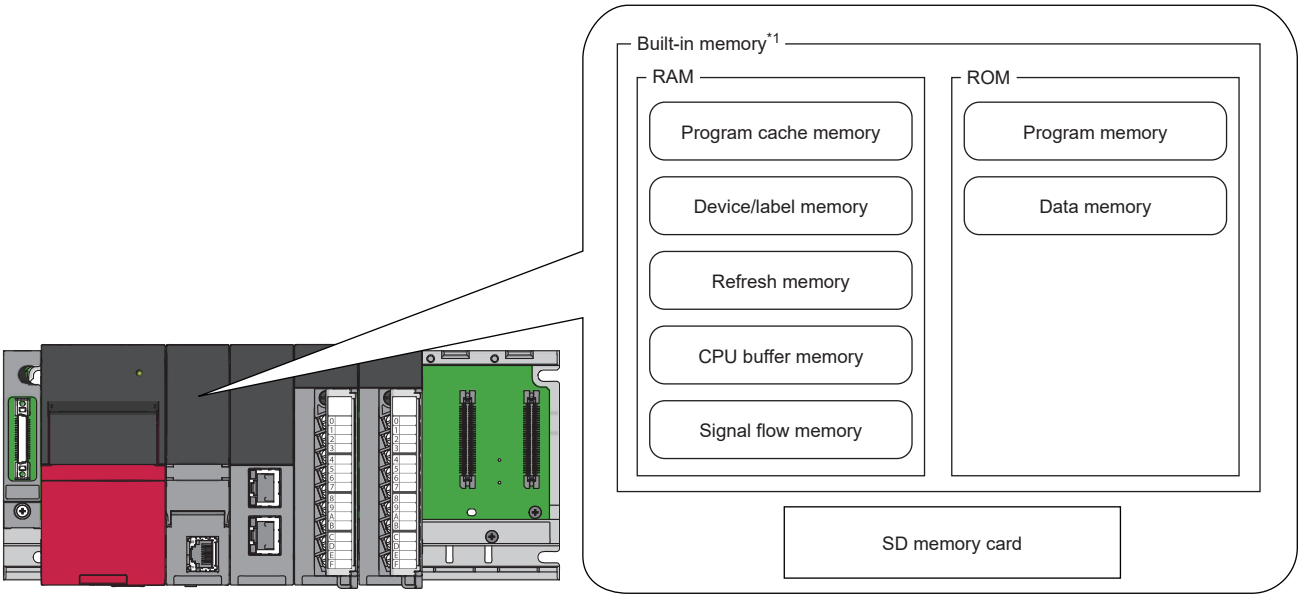
Momentary power failure exceeding the allowable momentary power failure time

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

10 MEMORY CONFIGURATION OF THE CPU MODULE

10.1 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.

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. (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 701 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.

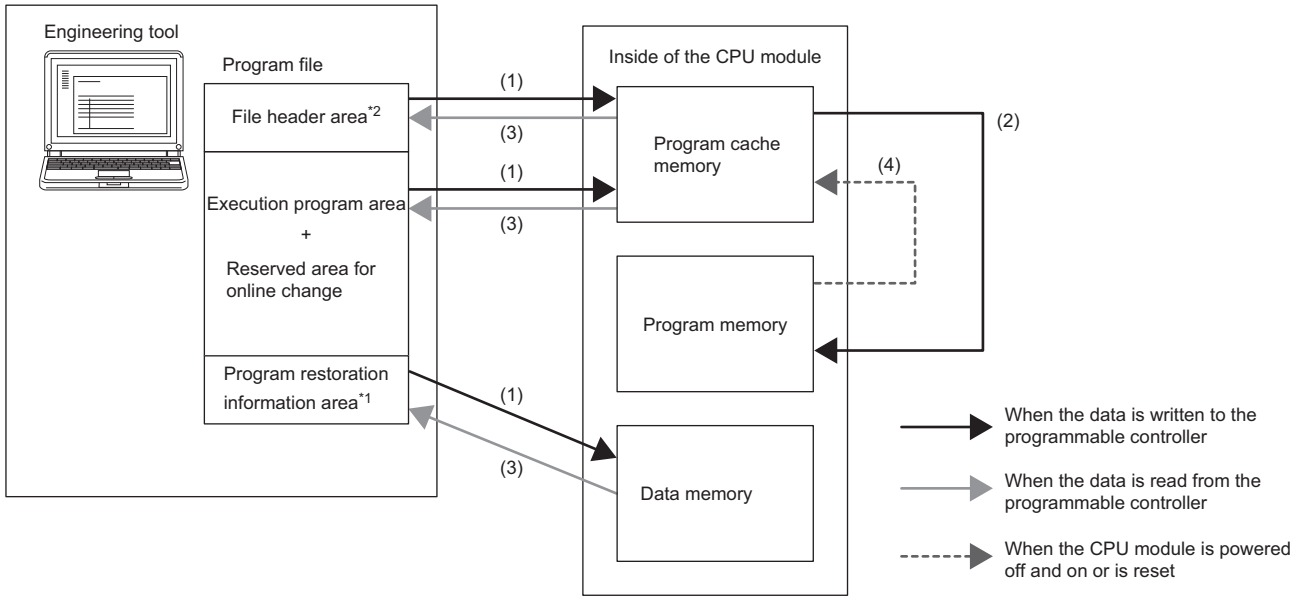
Data to be allocated

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 141 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 141 Destination of the file header area)

Point

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

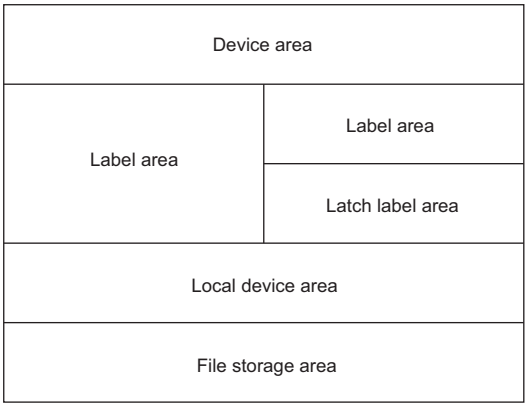
GX Works3 Operating Manual

■Destination of the file header area

In the case of a CPU module with firmware version "12" or earlier, the destination of the file header area is the data memory.

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 142 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 142 Device/label memory area setting)

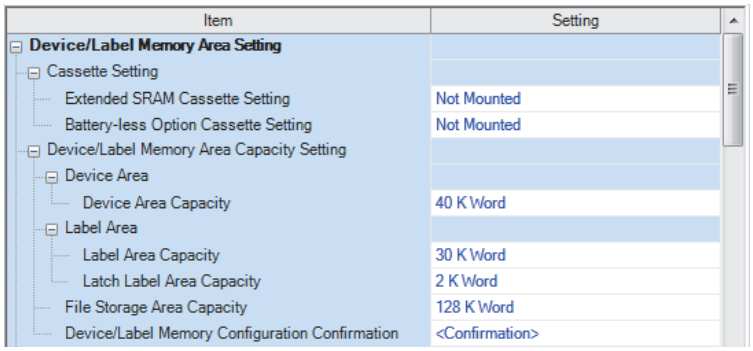
Device/label memory area setting

The capacity of each data area allocated within the device/label memory can be changed. (☞ Page 141 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 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	Extended SRAM Cassette Setting		When using an extended SRAM cassette, select the capacity of the cassette.	• Not Mounted • 1MB • 2MB • 4MB • 8MB	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 143 Default capacity of each area	☞ Page 143 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		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.		

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 (☞ Page 74 Hardware specifications). 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	R08PCPU	R16PCPU	R32PCPU	R120PCPU
Device area	40K words	40K words	40K words	40K words
Label area	40K words	50K words	90K words	110K words
Latch label area	2K words	2K words	4K words	4K words
Local device area	0K word	0K word	0K word	0K word
File storage area	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.)

- R08PCPU: 50K words
- R16PCPU: 60K words
- R32PCPU: 70K words
- 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. (☞ Page 78 Extended SRAM Cassette)

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

■R08PCPU

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)
Device area	2 to 594K words	2 to 1618K words	2 to 4690K words
Label area	0 to 592K words	0 to 1616K words	0 to 4688K words
Latch label area	0 to 544K words	0 to 1568K words	0 to 4640K words
File storage area	0 to 544K words	0 to 1568K words	0 to 4640K words

■R16PCPU

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)
Device area	2 to 860K words	2 to 1884K words	2 to 4956K words
Label area	0 to 858K words	0 to 1882K words	0 to 4954K words
Latch label area	0 to 800K words	0 to 1824K words	0 to 4896K words
File storage area	0 to 800K words	0 to 1824K words	0 to 4896K words

■R32PCPU

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)
Device area	2 to 1158K words	2 to 2182K words	2 to 5254K words
Label area	0 to 1156K words	0 to 2180K words	0 to 5252K words
Latch label area	0 to 1088K words	0 to 2112K words	0 to 5184K words
File storage area	0 to 1088K words	0 to 2112K words	0 to 5184K words

■R120PCPU

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)
Device area	2 to 1690K words	2 to 2714K words	2 to 5786K words
Label area	0 to 1688K words	0 to 2712K words	0 to 5784K words
Latch label area	0 to 1600K words	0 to 2624K words	0 to 5696K words
File storage area	0 to 1600K words	0 to 2624K words	0 to 5696K words

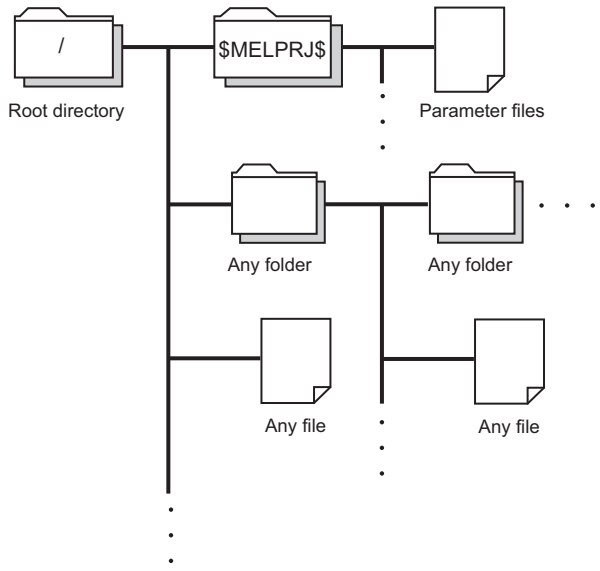
Point

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

☞ Page 78 Extended SRAM Cassette

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.

Refresh memory

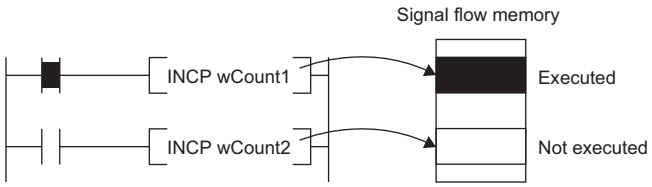
This memory is used to store data used to refresh communication with the intelligent function module. (Page 494 Refresh Data Register (RD))

CPU buffer memory

This memory is used by the Ethernet function or in data communication between multiple CPUs. (Page 722 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, 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

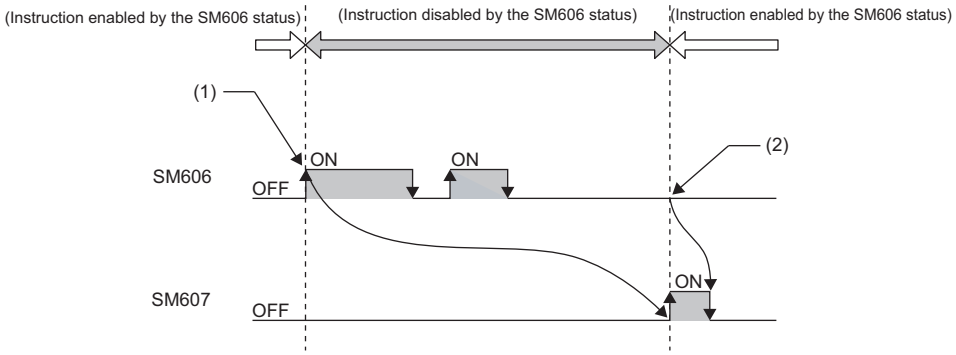
For details on how to insert or remove the SD memory card, refer to the following.

☞ Page 39 Inserting or Removing an Extended SRAM Cassette

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.

10.2 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
R08PCPU	128 bytes	512 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 762 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 the R08PCPU, it is written as 2048 bytes of data because the file size unit of the data memory is 2048 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	

10.3 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.
	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.

10.4 Files

This section lists the files used by the CPU module.

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*9		×	×	○*10	○*10	• UEXmmmnn.PRM*1 • UEXmmm00.PPR*5
Module-specific backup parameter*6		×	×	○*10*11	○*10*11	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
Remote password		×	×	○	○	00000001.SYP
Firmware update file		×	×	○*3	○	mmmm_vv.SYF
Firmware update prohibited file		×	×	○	○*3	FWUPDP.SYU
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
Backup data file for the iQ Sensor Solution data backup/restoration function		×	×	×	○	Depends on a connected device.QBR*7
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 156 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 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.

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

*11 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

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		—	—	—	○	○	○	—	—
Device data storage file		○*3*6	○	○*3*6	○*3*6	○	○*3*6	○	○
General-purpose data		○	○	○*4	○	○	○	○	○
Data logging setting file	Common setting file	○*5	○*5	○*5	○	○	○	—	—
	Individual setting file	○*5	○*5	○*5	○	○	○	—	—
Remote password		○*4	○	○*4	○*3*6	○	○*3*6	—	—
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	○	○	○	—	—
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 762 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.

File size

The following table lists the size of files that can be stored in the CPU module.

File type		File size
Program		Approx. 4050 bytes minimum (only the END instruction + 500 steps reserved for online program change)
FB file		Approx. 4900 bytes minimum (non-processing FB + 500 steps reserved for online program change)
CPU parameter		744 bytes minimum
System parameter		112 bytes minimum
Module parameter		Differs depending on the module used. For example, the size is 1036 bytes at minimum when the Ethernet function is used.
Module extension parameter		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		Differs depending on 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)
Initial device value		Approx. 140 bytes minimum (when the initial value of a word device is set)
Global label setting file		Approx. 660 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 file		2048 bytes minimum (when the setting capacity is 1K words)
Event history		1024 bytes minimum (when the setting capacity is 1K bytes (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.
Data logging setting file	Common setting file	142 bytes (When the setting is configured so that one-word data is collected in a binary file at each scan in continuous logging)
	Individual setting file	1192 bytes (When the setting is configured so that one-word data is collected in a binary file at each scan in continuous logging only for setting No.1)
Remote password		224 bytes minimum
Firmware update file		6816338 bytes minimum
Firmware update prohibited file		92 bytes minimum
System file for backing up CPU module data		20 + 12 + ((N1 × 34) + (N2 × 34) + (N3 × 34)) + M + 8 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: 30660 bytes • When the data logging setting has not been registered: 30232 bytes

File type	File size
Device/label data file for backing up CPU module data	• 1087398 + S1 + S2 + S3 + S4[*2] + S5 bytes[*1] · S1: N1 × 142 · S2: N2 × 134 · S3: (N3 + N4 × 2) × 18 + ((M1 ÷ 16) + M2 + (M3 × 2) + (M4 + (M4 ÷ 16) × 2) + ((M5 × 2) + (M5 ÷ 16) × 2) + (M6 × 8)) × 2 · S4: 16 + N5 × (12 + ((N6 + N7 × 2) × 18) + ((M7 ÷ 16) + M8 + (M9 × 2) + (M10 + (M10 ÷ 16) × 2)+ ((M11 × 2) + (M11 ÷ 16) × 2) + (M12 × 8)) × 2) · S5: (M13 + M14 + M15) × 2 · N1: Number of programs · N2: Number of file register files · N3: Number of device types used from M, L, B, F, SB, V, D, W, SW, U3En\HG, Z, LZ, and RD · N4: Number of device types used from T, ST, C, LC, LT, and LST · N5: Number of programs using local devices[*3] · N6: Number of local device types used from M, V, D, Z, and LZ · N7: Number of local device types used from T, ST, C, LC, LT, and LST · M1: Total number of points of M, L, B, F, SB, V, and S · M2: Total number of points of D, W, SW, U3En\HG, Z, and RD · M3: Number of points of LZ · M4: Total number of points of T, ST, and C · M5: Number of points of LC · M6: Total number of points of LT and LST · M7: Total number of points of the local devices M and V · M8: Total number of points of the local devices D and Z · M9: Number of points of the local device LZ · M10:Total number of points of the local devices T, ST, and C · M11:Number of points of the local device LC · M12:Total number of points of local devices LT and LST · M13:Module label capacity (words) · M14:Label area capacity (words) · M15:Latch label area capacity (words)
System file for the iQ Sensor Solution data backup/ restoration function	■CC-Link 36 + 20 × N bytes (N: No. of devices supporting iQSS whose data has successfully been backed up) ■Built-in Ethernet 52 + M bytes (M: Backup data file name size[*1]) The file name size differs depending on the device supporting iQSS.
Backup data file for the iQ Sensor Solution data backup/restoration function	The size differs depending on the device supporting iQSS.
System file for automatic restoration with the SD CARD OFF button	10 bytes

*1 Up to 3 bytes are added to the total file size so that the size will be a multiple of 4 bytes.

*2 S4 is added only when local devices are used.

*3 When the number of device points that is assigned to local devices is 0, N5 is 0.

Restriction

When the CPU module versions are different, a program which can be written to a CPU module with a certain version may not be written to the one with another version due to overcapacity. In this case, delete the steps reserved for online program change (default: 500 steps) and write the program to the CPU module.

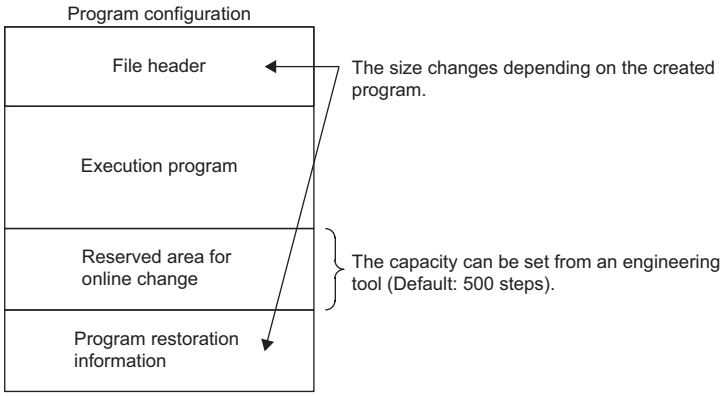
GX Works3 Operating Manual

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.

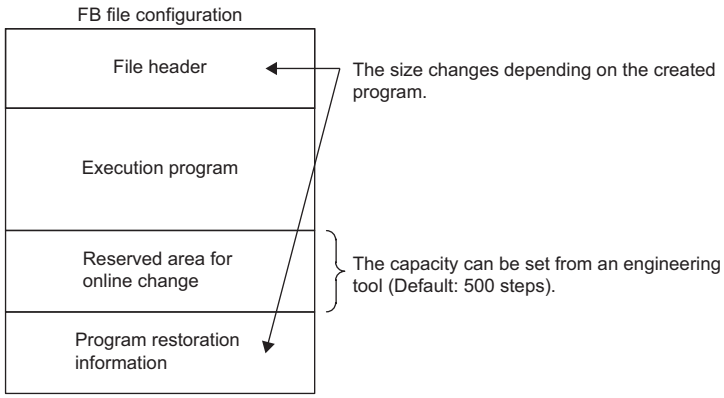


When the program is stored in the CPU module, file header, execution program, and reserved area for online change are allocated to the program memory. Program restoration information is allocated to the data memory. The following table shows the destination memory to which each area is allocated when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

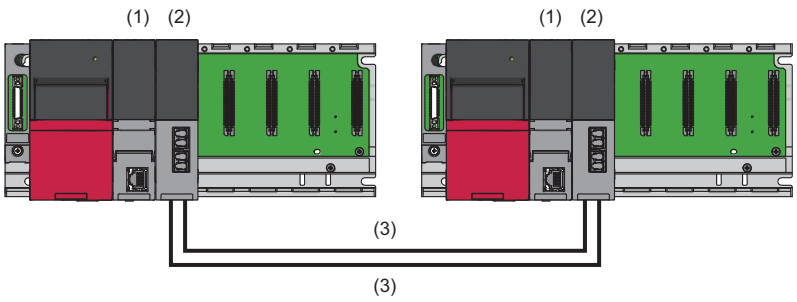
This file consists of a file header, execution program, reserved area for online change, and program restoration information.



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.

11 BASIC CONCEPT OF REDUNDANT SYSTEM

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 the CPU modules 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 762 Added and Enhanced Functions)

11.1 Concept of "Systems" in a Redundant System

System A and system B

A redundant system consists of two identical systems. One system is called "system A" and the other is called "system B" to distinguish between two systems connected with tracking cables. The system A or B is set using the engineering tool. (☞ Page 65 Setting the System (System A or System B))

Point

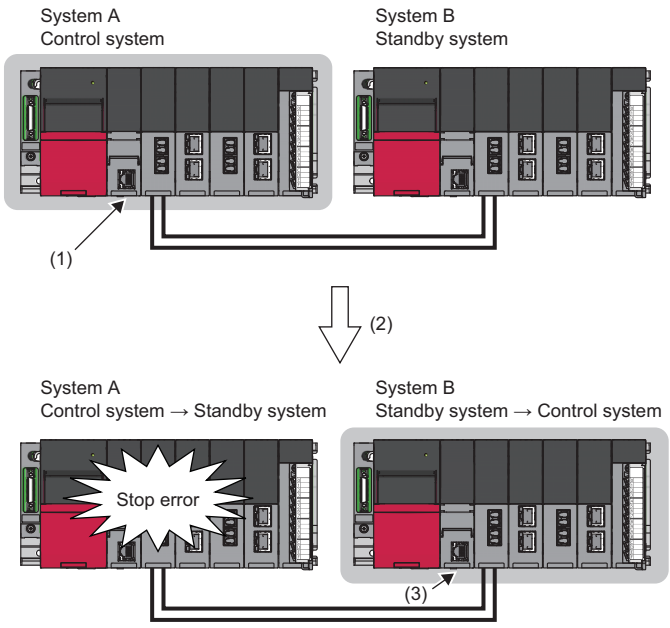
When one system is set as "system A", the other system is automatically set as "system B", and vice versa.

Control system and standby system

In a redundant system, the CPU module of one system executes programs and controls the entire redundant system. The other system will be in the standby state and does not perform control. The system that controls the entire redundant system is called "control system" and the system in the standby state is called "standby system".
The control or standby system is determined when both systems are started up and tracking communications between the two systems are established. (☞ Page 161 System Determination)

11.2 System Switching Between 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 from the control system to the standby system at every scan. If an error occurs in the control system, the standby system will function as the new control system and continue operation using the data that has been received.



(1) The control system is operating normally. (The control system transfers data to the standby system at every scan.)
(2) A stop error has occurred in the control system.
(3) The standby system functions as the new control system and continues operation using the data that has been received.

For details on the system switching, refer to the following.
☞ Page 377 System Switching

11.3 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 417 System Consistency Check)

11.4 Operation Modes of the Process CPU

The Process CPU operates in one of the following two modes.

Process mode

A mode to use the Process CPU in a system other than a redundant system. Select "Process" for "Mode" when creating a project using the engineering tool. The Process CPU is started up in process mode.

Redundant mode

A mode to use the Process CPU in a redundant system. Select "Redundant" for "Mode" when creating a project using the engineering tool. The Process CPU starts up in redundant mode. (☞ Page 64 Creating a Project)

Point

In a redundant system, set the redundant mode for the Process CPU in both systems.

11.5 Operation Modes of a Redundant System

A redundant system operates in one of the following two modes.

Operation mode	Description
Backup mode	A mode to operate a redundant system. When an error or a failure occurs in the control system, the standby system is switched to the new control system and continues operation. When a system failure occurs in the control system, the standby system can take over the control because data is transferred from the control system to the standby system at every scan.
Separate mode	A mode to perform system maintenance of a redundant system without stopping control. In separate mode, the CPU module of the control system and the CPU module of the standby system can execute different programs. In the standby system, programs and CPU parameters can be modified*1 and the operation of programs can be checked with the data transferred from the control system without stopping the control operation of the control system.

*1 When programs or CPU parameters are modified while labels (except those with a device assigned) are being used, perform the following operations when the redundant system is in backup mode.
When the program is modified: Execute the online change function for both systems.
When the CPU parameters are modified: Write data to the programmable controller when the CPU module is in the STOP state.
If programs or CPU parameters are modified only in the standby system, tracking transfer of labels is stopped. (☞ Page 408 When data is different between the control system and the standby system)

A redundant system starts up in backup mode. To switch the operation mode, use the engineering tool. (☞ Page 374 Operation Mode Change)

The following table summarizes the differences between the backup mode and the separate mode.

Item	Backup mode	Separate mode
Program operation	The control system executes programs. The standby system executes programs according to "Both Systems Program Executions Setting" in "Program Setting" of "CPU Parameter". The standby system does not execute programs by default. For how to change the mode to the backup mode, refer to the following. ☞ Page 375 Changing the operation mode to the backup mode	Both of the control system and the standby system execute programs. For how to change the mode to the separate mode, refer to the following. ☞ Page 374 Changing the operation mode to the separate mode
System switching	Both 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 is not transferred. • 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. • Auto memory copy • Memory copy using the engineering tool • Memory copy using the special relay and special register	The following copy operations can be performed. • Memory copy using the engineering tool • Memory copy using the special relay and special register
System consistency check	The system consistency is checked.	The system consistency is not checked.
Program execution type	■When the operation mode is changed to the backup mode The control system and the standby system take over the program execution type before the operation mode change.	■When the operation mode is changed to the separate mode The operation differs between the control system and the standby system. • The control system takes over the program execution type before the operation mode change. • The standby system executes the program with the execution type specified in 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 the interrupt program execution enable/disable status before the operation mode change.	■When the operation mode is changed to the separate mode The operation differs between the control system and the standby system. • The control system takes over the interrupt program execution enable/disable status before the operation mode change. • The standby system can no longer execute interrupt programs.
I/O refresh	The output (Y) refresh of the standby system depends on "Standby System Output Setting" of "CPU Parameter". The output (Y) of the standby system is not refreshed by default.	Even if "Standby System Output Setting" is set to "Disable" in "CPU Parameter", the output (Y) of the standby system is enabled.

Item	Backup mode	Separate mode
Link refresh	In the standby system, data is refreshed from devices of the CPU module to link devices (only the link special relay (SB) and link special register (SW)). (CC-Link is excluded.) This operation is the same in backup mode and separate mode.	In the standby system, data is refreshed from devices of the CPU module to link devices (only the link special relay (SB) and link special register (SW)). (CC-Link is excluded.) This operation is the same in backup mode and separate mode.
Writing data to the programmable controller	Data can be simultaneously written to both systems. Or, data can be written to only a system specified in the connection destination setting.	Data can be written to only a system specified in the connection destination setting.
Remote operation function	Remote operations can be performed on both systems.	Remote operations by 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 is not synchronized.

11.6 System Determination

This section describes how the control or standby system is determined.

When both systems are started up

The control or standby system is determined as described below.

Determination method

The control or standby system is determined when both systems are started up by powering on or resetting the system and tracking communications between the two systems are established.

■When both systems are simultaneously started up

The system A will be "control system" and the system B will be "standby system".
"Simultaneously" here means when one system is started up within three seconds of the other.

■When the system A or the system B is started up first

- In a redundant system without extension base units, the started system will enter the state that waits for the other system to start in three seconds. (☞ Page 163 State that waits for the other system to start)
- In a redundant system with redundant extension base unit, when the systems are powered on or reset one by one, the system that is started up first will be the control system and the other system will be the standby system. Settings for the state that waits for the other system to start in the CPU parameter (Control/Standby System Start-up Setting) become invalid.

Check method

Check the LED status of the redundant function module to check which system is the control system.

System	LED status of the redundant function module
Control system	
Standby system	

Point

Users can also check which system is the control system using the engineering tool. (GX Works3 Operating Manual)

- System monitor
- Monitor status bar

Precautions

■If the system A/B setting has not been set or the same system A/B setting has been set

A stop error occurs. To determine the control and standby systems, properly set the system A/B setting for both systems.

■If both systems are restarted when a stop error has occurred in the standby system (system A)

A stop error may occur in both systems.
If the stop error cause of the system A is a program error, a stop error occurs in 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 in the system B if a file mismatch is detected in the system consistency check after the restart. As a result, a stop error occurs in both systems.
In this case, eliminate the cause of the error that has occurred in the system A, and restart both systems.

■When the READY LED of the CPU module in one of the systems is flashing

Do not power off the other system.

The system may start up without checking the system consistency even when a mismatch exists between the two systems.

(☞ Page 418 Execution timing)

■If the system configurations differ between the two systems (for example, the insertion status of an SD memory card)

It may take time to start communications between the two systems. Check that the system configuration is the same between the two systems following the procedure for starting up both systems simultaneously, and then start up the redundant system.

(☞ Page 54 Starting up both systems simultaneously)

■When an SD memory card is used

- Do not power off the system, reset the CPU module, or remove the SD memory card during an access to the SD memory card. A stop error may occur in both systems and the systems may not be started as a redundant system.
- If the system is powered off, the CPU module is reset, or the SD memory card is removed during an access to the SD memory card, the data in the SD memory card may corrupt. 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. In this case, a stop error occurs in the other system as well. As a result, a stop error occurs in both systems and the systems cannot be started as a redundant system. Restart both systems.
- In a redundant system with redundant extension base unit, the control/standby system start-up setting is invalid. As a result, if the own system is started up while the SD memory card diagnostics is being executed in the other system, the own system may start up as a control system. Even when both systems are started up simultaneously, if the SD memory card diagnostics is executed in the other system, the own system may start up first as the control system, as in the case of the systems starting up one by one. In this case, when the SD memory card diagnostics of the other system finishes during initialization of the own system, a stop error is detected in the other system.

Point

If the SD memory card diagnostics of the other system takes time, only the own system can be started up.

(☞ Page 165 When one system is started automatically even though a tracking communication error has occurred)

When only one system is started up

- In a redundant system without extension base units, only one system, system A or system B, can be started up as the control system by the start-up methods described below.
- In a redundant system with redundant extension base unit, when the systems are powered on or reset one by one, the system that is started up first will be the control system and the other system will be the standby system. Settings for the state that waits for the other system to start in the CPU parameter (Control/Standby System Start-up Setting) become invalid.

Start-up method

To start up one system as the control system, perform one of the following operations to the CPU module that is waiting for the other system to start.

■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 RUN → STOP → RUN.

Set "Control System Start-up Setting (Switch Operation)" to "Enable" in the CPU parameter in advance. (☞ Page 429 Redundant System Operation Setting)

■Operation with the input (X)

Turn on the input (X) set in the parameter.

Set "Control System Start-up Setting (Input (X))" to "Enable" in the CPU parameter in advance. (☞ Page 429 Redundant System Operation Setting)

State that waits for the other system to start

In a redundant system without extension base units, if tracking communications are not established after the CPU module of the own system has started up, the own system will enter the state that waits for the other system to start in three seconds. The CTRL LED and the SBY LED of the redundant function module turn off because the control system cannot be determined at this point. In addition, the BACKUP LED of the redundant function module flashes because a system switching disable cause exists.

After tracking communications with the other system are established, the control system is determined and the system switching disable cause is eliminated.

■Operation of the CPU module in the state that waits for the other system to start

Even when the RUN/STOP/RESET switch of the CPU module is set to the RUN position, the CPU module will be in the STOP state while it is in the state that waits for the other system to start because the control system has not been determined yet. The CPU module does not execute programs.

The following table lists the refresh operations to be performed while the CPU module is in the state that waits for the other system to start.

Type	Operation
I/O refresh	Only the input refresh is performed. The output refresh is not performed.
Network module link refresh	Between the network module and the CPU module, SB/SW is refreshed but other link devices are not.
Intelligent function module refresh	Data is refreshed between the intelligent function module and the CPU module.

■Timeout of the state that waits for the other system to start

Time measurement starts for the state that waits for the other system to start upon completion of the initial processing.

Set the timeout time in "Other system Start-up Timeout Setting" of "CPU Parameter". (☞ Page 429 Redundant System Operation Setting)

When a timeout is detected, a stop error occurs. To not to generate a stop error, set "Other system Start-up Timeout Setting" of "CPU Parameter" to "Not Set".

Operation when the control or standby system has not been determined yet

In a redundant system without extension base units, the operation of the CPU module is the same as that in the state that waits for the other system to start (refresh operations to be performed while the CPU module is in the state that waits for the other system to start). (☞ Page 163 Operation of the CPU module in the state that waits for the other system to start)
The following tables list the refresh operations in a redundant system with redundant extension base unit.

■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	Between the network module and the CPU module, SB/SW is refreshed but other link devices are not.
Intelligent function module refresh	Data is refreshed between the intelligent function module and the CPU module.

■Refresh for the modules mounted on an extension base unit

Type	Operation
I/O refresh	Neither the input nor output refresh is performed.
Network module link refresh	Data is not refreshed between the network module and the CPU module.
Intelligent function module refresh	Data is not refreshed between the intelligent function module and the CPU module.

Precautions

- To start up the system that is waiting for the other system to start as the control system, check that the other system is not operating as the control system.
- When tracking communications are established by connecting tracking cables and powering on or resetting the other system, the CPU module that has been waiting for the other system to start enters the RUN state and starts executing programs. For this reason, check that program execution by the CPU module does not cause any problems before connecting tracking cables and starting up the other system.
- If a stop error has occurred in the other system when tracking communications are established between the two systems, a stop error occurs in the CPU module that has been waiting for the other system to start as well. In this case, take action against the error of the other system, and power off and on or reset both systems.
- When tracking communications cannot be established due to power-off of the other system or a problem with tracking cables, the CPU module enters the state that waits for the other system to start. Check that the other system is powered on and there is no problem with tracking cables. (☞ Page 549 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 162 When an SD memory card is used)
- To start up the systems one by one in a redundant system with redundant extension base unit, complete the start-up process of one system first. Then, start up the other system. (☞ Page 57 Starting up the systems one by one) During start-up of a system, the system cannot perform tracking communications. For this reason, a stop error may occur in the CPU module of the system that is started up later. In this case, restart the system having the CPU module in which a stop error has occurred. (When automatic recovery is set, the system can be automatically restarted.) (☞ Page 436 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 start-up, 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 start-up, 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.

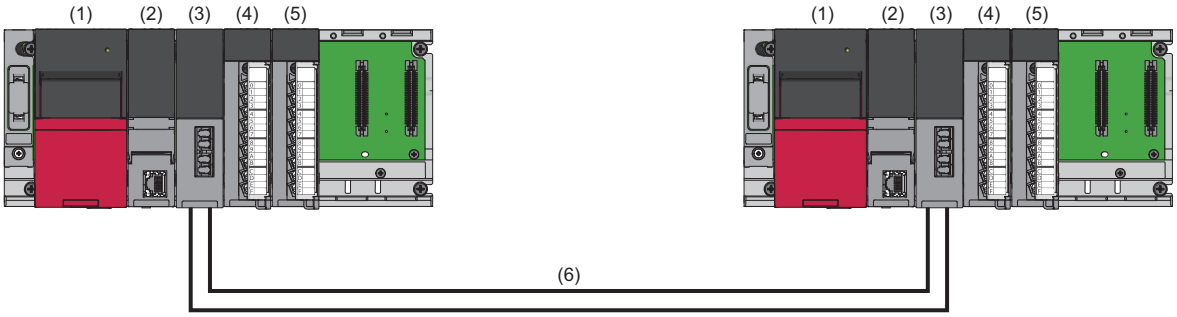
In a redundant system with redundant extension base unit, the "Control System Start-up Setting (Input (X))" is disabled.

Restriction

When using this program example, do not power off and on or reset the CPU module of each system in separate mode. Change the mode to the backup mode before powering off and on or resetting the CPU module.

System configuration

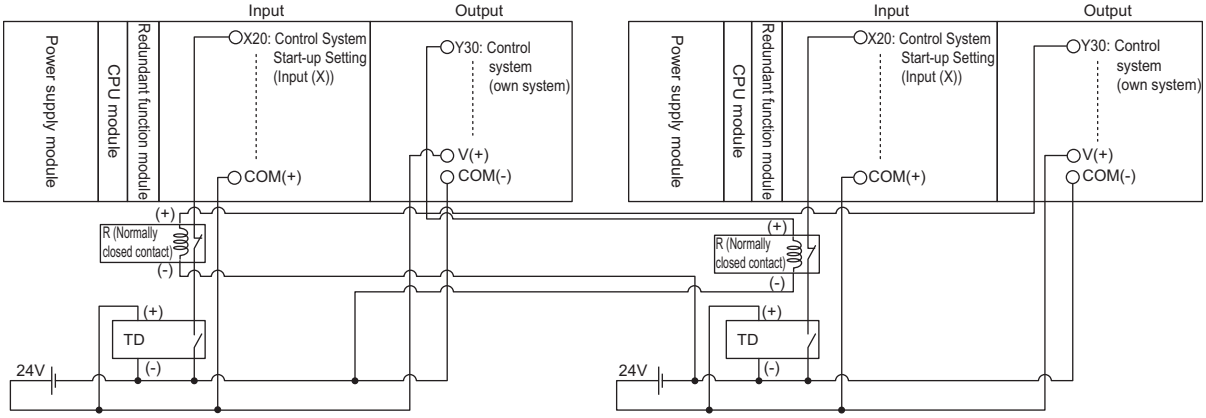
The following figure shows a configuration example of the system.



(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		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	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 "Global Device Setting" in "Device/Label Detailed Setting" as follows.

Tracking Block No. 1 File Register Setting
Total Global Device 36.2K Word File Name

No.	Device	Points (Decimal)	Start	End
1	X	12240	30	2FFF
2	Y	12224	40	2FFF

(5)

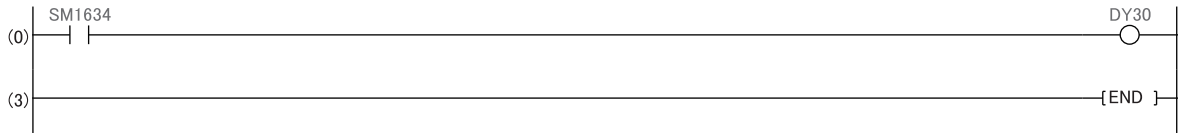
(5) Do not include X20 to X2F and Y30 to Y3F, which are used in the program example, in the tracking transfer range.

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 standby system or in the state that waits for the other system to start.

When the previous control system is started up as the control system

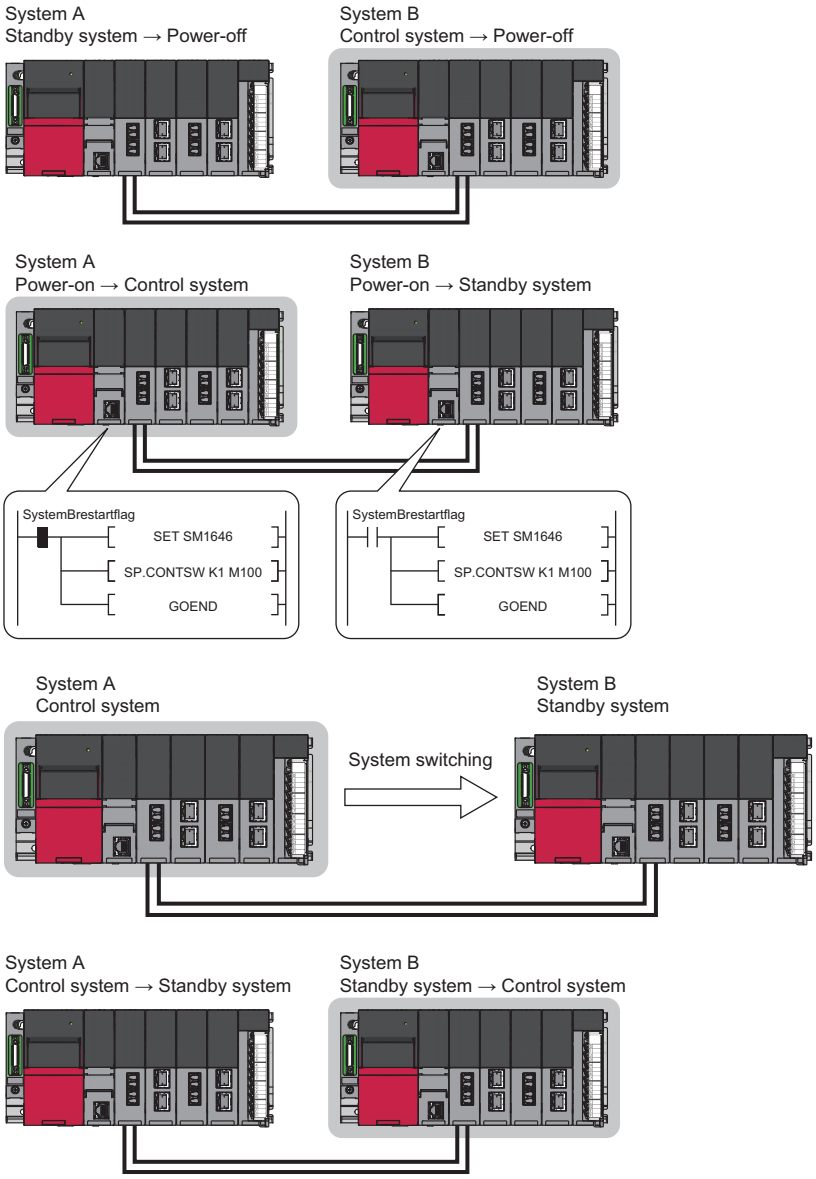
In a redundant system, when both systems are simultaneously started up, the system A always will be the control system. Even when both systems are temporarily powered off due to a power failure or other causes while the system B is operating as the control system, the system A will start up as the control system when both systems are powered on.
To start up the system B, which was operating as the control system before power-off, as the control system again, use SM1636 (Previous control system identification flag) in the program.
Note that if a network module is included in the system, wait until the network module of the other system starts up, and execute the SP.CONTSW instruction.

When network modules are not mounted on the main base units

■Program example

The following program switches the system B from the standby system to the control system by executing a system switching instruction when the operating status of the CPU module in the system A is changed to RUN for the first time. Turn on SM1646 (System switching by a user) in advance.

■Operation



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. When both systems are powered on, the system A starts up as the control system. SM1636 turns on for only one scan after the operating status of the CPU module has changed to RUN.

3. The systems are switched by the SP.CONTSW instruction.

4. The system B is switched from the standby system to the control system.

When network modules are mounted on the main base units

Wait until the network module of the system B starts up. Then, execute the system switching instruction.

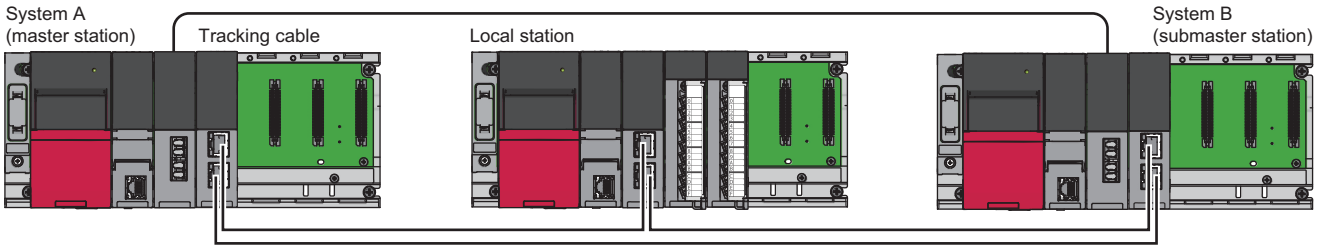
Point

When CC-Link modules are mounted on the 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.)

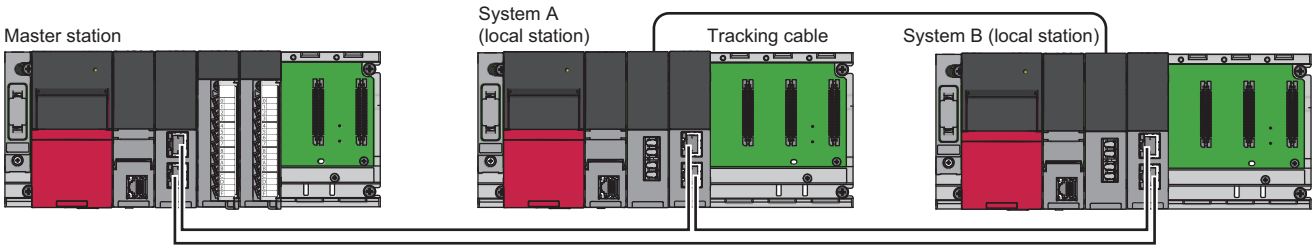
■In a redundant master station configuration or a redundant device station configuration

• System configuration

(Redundant master station)



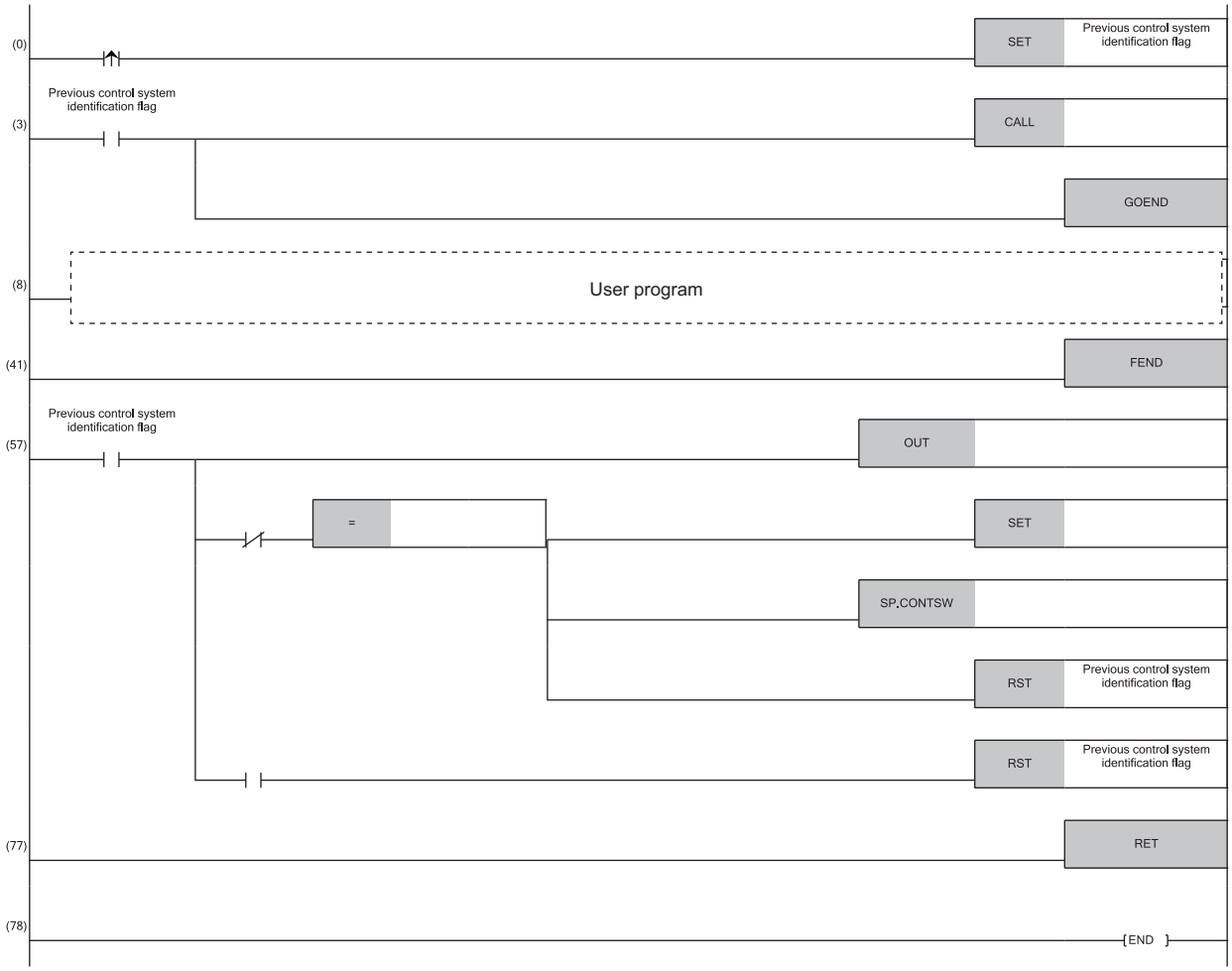
(Redundant device station)



Point

To start up the system B 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.

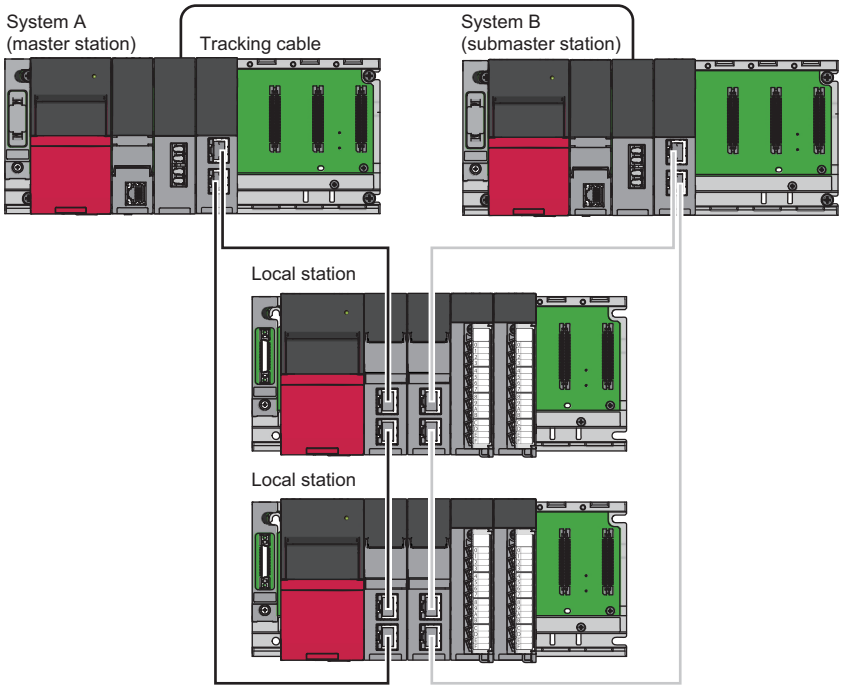
(41) Terminate the main routine program.

(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.)

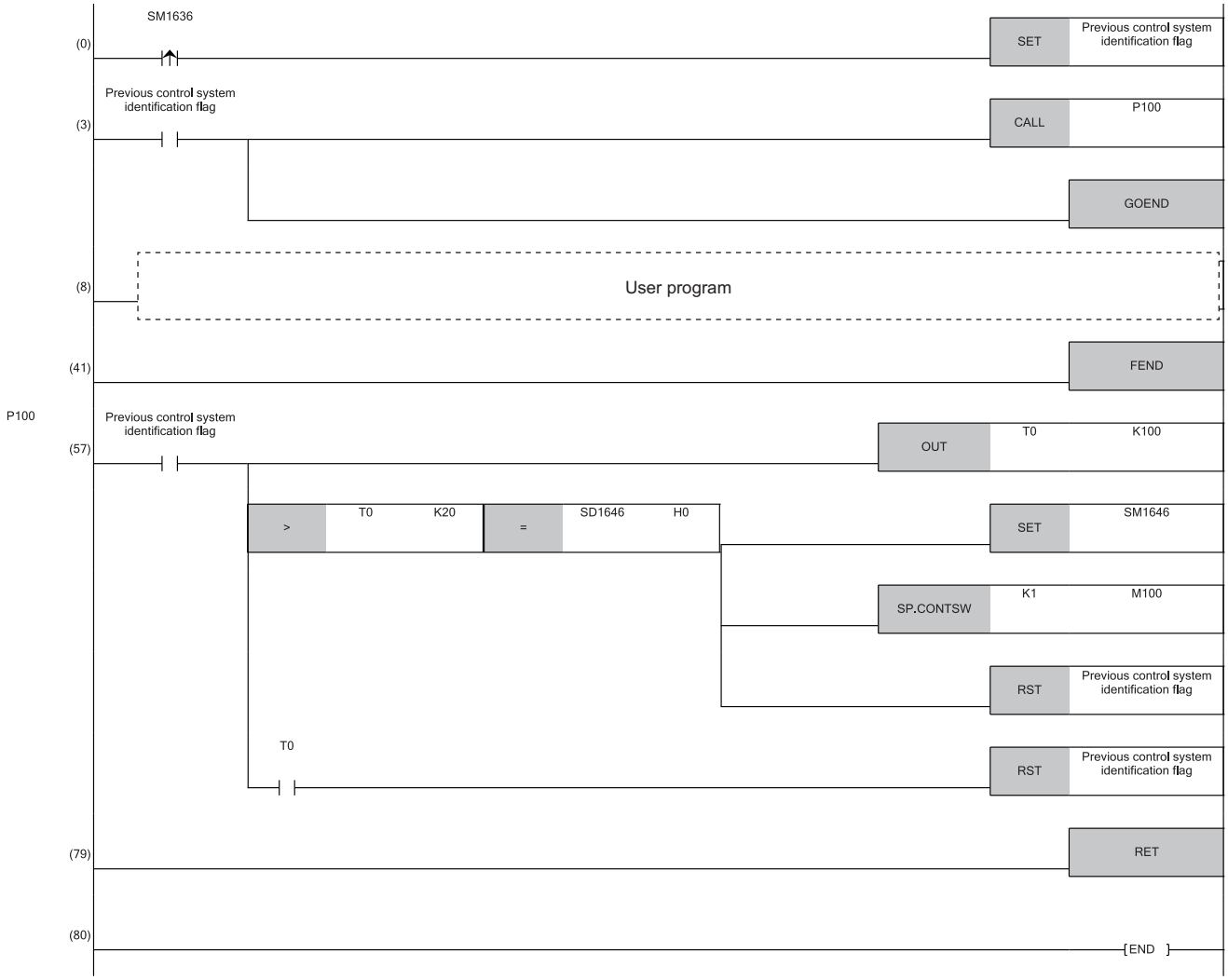
(77) Terminate the subroutine program.

■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.)

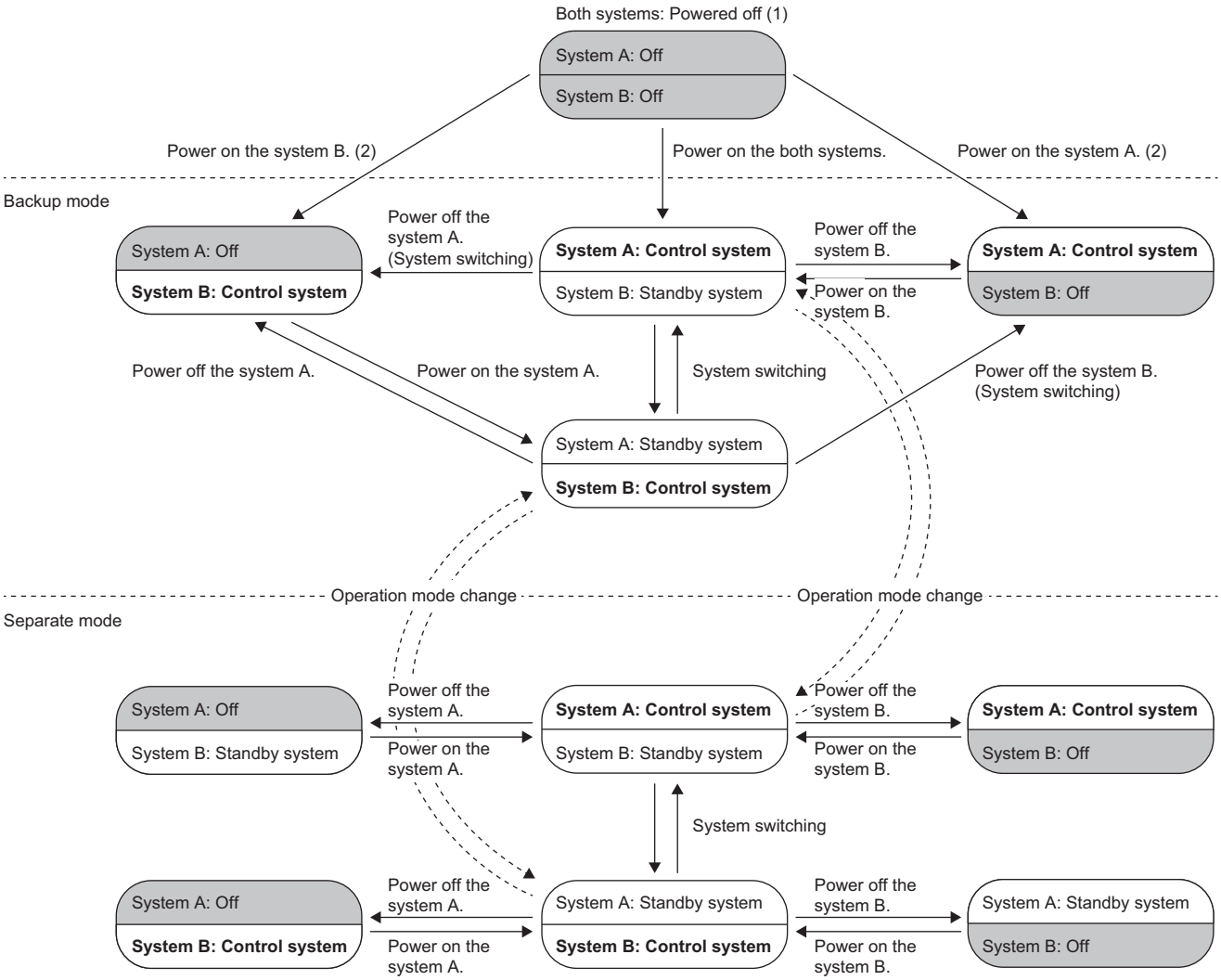
(79) Terminate the subroutine program.

■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.

11.7 State Transition of a Redundant System

The following figure shows the state transition of a redundant system after start-up based on the operation mode change and system switching.



(1) When both systems are powered off, both systems transition to the power-off state regardless of the operation mode or system type.
(2) A system that is powered on starts up as the control system. (☞ Page 163 When only one system is started up)

11.8 Access in a Redundant System with Redundant Extension Base Unit

Access to the extension base unit in a redundant system with redundant extension base unit 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 redundant extension base unit

Cable error between the main base unit of the control system 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.

Cable error between the main base unit of the standby system 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 redundant extension base units, 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 redundant extension base units 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 a cable error occurs on the inactive side between the extension base units

When the extension cable is redundant and an error occurs in the extension cable on the inactive side between redundant extension base units, 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 module of both systems.

PART 6 FUNCTIONS

This part consists of the following chapters.

12 FUNCTION LIST

13 CLOCK FUNCTION

14 WRITING DATA TO THE CPU MODULE

15 RAS FUNCTIONS

16 REMOTE OPERATION

17 BOOT OPERATION

18 MONITOR FUNCTION

19 TEST FUNCTION

20 DATA LOGGING FUNCTION

21 PID CONTROL/PROCESS CONTROL FUNCTION

22 CPU MODULE DATA BACKUP/RESTORATION FUNCTION

23 MULTIPLE CPU SYSTEM FUNCTION

24 SECURITY FUNCTION

25 ROUTING SETTING

26 REDUNDANT FUNCTION

12 FUNCTION LIST

The following table lists the functions of the CPU module.
○: Supported, ×: Not supported

Restriction

Some functions have restrictions on the firmware version of the CPU module used or the version of the engineering tool used. (☞ Page 762 Added and Enhanced Functions)

Function		Description	Availability according to operation mode		Reference
			Process mode	Redundant mode	
Constant scan		Repeatedly runs a program while retaining the scan time constant.	○	○	☞ Page 85 Constant scan
Device/label access service processing setting		Sets the time or execution timing of the device/label access service processing performed in the END processing in parameter.	○	○	☞ Page 87 Device/label access service processing setting
Device/label access service processing constant wait 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.	○	○	☞ Page 91 Device/label access service processing constant wait function
Interrupt function	Multiple interrupt function	When an interrupt occurs while an interrupt program triggered by another cause is running, stops the program if its priority is lower than that of the new interrupt, and runs the higher-priority program whenever its execution condition is satisfied.	○	○	☞ Page 129 Multiple interrupt function
Output mode setting when the status changed from STOP to RUN		Sets the output (Y) status when the operating status of the CPU module is switched from STOP to RUN.	○	○	☞ Page 137 Output mode at operating status change (STOP to RUN)
Device/label memory area setting		Sets the capacity of each area in the device/label memory.	○	○	☞ Page 142 Device/label memory area setting
Internal buffer capacity setting		Sets the capacity of the area (internal buffer) used by the system to temporarily store the results of data logging.	○	○	☞ Page 286 Internal buffer capacity setting
SD memory card forced disable		Forcibly disables an access to the SD memory card without powering off the system even when the function that accesses the SD memory card is being executed.	○	○	☞ Page 147 How to forcibly disable the SD memory card with a special relay
Clock function		Manages time for the system functions such as time stamps for the event history function and date information for the data logging function.	○	○	☞ Page 185 CLOCK FUNCTION
Writing data to the CPU module	Writing data to the programmable controller	Writes the specified data of the project with the engineering tool to the memory of the CPU module.	○	○	• ☞ Page 189 Writing Data to the Programmable Controller • 📖 GX Works3 Operating Manual
	Online change (ladder block)	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.	○	○	• ☞ Page 189 Online change (ladder block) • 📖 GX Works3 Operating Manual
	Online change (files)	Writes programs and other data to the running CPU module in units of files.	○	○	• ☞ Page 193 File batch online change • 📖 GX Works3 Operating Manual

Function		Description	Availability according to operation mode		Reference
			Process mode	Redundant mode	
RAS function	Scan monitoring function	Detects a hardware failure or program error by monitoring that the END processing is performed within a set scan time.	○	○	☞ Page 200 Scan Monitoring Function
	Self-diagnostics function	Self-diagnoses the CPU module to see whether an error exist or not.	○	○	☞ Page 202 Self-Diagnostics Function
	FB hierarchy information	Displays the FB hierarchy information of the function block with an error in the "Module Diagnostics" window.	○	○	🕮 GX Works3 Operating Manual
	Error clear	Batch-clears all the continuation errors being detected.	○	○	☞ Page 208 Error Clear
	Event history function	Collects operations executed and errors detected from the modules, and saves them in the CPU module. The saved logs can be checked in chronological order.	○	○	☞ Page 211 Event History Function
	Online module change function	Replaces a module without stopping the system.	○	○	🕮 MELSEC iQ-R Online Module Change Manual
	Program cache memory auto recovery function	Automatically detects and recovers the data stored in the program cache memory of the CPU module at the time of program execution if the data is overwritten due to the factors such as excessive electrical noise.	○	○	☞ Page 218 Program Cache Memory Auto Recovery Function
Remote operation		Remotely controls the operating status of the CPU module without switch operation.	○	○	☞ Page 219 REMOTE OPERATION
Boot operation		Transfers files in the SD memory card to the built-in memory of the CPU module when the CPU module is powered off and on or reset.	○	○	☞ Page 225 BOOT OPERATION
Monitor function	Circuit monitor	Checks the status of a running program on the program editor.	○	○	🕮 GX Works3 Operating Manual
	Device/buffer memory batch monitor	Monitors values in devices and buffer memory using the engineering tool connected.	○	○	🕮 GX Works3 Operating Manual
	Watch	Registers devices and labels, and checks the current values.	○	○	🕮 GX Works3 Operating Manual
	Program monitor list	Monitors the execution time of each program and the number of executions using the engineering tool connected.	○	○	🕮 GX Works3 Operating Manual
	Interrupt program monitor list	Monitors the number of executions of interrupt programs using the engineering tool connected.	○	○	🕮 GX Works3 Operating Manual
	Module error information collection function	Stores the error information of the module in the special register areas.	○	○	☞ Page 232 Module Error Information Collection Function
Test function	External input/ output forced on/ off function	Forcibly turns on/off the external input/output from the engineering tool.	○	○	• ☞ Page 235 External Input/Output Forced On/Off Function • 🕮 GX Works3 Operating Manual
	Device test with execution condition	Changes a device value for the specified step of a program, using the engineering tool.	○	○	• ☞ Page 244 Device Test with Execution Conditions • 🕮 GX Works3 Operating Manual
Data logging function		Collects data at a specified interval or a desired timing, and stores them as a file.	○	○	• ☞ Page 258 DATA LOGGING FUNCTION • 🕮CPU Module Logging Configuration Tool Version 1 Operating Manual
SFC function		Executes programs written in sequential function chart (SFC).	○	○	🕮 MELSEC iQ-R Programming Manual (Program Design)
PID control function		Performs PID control by using the PID operation instruction.	○	○	🕮 MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

Function		Description	Availability according to operation mode		Reference
			Process mode	Redundant mode	
Process control function		A process control program can be easily created by using process control function blocks.	○	○	MELSEC iQ-R Programming Manual (Process Control Function Blocks/ Instructions)
CPU module data backup/ restoration function		Backs up data such as program files, parameter files, and device/label data files in a CPU module to an SD memory card. The data backed up can be restored as necessary.	○	○	Page 311 CPU MODULE DATA BACKUP/ RESTORATION FUNCTION
Multiple CPU system function	Out-of-group I/O fetch	Enables loading input (DX) and buffer memory of the intelligent function module by accessing the non-controlled module of the CPU module.	○	×	Page 338 Out-of-group I/O Fetch
	Multiple CPU synchronized startup	Synchronizes the startup of CPU modules in a multiple CPU system so that operations of all the CPU modules start at the unified time.	○	×	Page 342 Synchronous startup setting
	Data communications between CPU modules	Communicates data between CPU modules in a multiple CPU system.	○	×	Page 347 Data Communication Between CPU Modules
	Multiple CPU synchronous interrupt	Executes an interrupt program at the fixed communication timing set in parameter.	○	×	Page 366 Multiple CPU Synchronous Interrupt
Security function	Block password function	Prevents unauthorized access to programs (in units of POUs).	○	○	GX Works3 Operating Manual
	Security key authentication function	Prevents unauthorized access to programs (in units of program files) or unauthorized execution of programs.	○	○	GX Works3 Operating Manual
	File password function	Prevents unauthorized reading/writing of data from/to files.	○	○	GX Works3 Operating Manual
	IP filter function	Identifies the IP address of external devices over Ethernet, and blocks access from an invalid IP address.	○	○	• GX Works3 Operating Manual • MELSEC iQ-R Ethernet/ CC-Link IE User's Manual (Startup)
	Remote password function	Limits access from external devices to the CPU module to the specific communication route via Ethernet.	○	○	• GX Works3 Operating Manual • MELSEC iQ-R Ethernet/ CC-Link IE User's Manual (Startup)
Label initialization function	Label initialization after converting all programs (reassignment)	All the 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.	○	○	Page 534 Initialization of Labels After Rebuilt All (Reassignment)
	Label initial value reflection setting	With the default settings, 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.	○	○	Page 536 Label Initial Value Reflection Setting
Routing setting		Sets the communication path required for transient communication to the stations on the different network.	○	○	Page 370 ROUTING SETTING
Label access setting from external device		Enables communications specifying the global label name from the GOT and external devices using SLMP.	○	○	Page 518 Label Access Setting from External Device
Latch function	Latch with a battery	Holds the device/label data in the CPU module even at power-off.	○	○	Page 523 Latch with Battery
Initial device/label value setting		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.	○	○	Page 530 DEVICE/LABEL INITIAL VALUE SETTINGS

Function		Description	Availability according to operation mode		Reference
			Process mode	Redundant mode	
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 374 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 377 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 391 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 409 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 417 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) respectively by executing a program that diagnoses external devices and networks in both systems.	×	○	☞ Page 422 Program Execution in Both Systems
	Redundant system operation setting	Sets the redundant system operation in the redundant system settings of the CPU parameter.	×	○	☞ Page 429 Redundant System Operation Setting
	Redundant function module communication test	Checks the redundant function module itself for errors when its communications are unstable.	×	○	☞ Page 434 Redundant Function Module Communication Test
	Settings for redundant system with redundant extension base unit	Sets the operation of the redundant system with redundant extension base unit in the redundant system settings of the CPU parameter.	×	○	☞ Page 435 Settings for Redundant System with Redundant Extension Base Unit
	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 436 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 561 Replacement/ Addition of Extension Cables (Online)
	Control system start-up at one system error	When one system detects an error, such as a CPU module failure, module failure, and module disconnection, at the power-on or reset and the redundant system cannot operate, the other system (normal system) starts as a control system.	×	○	☞ Page 438 Control System Start-up at One System Error
Ethernet function		Accesses the CPU module over Ethernet. Using this function, the CPU module can connect to MELSOFT products and GOTs, perform socket communications, and transfer FTP files.	○	○	🕮 MELSEC iQ-R Ethernet/ CC-Link IE User's Manual (Startup)
SLMP communication function		Enables accesses to devices/labels and remote operations from external devices other than the engineering tool by sending request messages using SLMP.	○	○	🕮 SLMP Reference Manual
Inter-module synchronization function		Controls multiple modules synchronously.	○	×	🕮 MELSEC iQ-R Inter-Module Synchronization Function Reference Manual

Function		Description	Availability according to operation mode		Reference
			Process mode	Redundant mode	
iQ Sensor Solution function		Functions provided by iQ Sensor Solution	○	○	📖 iQ Sensor Solution Reference Manual
Firmware update function	Update using the engineering tool	Enables users to update firmware versions of CPU modules and intelligent function modules using the engineering tool.	○	○	📖 MELSEC iQ-R Module Configuration Manual
	Update using an SD memory card	Enables users to update firmware versions of CPU modules and intelligent function modules using an SD memory card.	○	○	📖 MELSEC iQ-R Module Configuration Manual

13 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.

13.1 Time Setting

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.
The time of the standby system is synchronized with the time of the control system when the redundant system is in backup mode. The time setting using the time setting function (SNTP client) is available only for the control system.

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 185 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

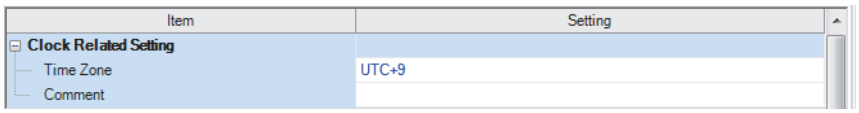
Date	Write operation to the CPU module	CPU module operating status
February 30	Executed	An error is not detected.
32nd of month 13	Not executed	• When the DATEWR instruction is executed, "Operation error" (error code: 3405H) is detected. • When SM210 is turned on, SM211 turns on.

13.2 Setting Time Zone

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.)

13.3 System Clock

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 666 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.
- In a redundant system, after system switching, SM420 (User timing clock No.0) to SM424 (User timing clock No.4) remain off in the CPU module of the new control system. To use SM420 to SM424 in the CPU module of the new control system, execute the DUTY instruction again.

Special register used for system clock

Special register used for system clock are as follows (☞ Page 698 System clock)

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

14 WRITING DATA TO THE CPU MODULE

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

14.1 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

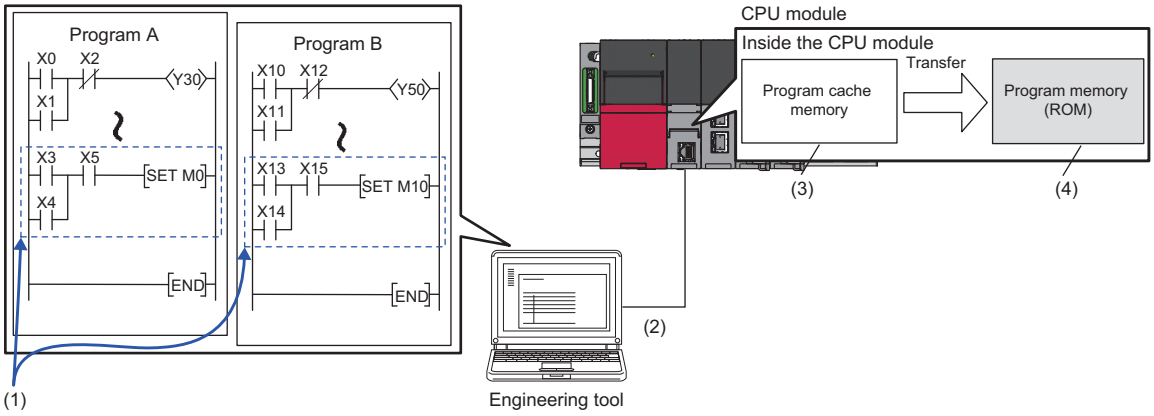
14.2 Online Change

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 189 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 193 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 146 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.

Firmware version of the CPU module	Required data memory space
"13" or later	Total increased size of the relevant program restoration information*1
"12" or earlier	Size of the program restoration information

*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 R120PCPU with a firmware version "12" 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	○	△*1
	Local label	○	△*1
FB definition	Local label	○*2	△*1
FUN definition	Local label	×*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.

In redundant mode

- 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.

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 762 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.

■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 redundant mode, configure this setting for both systems. Files can be written only when the firmware version of the CPU modules in both systems supports writing of the FB files and the global label setting file online.

■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.

14.3 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

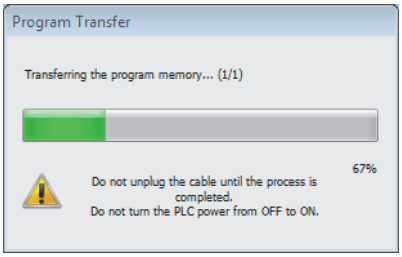
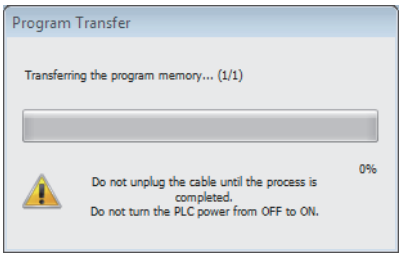
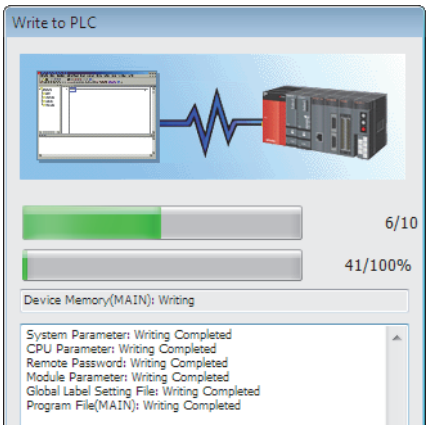
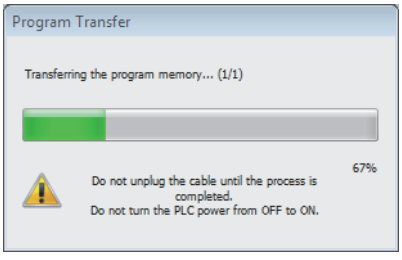
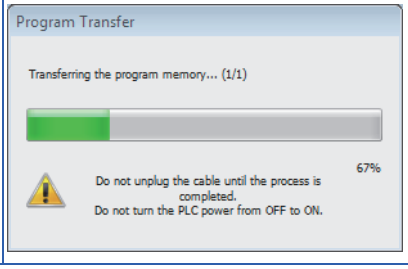
■Operations cannot be executed simultaneously

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

■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%.
		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.

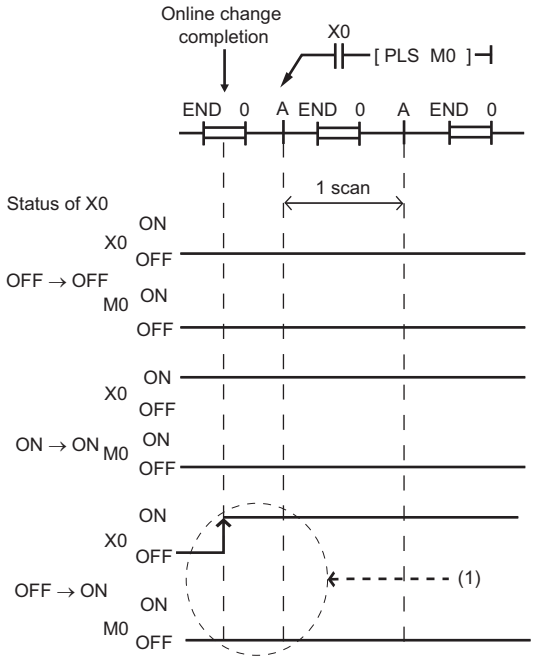
■Instructions which do not operate correctly

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

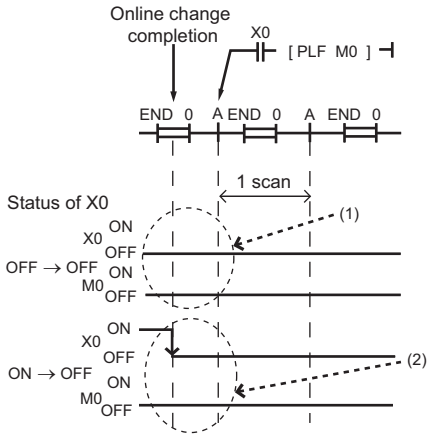
When a rising instruction exists within the range to be changed, the rising instruction will not be executed even if the execution condition (OFF to ON) is satisfied after completion of the online change (ladder block) or writing data to the programmable controller.



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

• Falling instruction

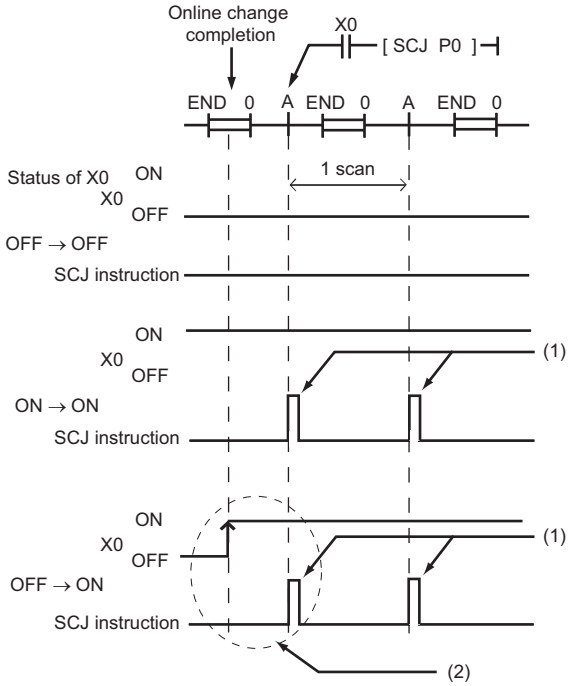
When a falling instruction exists within the range to be changed, the falling instruction will not be executed even if the execution condition (ON to OFF) is satisfied after completion of the online change (ladder block) or writing data to the programmable controller.



(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

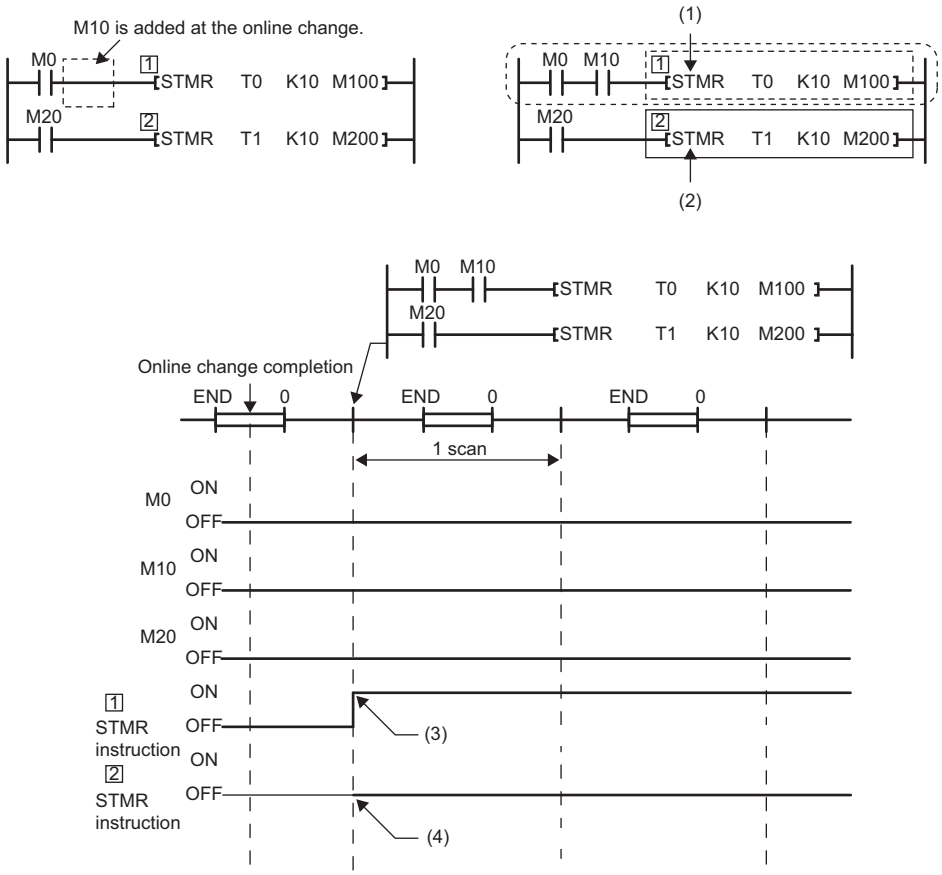
If an SCJ instruction exists within the range to be changed and the execution condition is satisfied at the completion of the online change (ladder block) or writing data to the programmable controller, the program will jump without waiting for a single scan.



(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 203 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.

■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 190 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.

In redundant mode

During an online change, avoid the following.

- Switching the operating status of the CPU module from STOP (PAUSE) to RUN
- System switching
- Changing the operation mode of the redundant system to backup mode
- Disconnecting tracking cables
- Powering off or resetting the CPU module of 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 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 CPU module of the control system 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 CPU module of the control system, 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 online 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 CPU module of the standby system and execute the memory copy again. (☞ Page 409 Memory Copy from Control System to Standby System)

4. Power off and on or reset the CPU module of the standby system.

■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 and retry the online change.

[Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Online Program Change" ⇨ "Operational Setting" ⇨ "Write Program Restore Information"

■Writing data to the CPU module in a redundant system with redundant extension base unit

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.

15 RAS FUNCTIONS

15.1 Scan Monitoring Function

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

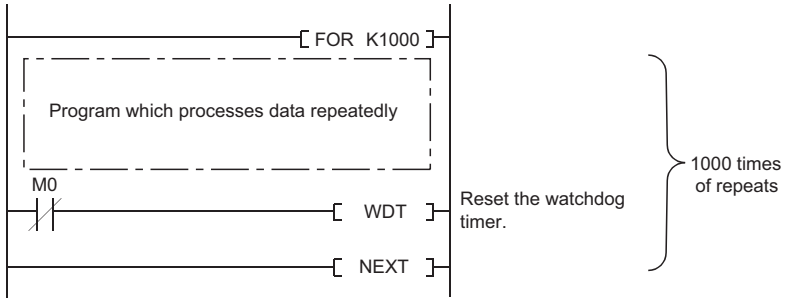
The following lists the precautions on the scan monitoring function.

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.

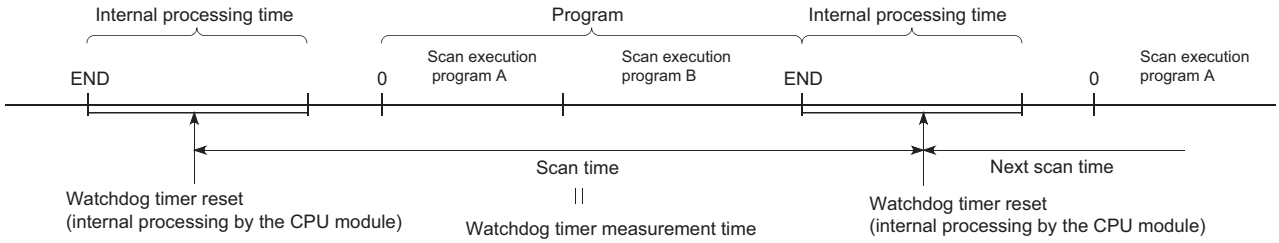
Resetting the watchdog timer when repeatedly executing a program

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.



In redundant mode

- 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 to be received. Thus, no error is detected even if the scan time monitoring time has elapsed while the standby system is waiting for tracking data to be received. Therefore, a time taken for tracking data to be received does not need to be considered in the scan time monitoring time setting.

15.2 Self-Diagnostics Function

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 code for the error content of 17th piece on and after will not be stored.)

Check method using LED

The error occurrence conditions can be checked through the lighting conditions of ERROR LED. (☞ Page 545 LED status of the CPU module)

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 is logged in the event history. (☞ Page 211 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

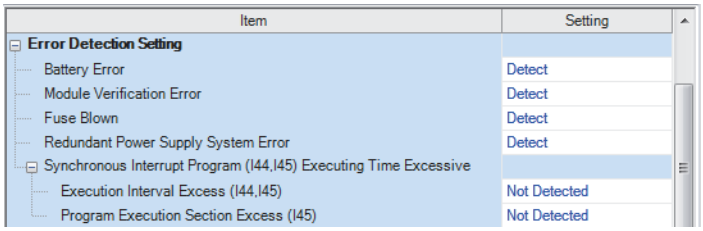
If an error has been detected, the program other than programs (instruction) where an error was generated is executed.

Error detection setting

Set whether or not to detect errors.

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

Window



Displayed items

Item		Description	Setting range	Default
Battery Error		Sets whether or not to detect the battery error.	• Detect • Not Detected	Detect
Module Verification Error*1		Sets whether or not to detect a module verification error.		Detect
Fuse Blown		Sets whether or not to detect a fuse blown of the controlled module.		
Redundant Power Supply System Error*2		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.*3		
Synchronous Interrupt Program (I44, I45) Executing Time Excessive	Execution Interval Excess (I44, I45)	Sets whether or not to detect the excessive execution time of synchronous interrupt program (I44, I45).		Not Detected
	Program Execution Section Excess (I45)	Sets whether or not to detect the program execution section excess errors of synchronous interrupt program (I45).		

*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 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.

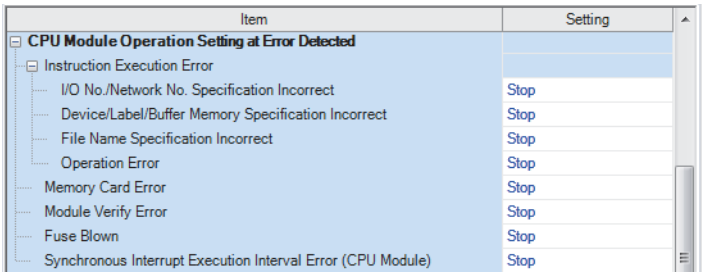
Error name	Error code
Power shutoff (either of the redundant power supply modules)	1010H
Failure (either of the redundant power supply modules)	1020H
Battery error	1090H
Module verification error	2400H, 2401H
Fuse blown error	2420H

CPU module operation 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 Error	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		Sets the CPU module operation upon a memory card error.		
Module Verification 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 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 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.

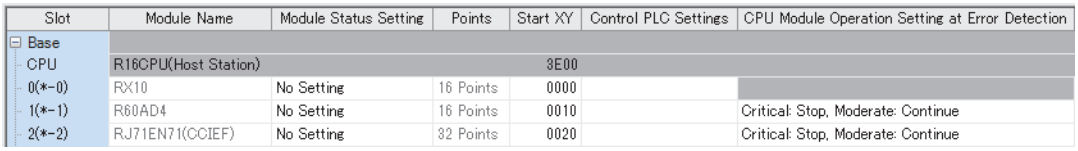
Error name	Error code
Memory card error	2120H, 2121H
Module verification error	2400H, 2401H
Fuse blown error	2420H
Inter-module synchronization signal error	2610H
Multiple CPU synchronization signal error	2630H
I/O number or network number specification error	2800H, 2801H, 2802H, 2803H, 2804H, 2805H, 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, 3424H, 3425H, 3426H, 3427H, 3430H, 3440H, 3441H, 34A0H

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



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 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

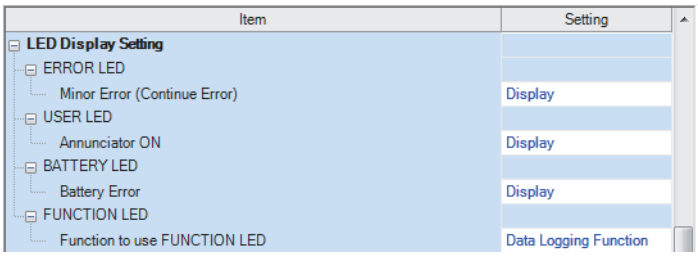
For the multiple CPU system configuration, it can be specified whether or not to stop all CPU modules when a major or moderate error occurs on a CPU module. (☞ Page 341 Stop setting)

LED display setting

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

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

Window



Displayed items

Item		Description	Setting range	Default
ERROR LED	Minor Error (Continue Error)	Set whether or not to display the ERROR LED when a minor error occurs.	• Display • Do Not Display	Display
USER LED	Annunciator ON	Set whether the USER LED is displayed or not when annunciator (F) ON is detected.		
BATTERY LED	Battery Error	Set whether the BATTERY LED is displayed or not when a battery error occurs.		

Error detection invalidation setting

Turning on the target bit of SD49 (Error detection invalidation setting) disables detection of the corresponding continuation error.*1 (☞ Page 683 Diagnostic information)

*1 When using the error detection invalidation setting, check the version of the CPU module used. (☞ Page 762 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.

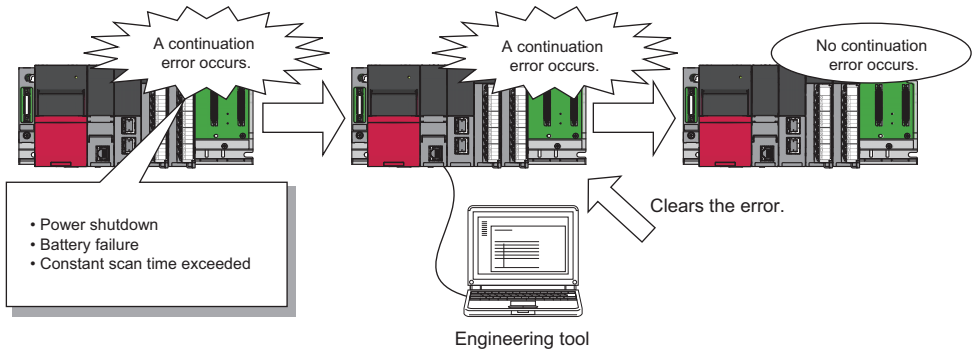
Applicable errors to the error detection invalidation setting

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

15.3 Error Clear

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
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

Error name	Error code
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
System consistency check error (system configuration)	3700H
System consistency check error (SD memory card mounting status)	3710H
System consistency check error (memory card write protect switch status)	3711H
Redundant system configuration error	3788H

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.

Clearing errors in the CPU module of the standby system from the CPU module of the control system

Errors in the standby system can be cleared from the CPU module of the control system by using SM1679 (Error reset (the other system)) in a program or external device.

Error clearing procedure

Use SM1679 to clear errors.

1. Eliminate the causes of all the continuation errors that have occurred in the standby system.
2. Turn on SM1679 of the CPU module of the control system to clear the errors. When multiple continuation errors have occurred, all the errors are cleared at once.

Precautions

- The error clear operation with SM1679 can be performed by the CPU module of the control system only. A continuation error in the CPU module of the standby system cannot be cleared by turning on SM1679 of the CPU module of the standby system.
- The 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 error clear target CPU module cannot be eliminated even though the error clear operation with SM1679 is performed.
- If the error cause has not been eliminated completely after the error clear operation, the same error will be detected again.
- The error clear processing is performed in the END processing. To clear an error, execute the END instruction while SM1679 is on.

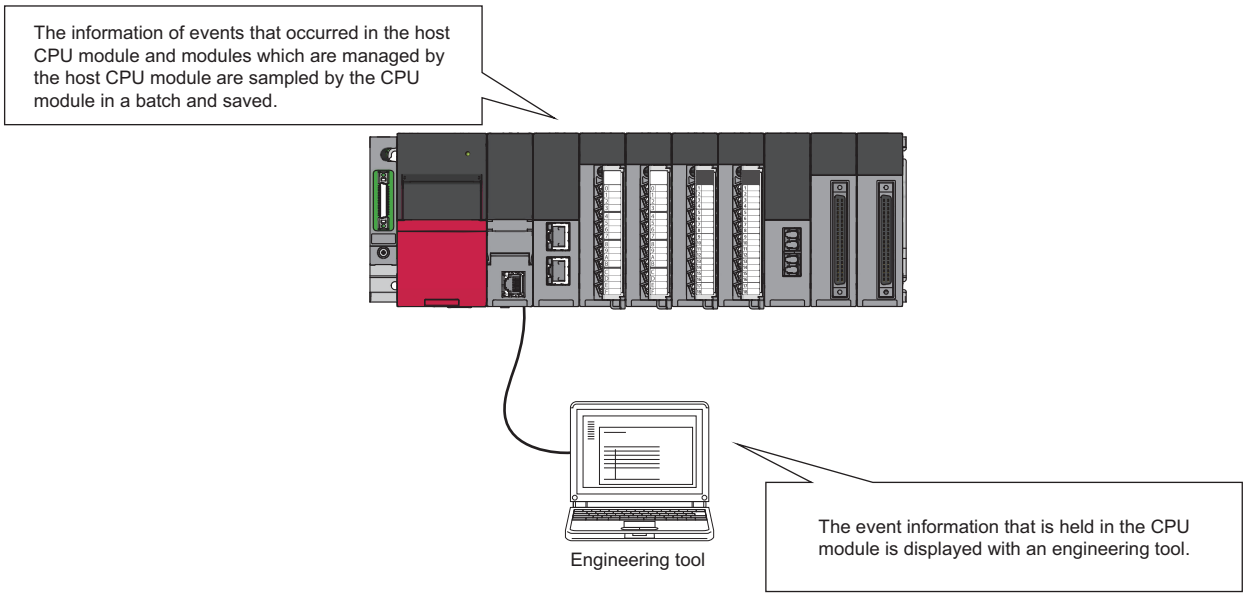
15.4 Event History Function

The CPU module collects and stores event information from each module, such as errors detected by the module, operations performed on the module, and network errors.*1

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.

In a redundant system, events that occur in a module on an 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.



*1 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

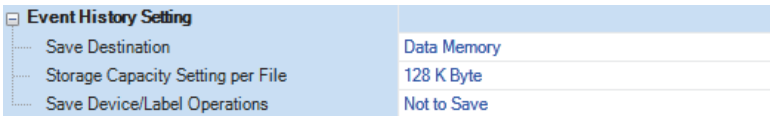
The event history information is constantly collected regardless of the operating state of the CPU module. There are occasions, however, when the event history information cannot be collected due to a major error in a module, a base unit error, a cable failure, or some other cause.

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 213 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 213 Storage memory)	• Data Memory • Memory Card	Data Memory
Set Save Volume of Per File	Specify the storage capacity per event history file. (☞ Page 213 File size)	1 to 2048K bytes (in 1K bytes)	128K Byte

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 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

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.

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 212 Event history setting)

■Storage memory

Choose either the data memory or SD memory card.

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.

■File size

The size for event history files can be changed in event history setting (☞ Page 212 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

Element		Size (byte)
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 a CPU module with firmware version "06" or later.

■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 215 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 215 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 762 Added and Enhanced Functions

*2 Event history logging is not restricted (stopped) for CPU modules with firmware version earlier than "22".

In a redundant system, the CPU module of the control system monitors the logging of events that occur in a module on an extension base unit and restricts the amount of events to save.

When systems are switched, the event history logging restriction state is cleared. Therefore, even if the old control system is in the logging restriction state before system switching, the new control system is in the state of no event history logging restriction.

■Events not supported by the logging restriction

The following events are not subject to the logging restriction.*1

- Events of the self-diagnostics for major errors from the intelligent function module

*1 For CPU modules with firmware version earlier than "22", CPU module events are not subjected to the logging restriction, either.

■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 647 Event List)

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 215 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.

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).
Set Save Volume of 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

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.

15.5 Program Cache Memory Auto Recovery Function

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.

16 REMOTE OPERATION

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

16.1 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.

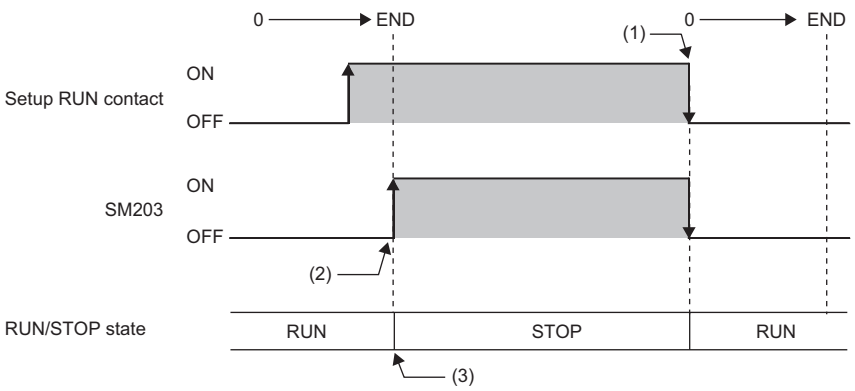
Executing method

The following methods are available to execute remote RUN/STOP:

Using a contact

Set RUN contact in the RUN-PAUSE contact setting. (☞ Page 222 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.

Point

In redundant mode, 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. This is because an operating status mismatch occurs depending on the timing.

Using an engineering tool

Perform remote RUN/STOP with the remote operation of the engineering tool. (GX Works3 Operating Manual)

[Online] ⇨ [Remote Operation]

In a redundant system, the operation is as follows.

■Backup mode

Selection in "Specify Execution Target"	Description
Currently Specified Station All Stations Specified Specify Group No.	The remote operation is performed only on the CPU module of a system specified in the transfer setup of the engineering tool.
Specify Both Systems	Remote operation is performed on the CPU modules of both systems.

Point

When "Specify Redundant CPU" is set to "Not Specified" in the transfer setup, select "All Stations Specified" or "Specify Group No." to perform remote operation on the CPU modules of both systems.

■Separate mode

The CPU module operating status of a system specified in the transfer setup of the engineering tool can be changed.

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 307 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.

16.2 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.

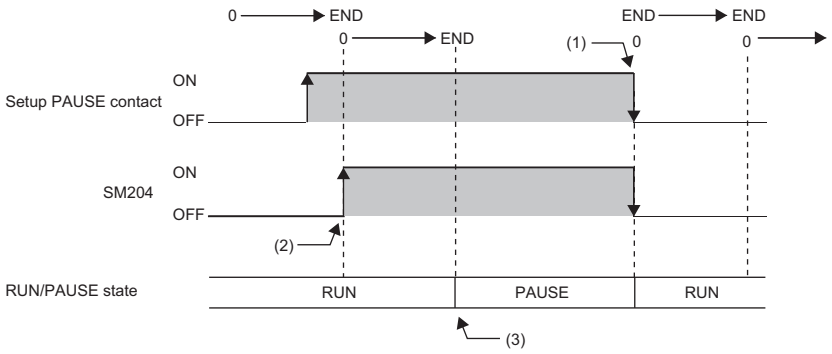
Executing method

The following methods are available to execute remote PAUSE:

Using a contact

Set PAUSE contact in "RUN-PAUSE Contact Setting". (☞ Page 222 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.

Point

In redundant mode, 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. This is because an operating status mismatch occurs depending on the timing.

Using an engineering tool

Perform remote PAUSE with the remote operation of the engineering tool. (GX Works3 Operating Manual)

The operation method in a redundant system is the same as that for remote RUN/STOP. (☞ Page 220 Using an engineering tool)

By an external device using SLMP

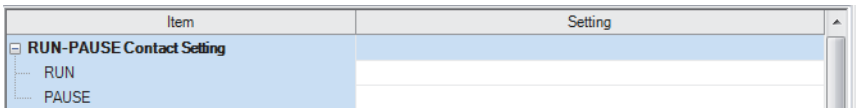
Perform remote PAUSE with the SLMP command. (SLMP Reference Manual)

16.3 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	Set the contact that controls RUN for the CPU module.	X0 to X2FFF	—
PAUSE	Set the contact that controls PAUSE for the CPU module.	X0 to X2FFF	—

Point

When setting the PAUSE contact, be sure to also set the RUN contact. (The PAUSE contact cannot be set alone.)

16.4 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

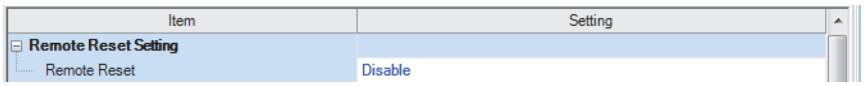
To perform the remote RESET operation when "Execution Target" of the remote operation is set to "All Stations Specified" or "Specify Group No.", ensure that the CPU module set in "Specify Connection Destination Connection" of the engineering tool is in the STOP state. (Even if a CPU module in the STOP 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	Set 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)

In a redundant system, the operation is as follows.

■Backup mode

The CPU modules of both systems can be reset by performing remote RESET operation on the CPU module of the control system. Only the CPU module of the standby system can be reset by performing remote RESET operation on the CPU module of the standby system.

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".

However, if remote RESET operation is performed on both systems in backup mode, note the following.

- When the CPU module of the control system is in the STOP state and the CPU module of the standby system is in the RUN state, performing remote RESET operation on the CPU module of the control system causes system switching. To prevent system switching in remote RESET operation, perform remote RESET operation after setting both of the CPU modules to the STOP state.
- When remote operation is performed on the CPU module of the control system or standby system through another path, the CPU module of the standby system cannot be reset by performing remote RESET operation on the CPU module of the control system. To perform 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 CPU module of the standby system. Cancel the remote operation first and perform remote RESET operation.
- When performing 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]

- 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.

■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.

By an external device using SLMP

Perform remote RESET with the SLMP command. (📖 SLMP Reference Manual)

16.5 Precautions for Redundant System

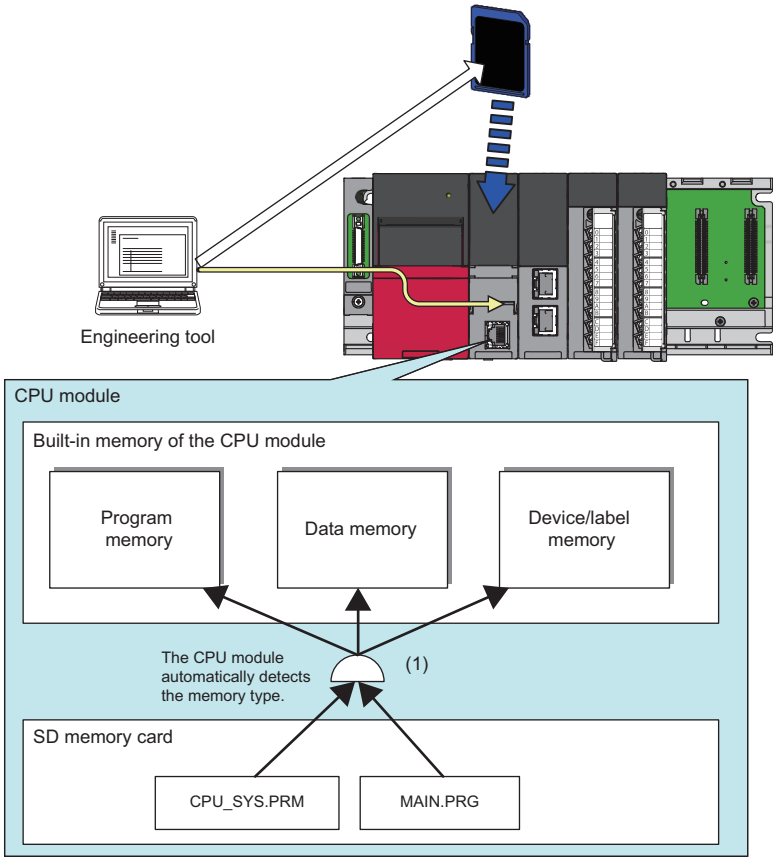
The following describes the precautions on the remote operation in the redundant system.

- 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 remote RESET is performed for both systems when the load of each CPU module is high in a redundant system with redundant extension base unit, 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 remote RESET for both systems again.

17 BOOT OPERATION

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 off and on or is reset.



17.1 Boot Operation Procedure

The boot operation procedure is as follows.

1. Configure the boot setting. (☞ Page 227 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 229 Writing Boot Settings and Boot Files)
4. Power off and on, or reset the CPU module.

17.2 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

17.3 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.

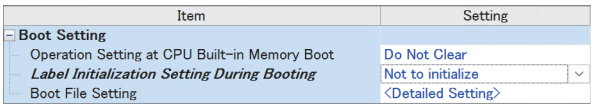
17.4 Configuring the Boot Setting

Configure the necessary settings for the boot operation.

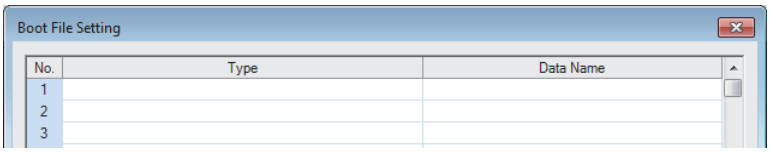
[Memory Card Parameters] ⇨ [Boot Setting]

Operating procedure

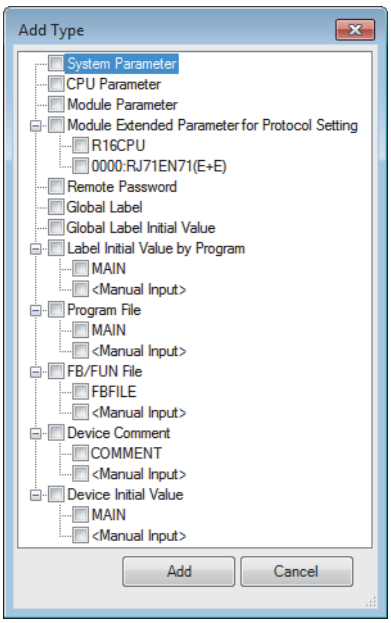
"Boot Setting" window



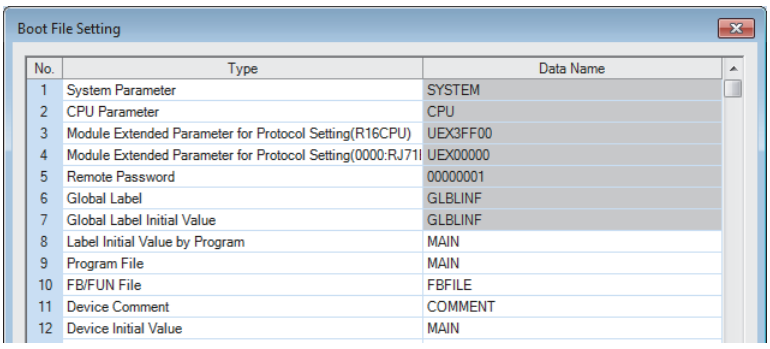
"Boot File Setting" window



"Add Type" window



"Boot File Setting" window



1. Click "Detailed Setting" on the "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.

3. Select type for the boot file. (Multiple selection is possible.)

4. Set the data name (file name).

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
Label Initialization Setting During Booting*1	Sets whether to initialize the latch type label value during the boot operation. (☞ Page 228 Label Initialization Setting During Booting)	• Not to initialize • Initialize	Not to initialize
Boot File Setting	Sets the file used for boot operation from the SD memory card.	—	—

*1 When using this setting, check the versions of the CPU module and engineering tool. (☞ Page 762 Added and Enhanced Functions)

Label Initialization Setting During Booting

The latch type label can be initialized during the boot operation (by reflecting the initial value, or clearing the label value to zero if no initial value is set).

Point

In a system where the process control extension setting is enabled or "Label Initial Value Reflection Setting at STOP to RUN" is disabled, perform the boot operation with this setting to reflect the label initial value during the boot operation.

■Latch type label value operation

When "Label Initialization Setting During Booting" is set to "Initialize", the latch type label will be initialized (by reflecting the initial value, or clearing it to zero if no initial value is set) during the boot operation regardless of the "Label Initial Value Reflection Setting at STOP to RUN" setting. The following table lists the operations of the latch type label value during the boot operation when "Label Initialization Setting During Booting" and "Label Initial Value Reflection Setting at STOP to RUN" are combined.

Label Initialization Setting During Booting	Label Initial Value Reflection Setting at STOP to RUN	Label initial value	Latch type label value operation
Not to initialize (default)*1	Disable (default)	Initial value provided	Value held
		No initial value provided	Value held
	Enable	Initial value provided	Initial value
		No initial value provided	Value held
Initialize*2	Disable (default)	Initial value provided	Initial value
		No initial value provided	Zero clear
	Enable	Initial value provided	Initial value
		No initial value provided	Zero clear

*1 When all data is rebuilt (reassigned) and written to the programmable controller, and then the power is turned off and on without the operating status change from STOP to RUN, the latch type label is initialized during the boot operation.

*2 Since the labels assigned to devices are assigned to device areas, they are not initialized.

■Non-latch type label value operation

This setting does not affect the operation of the non-latched label value.

■Precautions

In a system where "Label Initial Value Reflection Setting at STOP to RUN" is disabled because the process control extension setting is enabled, the following setting is recommended for the boot operation when the label initial value is reflected during the boot operation after the label initial value is changed.

- In "Boot File Setting" of the boot settings of the memory card parameters, set the global label initial value, the label initial value for each program (local label initial value file), the global label (global label setting file), the program file, and the FB/FUN file.
- Set "Label Initialization Setting During Booting" in the boot settings of the memory card parameters to "Initialize".
- Set "Target Memory" in the initial value settings of the CPU parameters to "Data Memory".

If the SD memory card is left mounted after the boot operation, the latch type label will be initialized every time the CPU module is powered off and on. Therefore, when initialization is not desired, remove the SD memory card from the CPU module.

17.5 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

17.6 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 existing	—		Yes
	Not set	Existing	Set		No
			Not set		Yes
		Not existing	—		Yes

17.7 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.

Redundant system

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 off and 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.

■In redundant system with redundant extension base unit

When starting up the CPU modules of both systems simultaneously by boot operation, 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.

18 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.	
Module error information collection function	Stores the error information of the module in the special register areas.	Page 232 Module Error Information Collection Function

Point

In a redundant system with redundant extension base unit, if systems are switched while the monitoring/test function is being used on the modules on the extension base units, the following operations are performed.

- If monitoring is performed 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.

18.1 Module Error Information Collection Function

This function collects error information from modules, including the CPU module, every 5 seconds and stores them in SD3696 to SD3825 (module error information) during END processing. These operations enable easy display of module error information equivalent to that of a system monitor on an external device such as GOT.
Also, collection and storage in the special register areas can be temporarily stopped by turning on and off SM3696 (module error information update setting). (The default is set to on: update.)
When SM3696 is turned off, module error information is not collected and the values of SD3696 to SD3825 are not updated. Also, when SM3696 is turned on and off, the values of SD3696 to SD3825 are not cleared to zero and the data is held.

Point

Use this function to check the error code detected by the module using an external device, or to acquire the module error information regardless of the CPU module operating status (including when a stop error occurs). Use the UNIINFRD instruction when acquiring the latest error status of the module from the program and performing the control immediately after a module error is detected. (MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))

Restriction

Before using the module error information collection function, check the version of the CPU module. (Page 762 Added and Enhanced Functions)

- Comparison of module error information collection function and UNIINFRD instruction function

Function	Error code acquisition	Collection timing	CPU module operating status and collection availability
Module error information collection function	○ (Acquisition possible)	Five second intervals (fixed)	RUN/PAUSE/STOP (including stop error): Collection possible
UNIINFRD instruction	× (Acquisition not possible)	At instruction execution	• RUN: Collection possible • PAUSE/STOP (including stop error): Collection not possible

Function details

This section describes details about the module error information collection function.

■When there is an empty slot or missing module

When there is an empty slot or a missing module (excluding the case of removal of a module during online module change), the latest error code and 0000H (as the module error status) are stored.

■When module error information cannot be collected

In the following cases, module error information for the target module cannot be collected. (No error code is stored in the special register area, and bit 13: module access restricted turns on.)

- When a module is undergoing online module change
- When a module is restarting
- When the standby system of a redundant system is collecting information about a module on an extension base unit

■When a module is occupying two slots

When a module is occupying two slots, module error information is stored in the special register area corresponding to the rear slot. The front slot is treated as an empty slot, so 0000H is stored in the corresponding special register area.

■For multiple CPU configuration

In a multiple CPU configuration, the module error information for the CPU module is stored in the special register areas of the CPU slot (SD3696 to SD3697) for Module 1, and in the special register areas of slot numbers 0 and later (SD3698 and later) for Modules 2 to 4, depending on the slot in which the CPU module is installed. The data structure of module error information when a multiple CPU configuration includes a CPU module that occupies multiple slots is shown below.

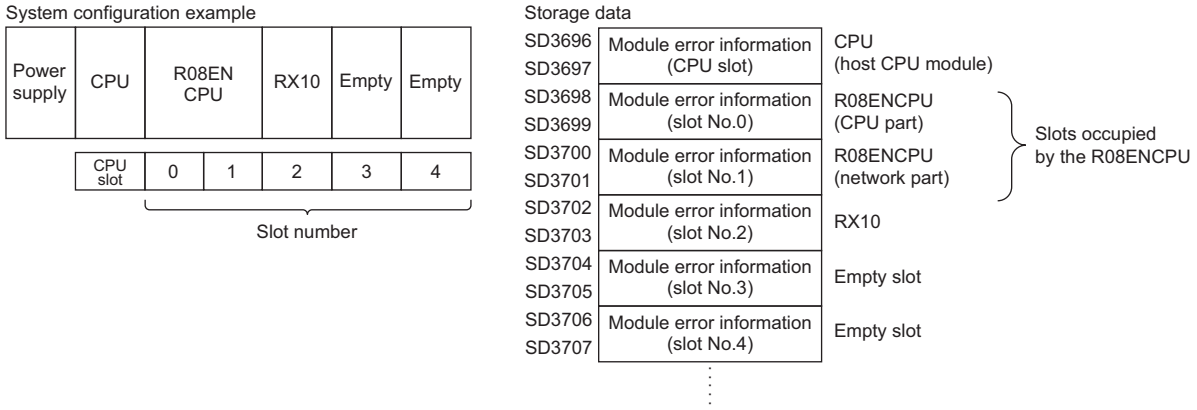
Even if the module is managed by another CPU module, module error information is collected and stored in the corresponding special register area.

• For RnENCPU

Among the occupied slots, the special register area corresponding to the front slot stores module error information for the CPU part. The special register area corresponding to the rear slot stores module error information for the network part.

Ex.

When the R08ENCPU is mounted in slot number 0

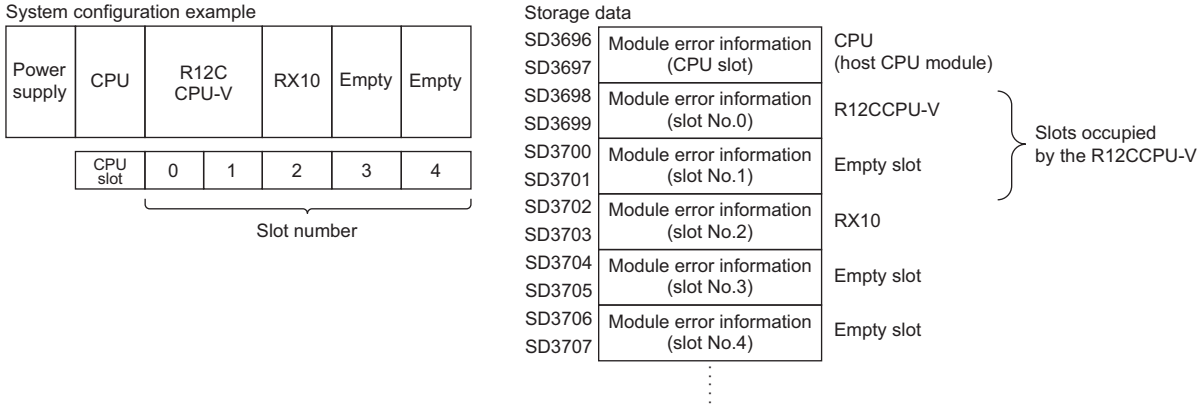


• For the R12CCPU-V and R102WCPU-W

Among the occupied slots, the special register area corresponding to the first slot stores module error information. Slots other than the first slot are treated as empty slots, so 0000H is stored in the corresponding special register area.

Ex.

When a CPU module (R12CCPU-V) occupying two slots is mounted in slot number 0



MEMO

19 TEST FUNCTION

Point

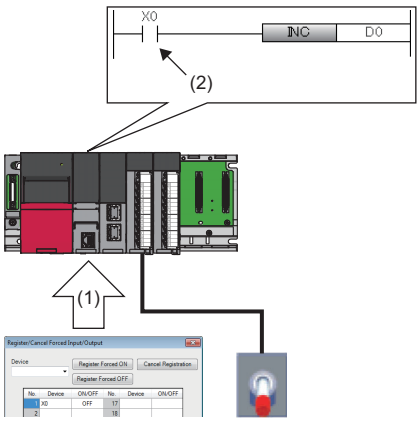
In a redundant system, changed values of devices and labels are transferred from the control system to the standby system in the END processing. If the systems are switched between the change and the tracking transfer, the changed values are not reflected to the CPU module of the new control system.

19.1 External Input/Output Forced On/Off Function

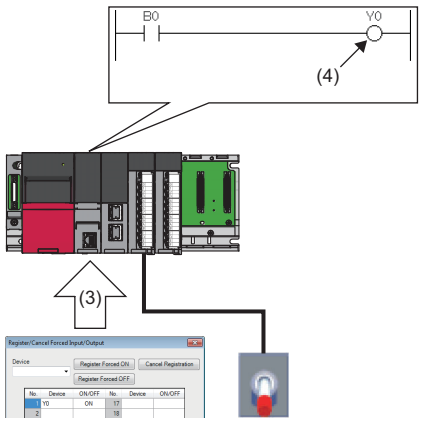
External inputs/outputs can be forcibly turned on or off from the engineering tool. This function enables input devices to be turned on or off regardless of the on/off state of the external inputs and enables the external outputs to be turned on or off regardless of the operation result of a program.

In the case of external input/output forced on/off in a redundant system, 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.)

(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 762 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	X0 to X2FFF (12288 points)
Output	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

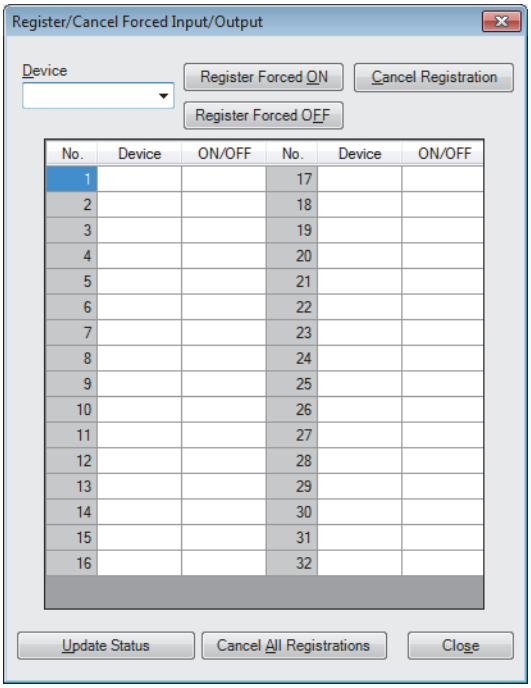
*1 When no modules are mounted on the base unit (no refresh targets), no data is output to the external outputs.

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	Enter 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).

■In redundant mode

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	○	○

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).

Behavior in forced on/off registration in redundant mode

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	Operation
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	Operation
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 control system is reflected.

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.)

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.

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[*6]
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] • 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 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) • At reflection of the tracking receive data to devices[*4]
Input/output of the CPU module assigned to LX and LY of the CC-Link IE Controller Network module or MELSECNET/H network module[*5]	• 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[*7]	• 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[*5]	• 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)	

*1 For the precautions for using the direct access device, refer to the following.
☞ Page 243 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.

*4 Forced on/off is reflected only to output devices. (Since refresh to external outputs is not executed.)

*5 In redundant mode, 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.)

*6 In redundant mode, whether external outputs to the standby system are reflected or not differs depending on the standby system output setting. (☞ Page 239 Reflection to external outputs of the standby system)

*7 In redundant mode, 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 in redundant system.

■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.

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 237 Operation method of forced on/off)

■FUNCTION LED

The FUNCTION LED of the CPU module flashes by registering forced on/off (every 200ms).

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 682 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 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 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 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 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 242 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 242 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.

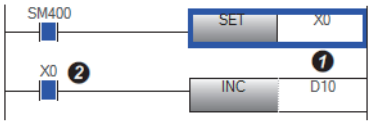
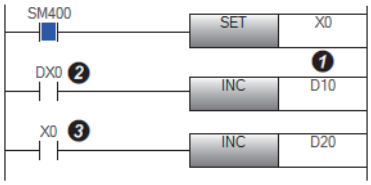
Precautions

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.

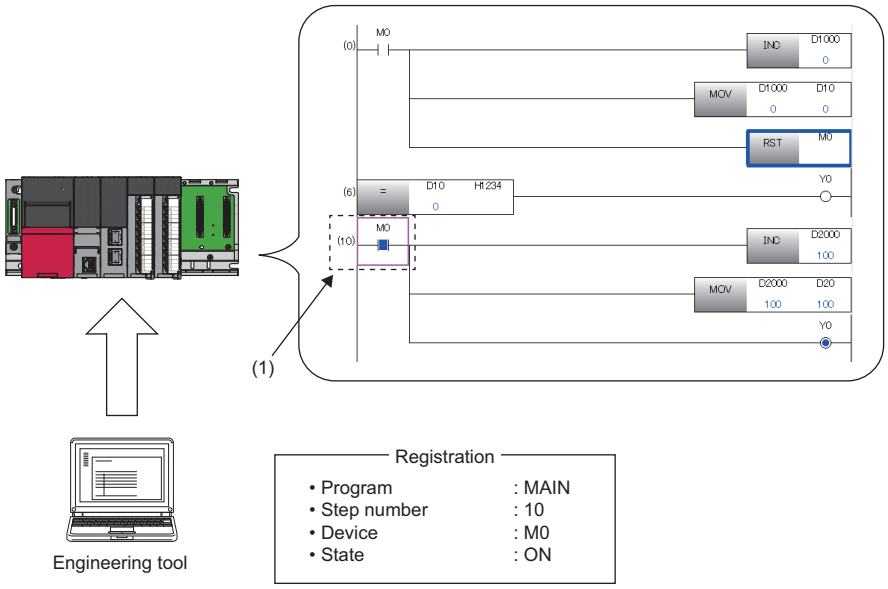
*1 For the out-of-group I/O fetch setting, refer to the following.
☞ Page 339 Out-of-group I/O fetch setting

*2 Except when the on/off state of the input/output devices is changed by using the output instruction

- In redundant mode, 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 a redundant system with redundant extension base unit, perform tracking transfer for the device.

19.2 Device Test with Execution Conditions

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.



Engineering tool

Registration

- Program : MAIN
- Step number : 10
- Device : M0
- State : ON

(1) The device is operated in accordance with the registered settings. (M0 of the step number 10 is on.)

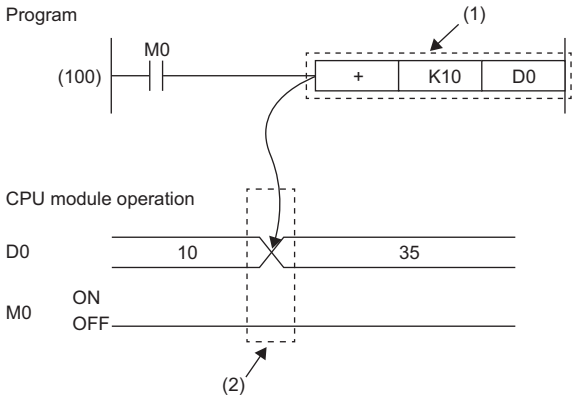
In redundant mode, using the engineering tool, device/label values can be set for each execution of specified steps of program.

Restriction

Before using the device test with execution conditions, check the version of the CPU module and the engineering tool used. (☞ Page 762 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.

■In redundant mode

- Even if the operation mode is changed, the registration information of the device test with execution conditions before changing the operation mode is retained.
- Even if the system is switched, the registration information of the device test with execution conditions before system switching is retained.
- 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.
- For the CPU module of the standby system in backup mode, 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.

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.

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

■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 206 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 711 Debug function)

Registration of device test with execution conditions

This section describes how to register the device test with execution conditions.

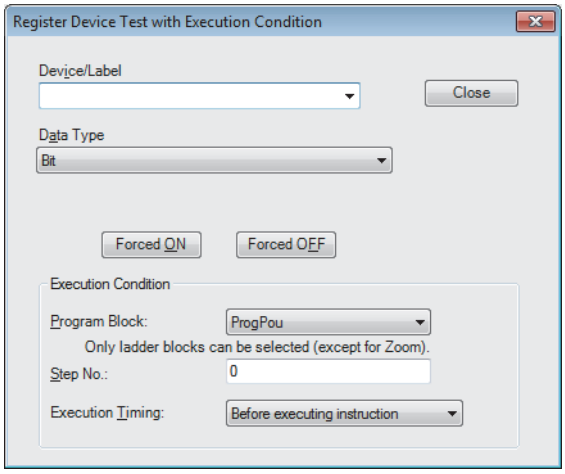
■Registration method

Specify each field in the "Register Device Test with Execution Condition" window.

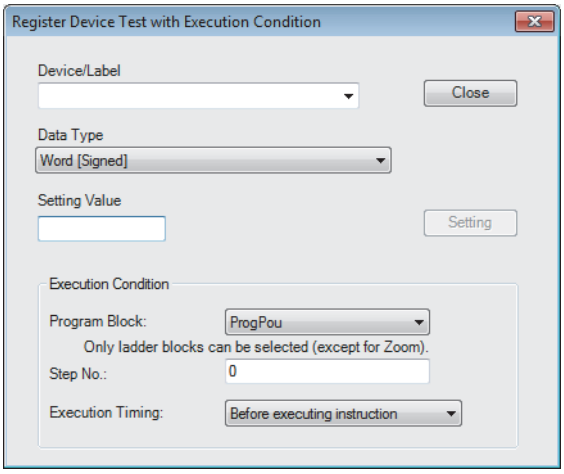
[Debug] ⇨ [Device Test with Execution Condition] ⇨ [Register]

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 246 Data that can be set)
Data Type		Specify a data type. (☞ Page 246 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 251 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

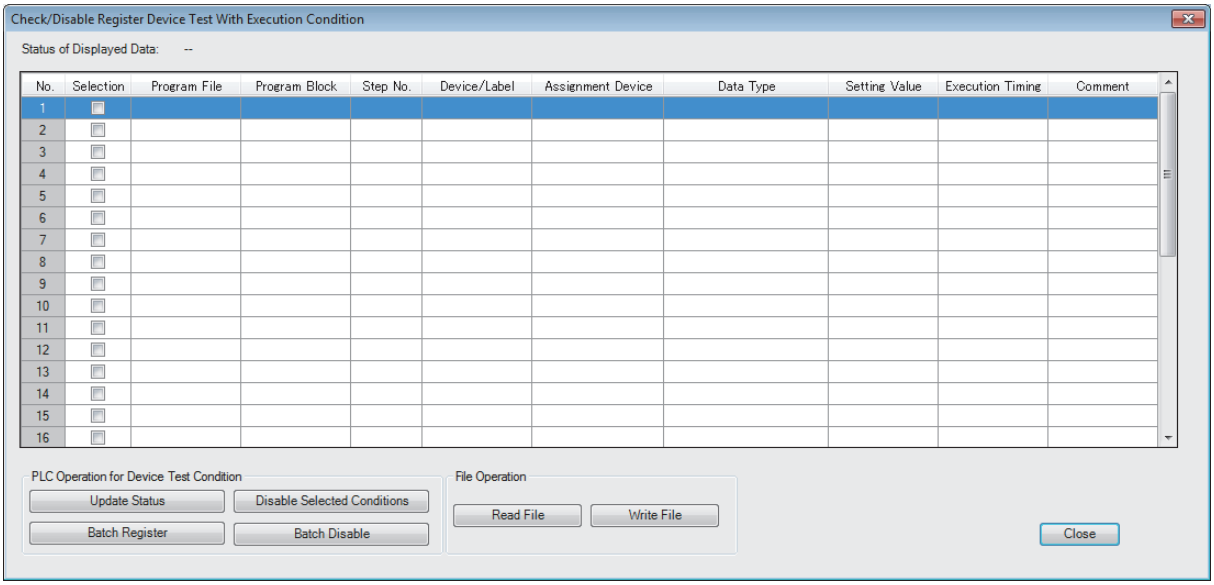
- Two device tests with execution conditions having different execution timing can be registered to one step even when they have the same device/label name.
- Different device modification (bit specification of word device, digit specification of bit device, or index modification) is handled as different 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 252 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.

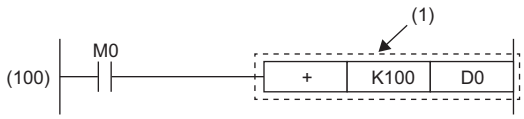
*5 All the registered settings of device test with execution conditions are disabled when the operating status of the CPU module is changed from STOP to RUN after the operation. (They are not disabled at the completion of the operation.)

*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.)

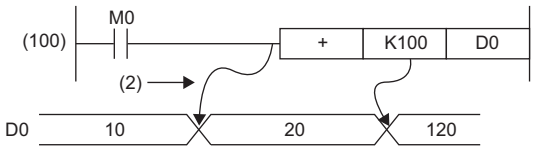
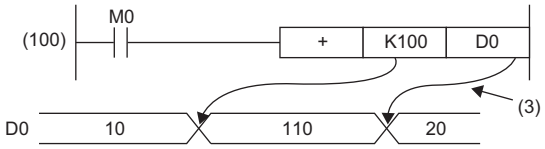
Execution timing

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".

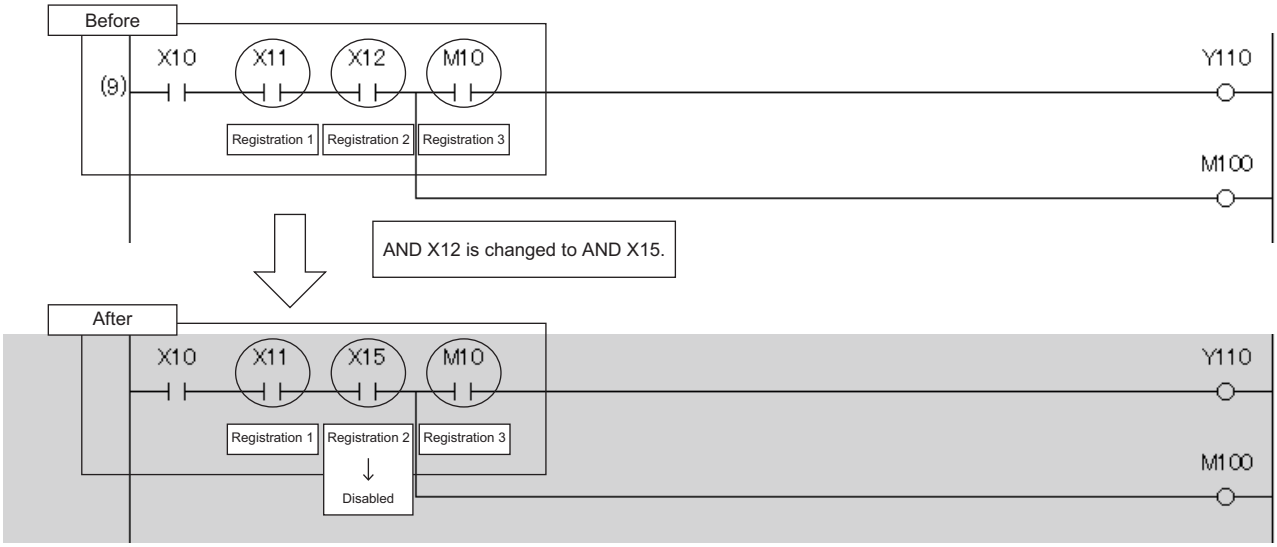
Operation during online change

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.

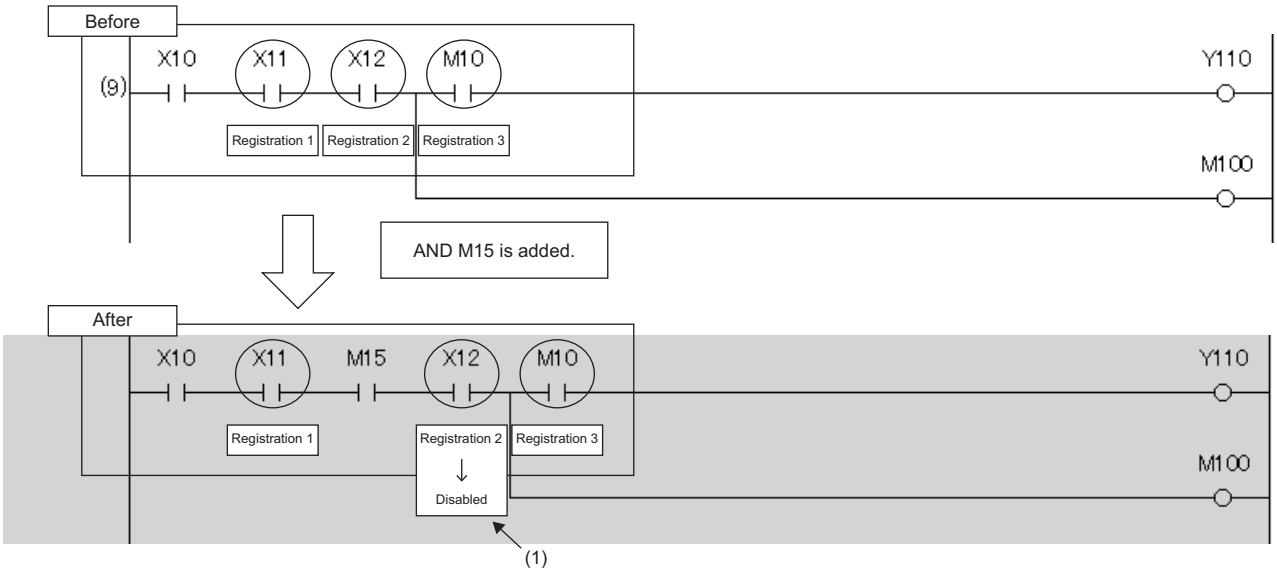
When the Registrations 1 to 3 of the device test with execution conditions are registered for the steps as shown below, the Registration 2 is disabled upon the execution of the online change (ladder block). The Registrations 1 and 3 are not disabled because the steps to which they are registered are 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.

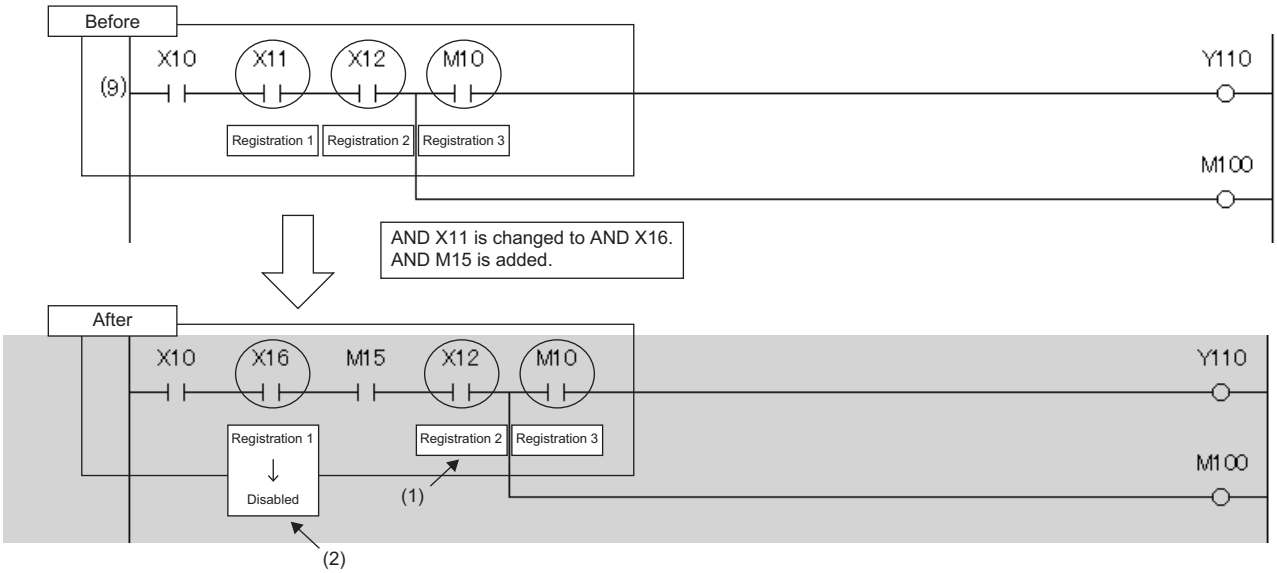
In the following example, an instruction is added by the online change (ladder block). In this case, when the device test with execution conditions is registered to the instruction immediately after the added instruction, the relevant registrations are disabled upon the execution of the online change (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).



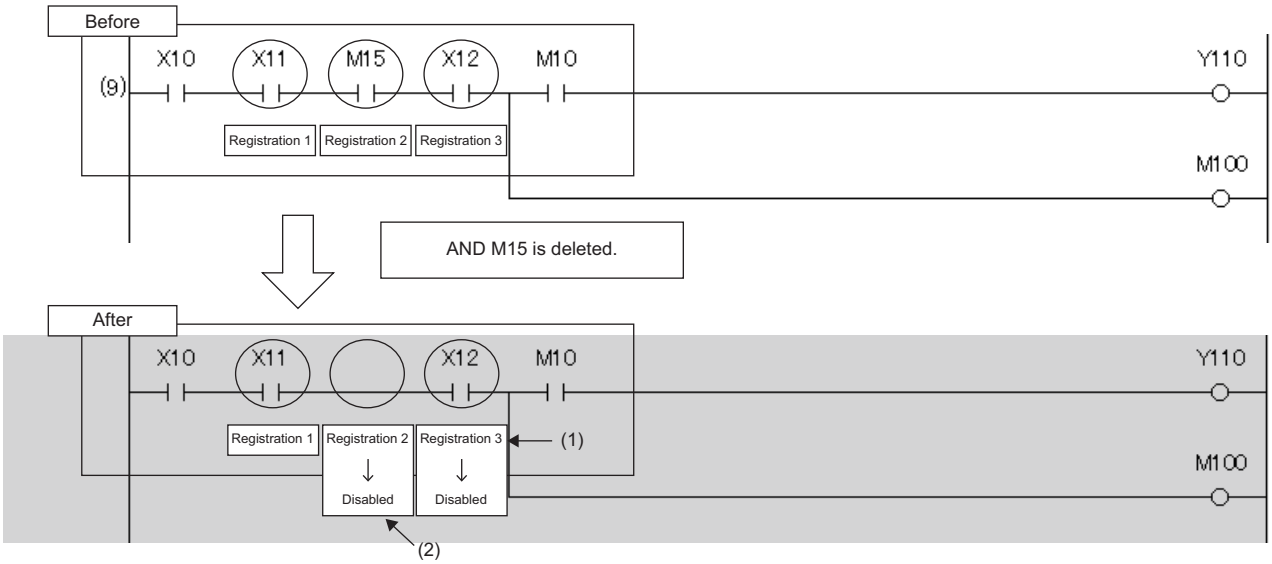
(1) Although this instruction is located immediately after the added instruction, the registration of the device test with execution conditions is not disabled because the added instruction is adjacent to the changed instruction.

(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.

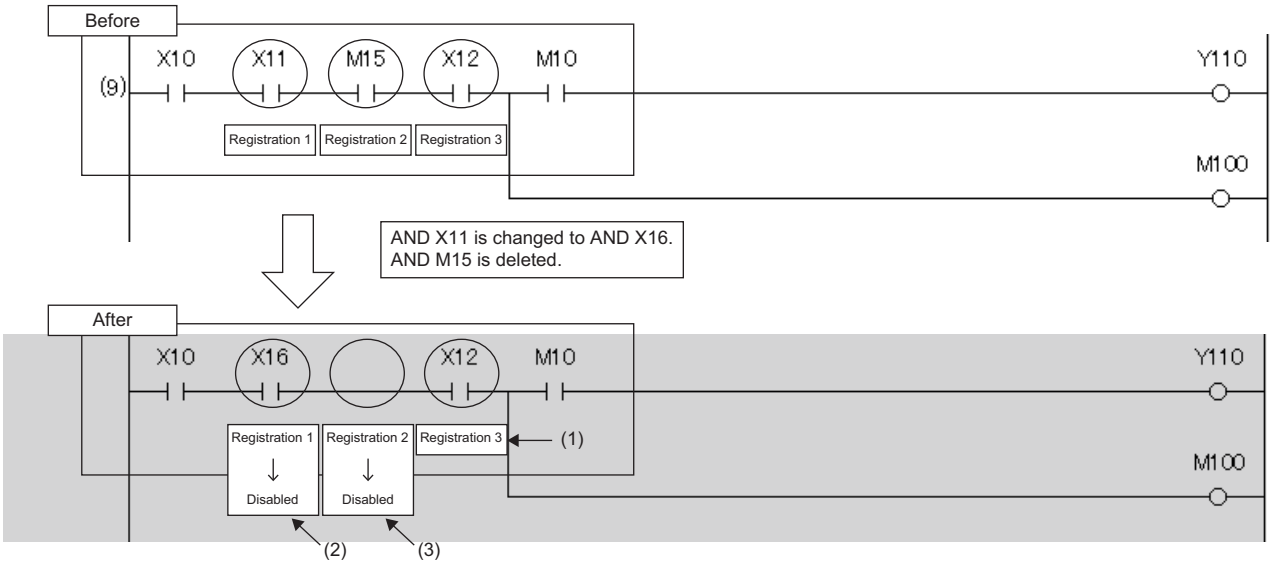
In the following example, an instruction is deleted by the online change (ladder block). In this case, when the device test with execution conditions is registered to the instruction immediately after the deleted instruction, the relevant registrations are disabled upon the execution of the online change (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.
Data logging	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.

■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 670 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 115 Interrupt Program)

Precautions in redundant mode

The following describes precautions on this function in redundant mode.

■When the program which is not executed in the CPU module of the standby system in 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 CPU module of the control system or standby system 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 CPU module of the standby system 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 CPU module of the standby system, take the tracking transfer into consideration.

■When the device test with execution conditions is registered to the standby system

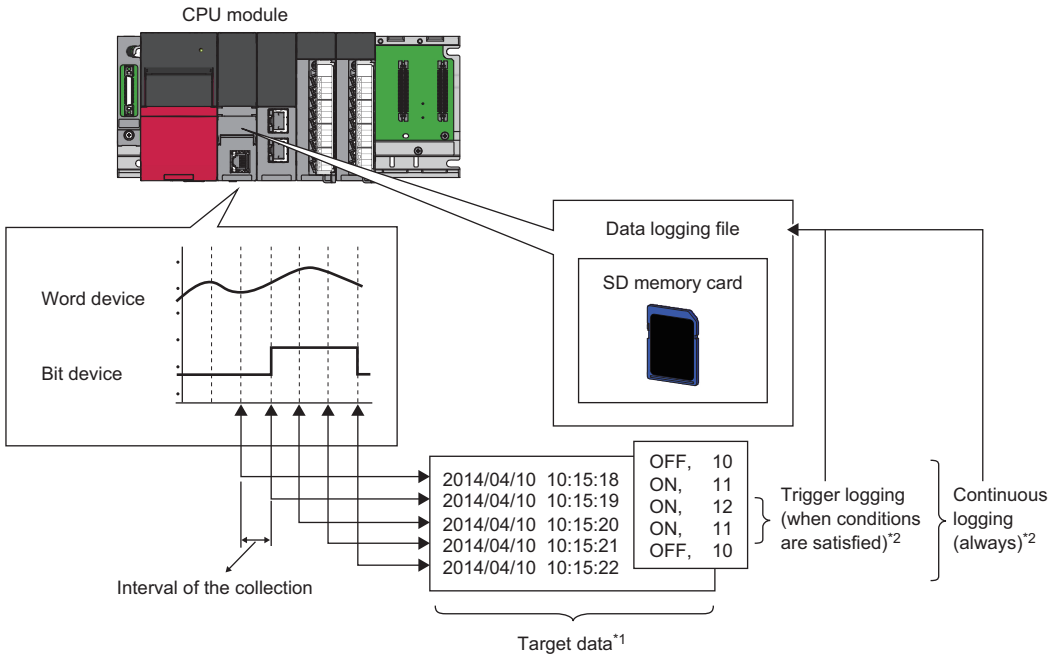
When the device test with execution conditions specifying buffer memory, link direct device, or direct access output of the module on the extension base unit is registered in the standby system in a redundant system with redundant extension base unit, the value of the device is not changed even if the execution conditions are satisfied.

20 DATA LOGGING FUNCTION

This function collects the specified data at the specified interval or any desired timing, and stores them as a file on the SD memory card.

The entire data logging function allows the entry of up to 10 data logging settings.

In a redundant system, the data logging function collects data only in the control system regardless of the operation mode.



*1 For details on the data to be collected, refer to the following. (☞ Page 261 Data to Be Collected)

*2 For details on continuous logging and trigger logging, refer to the following. (☞ Page 267 Logging Type)

Point

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

The data logging function can be configured with CPU Module Logging Configuration Tool of the supported version (listed in the table below) or any later version.

Operation mode	Supported version of CPU Module Logging Configuration Tool
Process mode	"1.43V" or later
Redundant mode	"1.58L" or later

20.1 Data Logging Procedure

This section describes the data logging procedure.

1. Install CPU Module Logging Configuration Tool.

CPU Module Logging Configuration Tool/GX LogViewer Installation Instructions (BCN-P5999-0506)

2. Launch CPU Module Logging Configuration Tool.
3. Configure the data logging setting.

CPU Module Logging Configuration Tool Version 1 Operating Manual

4. If auto logging is used, configure the common settings (auto logging common settings).

Page 294 Auto logging common setting

5. Insert an SD memory card to the CPU module, and power on the module.
6. Connect the CPU module to a personal computer.
7. Write the data logging setting to the SD memory card or data memory.
8. Switch the CPU module to RUN state to start the data logging.
9. Stop the data logging and read the data logging file.
10. Check the file into which data has been read.

Point

- To start data logging, always click the [Start] button in the "Logging Status and Operation" window of CPU Module Logging Configuration Tool. The data logging cannot be started even when the setting is written and power is turned off and on or the CPU module is reset (except for the auto logging).
- For the operation procedures for CPU Module Logging Configuration Tool, refer to the following.

CPU Module Logging Configuration Tool Version 1 Operating Manual

20.2 Specifications

This section describes the data logging specifications.

Item			Specifications	Reference
Number of data logging settings			10	—
Data collection	Collection interval		• Every scan operation • Time specification • Interrupt occurrence • Condition specification (device specification, step No. specification)	Page 262 Data Collection Conditions
	Number of points for collection		Maximum of 1280 (128 per setting)	
	AND conjunction		AND conjunction of the device specification and step No. specification is enabled.	
Logging type			• Continuous logging • Trigger logging	Page 267 Logging Type
Data processing	Trigger logging	Trigger condition	• Condition specification (device change specification, step No. specification) • When trigger instruction executed	Page 268 Operating procedure for trigger logging
		AND conjunction	AND conjunction of the device 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 time	
		Number of records	Maximum of 1000000	
File output	File name		Additional information plus file number	Page 272 Data Logging File
	File storage format		• Unicode text file • 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	
		Binary file	• Word (signed) • Double word (signed) • Word (unsigned) • Double word (unsigned) • Single-precision real number • Double-precision real number	
Storage location of data logging files			• SD memory card	—
Output file handling	Storage file switching	File switching timing	• Number of records • File size	Page 287 Switching to a storage file
		Maximum number of storage files	1 to 65535	
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.	Page 292 Setting Behavior at the Time of Transition to RUN
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.	Page 293 Auto Logging
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.	Page 297 SD Memory Card Replacement

20.3 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

*1 Duplicate data records are counted as distinct.

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

Global device data can be collected.

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, L, F, SM, V, B, SB, T (contact)*4, T (coil)*4, ST (contact)*4, ST (coil)*4, C (contact)*4, C (coil)*4, LT (contact)*4, LT (coil)*4, LST (contact)*4, LST (coil)*4, LC (contact)*4, LC (coil)*4, FX, FY, Jn\X, Jn\Y, Jn\SB, Jn\B, BLn\S*5
Word device*3	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
Double-word device	LT (current value), LST (current value), LC (current value), 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 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.

*5 The device can be collected by using the CPU module where the SFC function can be used. (☞ Page 762 Added and Enhanced Functions)

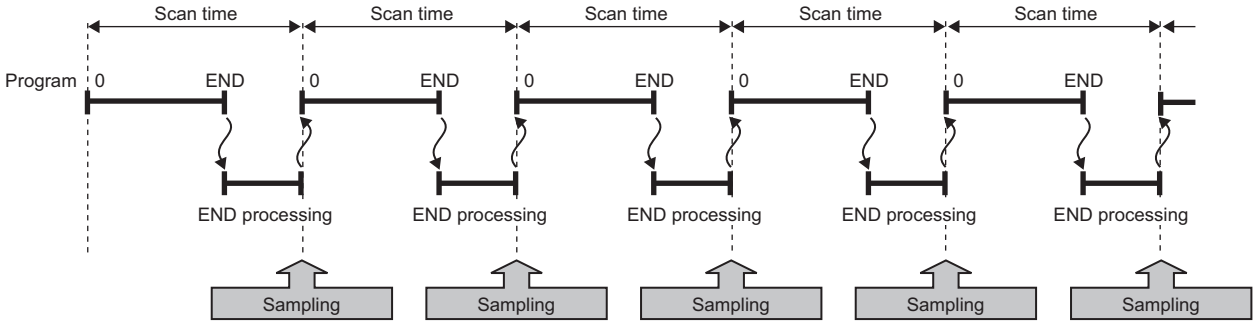
20.4 Data Collection Conditions

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 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

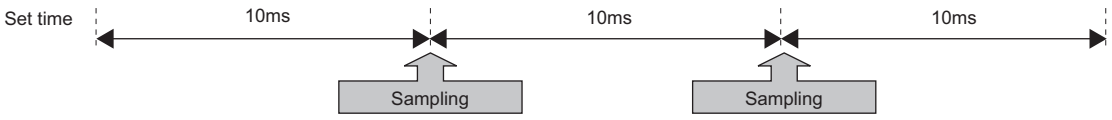
Specify the collection time interval. Specify the timing of data collection using advanced settings as well.

Data collection at specified time interval

Data are collected at specified time interval.

Ex.

10 milliseconds (10ms)



Point

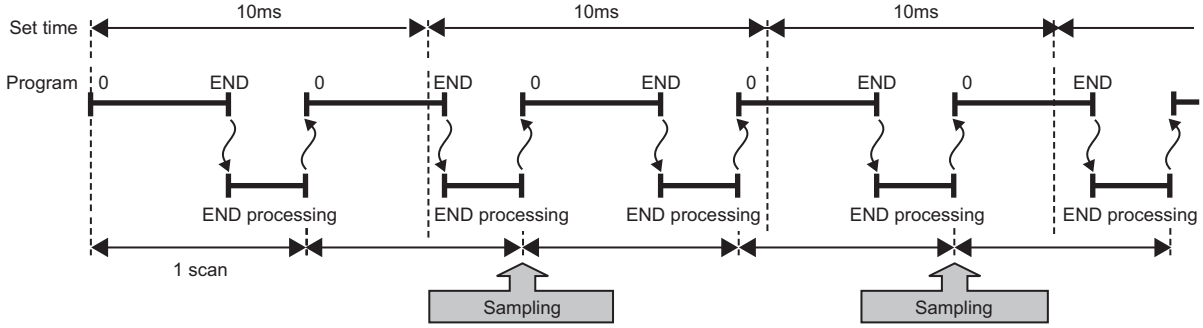
In redundant mode, when the systems are switched during data logging, time measurement starts from 0 in the new control system, and data is not collected at the specified interval.

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)

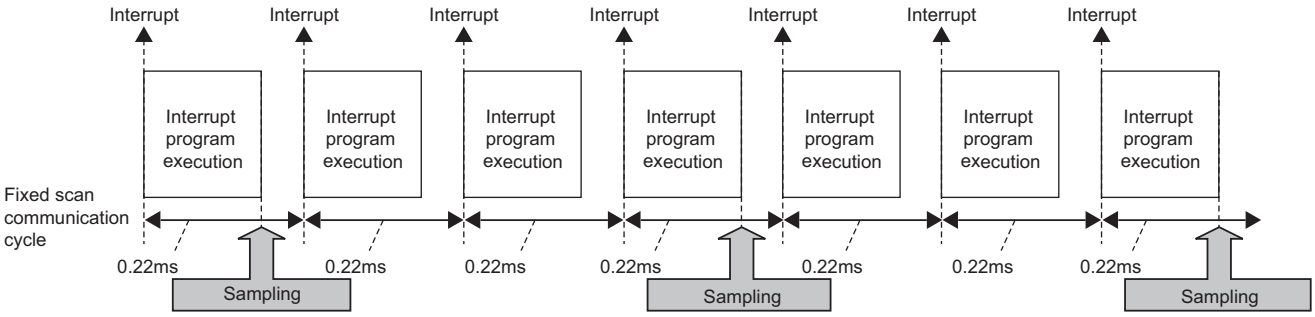


Interrupt occurrence

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 data conditions and step number. The AND condition using a combination of "Device specification" and "Step No. specification" results in the collection of data at the time when both conditions are established.

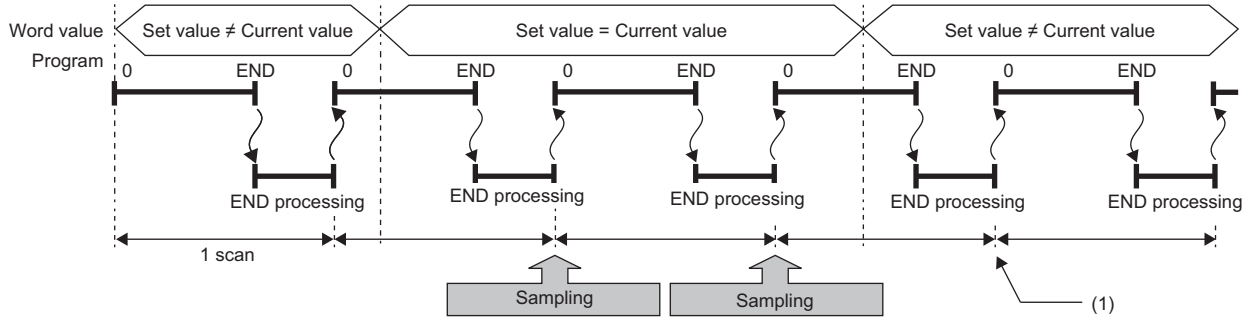
Device specification

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

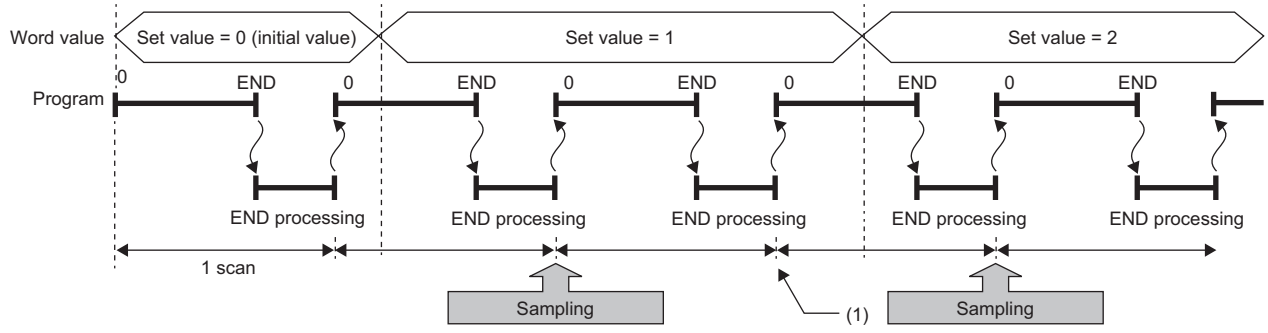


(1) During the END processing, the data logging function does not collect data because the conditions are not met.

■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 monitoring data, the following devices can be specified.

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, L, F, SM, V, B, SB, T (contact)[*4], ST (contact)[*4], C (contact)[*4], LT (contact)[*4], LST (contact)[*4], LC (contact)[*4], FX, FY, Jn\X, Jn\Y, Jn\SB, Jn\B, BLn\S[*5]
	Word device[*3]	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
	Double-word device	LT (current value), LST (current value), LC (current value), LZ

*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 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.

*5 The device can be collected by using the CPU module where the SFC function can be used. (☞ Page 762 Added and Enhanced Functions)

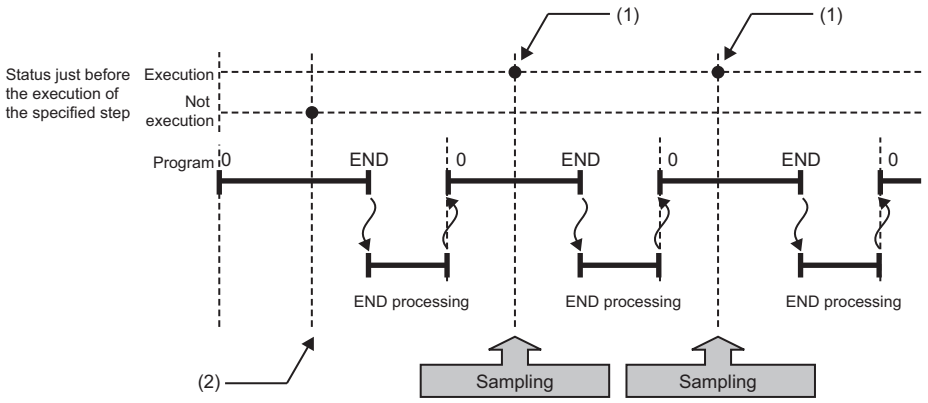
Step No. specification

Data are collected 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.

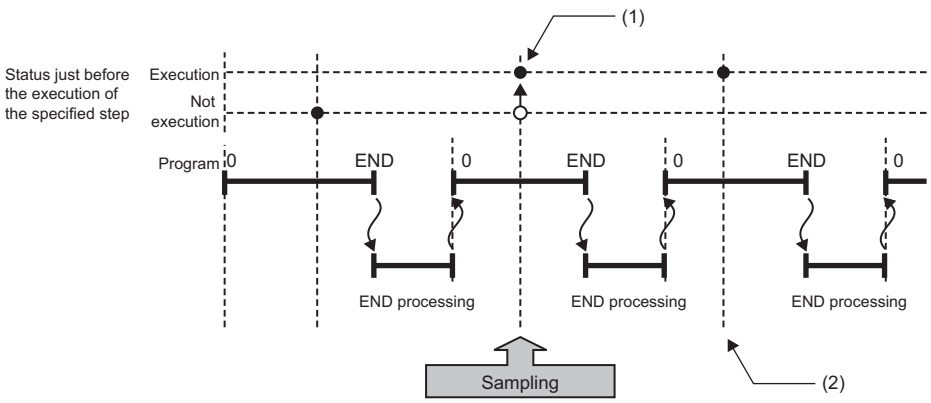


(1) The data logging function collects data because the state immediately before the execution of the specified step is a running state.
(2) It does not collect data because the state immediately before the execution of the specified step is not a running state.

■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

20.5 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.

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.)

Point

In redundant mode, if the systems are switched while trigger logging is being executed, logging data collected in the old control system is discarded.

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 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 change specification" and "Step No. specification" results in the collection of data at the time when both conditions are established.

■Device 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 change specification, monitored data can be configured to be collected from the devices listed in the following table.

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.

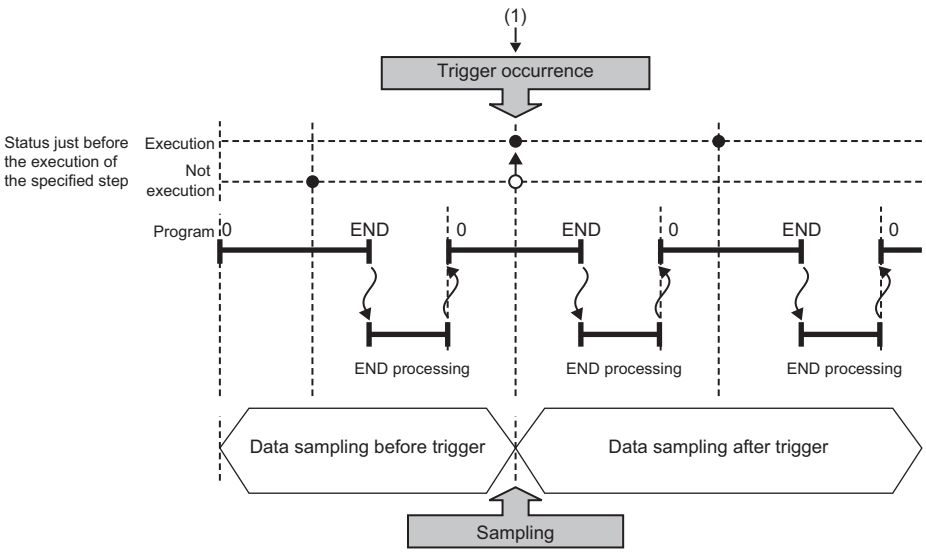
Type		Device*1
Device	Bit device*2	X, Y, M, L, F, SM, V, B, SB, T (contact)*4, ST (contact)*4, C (contact)*4, LT (contact)*4, LST (contact)*4, LC (contact)*4, FX, FY
	Word device*3	T (current value), ST (current value), C (current value), D, SD, W, SW, RD, R, ZR, Z, FD
	Double-word device	LT (current value), LST (current value), LC (current value)

*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 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.

■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:

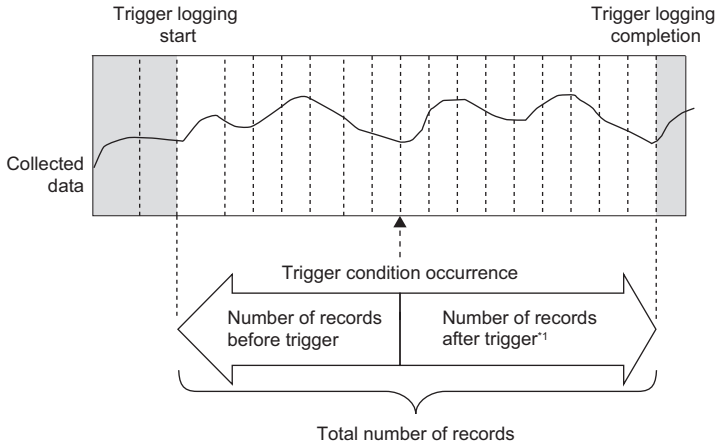
- Instruct CPU Module Logging Configuration Tool to suspend data logging (the special relay (data logging start) turns off).
- Turn on the special relay (Data logging suspend/resume flag).

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. (CPU Module Logging Configuration Tool Version 1 Operating Manual)



*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.

20.6 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.

File format	Application	Reference
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.	☞ Page 272 Unicode text file output type
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.	☞ Page 277 Binary file output format

Data output format

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.

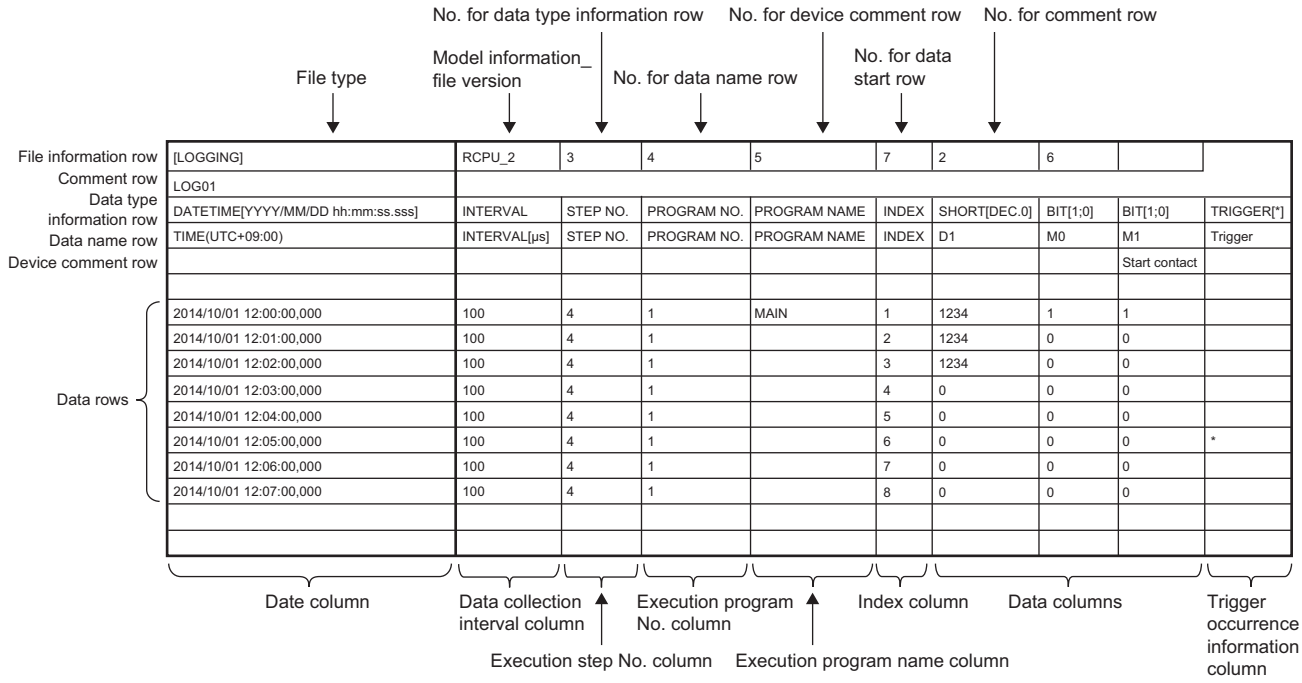
- Format specifications

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 the output data, 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. (CPU Module Logging Configuration Tool Version 1 Operating Manual)



• 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	"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

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.*2 When no comment is set, a blank row is output).	0 to 512 bytes

*2 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 bytes
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 bytes
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
Trigger occurrence information column	Trigger is output.	14 bytes

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

<Device comment row>

The device comment row for each column is displayed.

Item	Description	Size
Date column	No output (blank).	0 bytes
Data collection interval column		
Execution step No. column		
Execution program number column		
Execution program name column		
Index column		
Data column	When "Output device comments for data" is selected in the output setting, the comment with the specified comment number is displayed. 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 not output, this field is blank.	0 to 2048 bytes*3
Trigger occurrence information column	No output (blank).	0 bytes

*3 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 × 100 + 14 × 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 in 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*4 • 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

*4 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.

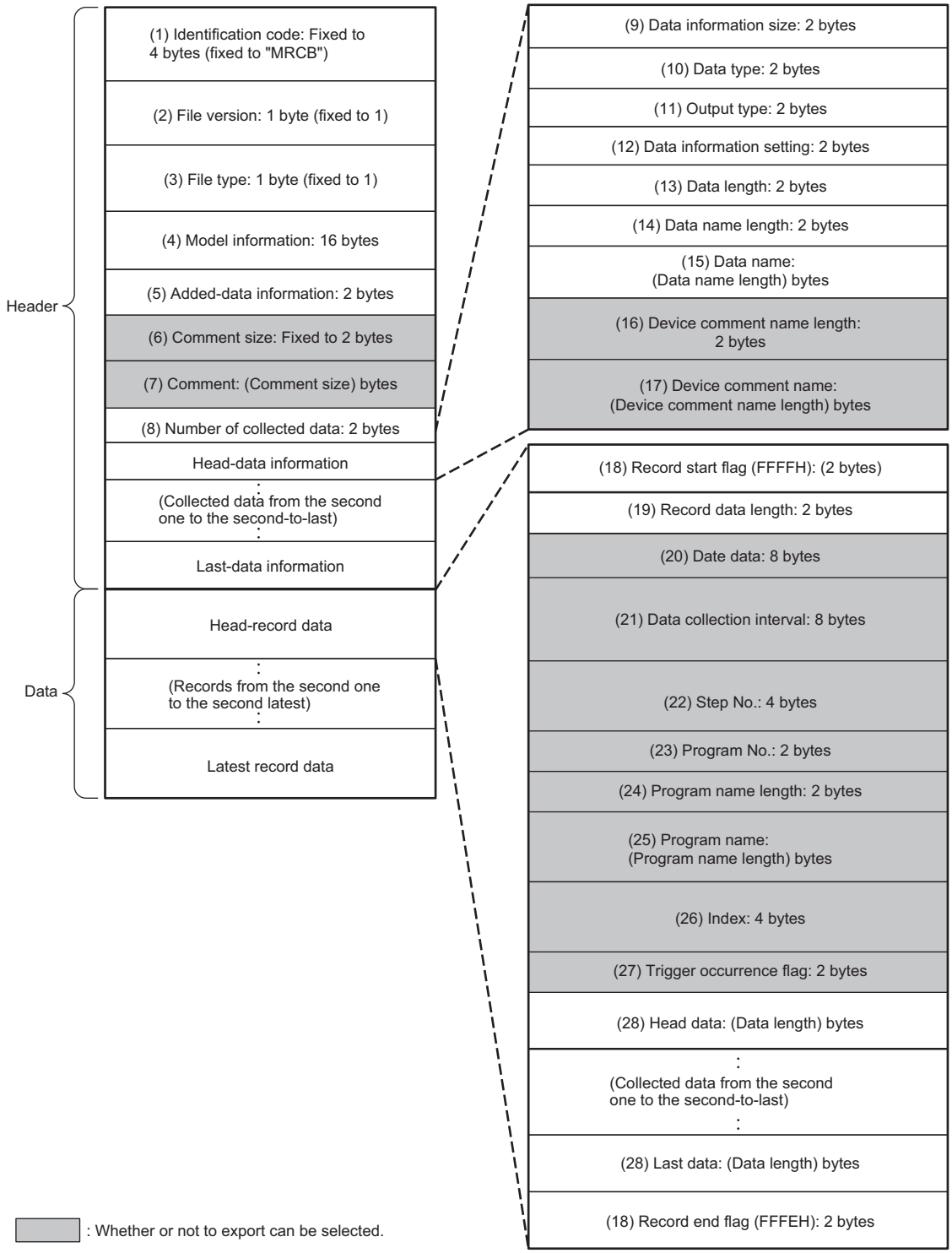
The size of the data type information row is determined by the following equation when data logging of 128 points of data from D100 to D227 (unsigned word type, decimal format) is performed (The following sections in the "Output" window are set to be output: "Date" (the output format is YYYY/MM/DD hh:mm:ss.sss), "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 No. 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

■Binary file output format

The following figure shows the configuration of the binary format and details of each data.



• Details of each data

No.	Item	Description	Size (byte)
(1)	Identification code	"MRCB" is output.	4
(2)	File version	"1" is displayed.	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. 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 device comments. 0: Do not output device comments. b7 1: Output comments. 0: Do not output comments. b9 1: Output device comments for each program. 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 format	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. b2 1: Device comment specified 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
(16)	Device comment name length	The length of the device comment name specified in the setting is output.	2
(17)	Device comment name	The device comment name specified in the setting is output in Unicode. 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.	0 to 2048

No.	Item	Description	Size (byte)
(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 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 step No. at the time of execution of END instruction in the last executed program in 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 280 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

*1 The following capacity values are output to ☐ based on the model.
Example: R08CPU, R08ENCPU: ☐ = 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.

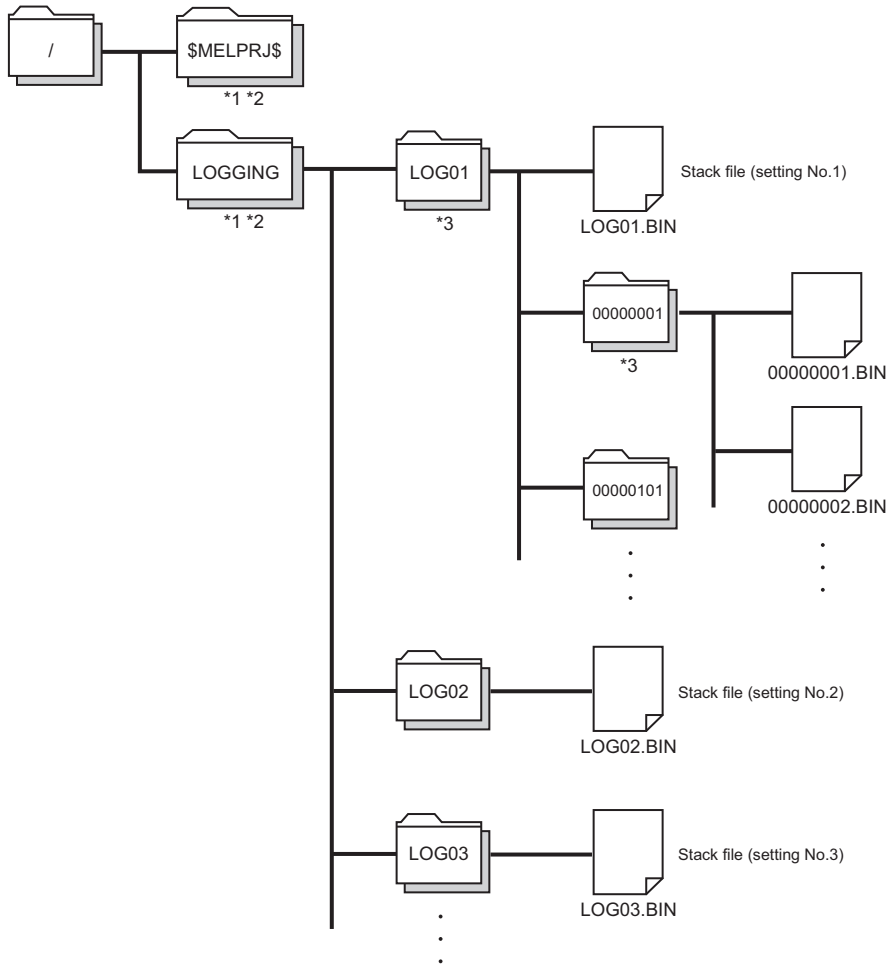
Output format	-Inf	0	+Inf
Single-precision real number	0xff800000	0x00000000	0x7f800000
Double-precision real number	0xfff0000000000000	0x0000000000000000	0x7ff0000000000000

Storage location of data logging files

The storage location of data logging files is an SD memory card. (CPU Module Logging Configuration Tool Version 1 Operating Manual)

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 (CPU Module Logging Configuration Tool Version 1 Operating Manual)

20.7 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. (📖 CPU Module Logging Configuration Tool Version 1 Operating Manual)

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

Data logging status in redundant system

The following table describes each data logging status in the redundant system.

Data logging states	Description
Stop	No data logging settings are registered and data collection is inactive.
Stop (after collection)	Transition from "Collection completed" to "Stop" has occurred due to the execution of another function*1.
Stop (after error)	Transition from "Error" to "Stop" has occurred due to the execution of another function*1.
RUN waiting (no collection)	• Data collection has not yet begun because the operating status of the CPU module is not in the RUN state. • After the operation mode of the redundant system was changed to the separate mode, the CPU module of the standby system enters the wait state for the RUN-transition instruction.
Start waiting (no collection)	Data collection is inactive, waiting for the start command.
Pause	Data logging is suspended and data collection has not yet been started.
Condition waiting (no collection)	Waiting for the first collection timing after the start operation
Collecting	Continuous logging is active and collecting data.
Trigger waiting (Collecting before trigger)	Trigger logging is active, 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	• 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 = the 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

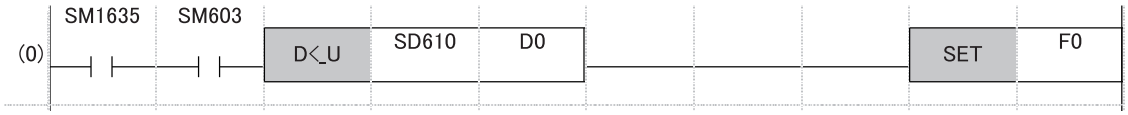
Before system switching (old control system)	After system switching (new standby system)
Start waiting (no collection)	Standby system start waiting
Pause	Standby system pause
Condition waiting (no collection)	Standby system no collection
Collecting	
Trigger waiting (Collecting before trigger)	
Collecting after trigger	

• 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	Start waiting (no collection)
Standby system pause	Pause
Standby system no collection	Condition waiting (no collection)

Point

Before using the data logging function, check if the SD memory card inserted in the CPU module of the standby system has sufficient free space with a program. When the SD memory card inserted in the CPU module of the standby system 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 422 Program Execution in Both Systems)
• Set the value of the criteria in D0 in increments of K of the double word [unsigned].

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	CARD ACCESS LED
• 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

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 241 FUNCTION LED)
- When program restoration information is not written (☞ Page 756 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 206 LED display setting)

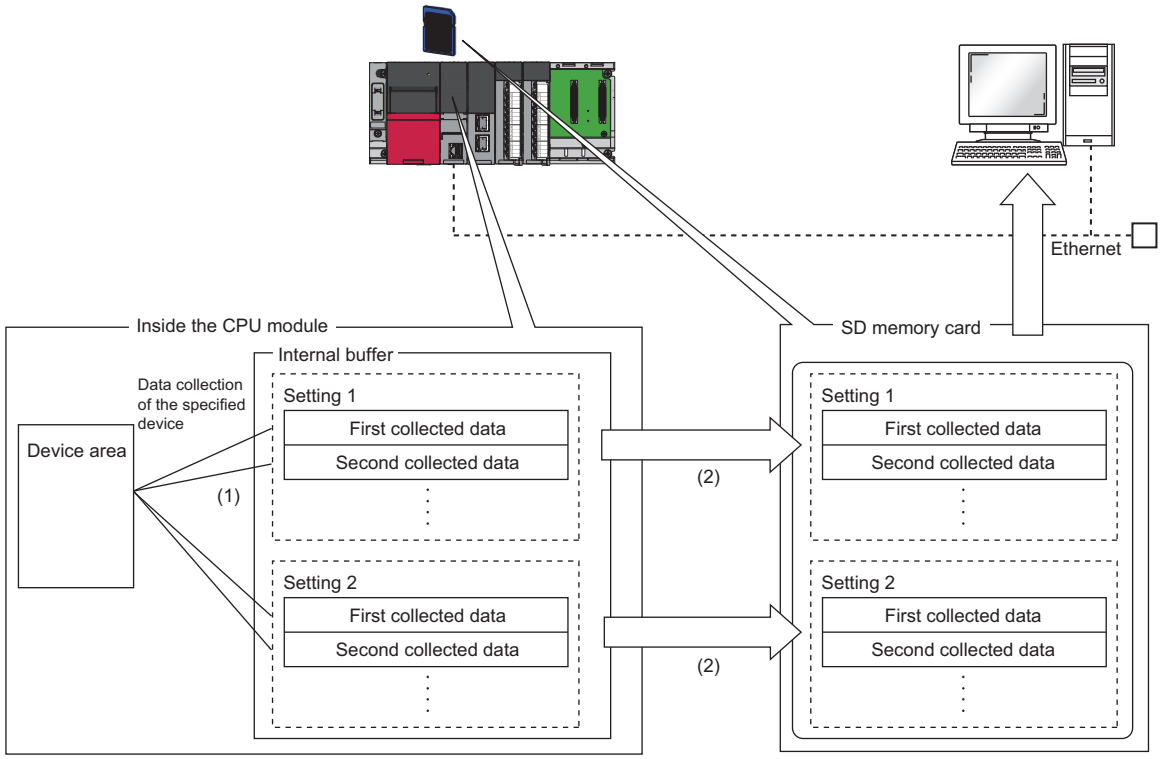
In redundant mode

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

20.8 Steps Until the Collected Data Is Saved

This section describes the steps until the collected data is saved.



(1) The collected data is temporarily stored in the specified internal buffer. (☞ Page 286 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.

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

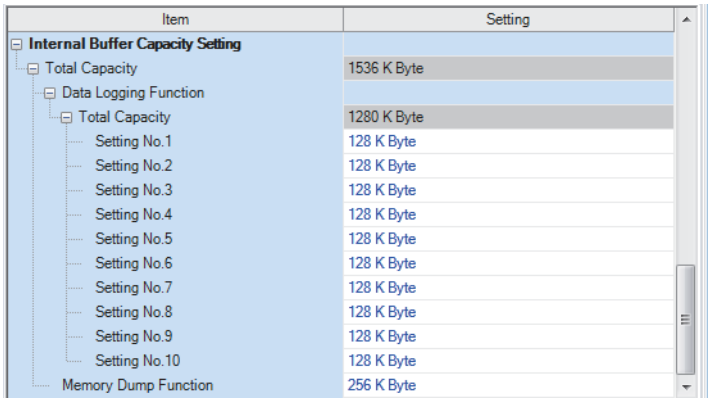
Set the capacity of the area (internal buffer) used by the system to temporarily store the results of data logging. The capacity can be set individually for each data logging setting number (1 to 10).

[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.	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	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

*1 Leaving this field blank allows the setting to be unused (0K bytes).

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

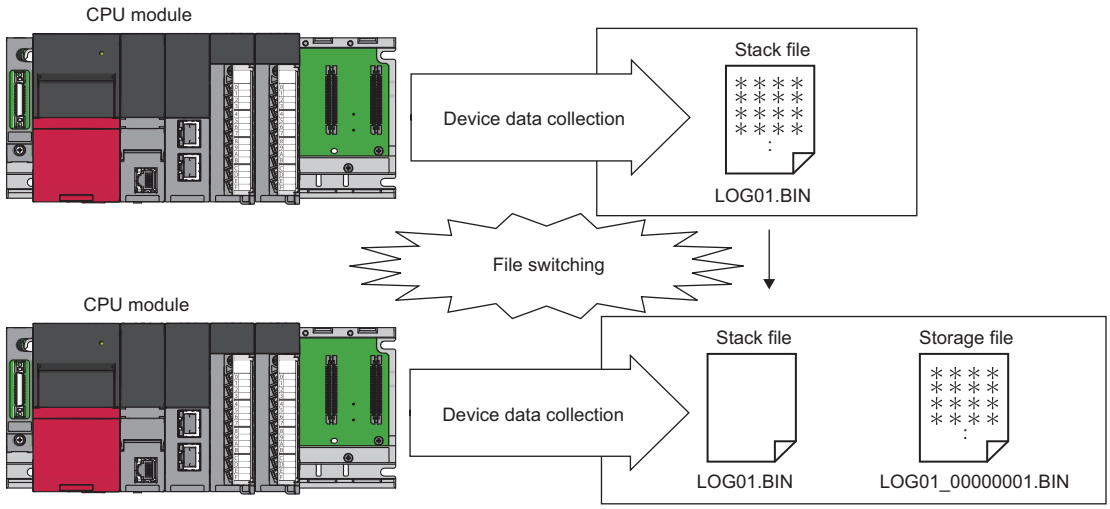
Switching to a storage file

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. (CPU Module Logging Configuration Tool Version 1 Operating Manual)
*2 The file number of the most recent storage file is stored in the special register (Latest storage file number).

In redundant mode

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). (CPU Module Logging Configuration Tool Version 1 Operating Manual)

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. (CPU Module Logging Configuration Tool Version 1 Operating Manual)

- 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.

File switching condition

The following table lists the setting items that can be used to specify the file switching condition.

Setting item	Description
Number of records	Specify the number of records within the following range. • 1 to 65500
File size*1	Specify the number of kilobytes within the following range. • 10 to 16384K bytes

*1 File switching occurs before the file grows beyond the specified size.

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.

■In redundant mode

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

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)	

*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.

• 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.

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.

20.9 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.

20.10 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). (CPU Module Logging Configuration Tool Version 1 Operating Manual)

- 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 293 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.

20.11 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 206 LED display setting)

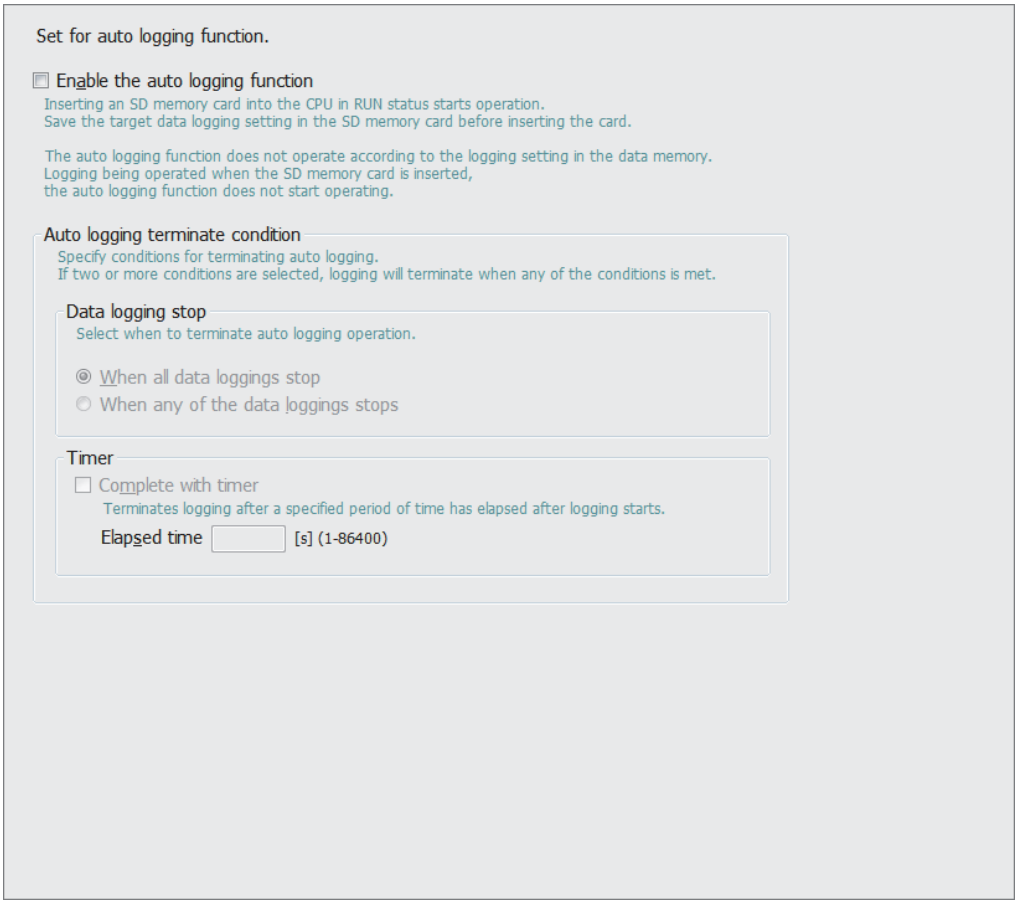
Point

- 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. (CPU Module Logging Configuration Tool Version 1 Operating Manual)

Auto logging common setting

The following window configures the required settings for using the auto logging function.

Window



Displayed items

Item			Description	Setting range	Default
Enable the auto logging function			Select this item to use auto logging.	—	Not checked
Auto logging terminate condition	Data logging stop		Select the condition to complete the auto logging operation. (☞ Page 295 Conditions for auto logging completion)	• When all data loggings stop • When any of the data loggings stops	When all data loggings stop
	Timer	Complete with timer	Select this checkbox to stop the auto logging operation by a timer.	—	—
		Elapsed time	Specify the time interval from the start of the data logging until stopping it.	1 to 86400 seconds	

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.

In redundant mode

- 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.

20.12 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.
☞ Page 147 How to forcibly disable the SD memory card with a special relay
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.)

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.

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
- Power off and on*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

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.

20.13 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 277 Binary file output format)

Unicode text file format: Refer to the data row. (☞ Page 272 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 277 Binary file output format)

Unicode text file format: Refer to the file information row to device comment row. (☞ Page 272 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)

20.14 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.

20.15 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 673 Data logging function)
- Special register: Special register areas relating to the data logging function (☞ Page 709 Data logging function)

20.16 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.
		The data conditions cannot be specified as the trigger conditions for multiple trigger conditions at the same time.
		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)	■When a step number is specified as the collection start condition or trigger condition • A data logging error occurs. • The online change (ladder block) is completed.
	File batch online change	■When a step number is specified as the collection start condition or trigger condition • A data logging error occurs. • File batch online change is completed.
	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.
	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.

*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.

■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	■When a step number is specified as the collection start condition or trigger condition • The online change (ladder block) is completed. • The data logging cannot be executed. (An error occurs when the start operation of the data logging is performed during the online change (ladder block).)
File batch online change		■When a step number is specified as the collection start condition or trigger condition • File batch online change is completed. • The data logging cannot be executed. (An error occurs when the start operation of the data logging is performed during the file batch online change.)
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.
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.

*1 The data logging function is not included here.

■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.

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 in condition specification

After registering data logging, do not change the file name of the file register file and the block No. of the file register when the file register is specified for the condition specification in the collection and trigger condition 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.

■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.

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 the SD memory card

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 occurs so frequently that a delay occurs in completing the access. To avoid such a delay, use the following workarounds:

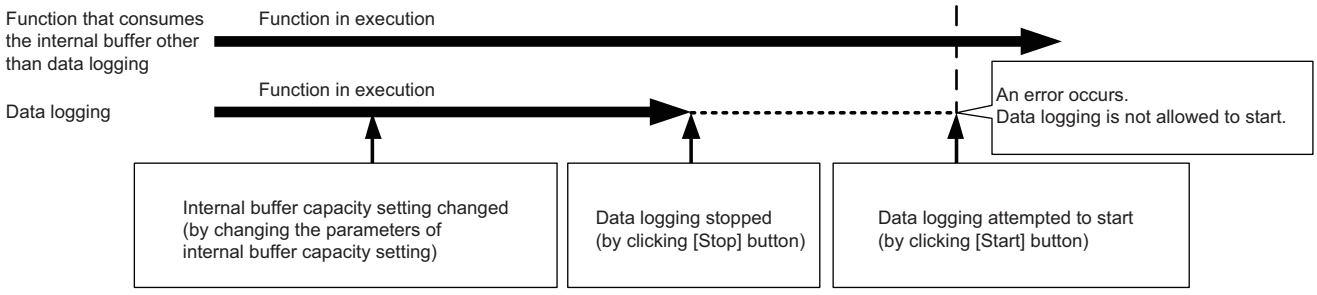
- Increase the data collection interval/timing.
- Reduce the number of data records to be collected.
- Lower the frequency of file switching.

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.

■File register in which "Use File Register of Each Program" is set

The program name cannot be specified for the following devices.

- 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)	SM1200 (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.
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 126 Saving/restoring of the file register (R) block number)

In redundant mode

When the data logging setting is written, please note the following.

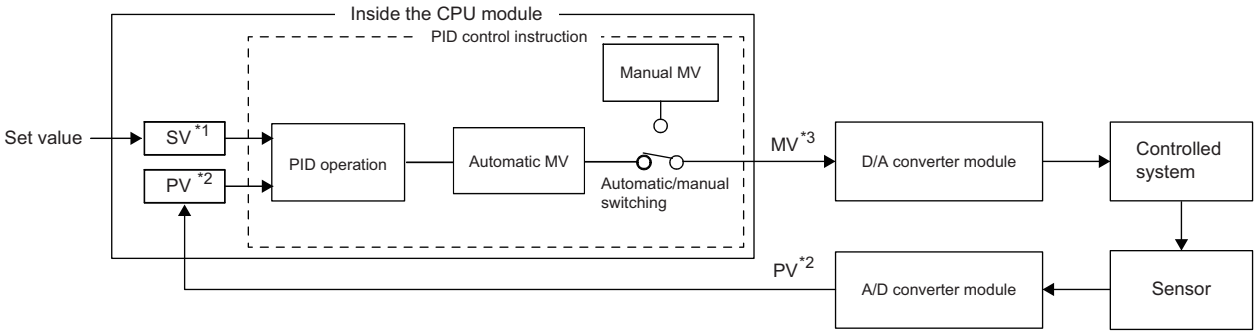
- While the data logging setting is being written, do not perform the following operations on both of the control system and standby system.
  - Turning off or resetting the CPU modules
  - Connecting/disconnecting the tracking cable
  - Online module change of redundant function modules
- In the following cases, the setting is written only to own system.
  - The CPU module of the other system 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 stop due to an error in the CPU module.

21 PID CONTROL/PROCESS CONTROL FUNCTION

This chapter describes the PID control/process control function.

21.1 PID Control Function

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

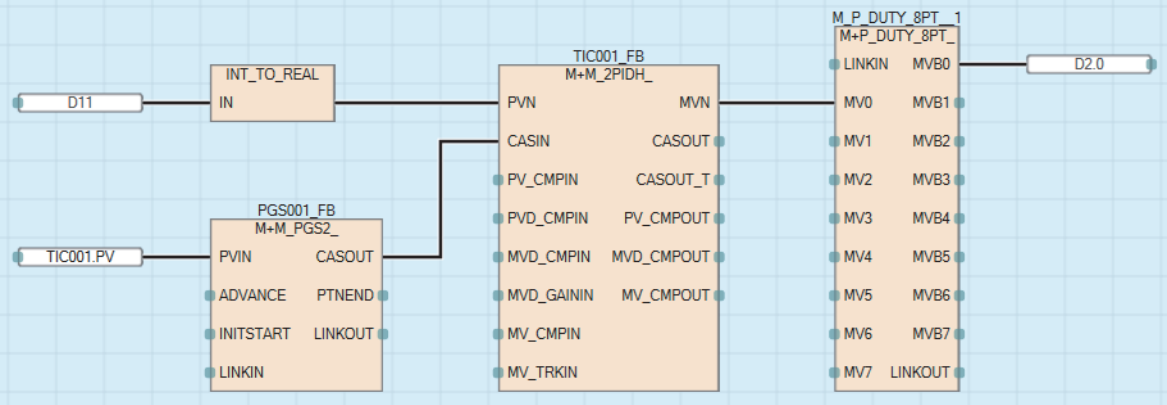
To execute the same PID control as that of the MELSEC-Q series and MELSEC-L series, use the PID control instructions.
For details on the instruction specifications, PID control, and programming, refer to the following.
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

21.2 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.

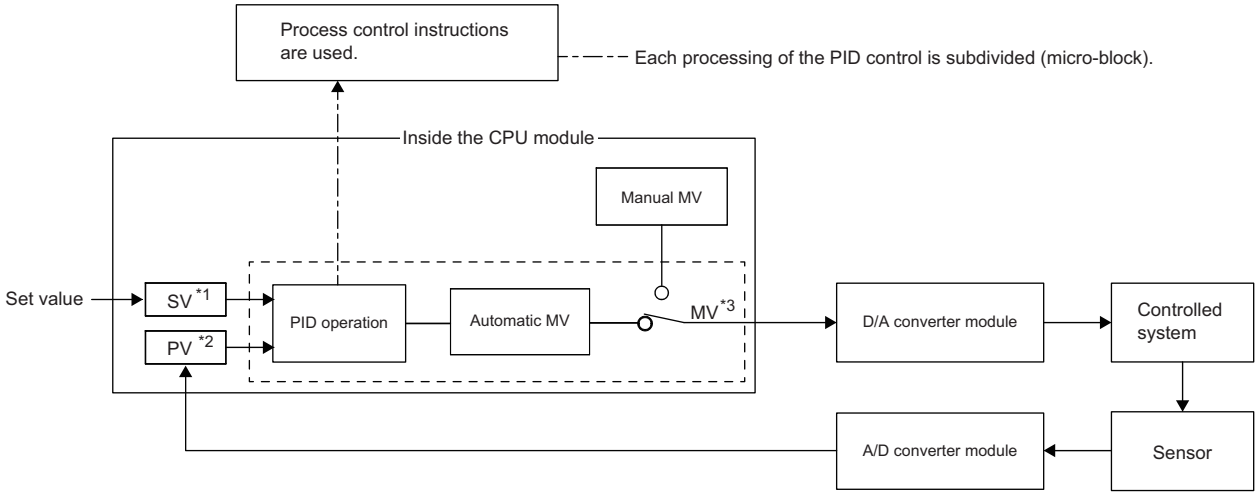
Point

For details on the process control function block, refer to the following.

MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions)

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)

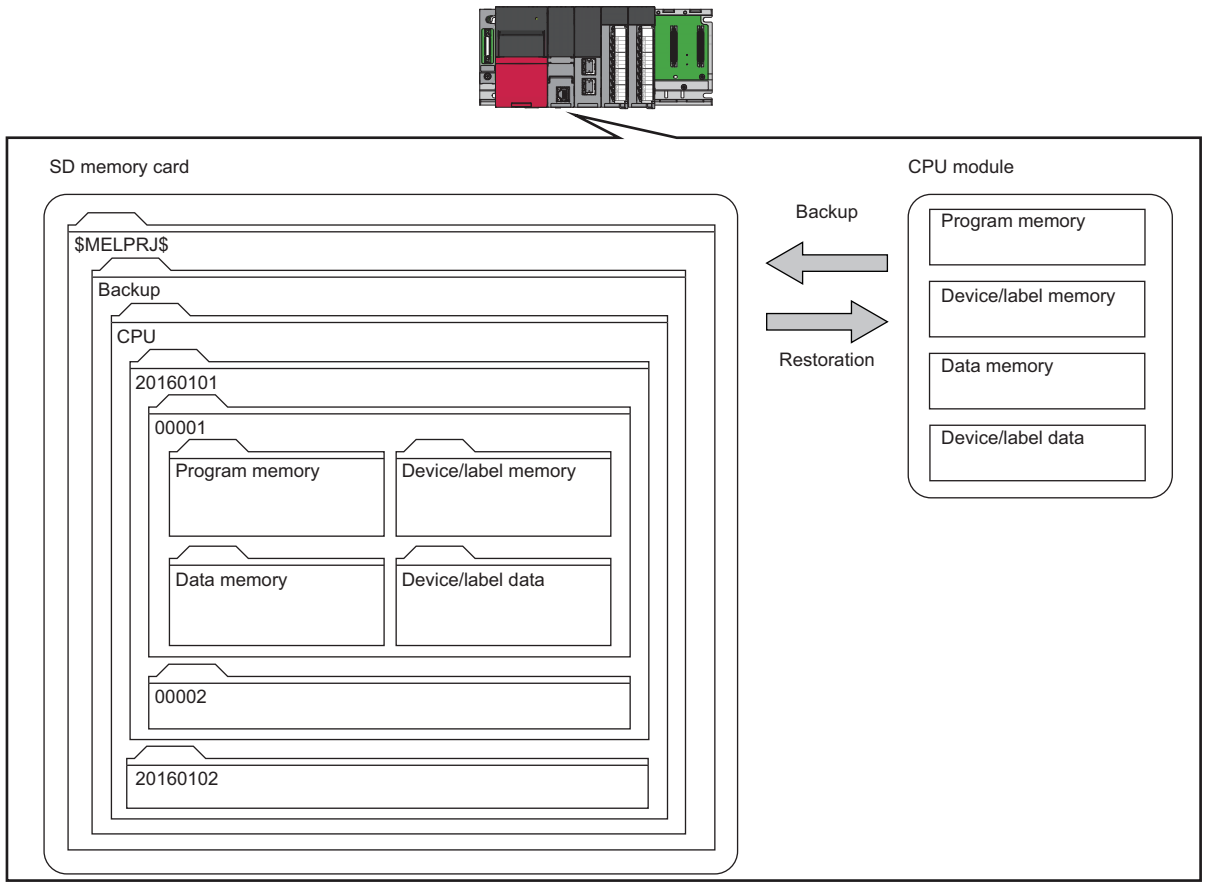
- For the specifications of process control instructions, basic loop types, and programming details, refer to the following.

MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions)

22 CPU MODULE DATA BACKUP/RESTORATION FUNCTION

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		Reference
Backup function	Backup processing triggered by turning on SM1351	Page 319 Backup processing triggered by turning on SM1351
	Automatic backup using SD944	Page 320 Automatic backup using SD944
Restoration function	Automatic restoration using SD955	Page 329 Automatic restoration using SD955
	Automatic restoration with the SD CARD OFF button[*2]	Page 331 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 322 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 762 Added and Enhanced Functions)

In redundant mode

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.

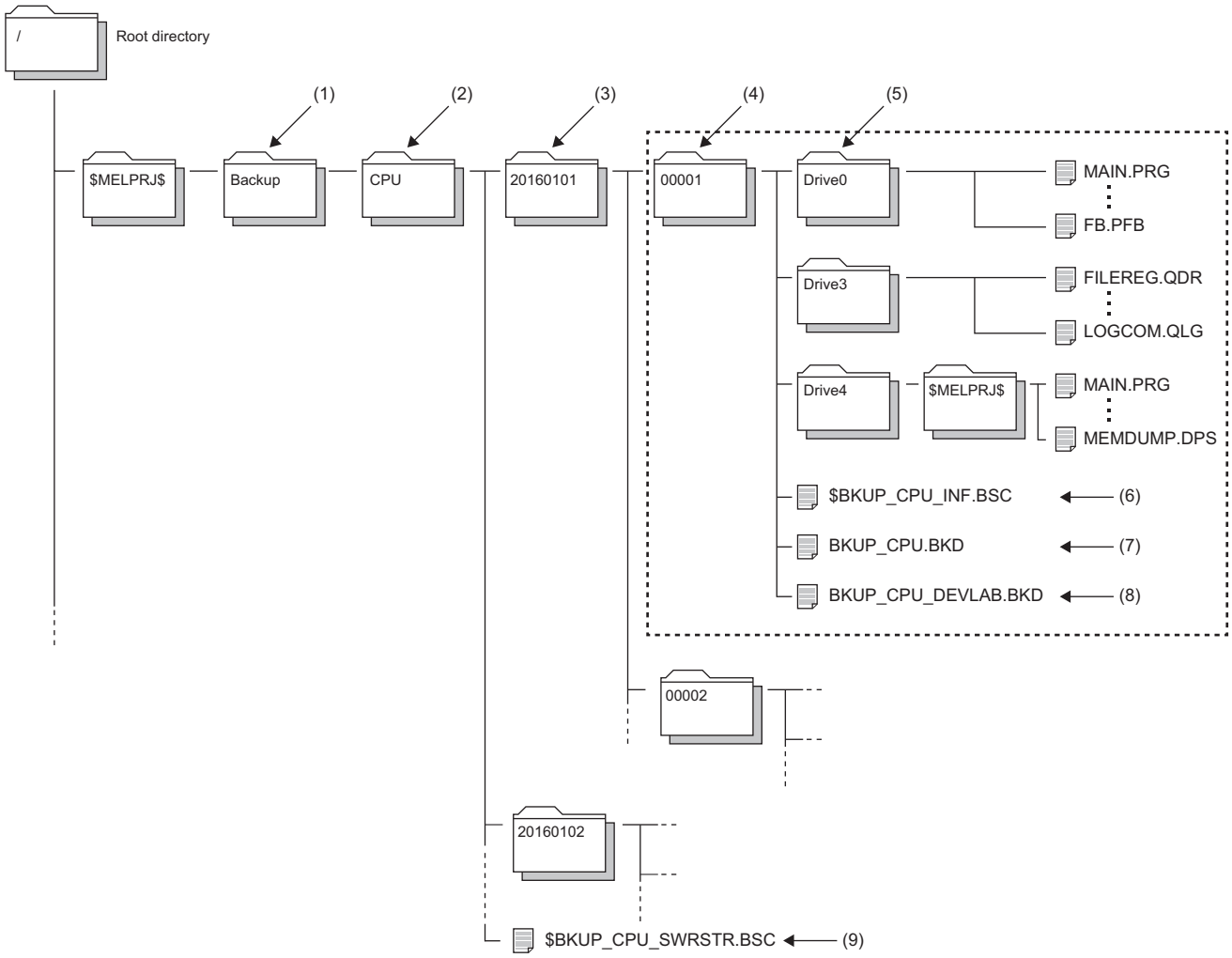
■Differences in usage from the memory copy function

When combining the data in the CPU module of the standby system with the data in the CPU module of the control system, 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 409 Memory Copy from Control System to Standby System)	When replacing only the CPU module of the standby system due to a failure and others
	When recovering the data due to error occurrence in the system consistency check

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 317 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 314 Backup/restoration target files)
Restoration target data is set with SD954 (Restoration target data setting). (☞ Page 327 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	○
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		×

All folders/files in the CPU built-in memory are backup/restoration targets.

■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).

22.1 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

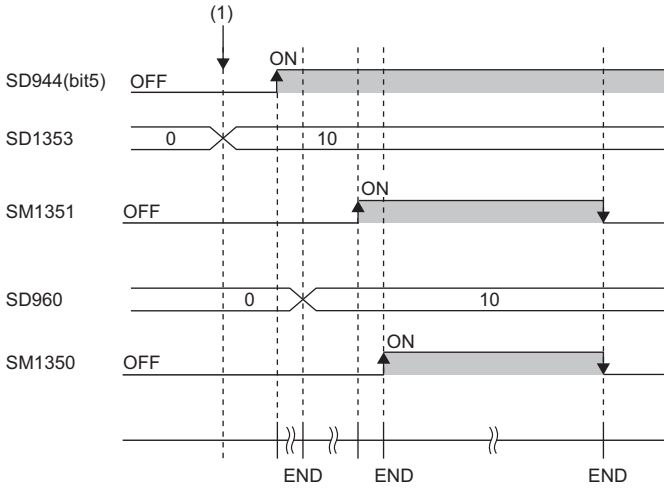
If data folders are created more than the capacity of the SD memory card before the number of them reaches the upper limit value for the number of CPU module backup data, the backup processing is completed with an error and the data cannot be backed up to reach the upper limit value.

■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)

In redundant mode

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 CPU module of the control system 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 136 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".

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.

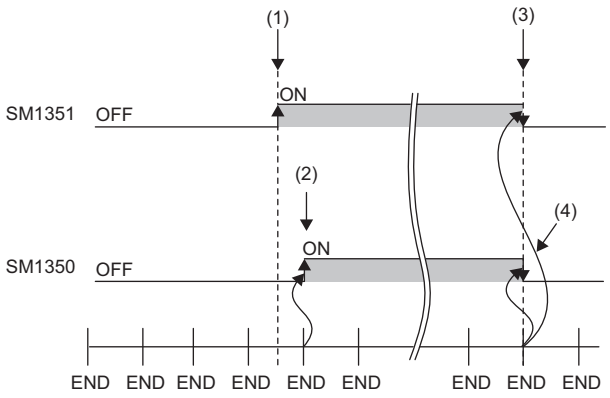
1. To set the upper limit value for the number of CPU module backup data, follow the steps below.

- 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 317 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 316 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 325 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 319 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 320 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.

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 319 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 320 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.

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 319 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 320 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.

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 327 Restoration target data, ☞ Page 327 Restoration of the special relay and special register, ☞ Page 328 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 319 Backup processing triggered by turning on SM1351, ☞ Page 320 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 565 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 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)	
	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	
	Online module change function	

*1 The operation performed only for opening a file to write data

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.)

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.

In redundant mode

The following describes the precautions for the backup function in redundant mode.

■Function that 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 tracking 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.

22.2 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.

- SM953, SM959, SM961, SM1350, SM1351, 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

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 136 Operation Processing When Operating Status Is Changed)

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.

In redundant mode

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 CPU module of the control system to the CPU module of the 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 CPU modules of both systems at the same time.)

1. Set the restoration function to the CPU module of each system. (☞ Page 329 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 both systems 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 329 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 329 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 a redundant system without extension base units, start up the CPU module of the own system as the control system. In a redundant system with redundant extension base unit, 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

Backup data is automatically restored when the CPU module is powered on or is reset while the SD CARD OFF button is being pressed.

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 322 Operating procedure (when a stop error has occurred in the CPU module))

*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 begins to flash. If the switch is pressed more than 10 seconds, restoration may fail.

In redundant mode

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 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 the control system to the CPU module of the 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 331 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 a redundant system without extension base units, start up the CPU module of the own system as the control system. In a redundant system with redundant extension base unit, 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.

Checking restoration errors

When an error occurs at the automatic restoration using SD955, a diagnostic error is detected. The error code is also stored in SD959. (☞ Page 565 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.

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.

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.)

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 timing of the restoration processing.

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)	
	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	
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	

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.

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 140 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.

In redundant mode

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 a redundant system with redundant extension base unit, 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.

23 MULTIPLE CPU SYSTEM FUNCTION

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

- This function cannot be used when the Process CPU is in redundant mode.
- 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.

23.1 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 339 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

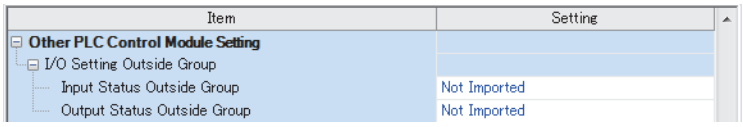
An error is not detected in reading input (DX) for the non-controlled module and buffer memory of the intelligent function module.

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	Specify whether or not out-of-group input status is fetched.	• Not Imported • Import	Not Imported
Output Status Outside Group	Specify 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.

23.2 Operation Settings

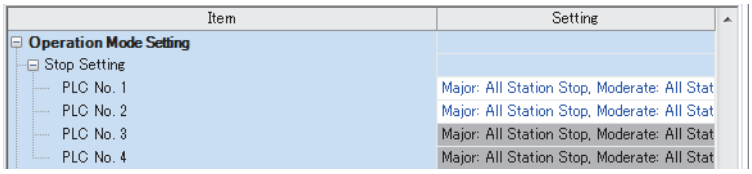
This section describes the operation settings of the multiple CPU system function.

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	Set 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	Set 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	Set 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	Set 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.

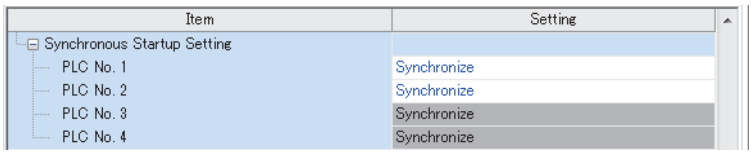
Error name	Error code
Another CPU module moderate error	1220H
Another CPU module major error	2461H, 2462H, 2470H

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	Set 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

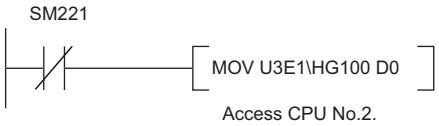
- Group setting for the synchronized start-up is available. For example, a setting in which only CPUs No.1 and No.2 start synchronously within a multiple CPU system with four CPU modules is possible.
- If a reserved (empty) CPU is specified to synchronize, it is skipped and the other CPUs in the group 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

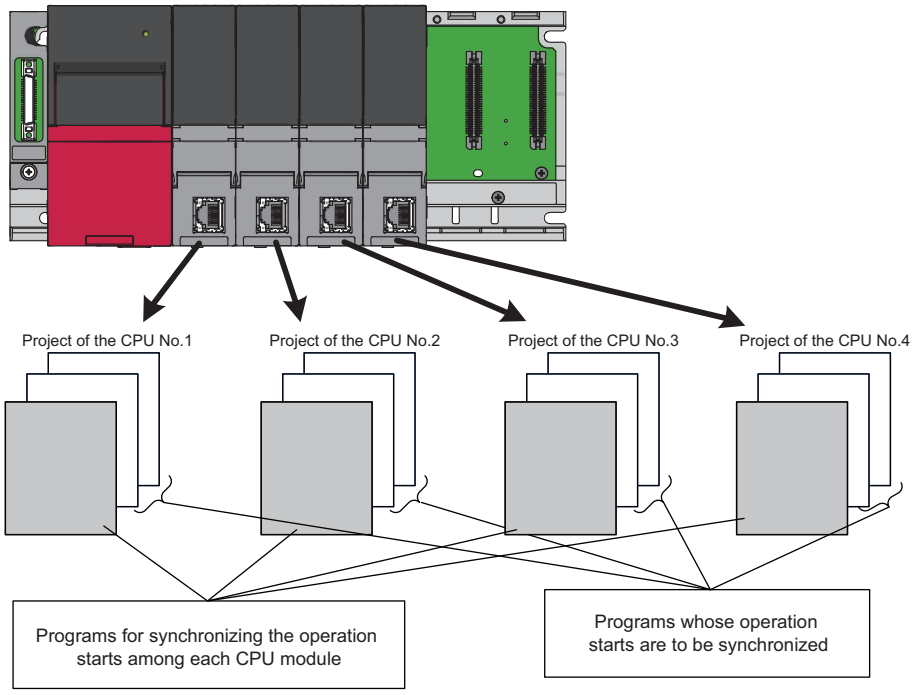
It is recommended to create a program that checks start-up of each CPU module using SM220 to SM223 (CPUs No.1 to No.4 preparation completed) when the multiple CPU synchronized startup is disabled. If a certain instruction is issued against a CPU module that has not started, the instruction executes no 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.

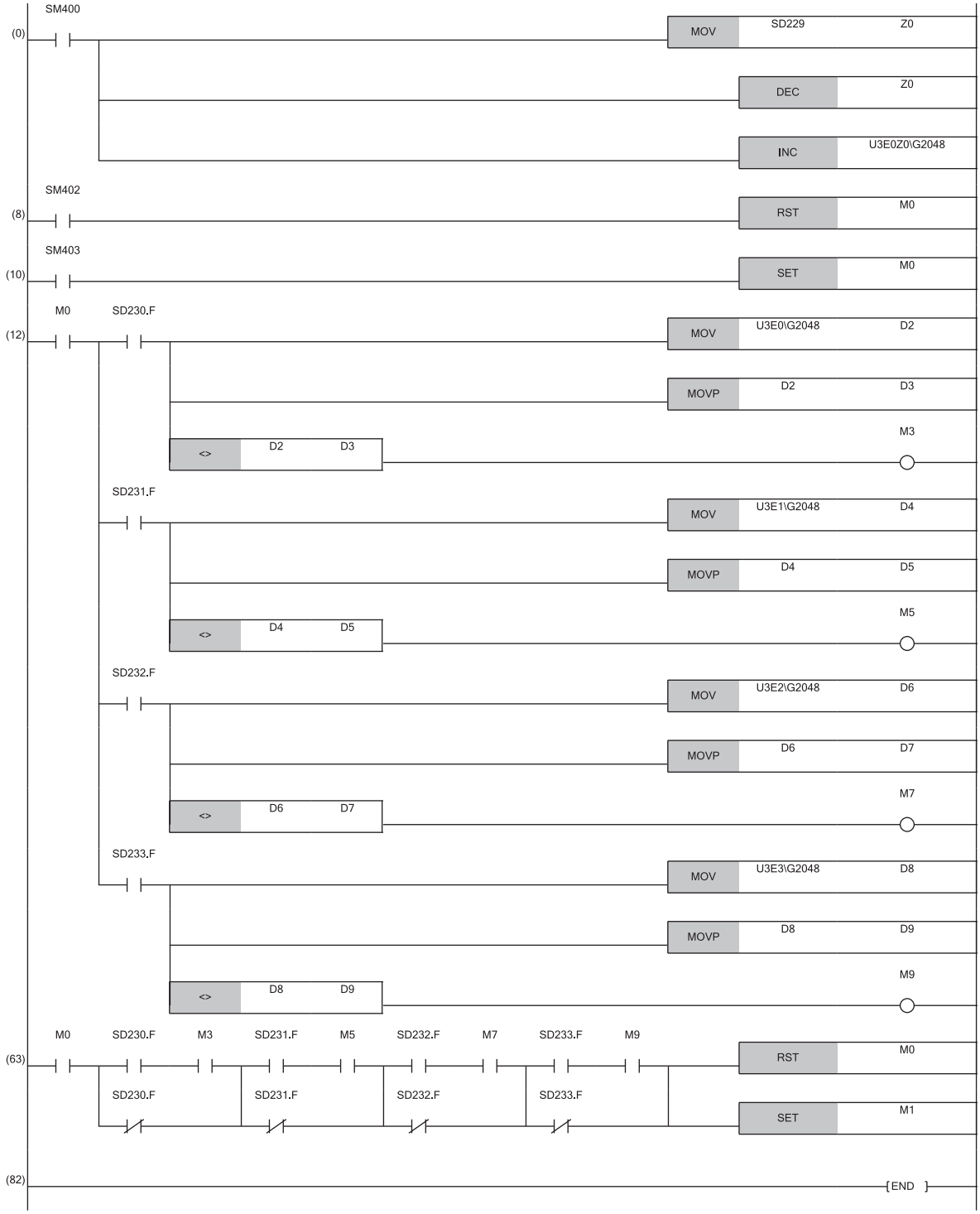
Program	Execution type	Description
Program to synchronize the start of operation processing	Scan	This is a program used to synchronize the start of operation processing across all CPU modules. This program must be specified as one for running at the beginning of the scan execution type of program. Also, 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.

■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 185 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 187 Setting Time Zone)

23.3 Multiple CPU Parameter Checking

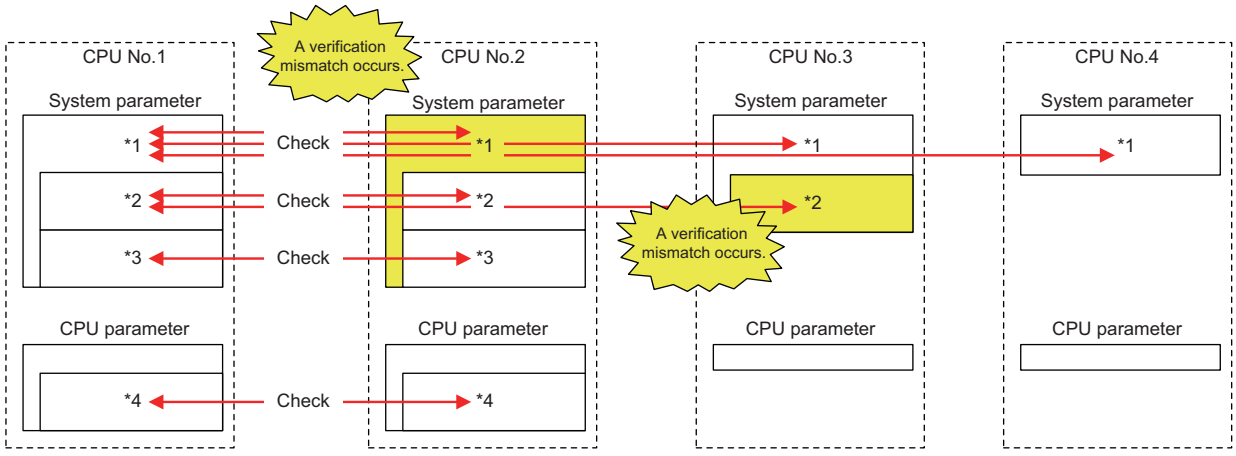
Whether the same setting is configured for between the system parameter of each CPU module and multiple CPU refresh number of points of CPU parameter is checked by the multiple CPU system at the timing shown below. However, as for the fixed scan communication setting and inter-module synchronization 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	×	×

23.4 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.

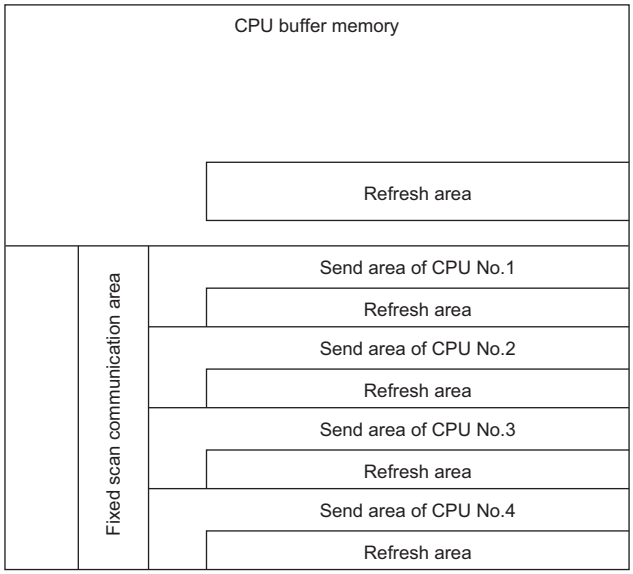
Communication method	Application	Description
Data communication with CPU buffer memory	This method is used when data is sent or received at the timing of each CPU module.	The CPU module for sending the data writes data into the CPU Buffer memory of the host CPU module. The CPU module for receiving data 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.

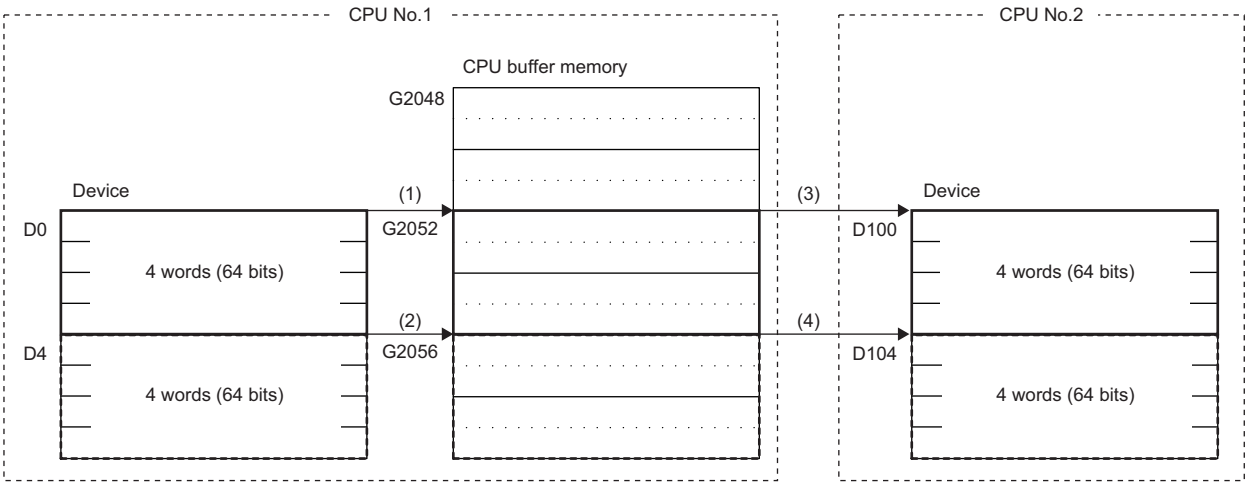


Memory	Communication method	Description	Area size
CPU buffer memory	Communication through direct access	This area is used to read/write data from/to the areas of the host CPU module or other CPU modules.*1 (☞ Page 359 Communication through direct access)	512K words fixed
Refresh area	Communication through refresh	Data communication is conducted through refresh at END processing. (☞ Page 355 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 359 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 350 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 355 Communication through refresh)	

*1 Data cannot be written to the areas of other CPU modules.

■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.



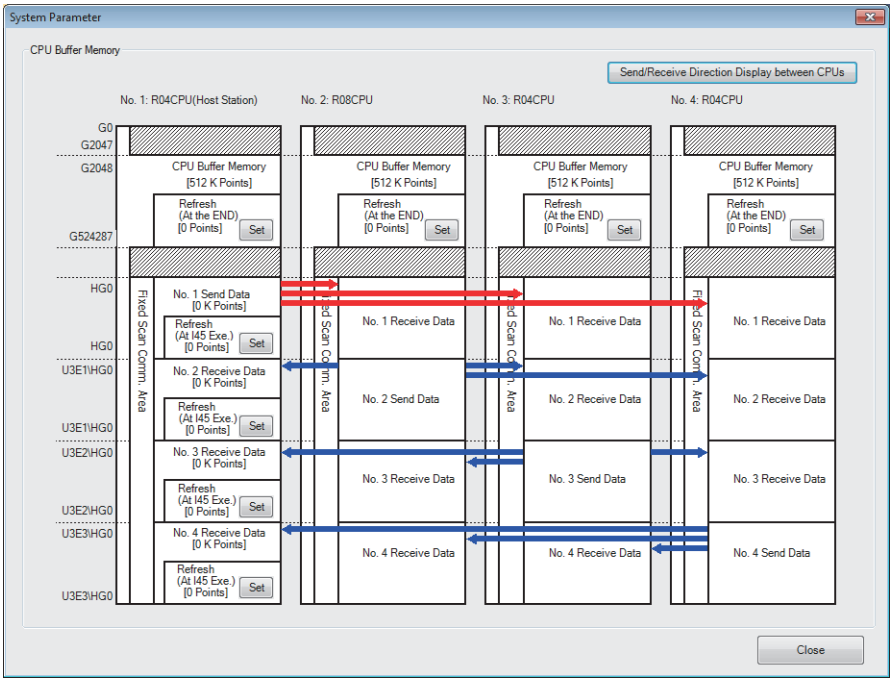
(1) The CPU module assures a 64-bit data and write the data to the CPU buffer memory. (TO U3E0 K2052 D0 K4)
(2) The CPU module assures a 64-bit data and write the data to the CPU buffer memory. (TO U3E0 K2056 D4 K4)
(3) The CPU module assures a 64-bit data and read the data from the CPU buffer memory. (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 358 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

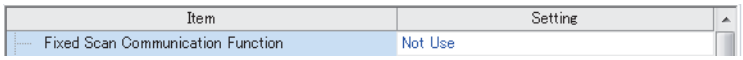
This section describes the setting for making the data communication with fixed scan communication area.

■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

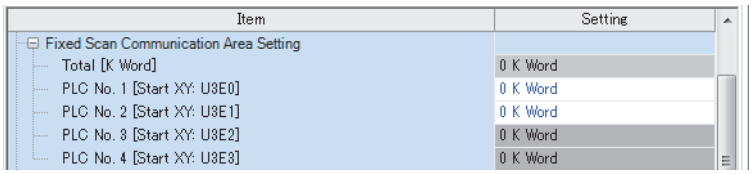
Item	Description	Setting range	Default
Fixed Scan Communication Function	Specify whether or not the fixed scan communication function should be used.	• Not Use • Use	Not Use

■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

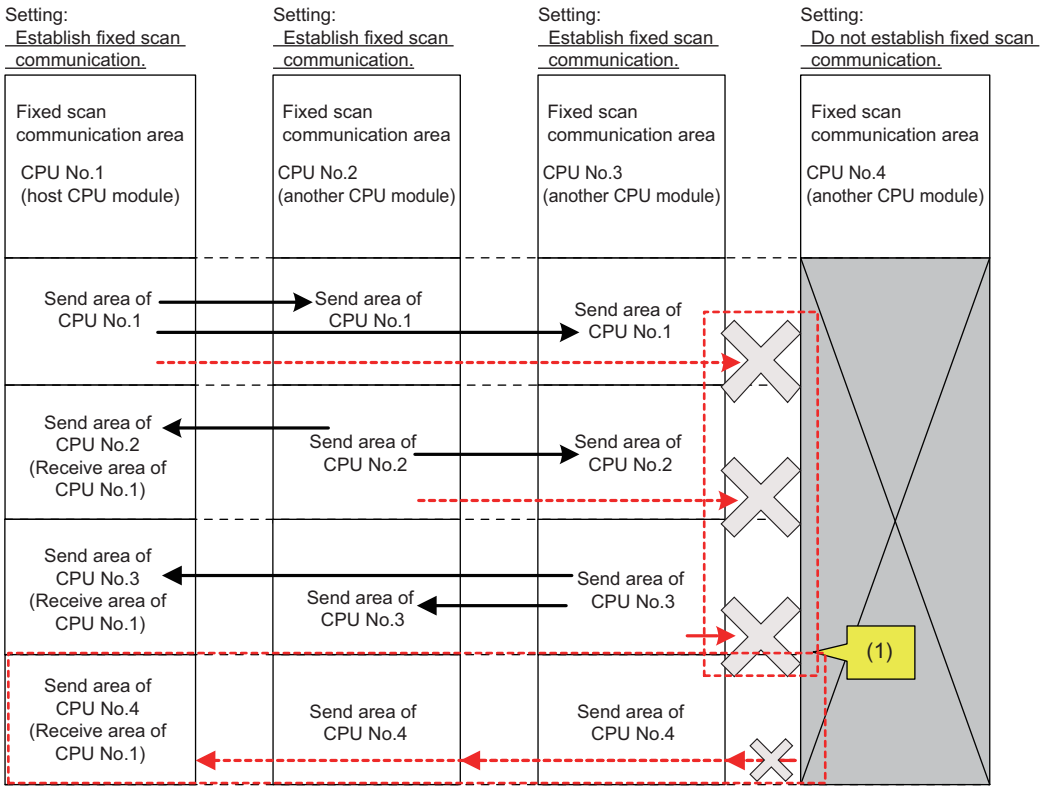
Item	Description	Setting range	Default
Total [K Word]	View the total value.	Entire system: 0 to 24K words*1	—
PLC No.1 [Start XY: U3E0]	Set the send area size for CPU No.1.	0 to 12K words	0K word
PLC No.2 [Start XY: U3E1]	Set the send area size for CPU No.2.	0 to 12K words	0K word
PLC No.3 [Start XY: U3E2]	Set the send area size for CPU No.3.	0 to 12K words	0K word
PLC No.4 [Start XY: U3E3]	Set 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"



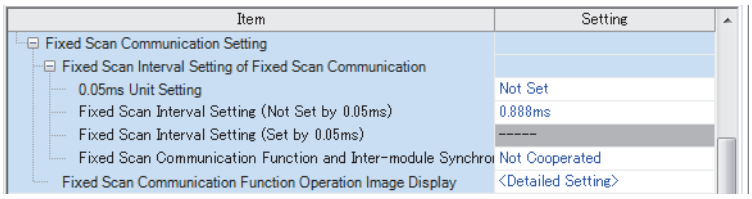
(1) Since the setting does not allow fixed scan communication, this CPU module does not send data to the other CPU modules although send areas have been reserved. This CPU module does not receive data from the other CPU modules.

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	Specify 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)	Select 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 • 0.444ms • 0.888ms • 1.777ms • 3.555ms • 7.111ms	0.888ms
Fixed Scan Interval Setting (Set by 0.05ms)	Specify the fixed scan communication cycle freely in increments of 0.05ms.	0.10 to 10.00ms	0.10ms
Fixed Scan Communication Function and Inter-module Synchronization Function	Set whether the fixed scan communication cycle cooperates with the inter-module synchronization cycle.	• Not Cooperated • Cooperate	Not Cooperated

Point

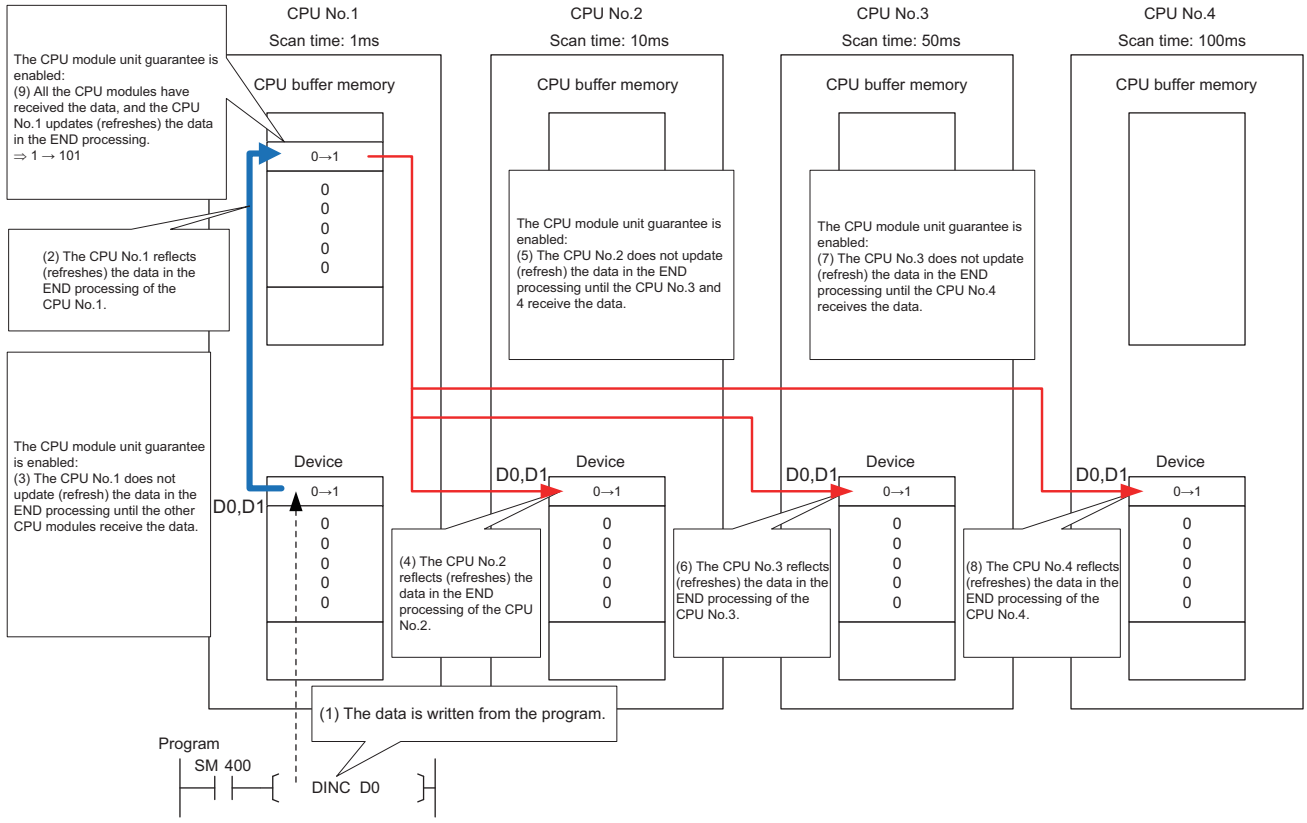
The send image for the fixed scan communication can be reviewed through the "Fixed Scan Communication Function Operation Image Display".

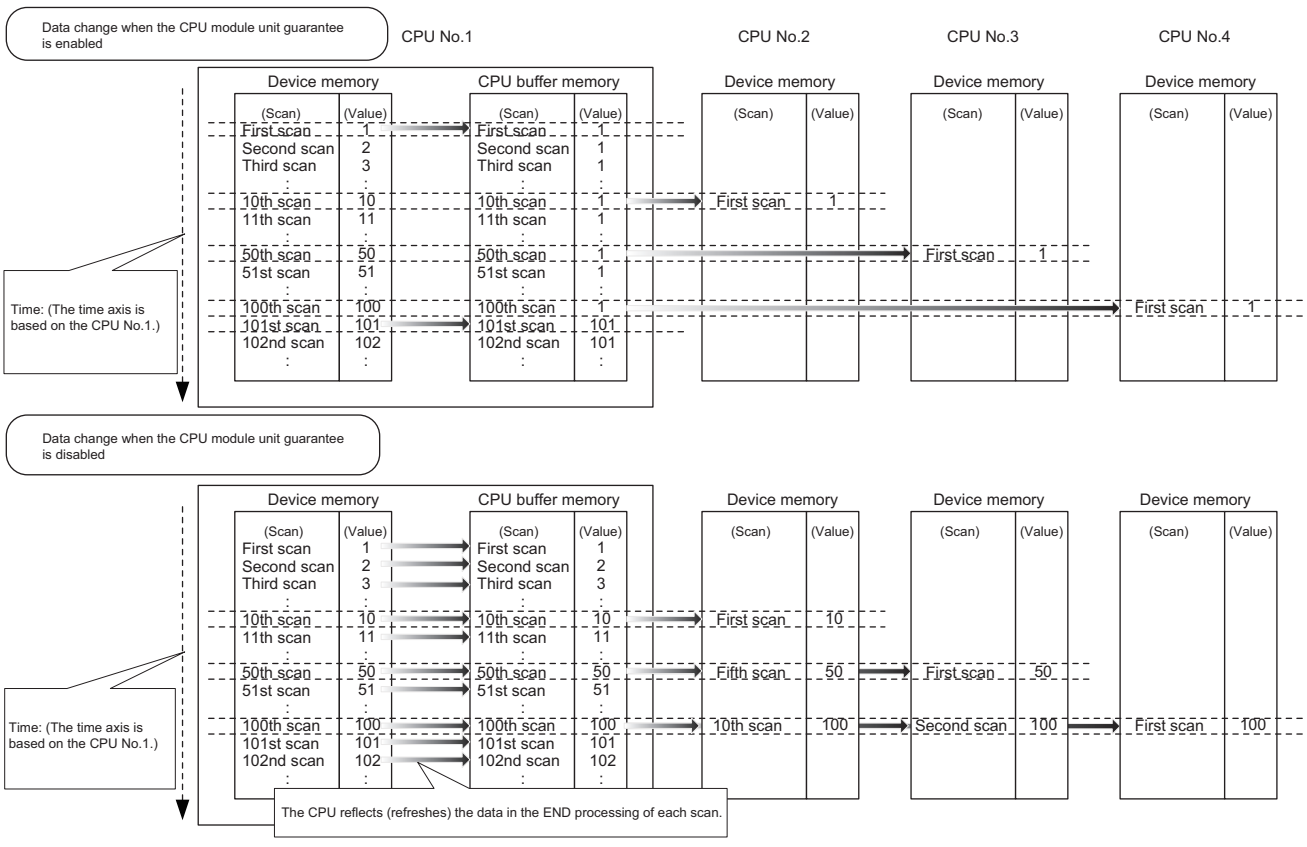
Error detection setting

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 203 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 362 Data assurance by program)

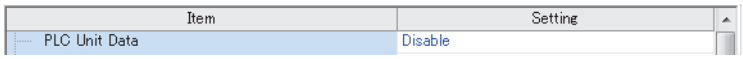
*2 Limited to the case of access within the multiple CPU synchronous interrupt program (I45). (☞ Page 364 When accessing fixed scan communication area)

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	Specify 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

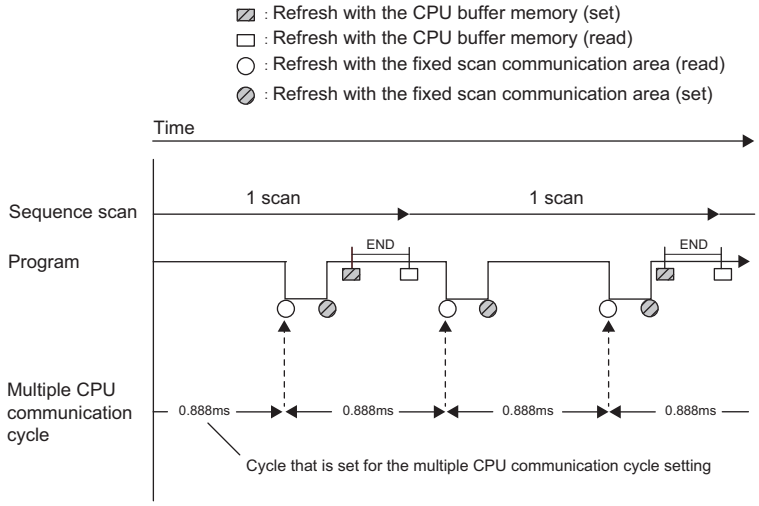
The following table shows the types of refresh.

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.

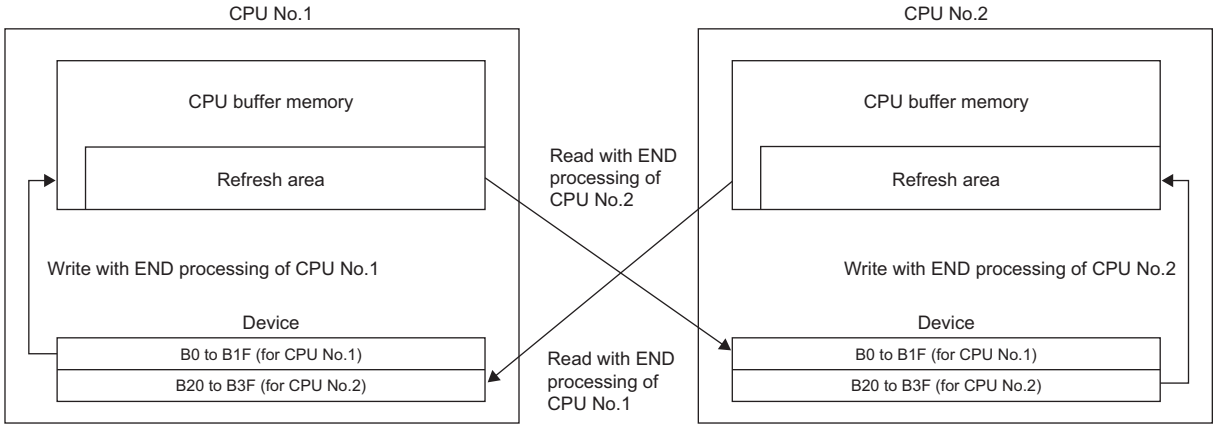


■Refresh using CPU buffer memory

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):

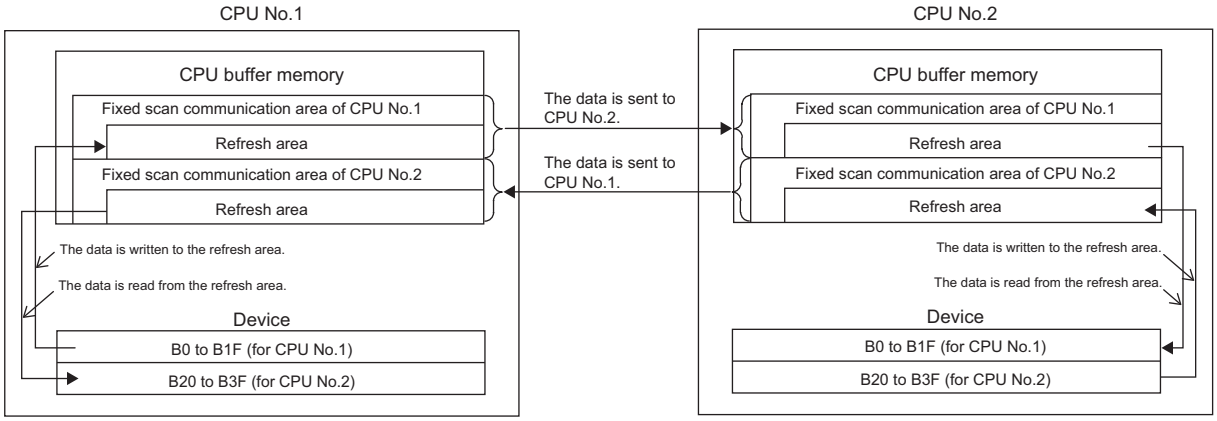


■Refresh using fixed scan communication areas

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):



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 204 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 366 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).

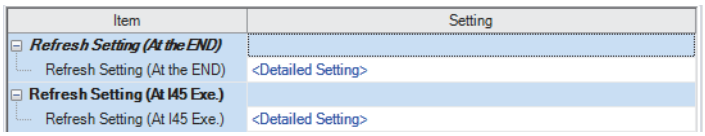
Refresh settings

The refresh can be set up with "Refresh Setting between Multiple CPUs" in "CPU Parameter".

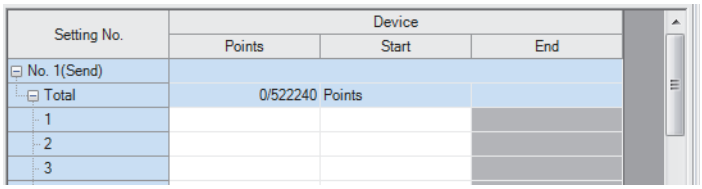
[CPU Parameter] ⇨ [Refresh Setting between Multiple CPUs]

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.

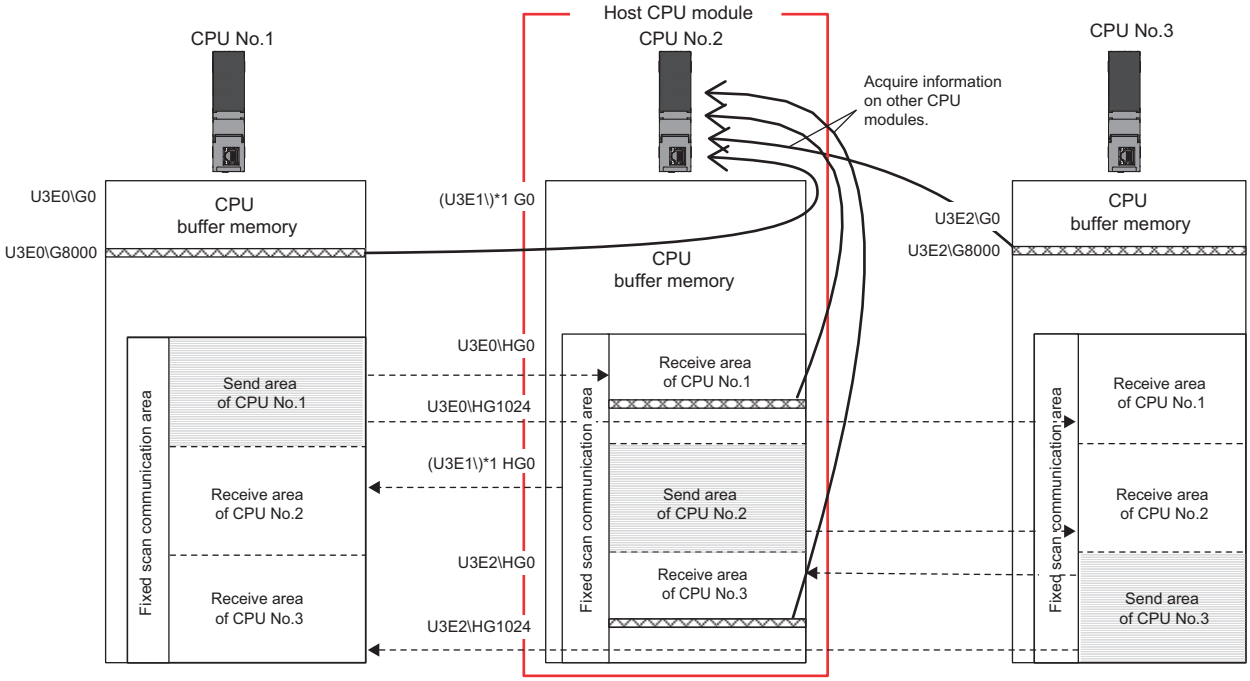
Communication through direct access

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 487 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.

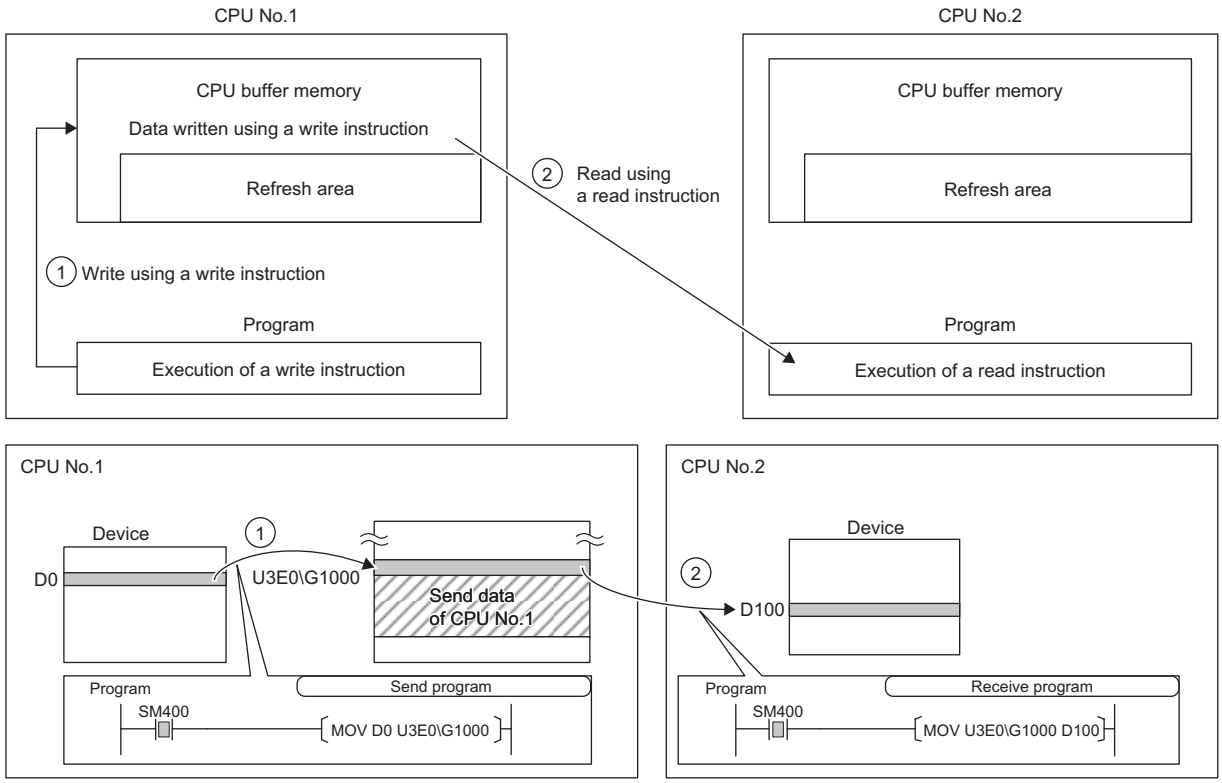
■Data communication behavior

- 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:

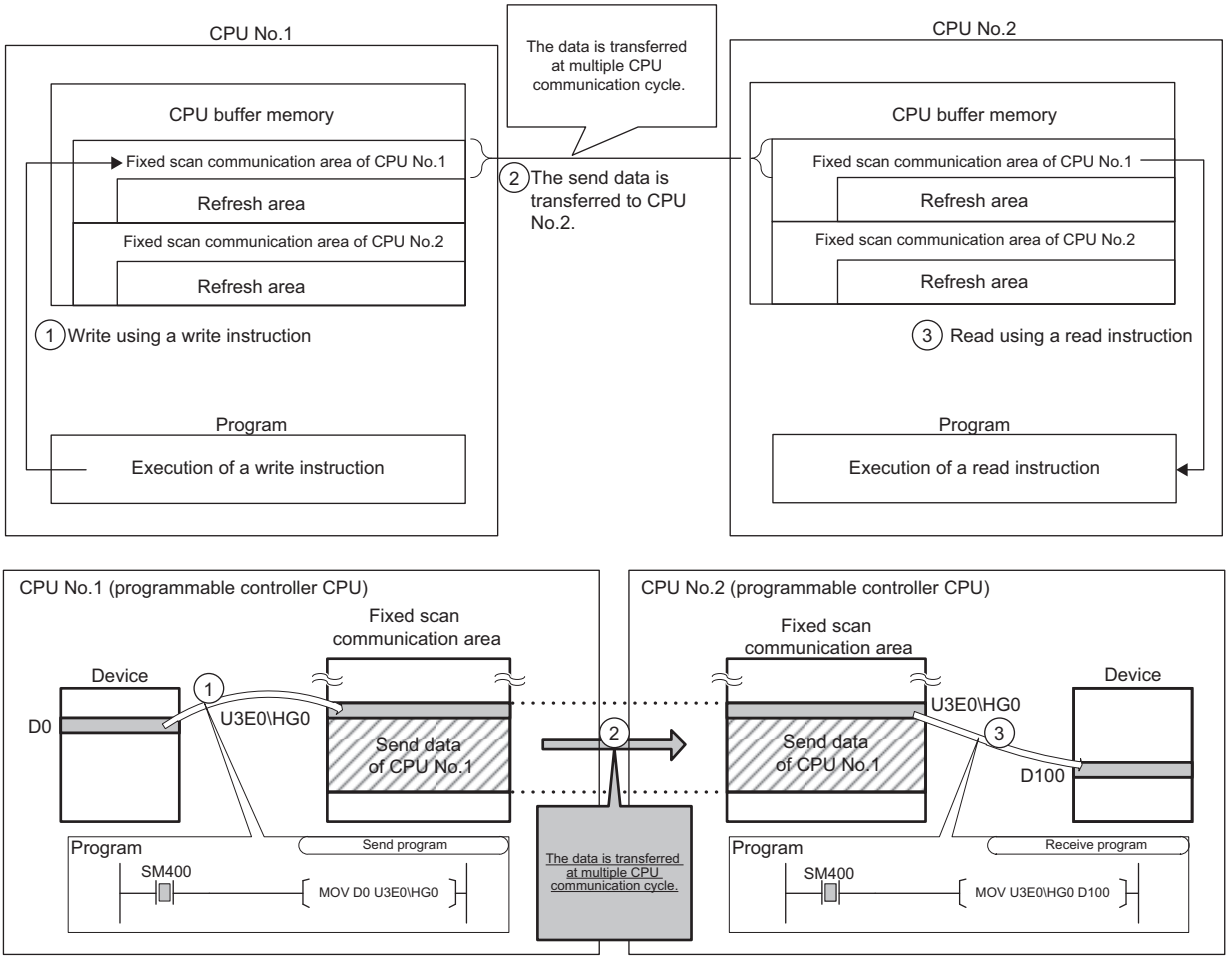


• When using an area within the fixed scan communication area

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 354 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.

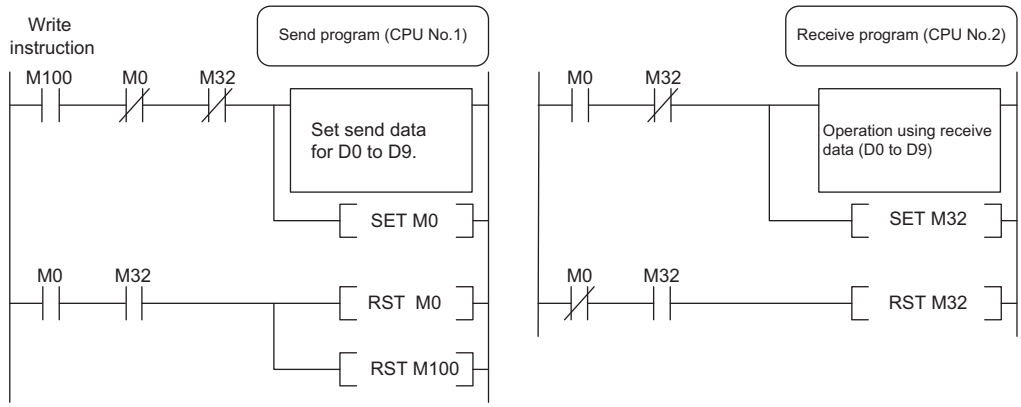
Ex.

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.

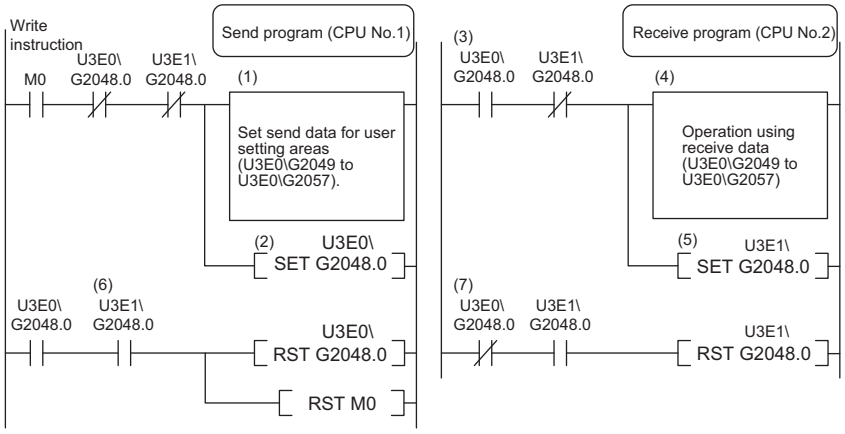
■When accessing CPU buffer memory:

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

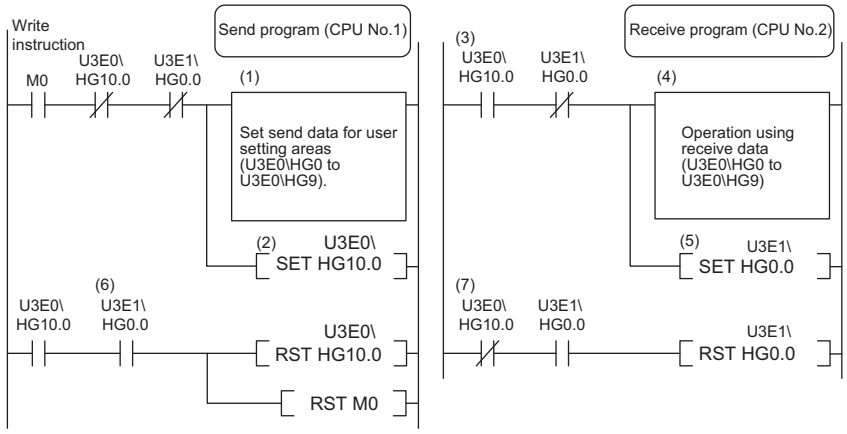
When accessing within the multiple CPU synchronous interrupt program (I45), enabling the setting of module-by-module data guarantee eliminates the need of an interlock circuit. When this refresh area is accessed within a program other than the above, or when the setting of module-by-module data guarantee is disabled, an interlock circuit is required, as with the access to the CPU buffer memory. (☞ Page 353 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.

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

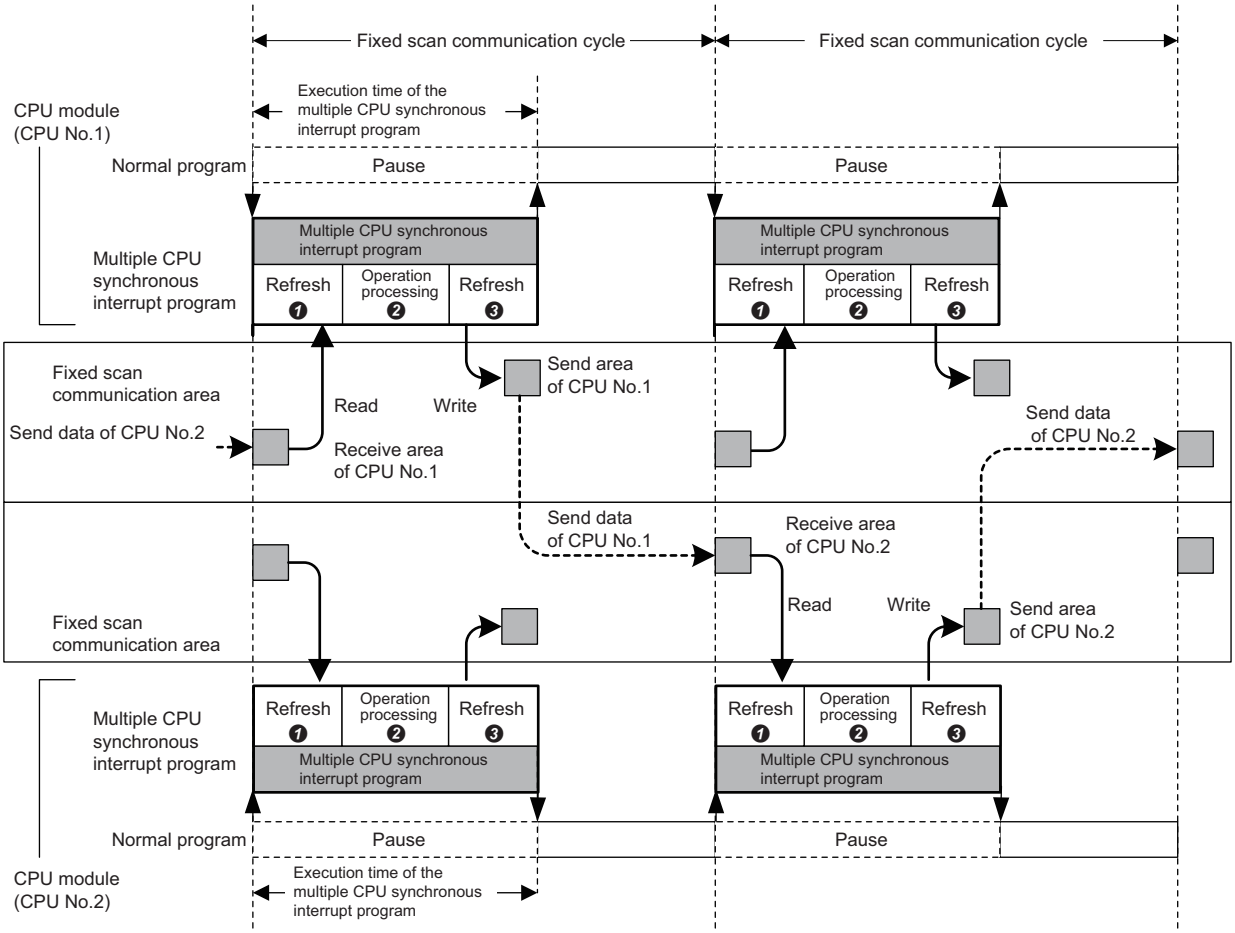
The following table lists the refresh and send/receive operation between CPU modules when the host CPU module detects a self diagnostic error. If one of the CPU modules enters into stop error state, the other CPU modules which are not in stop error state keep data stored before the stop error occurs.

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.

23.5 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 115 Interrupt Program)

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 352 Fixed scan communication setting)

Point

It is also possible to perform refresh during the multiple CPU synchronous interrupt program (I45) in execution. (☞ Page 355 Communication through refresh)

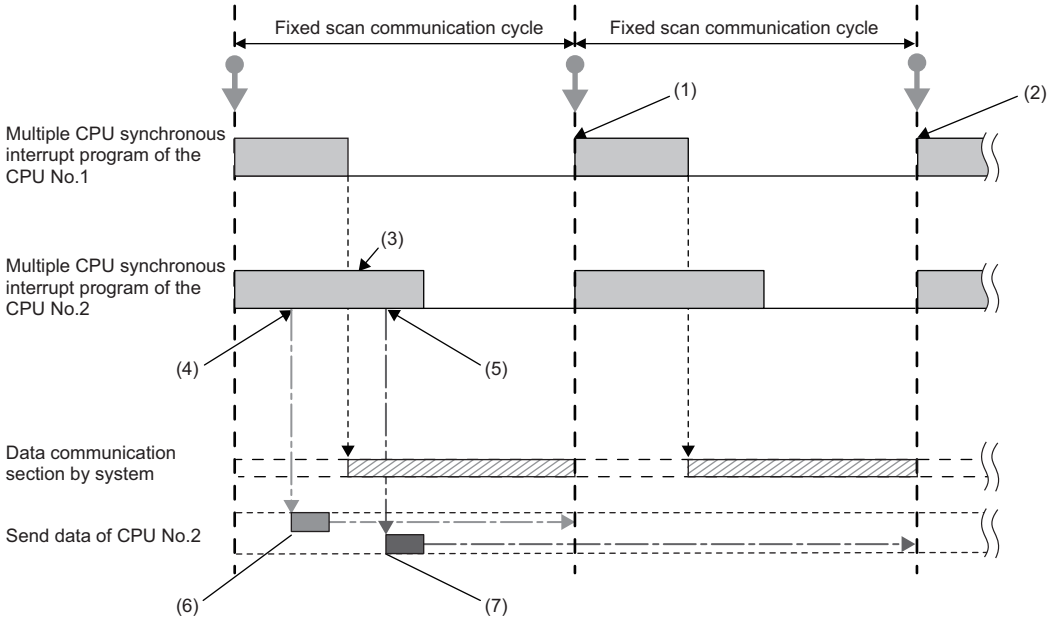
Multiple interrupt

For the multiple interrupt of the multiple CPU synchronous interrupt program, refer to the multiple interrupt function. (☞ Page 129 Multiple interrupt function)

Precautions

The precautions for the multiple CPU synchronous interrupt program are mentioned below.

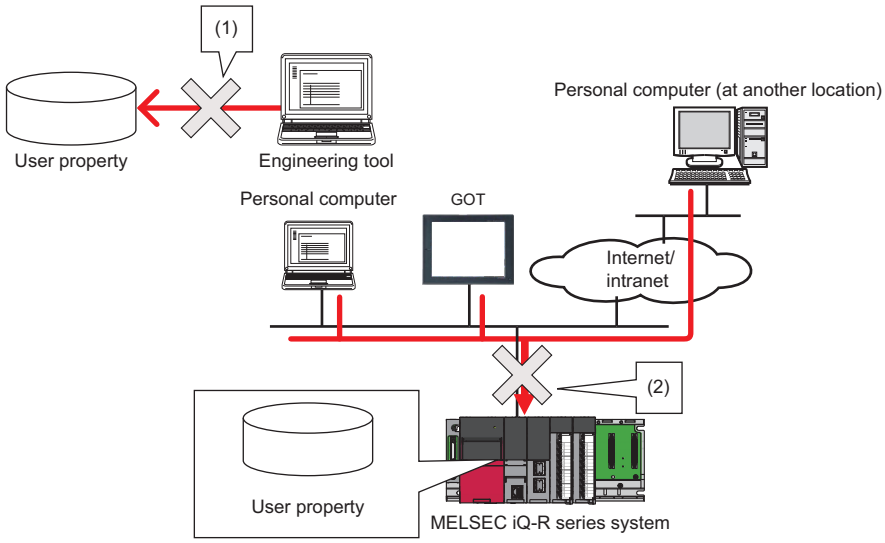
- Create a multiple CPU synchronous interrupt program so that it has the execution processing time shorter than the fixed scan communication cycle. If the interrupt program has the execution processing time equal to or longer than the cycle, the multiple CPU synchronous interrupt interval cannot be guaranteed. (☞ Page 116 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 203 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 203 Error detection setting) With this setting, when data is written with the multiple CPU synchronous interrupt program after the host CPU module starts data transfer, 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.

24 SECURITY 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	
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 369 Security key authentication for a CPU module
	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	

*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)

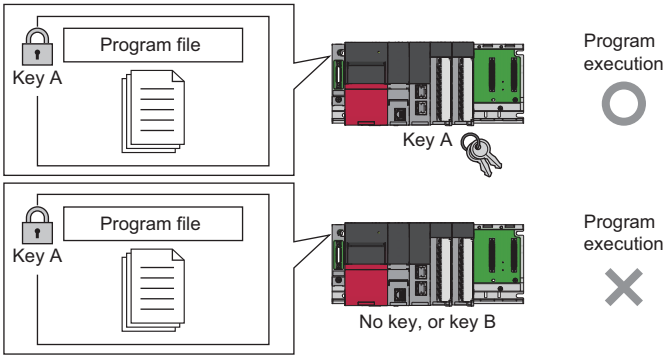
Furthermore, if a personal computer with a security key registered has failed, the locked project data cannot be viewed and edited. We assume no responsibility whatsoever for any damage or loss to the user and any other individual or organization, resulting from such a situation. Therefore, the user 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.

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.

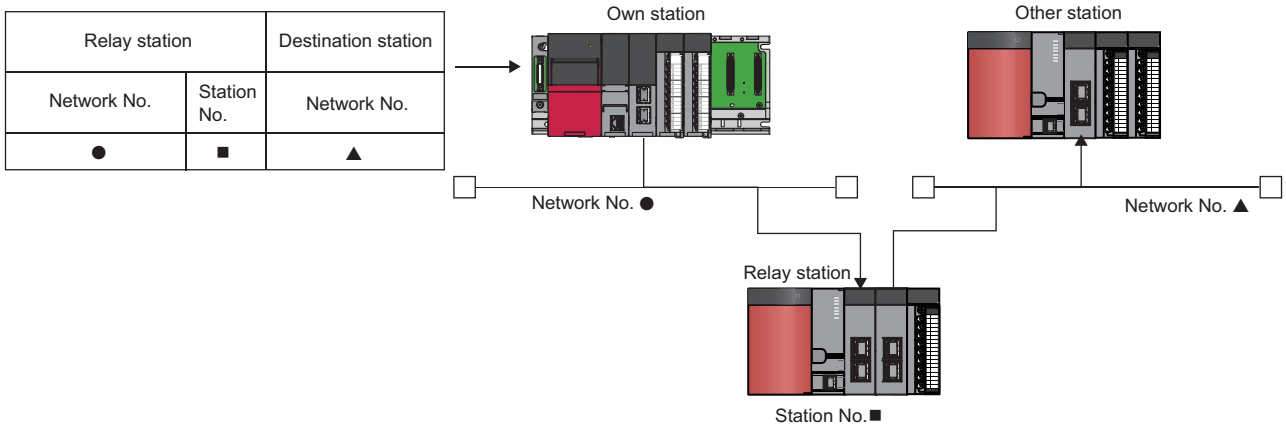
The security key written in the CPU module is retained even after power-off of the CPU module.

For details, refer to the following.

GX Works3 Operating Manual

25 ROUTING SETTING

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.

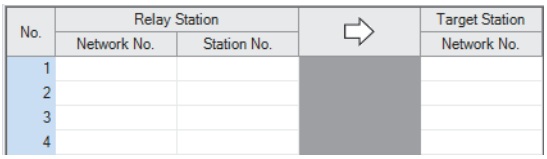


25.1 Setting Method

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.

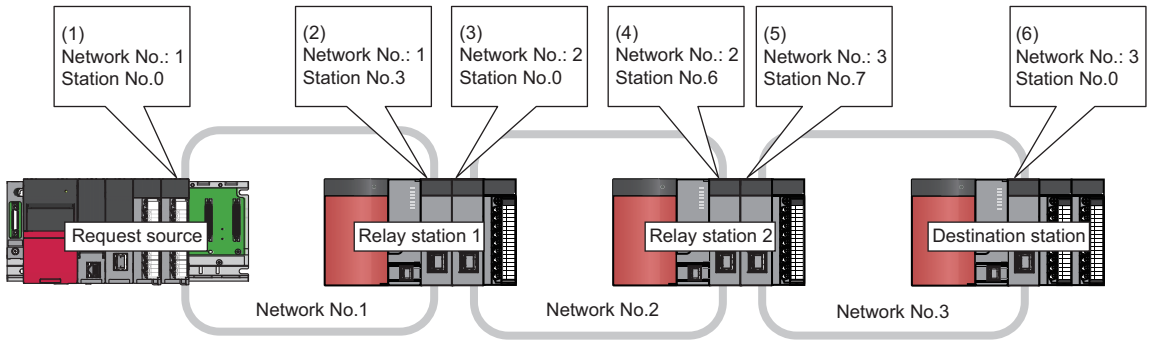
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

25.2 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.



Station	Request route	Response route
Request source	(1) Relay station: Network No. 1, Station No. 3; Destination station: Network No. 3 The data is transferred to the relay station (2) of the own network to go to the network No.3.	(1)
Relay station 1	(2) (3) Relay station: Network No. 2, Station No. 6; Destination station: Network No. 3 The data is transferred to the relay station (4) of the own network to go to the network No.3.	(2) (3) The data has been transferred to the network No.1, and is to be transferred to (1).
Relay station 2	(4) (5) The data has been transferred to the network No.3, and is to be transferred to (6).	(4) (5) Relay station: Network No. 2, Station No. 0; Destination station: Network No. 1 The data is transferred to the relay station (3) of the own network to go to the network No.1.
Destination station	(6)	Relay station: Network No. 3, Station No. 7; Destination station: Network No. 1 The data is transferred to the relay station (5) of the own network to go to the network No.1.

25.3 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.

26 REDUNDANT FUNCTION

This chapter describes the redundant system functions.

Function		Description	Reference
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 374 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 377 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 391 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 409 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 417 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 422 Program Execution in Both Systems
Redundant system operation setting		Set the redundant system operation in the redundant system settings of the CPU parameter.	Page 429 Redundant System Operation Setting
Redundant function module communication test		Checks the redundant function module itself for errors when its communications are unstable.	Page 434 Redundant Function Module Communication Test
Redundant system with redundant extension base unit	Settings for redundant system with redundant extension base unit	Set the operation of the redundant system with redundant extension base unit in the redundant system settings of the CPU parameter.	Page 435 Settings for Redundant System with Redundant Extension Base Unit
	Automatic standby system recovery	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 436 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 561 Replacement/Addition of Extension Cables (Online)
Control system start-up at one system error		When one system detects an error, such as a CPU module failure, module failure, and module disconnection, at the power-on or reset and the redundant system cannot operate, the normal system starts the control system.	Page 438 Control System Start-up at One System Error
SLMP communications		• There are notes for system switching when the system IP address matching function is not used. • When a remote operation command is executed, an operating status mismatch occurs between the CPU modules and the system switching is disabled. • In 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 different IP addresses in the system A and the system B.	Page 443 SLMP Communications

26.1 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.

Procedures

Execute the operation mode change to the CPU module of the control system from the "Redundant Operation" window of the engineering tool.

Changing the operation mode to the separate mode

This section describes how to change the operation mode to the separate mode.

To prevent tracking transfer in separate mode, turn off the tracking trigger in advance. (☞ Page 397 Tracking trigger)

1. Connect the engineering tool to the CPU module of the control system.
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 CPU module of the standby system enters the wait state for the RUN-transition instruction (same as the STOP state), and the PROGRAM RUN LED flashes.
5. Change the connection destination to the standby system on the engineering tool. Change the operating status of the CPU module using the remote operation (remote STOP → remote RUN).
6. The CPU module enters the RUN state and executes a program, and the PROGRAM RUN LED turns on.

Point

For the step 5 (changing the operating status of the CPU module to the RUN state), the following methods are also available.

- Changing the RUN/STOP/RESET switch from the STOP position to the RUN position
- Changing the operating status from the STOP state to the RUN state 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 if the control system CPU module is turned off and on or reset. *1

- When the CPU module of the control system is powered off and on or reset, the system starts up as the control system in separate mode.
- When the CPU module of the standby system is powered off and on or reset, the system starts up as the standby system in separate mode.
- When the CPU modules of both systems are simultaneously powered off and on or reset, the systems start up in backup mode.

*1 In a redundant system with redundant extension base unit, do not power off or reset the CPU module of the control system during operation in separate mode because control cannot be continued.

Before powering off or resetting the CPU module of one system in separate mode, ensure the following:

- The other system has been started up and the READY LED is on.
- When the operating status of the CPU module in the other system has been changed to the RUN state, the PROGRAM RUN LED is on.

If the CPU module of one system is powered off or reset during initial processing (when powered on/switched to RUN) of the CPU module of the other system, the operation mode mismatch occurs between two systems and a stop error occurs.

Changing the operation mode to the backup mode

The operation mode can be changed to the backup mode only in the communication path where the operation mode was changed to the separate mode.

This section describes how to change the operation mode to the backup mode.

1. Set the CPU modules of the control system and the standby system to have the same file configuration and operating status. (To prevent an error from occurring after the operation mode is changed from the separate mode to the backup mode and the system consistency check is restarted.)
2. Connect the engineering tool to the CPU module of the control system.
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 397 Tracking trigger)

Point

- When the CPU modules of both systems are simultaneously powered off and on or reset, the systems start up in backup mode.
- Even in a different communication path, when the CPU module of the control system has been powered off or reset in separate mode, the operation mode can be changed from the separate mode to the backup mode using the engineering tool.
- When the operation mode is changed to the separate mode, the CPU module of the standby system enters the wait state for the RUN-transition instruction (the PROGRAM RUN LED flashes). If the operation mode is changed to the backup mode in this state, the CPU module enters the RUN state (the PROGRAM RUN LED stops flashing).

Precautions

The following describes the precautions for changing the operation mode of a redundant system.

Operation mode change in the wait state for the RUN-transition instruction

When the operation mode is changed to the separate mode, the CPU module of the standby system enters the wait state for the RUN-transition instruction (same as the STOP state). If the operation mode is changed to the backup mode in this state, the CPU module of the standby system enters the RUN state.

The following describes the precautions for this state.

- If the SP.CONTSW instruction is executed before the CPU module of the standby system enters the RUN state (during initial processing (when switched to RUN)), an operating status mismatch occurs between two systems and a continuation error occurs. (☞ Page 390 System switching during initial processing/initial processing (when switched to RUN))
- If the CPU module of the standby system takes time to change its operating status to the RUN state due to causes such as access to a file in the CPU module of the standby system, an operating status mismatch occurs between two systems and a continuation error occurs.

Operation mode change during initial processing (when powered on/switched to RUN)

When the operation mode is changed during initial processing (when powered on or switched to RUN), operation mode change occurs after the initial processing (when powered on or switched to RUN) is completed. Even if the initial processing (when powered on or switched to RUN) takes time and a communication error occurs, operation mode change occurs after the initial processing (when powered on or switched to RUN) is completed.

Consecutive executions of operation mode change

To consecutively execute the operation mode change, certain time intervals between executions are required. If the operation mode of the redundant system is changed again before the operating status of the CPU module is changed by the previous operation mode change, an operating status mismatch occurs between two systems and a continuation error may occur.

26.2 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. For debugging and maintenance purpose, users can switch the systems at any desired timing.

System switching method

The following two methods are available for system switching: automatic system switching (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 of the redundant system, 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).
- A manual system switching request is executed for the CPU module of the control system.

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

When a redundant system is in backup mode, it judges whether system switching is required or not. Then, 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

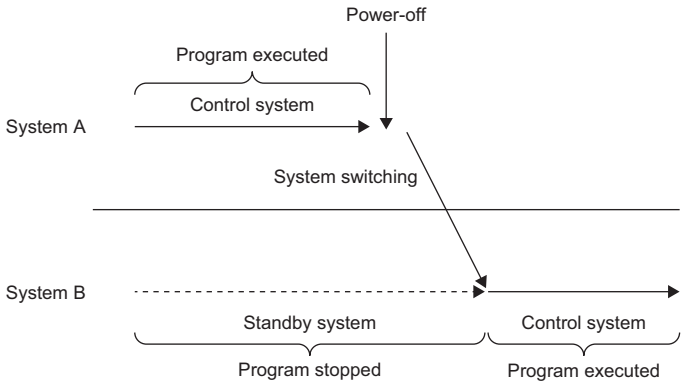
In a redundant system, the CPU module of the standby system monitors the status of the control system. If the control system is unable to control the redundant system in the following cases, the CPU module of the standby system is switched to the control system and the CPU of the new control system continues the control over the redundant system.

- The CPU module of the control system has been turned off.
- The CPU module of the control system has been reset.
- A hardware failure has occurred in the CPU module of the control system.*1

*1 The CPU module of the control system is switched to the standby system even when the current standby system is not ready to switch. (☞ Page 384 Execution availability of system switching)

Ex.

Operation when the CPU module of the control system (system A) is powered off



Point

When the redundant function module has been removed from the base unit or the base unit has failed under any of the following conditions, system switching occurs.

- When the redundant system with redundant extension base unit is configured
- When a network is built using a CC-Link IE Controller Network module*2
- When a network is built using a CC-Link IE Field Network module

*2 This applies when the CPU module with the firmware version of "09" or later is used in both systems.

■Precautions when systems are switched by reset operation using the RUN/STOP/RESET switch

When system systems are switched by reset operation using 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 using the RUN/STOP/RESET switch of the CPU module in the control system while 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 using the RUN/STOP/RESET switch.

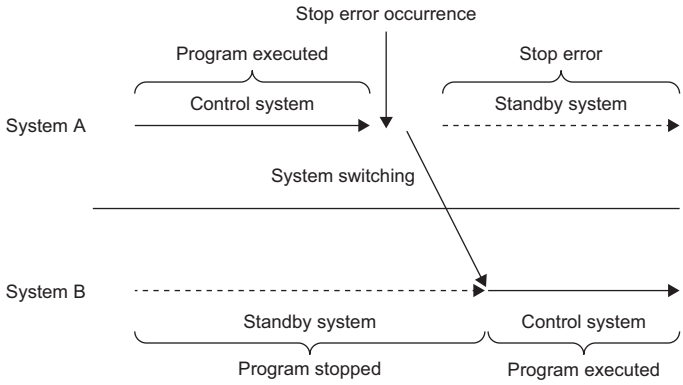
■Stop error of the CPU module

When a stop error occurs in the CPU module of the control system, the CPU module of the standby system is notified of a system switching due to the stop error, and the CPU module of the standby system is switched to the control system. The CPU module of the control system in which the stop error has occurred is switched to the standby system. *1

*1 If a WDT error has occurred, the CPU module of the control system is switched to the standby system even when the current standby system is not ready to switch. (☞ Page 384 Execution availability of system switching)

Ex.

Operation when a stop error occurs in the CPU module of the control system (system A)



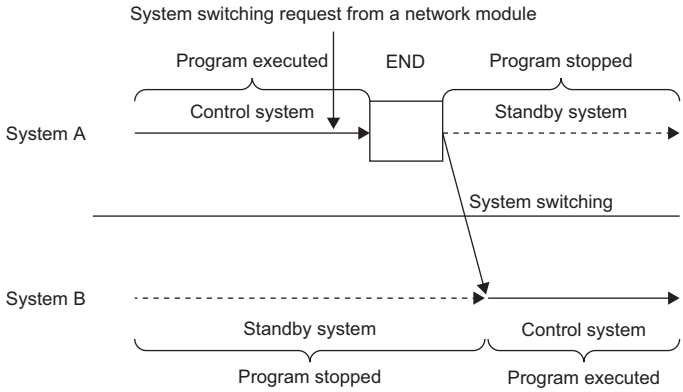
■System switching request from a network module

The network module on the main base unit of the control system requests the CPU module to switch systems when a communication error or a disconnection is detected. (The network module on an extension base unit does not request the CPU module to switch systems when an error is detected.)

When the CPU module of the control system receives a system switching request from the network module, the systems are switched after the END processing.

Ex.

Operation when the CPU module receives a system switching request from a network module



Point

When disconnecting the network cable of the control system for maintenance or other reasons, disconnect the network cable of the standby system in advance. 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 390 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 at a desired timing.

Point

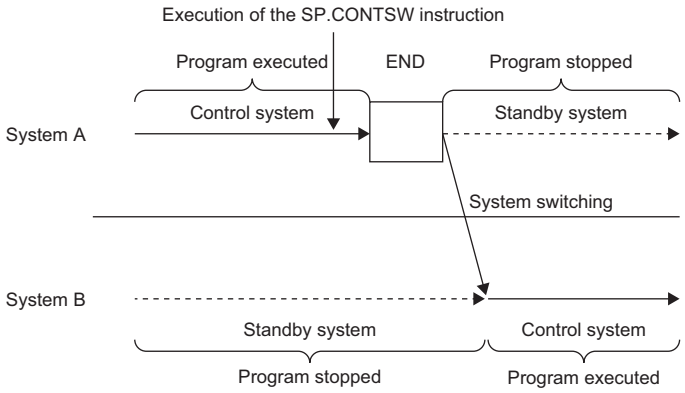
- After turning on SM1646 (System switching by a user), perform the manual system switching in the control system.
- When the manual system switching is disabled by the DCONTSW instruction, execute the ECONTSW instruction. The system switching is enabled in the initial status.

■System switching by the SP.CONTSW instruction

When the SP.CONTSW instruction is executed in the CPU module of the control system, the systems are switched at the END processing after the instruction execution.

Ex.

Operation of system switching by the SP.CONTSW instruction



Restriction

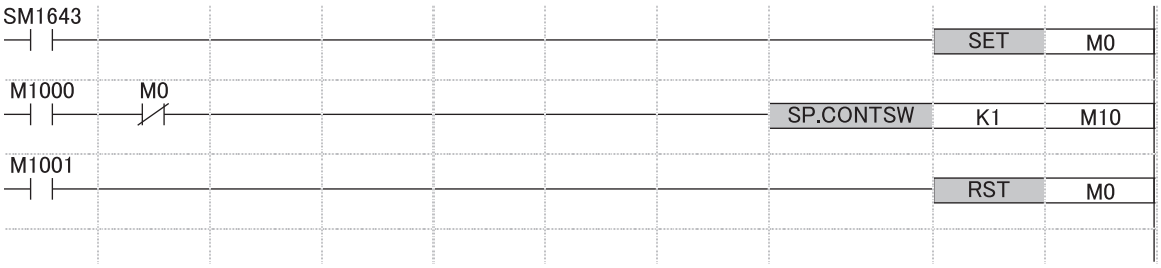
When the SP.CONTSW instruction is executed while the tracking transfer is in asynchronous transfer mode, such as immediately after the transition to RUN in backup mode or in separate mode, several scans may be required for the system switching.

Point

Once the devices and labels specified in "Tracking Setting" are transferred, the CPU modules of both the control system and the standby system have the same specified devices and labels. Thus, if the systems are switched by using the SP.CONTSW instruction in the CPU module of the control system, switching may also be performed on the CPU module of new control system.

When using the SP.CONTSW instruction, create a program that does not execute the SP.CONTSW instruction again on the CPU module of the new control system 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

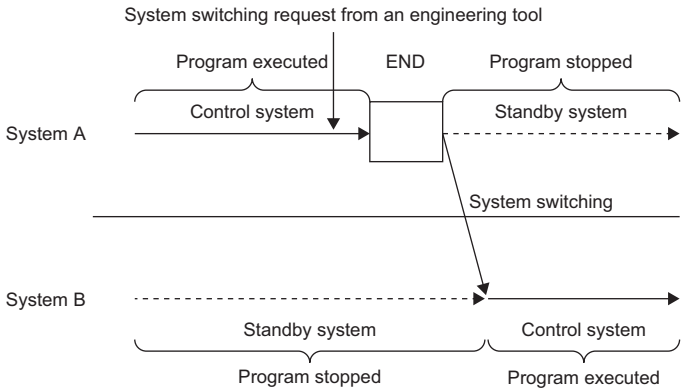
When the engineering tool sends a system switching request to the CPU module of the control system, the systems are switched after the END processing.

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 status of the CPU modules are the same.

Item		CPU module of the new control system	CPU module of the new standby system
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 the start.	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 399 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)		Data is input when an instruction using the direct access input (DX) is executed in the program after system switching.	No operation is performed because the program does not operate.*1
Direct access output (DY)		Data is output when an instruction using the direct access output (DY) is executed in the program after system switching.	
FROM/TO instructions		These instructions are executed when the instruction execution condition is satisfied in the program after system switching.	
Instruction that requires several scans		These instructions are executed when the instruction execution condition is satisfied in the program after system switching.	• 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 backup mode. • The constant scan function is enabled in separate mode.

*1 The operation varies in a program executed in both systems. (☞ Page 425 Operation at system switching)

*2 The operation varies when the signal flow memory is not transferred. (☞ Page 447 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 429 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 CPU module of the standby system		×	○[*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 program change		○	○	○	○	○[*2*5]	×	×
Operating status mismatch of the CPU modules in both systems		○	○	○	○	×	×	×
During system switching		○	○	○	○	○	×	×
During online module change	Redundant function module	×	○[*6]	○[*6]	×	×	×	×
	Module on the main base unit in a redundant system with redundant extension base unit	○	○	○	○	○	×	×
	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 program 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 program 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 419 Files)

*6 Although the CPU module of the control system is switched to the standby system, the current 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 CPU module of the standby system		×	×	×	×	×
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 program change		×	×	×	○	○
Operating status mismatch of the CPU modules in both systems		×	×	×	○	○
During system switching		×	×	×	×	×
During online module change	Redundant function module	×	×	×	×	×
	Module on the main base unit in a redundant system with redundant extension base unit	×	×	×	×	×
	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 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).

*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 In a redundant system with redundant extension base unit, If the system switching timing and the interrupt timing of the internal timer overlap, 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 386 Event history
Special relay (SM)/Special register (SD)	System switching result and detailed information	Page 386 Special relay/Special register
CTRL LED and SBY LED of the redundant function module	System switching result	Page 387 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) of the control system.

For the values stored in the special relay and special register, refer to the following.

- Special relay (☞ Page 679 Redundant function)
- Special register (☞ Page 715 Redundant function)

■Special relay

The following table shows the special relay numbers for system switching and the storage status of the CPU modules in the control system and standby system.

○: Stored, ×: Not stored

SM No.	Name	Storage status at system switching	
		CPU module of the new control system	CPU module of the new standby system
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 numbers for system switching and the storage status of the CPU modules in the control system and standby system.

○: Stored, ×: Not stored

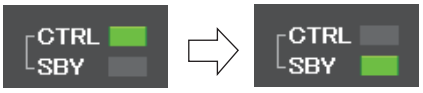
SD No.	Name	Storage status at system switching	
		CPU module of the new control system	CPU module of the new standby system
SD1642	BACKUP/SEPARATE LED flashing cause	×	×
SD1643	System switching cause	×	○
SD1644	Cause of failure of system switching*1	×	×
SD1645	System switching request status from a network 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.

Control system → Standby system



Standby system → Control 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 communications between the redundant function module and a CPU module, the systems are switched.	Page 389 Error in the redundant function module
When both systems operate as standby systems	• If a communication error has occurred 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 389 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 389 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 390 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 390 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 a redundant system with redundant extension base unit, 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 redundant function module has been removed from the base unit or the base unit has failed under any of the following conditions, system switching occurs.
Therefore, build a network that meets any of those conditions when system switching is required at the redundant function module removal from the base unit or the main base unit failure.

- When the redundant system with redundant extension base unit is configured
- When a network is built using a CC-Link IE Controller Network module*1
- When a network is built using a CC-Link IE Field Network module

*1 This applies when the CPU module with the firmware version of "09" or later is used in both systems.

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.

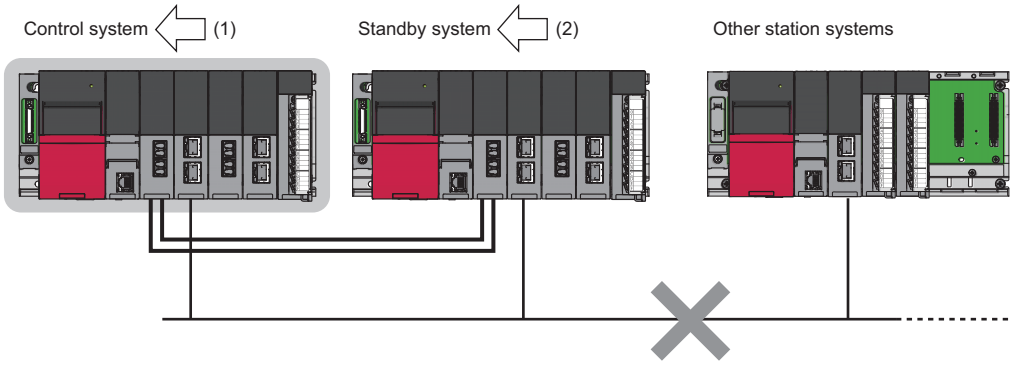
Item		Operation of both systems
LED of the redundant function module	BACKUP LED	Flashing (in backup mode)
	SEPARATE LED	Flashing (in separate mode)
	CTRL LED	Off
	SBY LED	On
Special relay	SM1635 (Standby system judgment flag)	On
System switching request	System switching request from a network module	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 199 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 in the CPU module of the control system and the BACKUP LED of the redundant function module flashes because a system switching disable cause exists.

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 a redundant system with redundant extension base unit, 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 a redundant system with redundant extension base unit

■When the extension cable is redundant

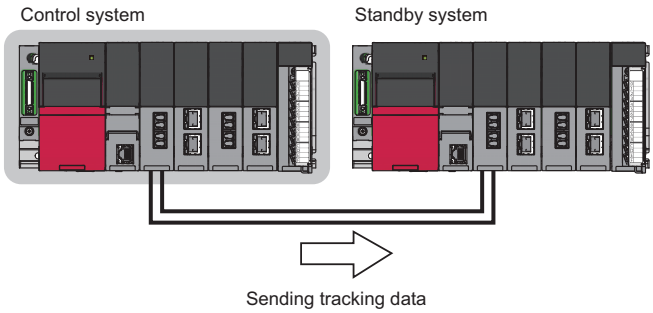
When the extension cable between the main base module of the control system and the extension base unit or the extension cable (on the active side) between extension base units fails or is disconnected, a bus access error occurs. After the systems are switched due to a stop error, control is continued 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.

26.3 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 CPU module of the control system sends tracking data to the CPU module of the standby system 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 CPU module of the standby system receives the tracing 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 395 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 400 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 398 When the "Process Control Extension Setting" is enabled)

Restriction

When a tracking transfer is not performed, the possible causes are the following:

- The CPU module of the standby system 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 393 Devices that can be specified
	Special relay	○ (Auto transfer)	×	Page 394 Special relay
	Special register	○ (Auto transfer)	×	Page 395 Special register
Global label*2 (Except module labels)		○	○	Page 401 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 399 Tracking transfer setting for the signal flow memory
SFC information		○ (Auto transfer)	×	—
PID control instruction information		○ (Auto transfer)	×	Page 395 PID control instruction information
CPU buffer memory		×	×	—
System data		○ (Auto transfer)	○ (Auto transfer)	Page 395 System data

*1 The data that is used in a program executed in both systems is not transferred. (☞ Page 422 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 394 Auto tracking data)

Point

When using the SFC program, set all points of step relay (S) within the device transfer range.

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 659 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 682 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.

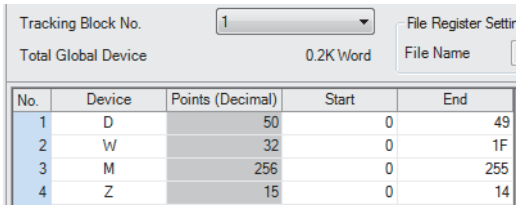
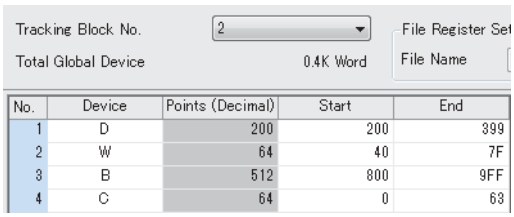
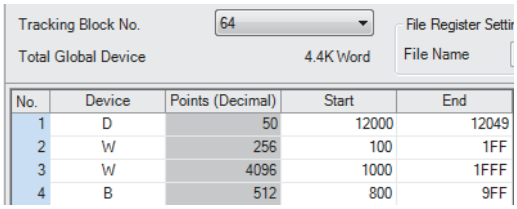
Tracking block

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

Tracking block No.	Setting example
1	 No. / Device / Points (Decimal) / Start / End 1 D 50 0 49 2 W 32 0 1F 3 M 256 0 255 4 Z 15 0 14
2	 No. / Device / Points (Decimal) / Start / End 1 D 200 200 399 2 W 64 40 7F 3 B 512 800 9FF 4 C 64 0 63
3	Not set
⋮	⋮
64	 No. / Device / Points (Decimal) / Start / End 1 D 50 12000 12049 2 W 256 100 1FF 3 W 4096 1000 1FFF 4 B 512 800 9FF

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 398 Tracking transfer setting)
2. Add a program that controls tracking triggers. (☞ Page 397 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

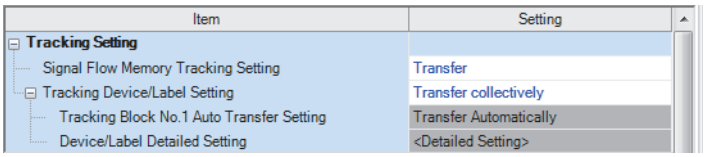
To continue the same control as before even after system switching, configure the settings in "Tracking Setting" so that all data necessary for the program is transferred.

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	Set whether to transfer the signal flow memory or not. (☞ Page 399 Tracking transfer setting for the signal flow memory)	• Do Not Transfer • Transfer	Transfer
Tracking Device/Label Setting	Set "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 400 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	Set whether to automatically transfer the tracking block No.1 or not. (☞ Page 396 Tracking block and tracking trigger)	• Do Not Transfer Automatically • Transfer Automatically	Transfer Automatically
Device/Label Detailed Setting	Set devices and labels to be transferred. (☞ Page 401 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 402 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 447 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 setting of the assigned devices. To transfer global labels with devices assigned, specify the assigned global devices in the global device setting. (☞ Page 402 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 401 Detailed setting)

Restriction

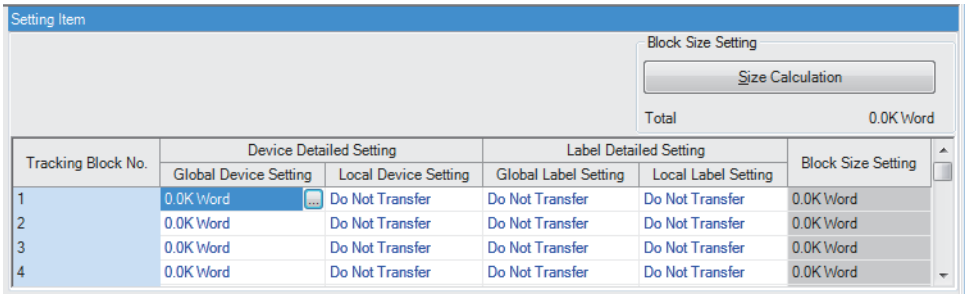
After setting "Transfer collectively", perform a test operation in the system design phase and check if the size of tracking data is 1M words or smaller. If the size of the tracking data is larger than 1M words, a stop error occurs when the CPU module is powered off and on or reset.

■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	Set global devices to be transferred. (☞ Page 402 Global device setting)	—	0.0K Word
	Local Device Setting	Set 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 393 Devices that can be specified	• Do Not Transfer • Transfer	Do Not Transfer
Label Detailed Setting	Global Label Setting*1	Set 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	Set 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	Set 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

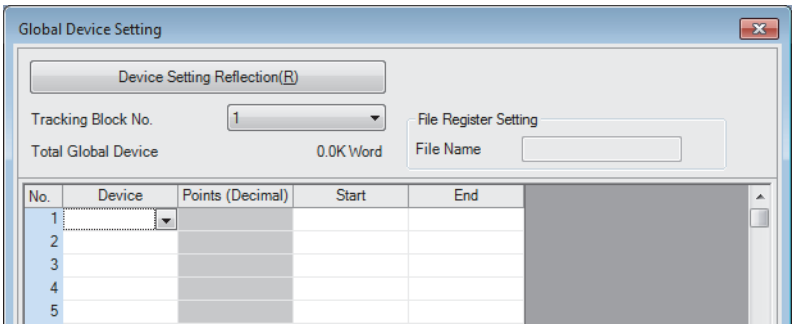
When setting devices and labels for tracking transfer, click the [Size Calculation] button to check if the tracking data capacity in one scan is equal to or less than 1M words. (☞ Page 407 Data capacity for tracking transfer)

■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.	Select 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	Select a device to be transferred. (☞ Page 393 Devices that can be specified)
Start/End	Specify 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

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.

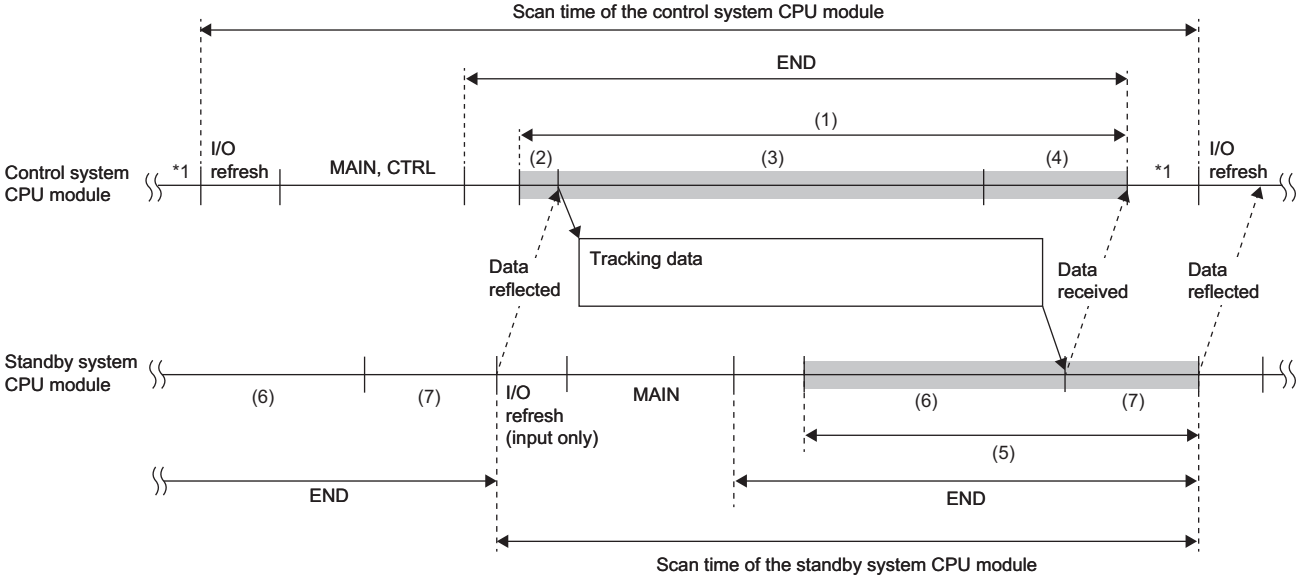
☞ Page 739 Increase in the scan time due to tracking transfer

■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 a redundant system without extension base units*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 81 In redundant mode), but the operation is the same as in a configuration with main base units 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.

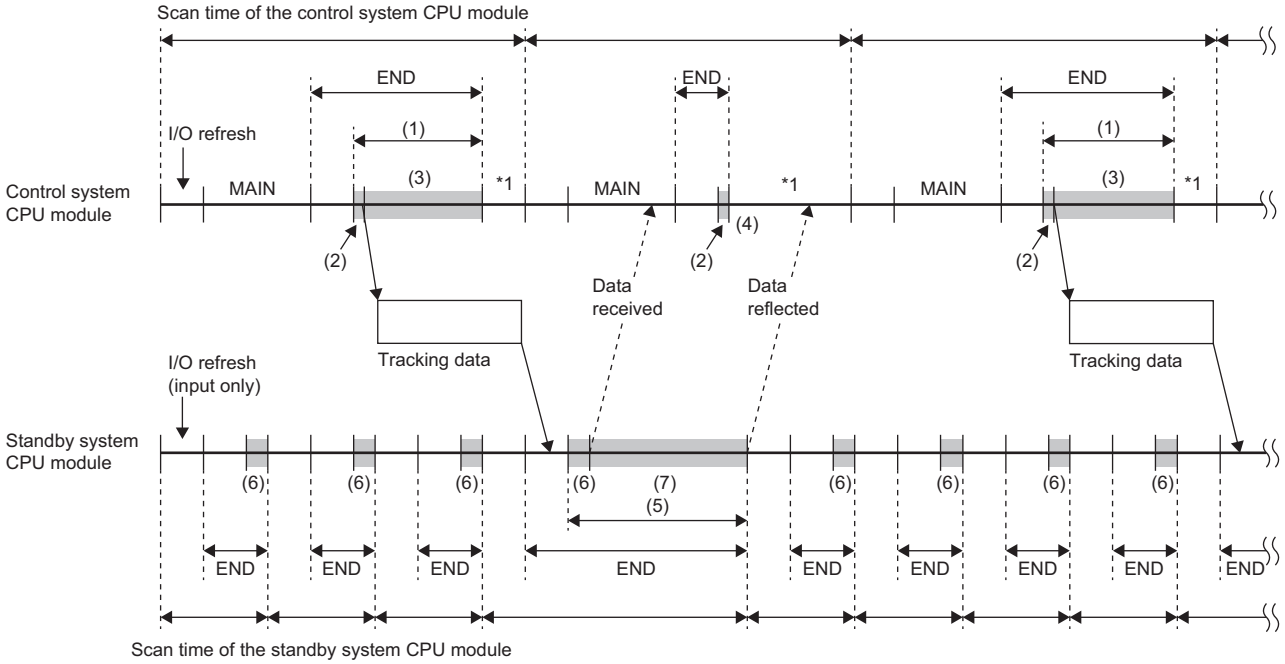
■Asynchronous tracking mode

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 CPU module of the control system is in the RUN state and the CPU module of the standby system is in the STOP state (in a redundant system without extension base units*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 81 In redundant mode), but the operation is the same as in a configuration with main base units 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 a redundant system without extension base units, 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 CPU module of the control system starts in the STOP state and is switched to the RUN state after reflecting the tracking data is completed in the CPU module of the standby system. (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 module in the control system and that of the CPU module in the standby system do not march until the operating status of the CPU module in the control system is switched from the STOP state to RUN state, the BACKUP LED of the redundant function module flashes.

When the time of the initial processing and that of the initial processing (when switched to RUN) of the CPU module in the standby system are longer than those of the CPU module in the control system, the operating status of the CPU module in the control system may not be immediately switched to the RUN state after power-on.

In a redundant system with redundant extension base unit, 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.

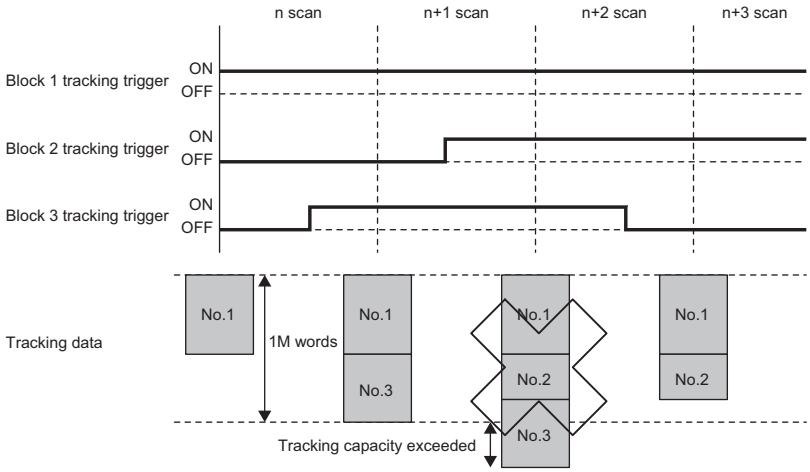
Device data used by the CPU module of the new control system

After system switching, the CPU module of the new control system starts operations by using the device/label data received in tracking transfer. The following table shows device/label data used by the CPU module of the new control system 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, 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 is 1M words or less and transferred in one scan. (☞ Page 401 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 CPU module of the control system and the CPU module of the standby system for tracking transfer. (☞ Page 409 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 208 Error Clear)

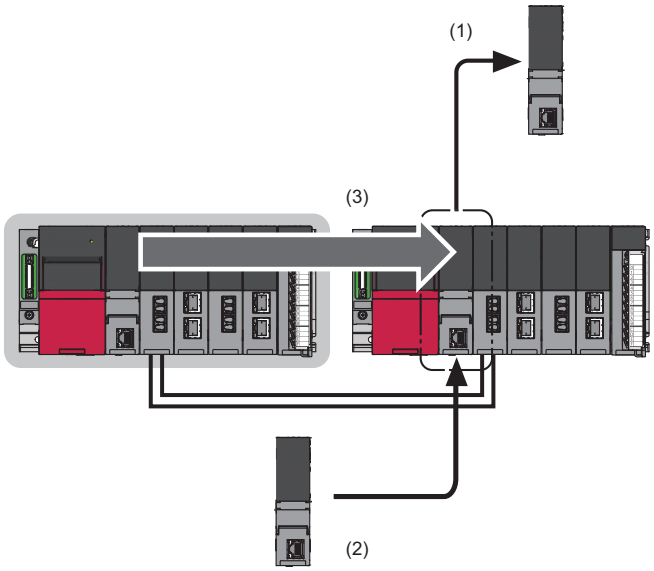
To check whether the redundant function module has restarted, check the event history. (☞ Page 647 Event List)

26.4 Memory Copy from Control System to Standby System

This function transfers data, such as programs and parameters, of the CPU module in the control system to the CPU module in the standby system to match the memory data between the two CPU modules.

Ex.

Replacing the CPU module of the standby system using this function



(1) Remove the CPU module of the standby system.
(2) Mount a new CPU module.
(3) Execute the memory copy function to match the memory data between the two CPU modules.

Data in the following memory is copied.

- Program memory
- Device/label memory (only the file register file area)
- Data memory (only the files in the system folder ($MELPRJ$))
- SD memory card (only the files in the system folder ($MELPRJ$))
- System memory (copying only the system operation setting[*1])[*2]

Data is copied only when data in the memory differs between the control system and the standby system. When data is the same between the two systems, data is not copied.

*1 Set in SM384 (System operation setting request) and SD384 (System operation setting).
*2 Check the version of the CPU module used. (☞ Page 762 Added and Enhanced Functions)

Point

For the module replacement method and the maintenance procedure using this function, refer to the following.
☞ Page 552 Module Replacement in a Redundant System

Files to be copied

The following table lists the files to be copied by the memory copy function.

○: Copied, ×: Not copied, —: Cannot be stored

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

Data is copied only when data in the memory differs between the control system and the standby system. The event history is stored only when the memory copy is completed successfully. (☞ Page 416 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 using the CPU module with the security key written, the program file and its security key are copied but the security key of the CPU module is not copied. If the replaced CPU module is started up without writing the security key after completion of the memory copy, an error occurs in the CPU module because the security keys do not match between the program file and the CPU module.

Point

If the security key of the CPU module is written to an extended SRAM cassette, the replaced CPU module can take over the security key by simply inserting the extended SRAM cassette to which the security key has been written. In this case, writing the security key to the replaced CPU module after completion of the memory copy is not required.

Execution methods

There are three methods for executing the memory copy.

Item	Description	Application
Auto memory copy	The system automatically executes the memory copy. CPU parameter settings are required in advance.	For executing the memory copy without using an engineering tool or an external device (such as a GOT)
Memory copy using the engineering tool	The memory copy is executed by an online operation of the engineering tool connected to the CPU module of the control system.	For executing the memory copy using the engineering tool
Memory copy using the special relay and special register	The memory copy is executed by operations of the special relay and special register.	For executing the memory copy using an external device (such as a GOT)

Execution conditions

The memory copy can be executed regardless of the operating status (RUN, STOP, PAUSE, stop error) of the CPU module in both control and standby systems.

The auto memory copy can be executed only when the redundant system is in backup mode.

Auto memory copy

The execution procedure for the auto memory copy is described below.

The auto memory copy is executed when the standby system is powered off and on or reset while the control system is operating.

Point

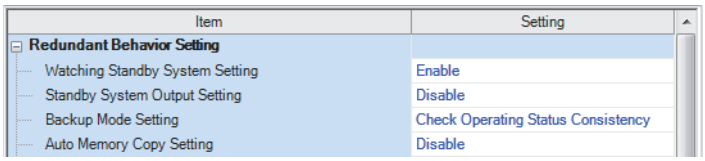
- The auto memory copy needs to be set in the CPU parameter in advance. For this reason, consider whether to use the auto memory copy or not when determining the maintenance policy in the system design phase.
- The auto memory copy is executed when the parameter has been set for the CPU module of the control system. Even when the parameter has not been set for the CPU module of the standby system, the auto memory copy is executed when the parameter has been set for the CPU module of the control system.

Parameter

To execute the auto 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	Set whether to enable the auto memory copy function or not.	• Disable • Enable	Disable

Restriction

- When the control system and the standby system are simultaneously powered on, the auto memory copy is not executed.
- The CPU module of the standby system does not perform boot operation using an SD memory card, but operates based on the files copied by the auto memory copy. However, when the auto memory copy cannot be executed due to a communication error or other causes at power-on or reset of the standby system, the CPU module of the standby system operates with the files transferred by the boot operation.

Execution procedure

1. When the redundant system is in separate mode, switch the operation mode to the backup mode. (☞ Page 374 Operation Mode Change)
2. Power off and on or reset the standby system. The system executes the memory copy. During the memory copy, the MEMORY COPY LED of the redundant function module flashes (in 200ms intervals) in both systems.

Control system | Standby system

3. The standby system is automatically reset and restarted, and the memory copy is completed. The MEMORY COPY LED turns off in both systems.

Control system | Standby system

Memory copy using the engineering tool

The execution procedure for the memory copy using the engineering tool is described below.

Execution procedure

1. Connect the engineering tool to the CPU module of the control system.
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 the [Execute] button. During the memory copy, the MEMORY COPY LED of the redundant function module flashes (in 200ms intervals) in both systems.

Control system | Standby system

4. When the memory copy is completed, the MEMORY COPY LED turns off in the control system and turns on in the standby system.

Control system | Standby system

5. Power off and on or reset the standby system. The MEMORY COPY LED turns off in the standby system.

Control system | Standby system

Memory copy using the special relay and special register

The execution procedure for the memory copy using the special relay and special register is described below.

Special relay to be used

The memory copy function uses the following special relay areas.

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 659 List of Special Relay Areas

Special register to be used

The memory copy function uses the following special register areas.

SD number	Name
SD988*1	Memory copy completion status (latch)
SD1653	Memory copy destination I/O number
SD1654	Memory copy completion status

*1 This is a latch area. Even when the system is powered off and on or reset, the result of the memory copy executed immediately before power-on or reset can be checked.

For details on the special register (SD), refer to the following.

☞ Page 682 List of Special Register Areas

Execution procedure

To execute the memory copy using the special relay and special register, use the special relay and special register of the CPU module in the control system.

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 current memory copy is completed.
   - When SM1655 is on, the memory copy using the special relay and special register cannot be started. Turn off SM1655, and execute the memory copy.
2. Store 03D1H (CPU module of the standby system) 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 LED of the redundant function module flashes (in 200ms intervals) in both systems.

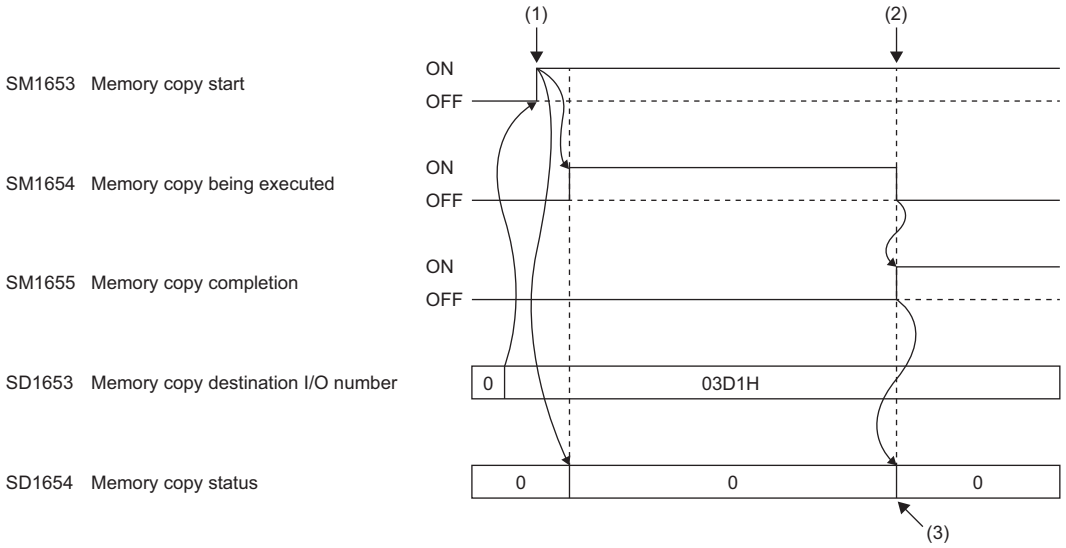
   Control system: MEMORY COPY (flashing)　　Standby system: MEMORY COPY (flashing)
4. When the memory copy is completed, the MEMORY COPY LED turns off in the control system and turns on in the standby system.

   Control system: MEMORY COPY (off)　　Standby system: MEMORY COPY (on)
5. Power off and on or reset the standby system. The MEMORY COPY LED turns off in the standby system.

   Control system: MEMORY COPY (off)　　Standby system: MEMORY COPY (off)

Operation of the special relay and special register

The following chart shows the operation of the special relay and special register.



(1) The memory copy starts.
(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 are the precautions for the memory copy function.

Restrictions

■Restrictions on memory copy execution

Do not execute the memory copy in the following cases.

- The CPU module of the standby system is powered off or being reset.
- The initial processing (when switched to RUN) is being performed.
- The tracking cable has an error or is not connected.
- The redundant function module has an error.
- The two systems cannot perform communications via the redundant function module.
- The model of the CPU module in the control system differs from that of the CPU module in the standby system.
- An online operation is being performed by the engineering tool.
- The systems are being switched.
- A module (a redundant function module or any module of the standby system) is being changed online.
- The CPU module is being backed up. (CPU module data backup/restoration function)
- The redundant system is in separate mode. (Restriction for the auto memory copy)
- The CPU module is set as the standby system.
- The memory copy is being executed.
- A value other than 03D1H is stored in SD1653. (Restriction for the memory copy using the special relay and special register)
- The CPU module of the control system to which the program restoration information is not written has executed the memory copy to the CPU module of the standby system that does not support the program restoration information write selection.
- A label in the CPU module of the standby system is being accessed from an external device.

■Restrictions during memory copy execution

Do not execute the following operations during memory copy execution.

- Online operation from the engineering tool
- RUN/STOP/RESET switch operation
- Removing the SD memory card
- Online module change
- Backing up the CPU module (CPU module data backup/restoration function)
- Powering off or resetting the systems
- Disconnecting tracking cables

Whether to delete files before memory copy

When memory copy is performed, the CPU module of the standby system may detect a stop error because the files are deleted from the memory of the CPU module of the standby system to which the data is copied. 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 execution

When the memory copy is completed with an error, the MEMORY COPY LED of the redundant function module turns off in the control system and the MEMORY COPY LED flashes (in 1s interval) in the standby system. In this case, the memory copy has not been normally executed to the CPU module of the standby system.

Check the error code stored in SD1654 (Memory copy completion status), eliminate the error cause, and execute the memory copy again. (☞ Page 682 List of Special Register Areas)

When labels accessible from external devices are used

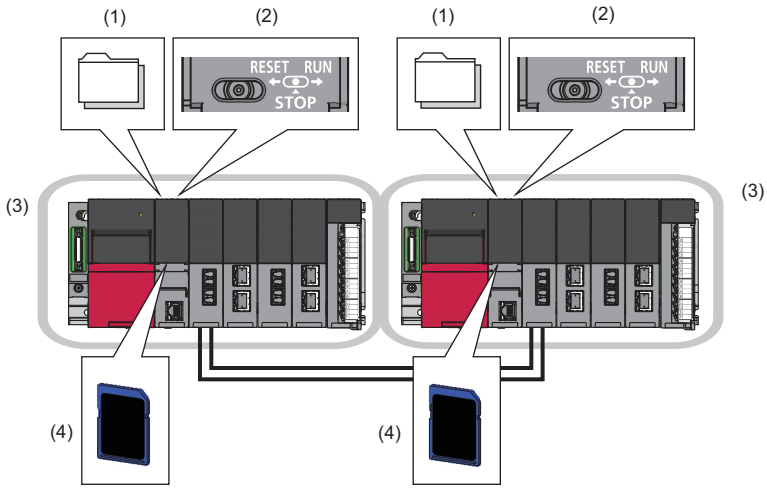
When executing the memory copy, pay attention to the firmware version of the CPU modules used.

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.

26.5 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 function checks the following items.

No.	Item	Description	Reference
(1)	Files	Parameter file, program file, and other files	Page 419 Files
(2)	Operating status	Operating status of the CPU module (RUN/STOP/PAUSE)	Page 420 Operating status
(3)	Modules on the main base unit	CPU module and other modules mounted on the main base unit	Page 420 Modules on the main base unit
(4)	SD memory card	Insertion status of an SD memory card and status of the write protect switch	Page 421 SD memory card

Execution timing

The following table lists the execution timing of the system consistency check.

Item	Execution timing
Files	• When both systems are simultaneously powered on or reset • When the other system is powered on or reset while one system is waiting for the other system to start • When the standby system is powered on or reset while the control system is operating • When the operation mode is changed to the backup mode • When the operating status of the CPU module is switched from STOP to RUN • When the END processing is performed • When the online program change is completed • When the systems are switched • When tracking cables are reconnected
Operating status*1	• When the standby system is powered on or reset while the control system is operating • When the operation mode is changed to the backup mode • When the operating status of the CPU module is switched from STOP to RUN • When the END processing is performed • When tracking cables are reconnected
Modules on the main base unit	• When both systems are simultaneously powered on or reset • When the other system is powered on or reset while one system is waiting for the other system to start • When the standby system is powered on or reset while the control system is operating • When the operation mode is changed to the backup mode • When tracking cables are reconnected
SD memory card	• When both systems are simultaneously powered on or reset • When the other system is powered on or reset while one system is waiting for the other system to start • When the standby system is powered on or reset while the control system is operating

*1 Checking the operating status can be disabled in "Backup Mode Setting" of "Redundant Behavior Setting". (☞ Page 429 Redundant System Operation Setting)

The function is not executed in the following cases.

- 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 redundant system is in separate mode.
- The memory copy is being executed.

Files

Whether files are the same between the control system and the standby system 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 the program restoration information write status are also checked. (☞ Page 156 Configuration of a program file, Page 756 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 exists or not is checked. The file contents are not checked.

*3 Files are not checked at the point when an SD memory card is 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.)

When a mismatch is detected

A stop error occurs in the CPU module of the standby system.

Perform either of the following operations to match the files in the CPU modules of both systems.

- Write files to the CPU modules of both systems.
- Execute the memory copy function to match the files stored in both systems. (☞ Page 409 Memory Copy from Control System to Standby System)

Operating status

Whether the operating status (RUN/STOP/PAUSE) of the CPU module is the same between the control system and the standby system is checked.

When a mismatch is detected

A continuation error occurs in the CPU module of the standby system. In addition, the BACKUP LED of the redundant function module flashes in both systems because a system switching disable cause exists.

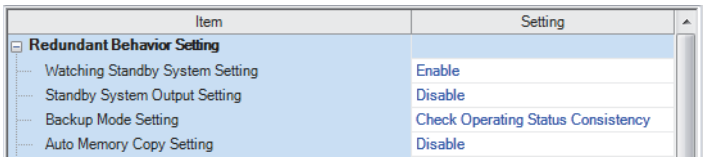
Backup mode setting

To not to generate a continuation error when the operating status is switched during operation, checking the operating status can be disabled.

When disabled, a continuation error will not occur in the CPU module of the standby system even when the operating status of the CPU module differs in the two systems.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Redundant Behavior Setting]

Window



Displayed items

Item	Description	Setting range	Default
Backup Mode Setting	Set 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

Modules on the main base unit

Whether the following items are the same between the control system and the standby system is checked.

- Model of the CPU module
- Model and type 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 762 Added and Enhanced Functions

When a mismatch is detected

A stop error occurs in the CPU module of the standby system. If a stop error is detected when both systems are simultaneously powered on or reset, a stop error occurs in the CPU module of the control system as well. *1

("Simultaneously" here means when one system is started up within three seconds of the other.)

*1 If "Control System Start-up Setting at One System Error" is set to "Enable", this stop error is treated as a continuation error.

Checking the system configuration in a redundant system with redundant extension base unit

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 unit, and the model name and start I/O number of the mounted module 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 setting for the system parameters and module parameters are the same in both systems.

Point

- If a stop error is detected when both systems are simultaneously powered 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 units are not connected to the same redundant extension base unit, a stop error occurs at the following timing.

- At power-on or reset
- At tracking cable connection

Point

- If a stop error is detected when both systems are simultaneously powered 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

Whether the insertion status of an SD memory card and the status of the write protect switch are the same between the control system and the standby system is checked. The type and capacity of an SD memory card is not checked.

Point

The insertion status of an SD memory card is checked even when only the standby system is powered off and on or reset while the control system is operating. When an SD memory card is used, it is recommended to remove it only when necessary.

When a mismatch is detected

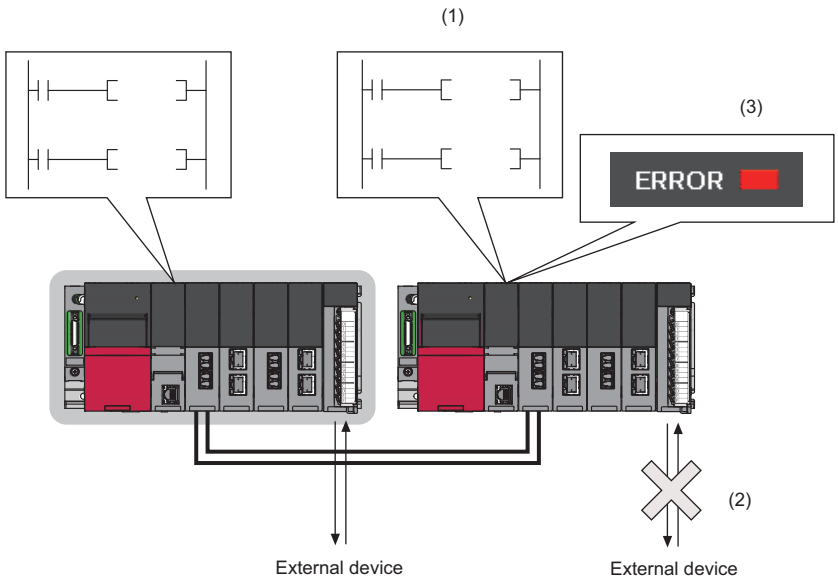
A stop error occurs in the CPU module of the standby system. If a stop error is detected when both systems are simultaneously powered on or reset, a stop error occurs in the CPU module of the control system as well. *1

*1 If "Control System Start-up Setting at One System Error" is set to "Enable", this stop error is treated as a continuation error.

26.6 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.
When "Both Systems Executions" is set to a program in "Both Systems Program Executions Setting", the program is executed by the CPU modules of both systems.

Ex.
When an error is detected in the external device connected to the standby system, the error is notified by generating a continuation error.



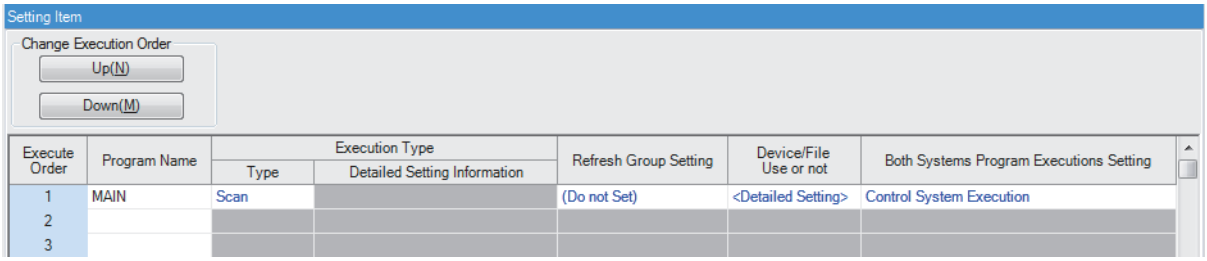
(1) A diagnostic program that is set as a program executed in both systems is executed.
(2) An error is detected in the external device connected to the standby system by the diagnostic program.
(3) The PALERT instruction is executed, and the detected error is displayed in the detailed information of the continuation error.

Both systems program executions setting

Set the parameter for each program to be executed in both systems.

[CPU Parameter] ⇨ [Program Setting]

Window



Displayed items

Item	Description	Setting range	Default
Both Systems Program Executions Setting	Set 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, and standby type. • "Control System Execution" can be set for programs whose execution type is the fixed scan execution type and event execution type.	• Control System Execution • Both Systems Executions	Control System Execution

Point

To enable the output (Y) to external devices connected to the standby system in a program executed in both systems, set "Enable" in "Standby System Output Setting" of the CPU parameter. (☞ Page 429 Redundant System Operation Setting)

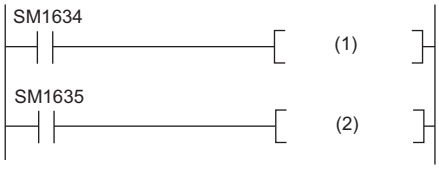
Operation of a program executed in both systems

The following table summarizes the operation of a program executed in both systems.

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 when the operating status of the CPU module is changed to RUN. When an initial execution type program is not set as a program executed in both systems, a scan execution type program that is set as a program executed in both systems is executed in the first scan when the operating status of the CPU module is changed to RUN.	

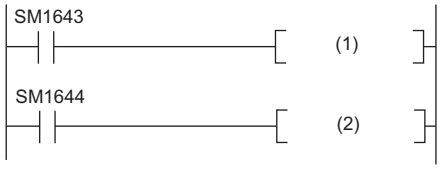
Point

- To perform different diagnostic processing in the control system and the 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) turns on only in the standby system. The following is a program example that performs different processing by using the special relay. For the processing (1) and (2), use the SET instruction or another instruction to prevent duplication of a coil in the processing.



(1) Diagnostic processing in the control system, (2) Diagnostic processing in the standby system

- In a program executed in both systems, device/label values may differ between the control system and the standby system. If the systems are switched in this state, the program is executed using the different data and an unintended operation may be performed. To prevent this, 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 the processing (1) and (2), use the SET instruction or another instruction 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 of a program executed in both systems differs at system switching. The following table summarizes the operation at system switching.

Item		CPU module of the new control system	CPU module of the new standby system
Program execution	Initial execution type program	When an initial execution type program has not been completed in the old control system at system switching, it is executed again from the start.	When an initial execution type program has not been completed in the old standby system at system switching, it is not executed in the new standby system because the systems are switched after the program is completed in the old standby system.
	Scan execution type program	A scan execution type program is executed from the step 0.	
Direct access input (DX)		Data is input when an instruction using the direct access input (DX) is executed in the program after system switching. However, no data is input for the direct input for the module mounted on the extension base unit.	
Direct access output (DY)		Data is output when an instruction using the direct access output (DY) is executed in the program after system switching. 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		These instructions are executed when the instruction execution condition is satisfied in the program after system switching. 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		• The instruction is executed when the instruction execution condition is satisfied in the program after system switching. • When the systems are switched while the instruction is being executed, the instruction execution continues and the completion device turns on upon completion of the instruction execution.	

For the operation at system switching as for programs which are not set as those executed in both systems, refer to the following.

☞ Page 383 Operation at system switching

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. In a redundant system with redundant extension base unit, set a program execution time of the standby system to be within 200ms.	Page 427 Program execution time
Constant scan		The constant scan is invalid in the standby system when the redundant system is in backup mode.	—
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 in the CPU module of the new control system. 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.	—
	Event execution type	In a redundant system with redundant extension base unit, an event execution type program in which the trigger type is set to "ON of bit data (TRUE)" and the module access device (Un\G) of the module on an extension base unit is specified cannot be executed by the standby system when the redundant system is in separate mode. Therefore, be careful when performing system maintenance in separate mode.	—
SFC program		The SFC program cannot be set to be executed in both systems.	—
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 427 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 in backup mode.	—
Buffer memory address		In a redundant system with redundant extension base unit, do not access the buffer memory of the module on the extension base unit from a program executed in the standby system in 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).	—
Restricted instructions		Some instructions have restrictions when they are used in a program executed in both systems.	Page 428 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) are 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 CPU module of the control system 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 a redundant system with redundant extension base unit, 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.

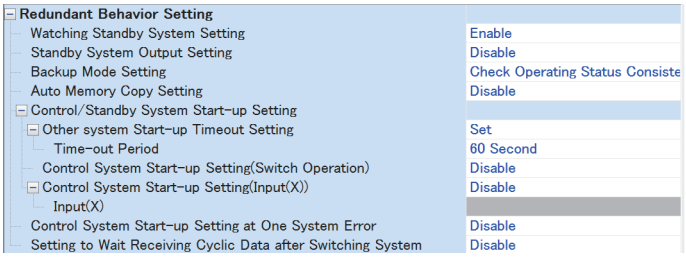
Classification	Notation	Description
Calling a subroutine program	CALL(P)	These instructions may not normally operate if a subroutine with any of a rising instruction, falling instruction, or SCJ instruction in a program which is not set as a both systems execution program is called in the standby system 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.

26.7 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		Set 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 CPU module of the standby system is off, has been reset, or a stop error has occurred. This item is enabled only for the CPU module of the control system.	• 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 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 when the CPU module of the standby system is in the STOP state, the output (Y) received in tracking transfer is output.)	• Disable • Enable	Disable
Backup Mode Setting		☞ Page 420 Operating status		
Auto Memory Copy Setting		☞ Page 411 Auto memory copy		
Control/ Standby System Start-up Setting*2	Other system Start-up Timeout Setting	Set 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	—
Control System Start-up Setting at One System Error*3		Set whether to start the normal system as a control system when the other system detects an error, such as a CPU module failure, module failure, and module disconnection, at the power-on or reset and the redundant system cannot operate. For the control system start-up at one system error, refer to the following. ☞ Page 438 Control System Start-up at One System Error	• Disable • Enable	Disable
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 432 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 762 Added and Enhanced Functions)

*2 This setting is invalid in a redundant system with redundant extension base unit. (The system that has started up first starts up as a control system regardless of this setting.)

*3 When using this setting, check the versions of the CPU module and engineering tool used. (☞ Page 762 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 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 113 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 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 in backup mode, set "Both Systems Program Executions Setting". (☞ Page 422 Program Execution in Both Systems)

Operation at system switching

The operation performed at 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	CPU module of the new control system	CPU module of the new standby system
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 422 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.

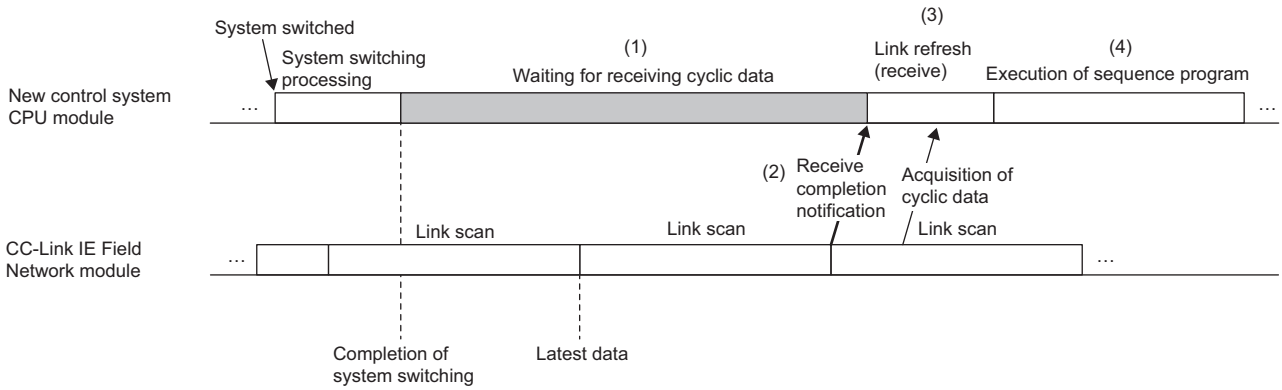
Point

In a redundant line structure, the cyclic transmission delay time of one system may differ by up to one link scan from that of the other system. Enable this setting to prevent the execution of a program with old cyclic data in the new control system following a 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 762 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 745 Extended time until initial output after system switching (Tjo))
- When this setting is disabled, the sequence program is executed without waiting for the latest cyclic data reception after system switching. Therefore, in a redundant line structure, if the new control system has not received the latest cyclic data, the sequence program may be executed with the cyclic data of old control system that is one link scan before.
- 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 746 Waiting time for cyclic data receive after system switching (Twcyc))

26.8 Redundant Function Module Communication Test

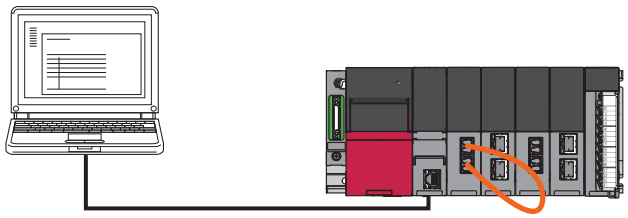
The hardware of the redundant function module is checked for an error when its communication is unstable.

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 CPU module of the standby system.
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 a 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 62 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.
- Ensure that the module communication test is performed on the CPU module of the standby system. Performing the test on the CPU module of the control system may cause an unintended operation.

26.9 Settings for Redundant System with Redundant Extension Base Unit

Set the operation of the redundant system with redundant extension base unit in the redundant system settings of the CPU parameter.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Redundant system with extension base unit]

Restriction

Before using this function, check the versions of the CPU module and engineering tool used. (☞ Page 762 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 436 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 436 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 a redundant system without extension base units.

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 436 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 437 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 437 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 redundant extension base units are connected in the second and later extension levels), 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 an automatic recovery cause is detected while an external file operation, such as writing data to the programmable controller, is being performed on the CPU module of the standby system, the CPU module may be reset depending on the timing, and may not start normally. Therefore, before performing an external file operation on the CPU module of the standby system, check that the READY LED of the CPU module is on. If the CPU module is reset by this function while the external file operation is being performed, perform the external file operation again after the CPU module is reset.

26.10 Control System Start-up at One System Error

When one system detects an error, such as a CPU module failure, module failure, and module disconnection, at the power-on or reset and the redundant system cannot operate, the normal system starts the control system.

Restriction

When performing the control system start-up at one system error, check the versions of the CPU module and engineering tool used. (☞ Page 762 Added and Enhanced Functions)

Setting method

Enable "Control System Start-up Setting at One System Error" in "Redundant System Operation Setting". (☞ Page 429 Redundant System Operation Setting)

When enabling this function, set all modules to be connected in "I/O Assignment Setting", and write the system parameters and module parameters.

Starting a control system

When this function is enabled, the control system starts up if one of the following errors occurs.

- System configuration mismatch between both systems (☞ Page 438 System configuration mismatch between both systems)
- Error detection in the other system (☞ Page 439 Error detection in the other system)

System configuration mismatch between both systems

When the system configurations of both systems become mismatched due to failure or disconnection of a module on the main base unit (power supply module, CPU module, or redundant function module) or an extension cable, the normal system starts as the control system without detecting a stop error.

In addition, when the system configurations become mismatched due to failure or removal of an SD memory card inserted in the CPU module, the system A starts as the control system without detecting a stop error.

■Operation when the mounting states of the modules on the main base unit are mismatched

Even if the system configurations mismatch due to failure or disconnection of a module on the main base unit (power supply module, CPU module, or redundant function module) causes a system consistency check error (error code: 3700H), the normal system starts as the control system without a stop error detected in the CPU module. (The control system detects a continuation error.)

■Operation when the states of extension cables connected to the main base unit are mismatched

When the system configurations mismatch occurs due to failure or disconnection of extension cables connected to the main base unit, the system with the failure or disconnection detects a redundant system error (error code: 3748H) and stops the CPU module operation. The system with normal connection of extension cable starts as the control system without a stop error detected in the CPU module. (The control system detects a continuation error.)

■Operation when the states of SD memory card insertion are mismatched

Even if the system configurations mismatch due to failure or removal of an SD memory card inserted in the CPU module causes a system consistency check error (error code: 3710H), the system A starts as the control system without a stop error detected in the CPU module. (The control system detects a continuation error.)

Error detection in the other system

When a stop error occurs in the other system due to failure of the CPU module or failure or disconnection of a module and the normal system detects that error, the normal system starts as the control system without a stop error detected in the CPU module. Or the normal system starts as the control system with an automatic reset of both systems.

■When the normal system operates as a control system by the control system start-up

At simultaneous power-on/reset of both system or during a start-up of the control system that is waiting for a start-up of the other system, if an error detection in the other system causes a stop error, the normal system starts as the control system without a redundant system error (3747H) detected.

In that case, the normal system that has started as the control system detects an error in the CPU module of the standby system.

When this function is enabled, a redundant system error (3747H) is not detected at the power-on/reset or during a start-up of the control system that is waiting for a start-up of the other system.

■When the normal system operates as a control system by reset

In the following cases, the system automatically resets both systems and then starts the normal system as the control system.

- The system switching cannot be performed even if an error (such as failure or disconnection of a module) is detected in a system that is intended to start as the control system
- The normal system, which is powered on while only the error system (with failure or disconnection of a module) is operating, cannot be started as the control system

However, this function does not perform the automatic reset in separate mode.

Note that the system starts the automatic reset after the READY LED flashes for three seconds. (☞ Page 28 CPU MODULE)

When this function is enabled, the automatic reset is performed if one of the following errors is detected.

- System consistency check error (file): 3714H
- Redundant system error: 374CH, 374DH
- Control system CPU module error: 3760H
- Redundant system configuration error: 3761H

The automatic reset may not be performed depending on an error. For details, refer to the following.

☞ Page 442 Operation when files in both systems are mismatched

The availability of automatic reset depends on the system configurations.

System configuration	Start-up method	Normal system	Automatic reset
Main base unit	Simultaneously power on both systems.	System A	Not available*2
		System B	Not available*2*3
	Power on the system with an error (failure or disconnection of a module), and then the normal system.	System A	Not available*2
		System B	Available
	Power on the normal system, and then the system with an error (failure or disconnection of a module).*1	System A	Not available*2
		System B	Not available*2*3
	Power on the normal system while the system with an error (failure or disconnection of a module) is operating as the control system.	System A	Available
		System B	Available
	Power on the system with an error (failure or disconnection of a module) while the normal system is operating as the control system.	System A	Not available*2
		System B	Not available*2
Redundant extension base unit	Simultaneously power on both systems.	System A	Not available*2
		System B	Available
	Power on the system with an error (failure or disconnection of a module), and then the normal system.	System A	Available
		System B	Available
	Power on the normal system, and then the system with an error (failure or disconnection of a module).	System A	Not available*2
		System B	Not available*2
	Power on the normal system while the system with an error (failure or disconnection of a module) is operating as the control system.	System A	Available
		System B	Available
	Power on the system with an error (failure or disconnection of a module) while the normal system is operating as the control system.	System A	Not available*2
		System B	Not available*2

*1 When the system to be powered on first is a normal system and the other system start-up timeout time has elapsed, the automatic reset is not performed and a stop error occurs in both systems. To prevent this, it is necessary to perform a specific operation to start up the control system within the timeout time or set "Other system Start-up Timeout Setting" to "Not Set".

*2 The automatic reset is not required because the normal system starts as the control system.

*3 When a stop error, such as a file mismatch, is detected in the system intended to start as the standby system and the systems cannot be switched due to a both system stop error, the automatic reset is performed.

However, when one of the following errors is detected, an automatic reset of both systems is not performed. This is because a stop error will occur in both systems until the error is cleared, even if an automatic reset is attempted.

In this case, restore the systems by manual operation (powering off and on or resetting the systems by users) after the error is cleared.

- Module configuration error: 2006H
- Base unit configuration error: 2060H
- System consistency check error (CPU module): 3701H
- Extension cable error: 3750H
- Redundant system configuration error: 3785H, 3786H, 3787H, 3789H, 378AH

■Cause of failure of automatic reset

When the automatic reset of both systems is performed using this function, the automatic reset fails if any of the following causes exist. When the automatic reset fails, restore the systems by manual operation (powering off and on or resetting the systems by users).

- Tracking communication error (cable disconnection, control system power-off, tracking communication data error, or another cause)
- A file on the CPU module of the own system being accessed
- A file on the CPU module of the other system being accessed
- A CC-Link module is mounted on the main base unit of the own system when a redundant system is configured with redundant extension base unit.
- A CC-Link module is mounted on the main base unit of the other system when a redundant system is configured with redundant extension base unit.
- During memory copy from control system to standby system
- During execution of the backup function (by turning on SM1351) on the CPU module of the own system
- During execution of the backup function (by turning on SM1351) on the CPU module of the other system
- During remote reset on the CPU module of the own system
- During remote reset on the CPU module of the other system
- During online module change in the own system
- During online module change in the other system

■Check method of automatic reset results in the event history

Whether the automatic reset was performed or not can be checked by referring to the event history. (☞ Page 647 Event List)

Precautions

The following describes precautions when this function is enabled.

When the redundant system is configured with redundant extension base unit

■Stop error in both systems

A stop error occurs in both systems if one of the following errors is detected when the redundant system is configured with redundant extension base unit: failure of an extension base unit, failure or disconnection of a module on an extension base unit, failure or disconnection of an extension cable[*1] connecting extension base units (cable between extension base units).

*1 Refers to both of redundant extension cables or a non-redundant extension cable.

■When a CC-Link module is mounted on the main base unit

The automatic reset cannot be performed. Check the system configuration and mount the CC-Link module on the redundant extension base unit.

When tracking communications with the other system cannot be performed at startup

If tracking communications cannot be performed at system start-up due to a failure or disconnection of a module on the main base unit (power supply module, CPU module, or redundant function module) or a tracking cable, a waiting state for the other system to start up or a stop error will occur. This prevents the normal system from starting as the control system.

A tracking communication failure can be confirmed with whether a continuation error (Tracking communication disabled, error code: 1B70H) occurs or whether bit 1 of SD1642 (BACKUP/SEPARATE LED flashing cause) is on.

Automatic reset during file operation

When this function is enabled, an execution of external file operation (such as writing data to the programmable controller) before the start-up of both systems may cause automatic reset, preventing the normal system from starting as the control system. (Whether the automatic reset is performed or not depends on the timing of the external file operation.) Therefore, execute the external file operation after confirming the READY LEDs in both systems are on.

If the automatic reset is performed during the external file operation, perform the file operation again after the automatic reset.

Combination of automatic reset and other functions

The following operations cannot be performed while the automatic reset is being performed using this function. Or the automatic reset using this function does not operate while any of the following operations are being performed. Therefore, start the following operations after confirming that the READY LEDs in both systems are on.

- Memory copy from the control system to the standby system (using the engineering tool, or the special relay and special register)
- Backup function (by turning on SM1351) on the CPU module
- Remote reset
- Online module change function

Timeout time set in "Other system Start-up Timeout Setting"

If a tracking communication failure at the system start-up causes other system start-up timeout and the tracking communications disabled (error code: 3746H) is detected, the normal system may not start as the control system.

To prevent the other system start-up timeout, set "Other system Start-up Timeout Setting" of "CPU Parameter" to "Not Set".

When "Other system Start-up Timeout Setting" is set to "Set", perform a specific operation to start up the control system within the timeout time. For specific operation, refer to the following.

☞ Page 161 System Determination

Operation when files in both systems are mismatched

If the following data is deleted from only one system and a file mismatch occurs between both systems at the system start-up, the normal system may not start as the control system.

- Program file
- FB file

A stop error caused by deleting files set in "Program Setting" and "FB/FUN File Setting" is detected at the time of a CPU module status transition to RUN, not during initial processing. Therefore, this stop error is not subject to the reset that enables the normal system to start as the control system.

26.11 SLMP Communications

The SLMP frame send instruction (SP.SLMPSND) performs communications using different IP addresses in the system A and the system B.
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

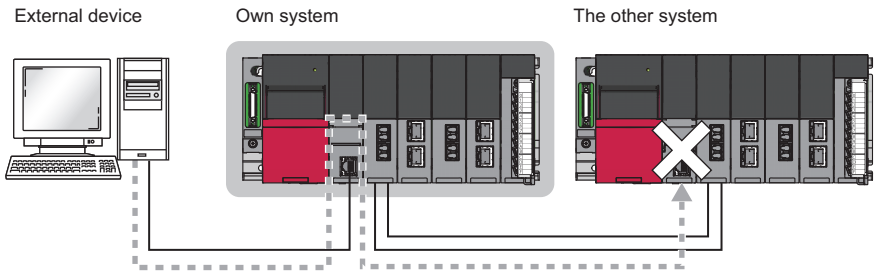
When "Control System" or "Standby System" has been selected in the transfer setup and the systems are switched, a target system mismatch is detected by a command issued during the system switching, and a communication error occurs. If a communication error occurs while a data write command is being issued, the data write command needs to be issued for the new control system.

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.



26.12 Precautions on Programming

This section describes the precautions on programming for a redundant system.

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 when the redundant system is 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 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
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 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 station number 10 of network number 1.

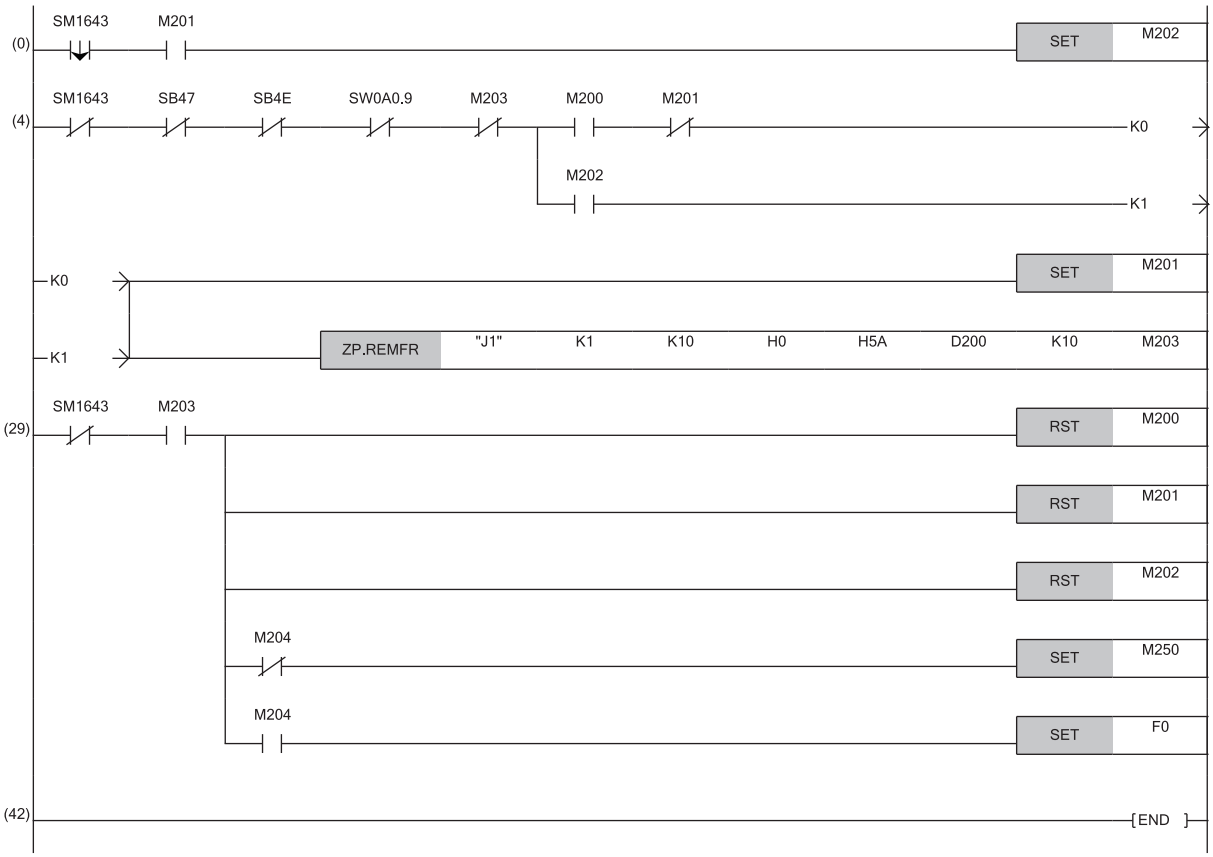
• Devices used

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.

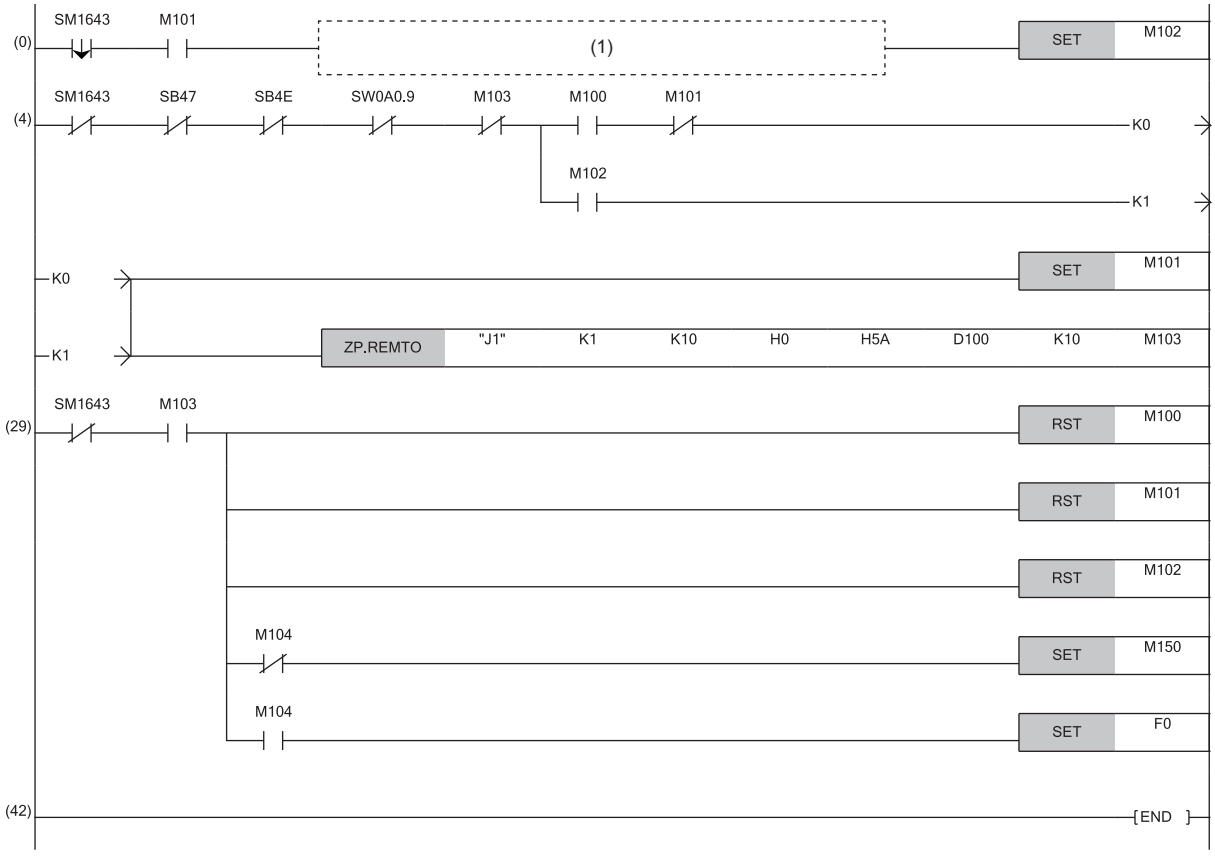
• Devices used

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

The following describes the instructions whose operations after system switching vary depending on whether the signal flow memory is tracked or not when the redundant system is 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 (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.

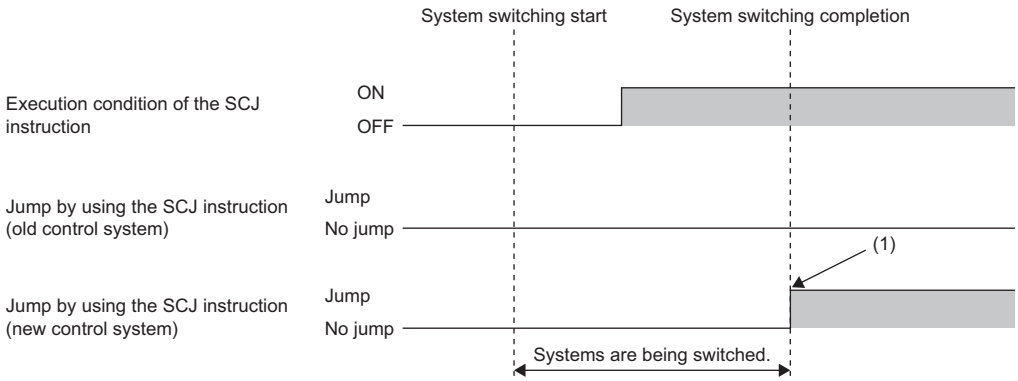
■Falling instruction

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

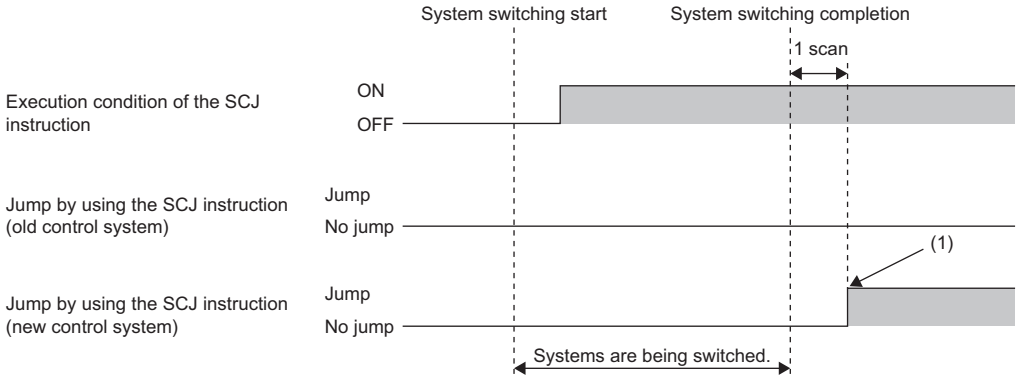
Once the system switching is performed, the execution condition of the SCJ instruction turns on after the signal flow memory of the new control system turns on. In the new control system, processing jumps to the pointer specified by the SCJ instruction in the first scan.



(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

Instructions causing different operation results between both systems

This section describes the instruction that causes different operation results between the control system and standby system after system switching.

■PID control instructions

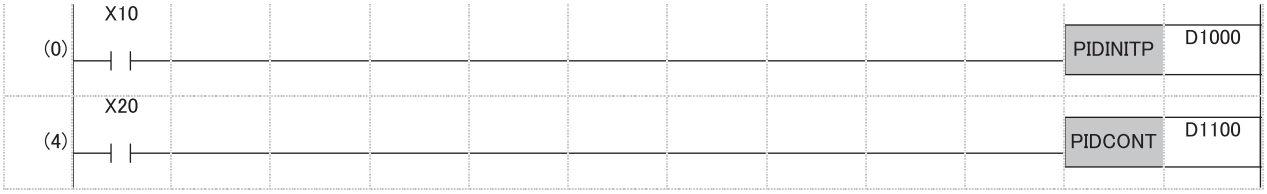
When using the following PID control instructions, include the number of device points used by the systems in tracking target data. Otherwise, the instructions cause different operation results between the control system and standby system after system switching.

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 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.

Interrupt from modules

This section describes the precautions for interrupts from modules.

System switching when redundant system is 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 when redundant system is in separate mode

■When the old control system is switched to the new standby system

The old control system executes the interrupt program of an interrupt factor 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 operation mode is changed to separate mode

■For the control system

The control system retains the interrupt factors before operation mode change to the separate mode. After operation mode change to the separate mode, the control system executes the interrupt programs of the interrupt factors that have occurred when the redundant system was in backup mode.

■For the standby system

The interrupt factors that the standby system has retained before operation mode change to the separate mode will be discarded. Therefore, after operation mode change to the separate mode, the interrupt programs of the interrupt factors that have occurred before operation mode change will not be executed.

When operation mode is changed to backup mode

■For the control system

When the interrupt programs of the interrupt factors before operation mode change to the backup mode have not been executed yet, the control system will execute the programs retained when the redundant system is in separate mode.

■For the standby system

When the interrupt programs of the interrupt factors before operation mode change to the backup mode have not been executed yet, the interrupt factors retained when the redundant system is in separate mode will remain retained. Interrupt programs will not be executed.

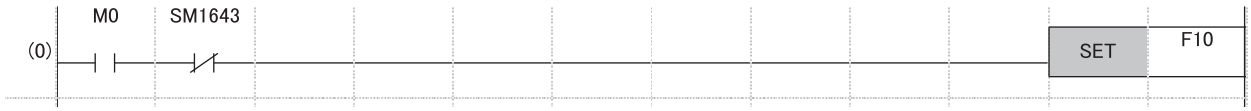
Precautions for using the annunciator (F)

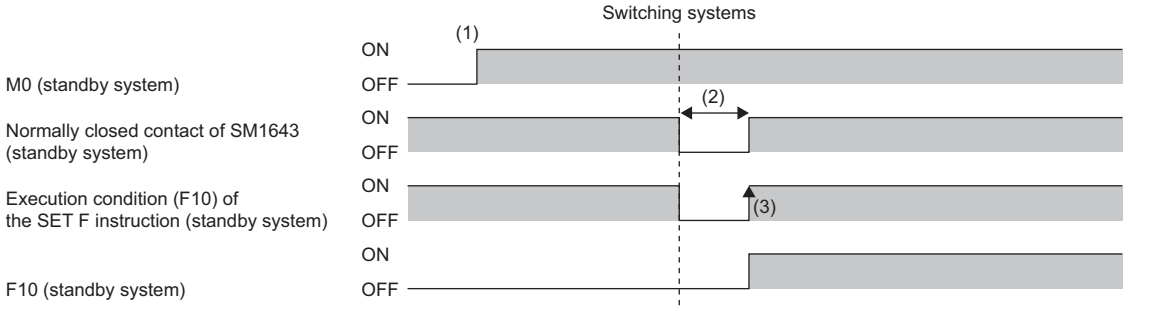
This section describes the precautions for using the annunciator (F).

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.

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

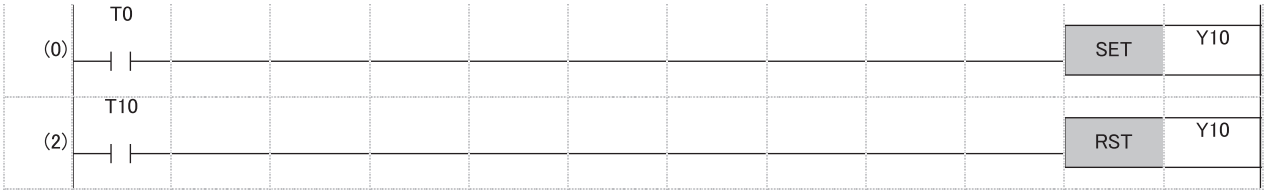
Depending on the timing to perform system switching such as power-off, tracking transfer processing is suspended and tracking data may not be reflected to the CPU module in the new control system. For the timer (T), retentive timer (ST), long timer (LT), long retentive timer (LST), and timer function 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.

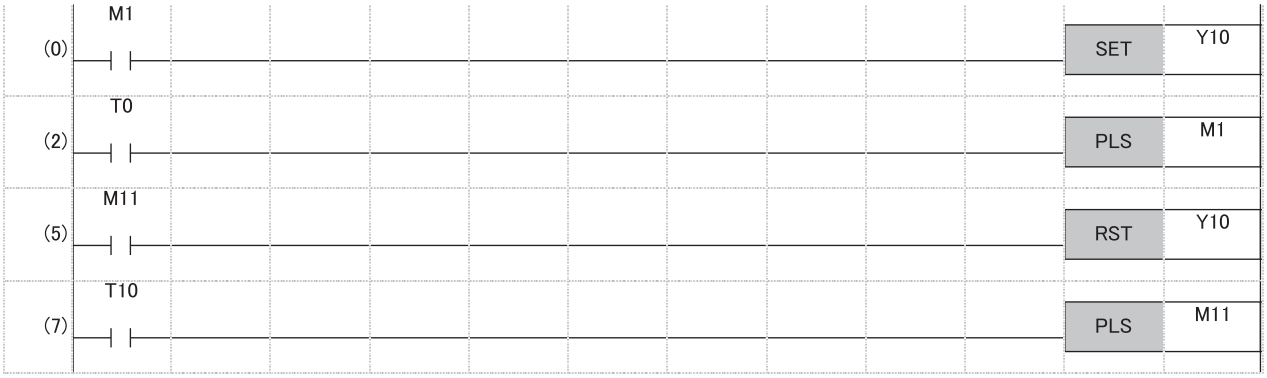
Ex.

Program that delays outputting values by one scan after the time of the timer (T) is up

[Without measures]



[With measures]



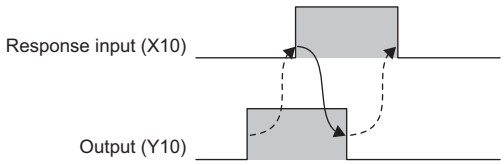
Point

In the CPU parameter, select "Transfer" (default setting) in "Signal Flow Memory Tracking Setting" of "Redundant System Settings". (☞ Page 399 Tracking transfer setting for the signal flow memory)

Precautions on access to intelligent function module or external devices

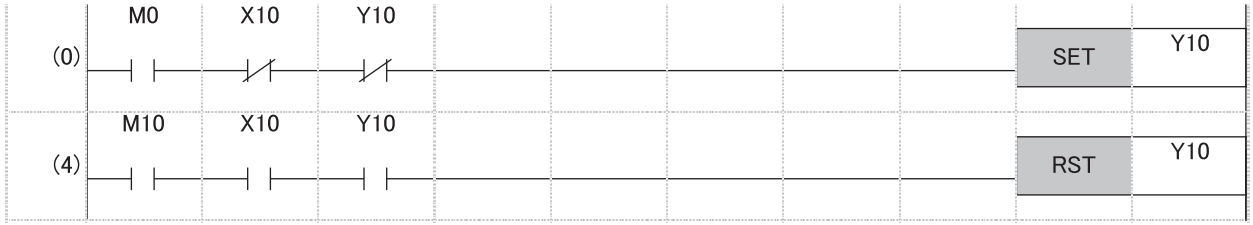
Depending on the timing of system switching cause to be caused, such as power-off, tracking processing is suspended and device/label data may not be applied to the CPU module in the new control system after the system switching.
Consequently, output data may differ from device/label data of the CPU module 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

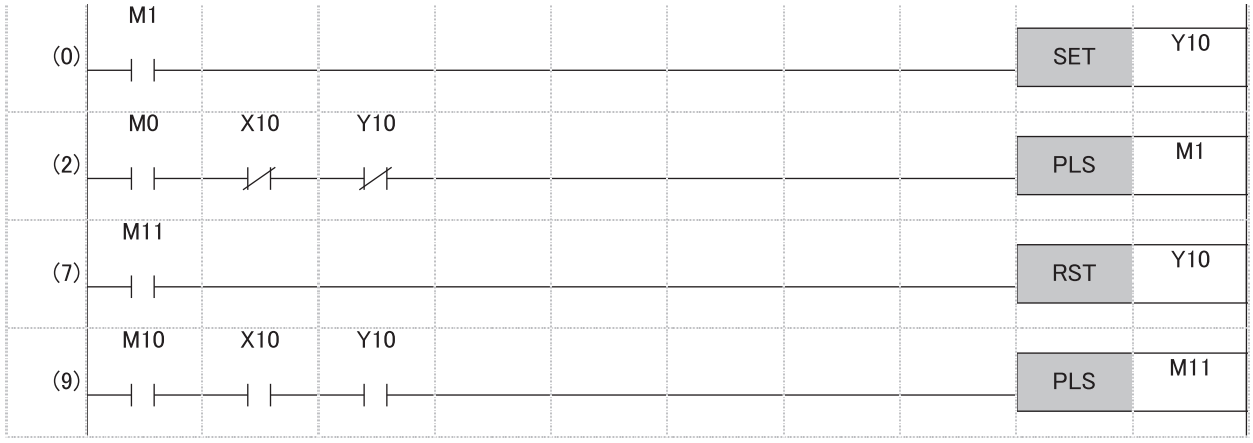


In the following program, turning on M0 turns on the output (Y10) and turning on M10 turns off the output (Y10).

[Without measures]



[With measures]



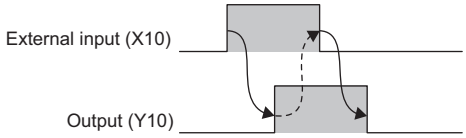
(2) PLS M1 delays SET Y0 by one scan.
(9) PLS M11 delays RST Y10 by one scan.

Point

In the CPU parameter, select "Transfer" (default setting) in "Signal Flow Memory Tracking Setting" of "Redundant System Settings". (☞ Page 399 Tracking transfer setting for the signal flow memory)

Ex.

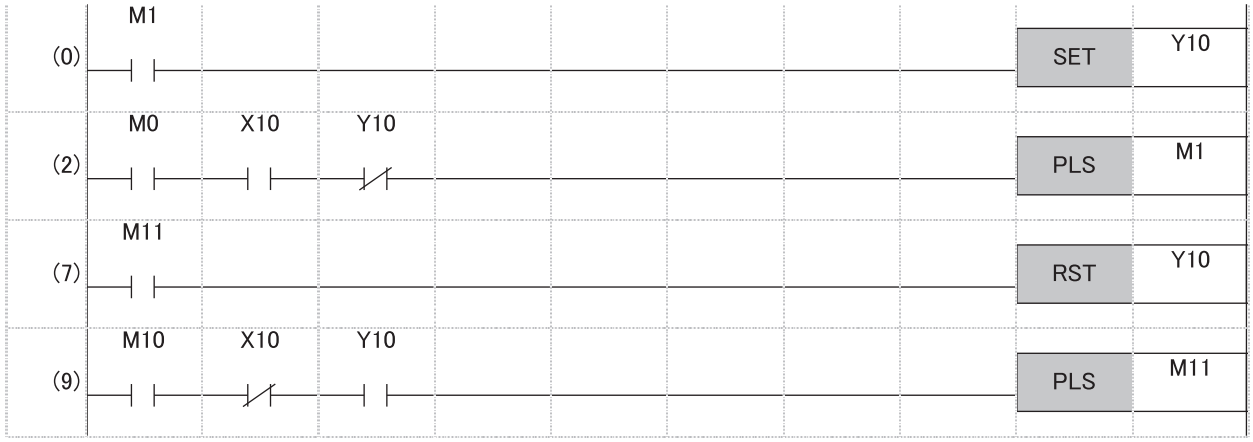
When output is returned to external input



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 399 Tracking transfer setting for the signal flow memory)

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.

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.

Precautions for the redundant extension base unit configuration

This section 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 204 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 391 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 391 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

MEMO

PART 7 DEVICES, LABELS, AND CONSTANTS

This part consists of the following chapters.

27 DEVICES

28 LABELS

29 LATCH FUNCTION

30 DEVICE/LABEL INITIAL VALUE SETTINGS

31 LABEL INITIALIZATION FUNCTION

32 CONSTANTS

27 DEVICES

This chapter describes the devices.

27.1 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	Unchangeable	Hexadecimal
	Bit	Output	Y	12K points		Hexadecimal
	Bit	Internal relay	M	12K points	Changeable (☞ Page 462 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		Decimal
	Bit/word	Retentive timer	ST	0 points		Decimal
	Bit/double word	Long timer	LT	1K points		Decimal
	Bit/double word	Long retentive timer	LST	0 points		Decimal
	Bit/word	Counter	C	512 points		Decimal
	Bit/double word	Long counter	LC	512 points		Decimal
	Word	Data register	D	18K points		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*5	Unchangeable	Hexadecimal
	Bit	Link output	Jn\Y	160K points (Max.)*1*5		Hexadecimal
	Bit	Link relay	Jn\B	640K points (Max.)*1*5		Hexadecimal
	Bit	Link special relay	Jn\SB	5120 points (Max.)*1*5		Hexadecimal
	Word	Link register	Jn\W	2560K points (Max.)*1*5		Hexadecimal
	Word	Link special register	Jn\SW	5120 points (Max.)*1*5		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 489 Index register setting)	Decimal
	Double word	Long index register	LZ	2 points		Decimal
File register	Word	File register	R/ZR	0 points	Changeable	Decimal

Classification	Type	Device name	Symbol	Number of points of Default	Parameter-set range	Notation
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 498 Pointer setting)	Decimal
	—	Interrupt pointer	I	1024 points	Unchangeable	Decimal
Other devices	—	Network No. specification device	J	—	Unchangeable	Decimal
	—	I/O No. specification device	U	—		Hexadecimal
	—	SFC block device*3	BL	320 points		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 R120PCPU.

*3 Version restrictions vary according to the operation mode. (☞ Page 762 Added and Enhanced Functions)

*4 Can be used as a device comment in the SFC program.

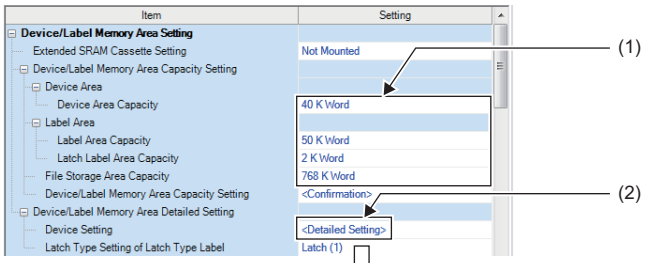
*5 The maximum number of points differs depending on the "Link Direct Device Setting" of the engineering tool. Before using the "Link Direct Device Setting", check the version of the CPU module and engineering tool used. (☞ Page 762 Added and Enhanced Functions)

27.2 Device Setting

The number of points of each user device can be changed (☞ Page 464 User Device)

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

Window



Item	Symbol	Device	
		Points	Range
Input	X	12K	0 to 2FFF
Output	Y	12K	0 to 2FFF
Internal Relay	M	12K	0 to 12287
Link Relay	B	8K	0 to 1FFF
Link Special Relay	SB	2K	0 to 7FF
Annunciator	F	2K	0 to 2047
Edge Relay	V	2K	0 to 2047
Step Relay	S	0	
Timer	T	1K	0 to 1023
Long Timer	LT	1K	0 to 1023
Retentive Timer	ST	0	
Long Retentive Timer	LST	0	
Counter	C	512	0 to 511
Long Counter	LC	512	0 to 511
Data Register	D	18K	0 to 18431
Link Register	W	8K	0 to 1FFF
Link Special Register	SW	2K	0 to 7FF
Latch Relay	L	8K	0 to 8191
File Register	ZR		
Total Device			38.4K Word
Total Word Device			34.5K Word
Total Bit Device			62.0K Bit

(1) The capacity of each area can be changed. (☞ Page 142 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 142 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 M94773247	64 points
Bit	Link relay	B	B0 to B5A61FFF	64 points
Bit	Annunciator	F	F0 to F32767	64 points
Bit	Link special relay	SB	SB0 to SB5A61FFF	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 T5265151	32 points
Word	Retentive timer	ST	ST0 to ST5265151	32 points
Word	Long timer	LT	LT0 to LT1480831	1 point
Word	Long retentive timer	LST	LST0 to LST1480831	1 point
Word	Counter	C	C0 to C5265151	32 points
Word	Long counter	LC	LC0 to LC2787391	32 points
Word	Data register	D	D0 to D5923327	4 points
Word	Link register	W	W0 to W5A61FF	4 points
Word	Link special register	SW	SW to SW5A61FF	4 points
Bit	Latch relay	L	L0 to L32767	64 points

*1 This is the maximum number of points for the R120PCPU with an extended SRAM cassette (8MB) (NZ2MC-8MBSE). 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.

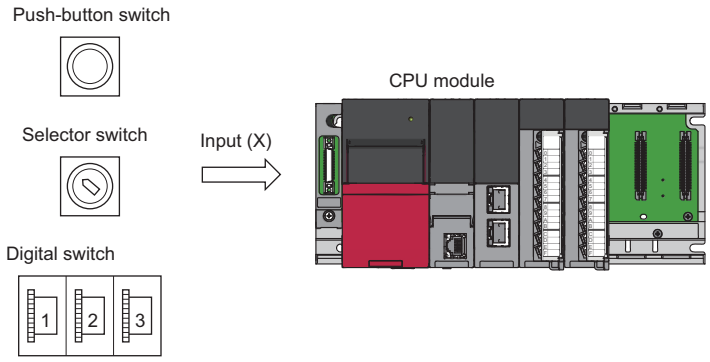
*2 Version restrictions vary according to the operation mode. (☞ Page 762 Added and Enhanced Functions)

27.3 User Device

This section describes the user device.

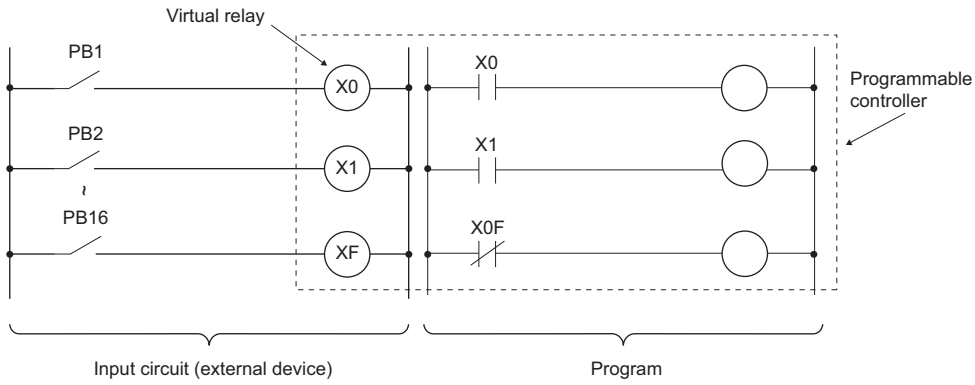
Input (X)

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

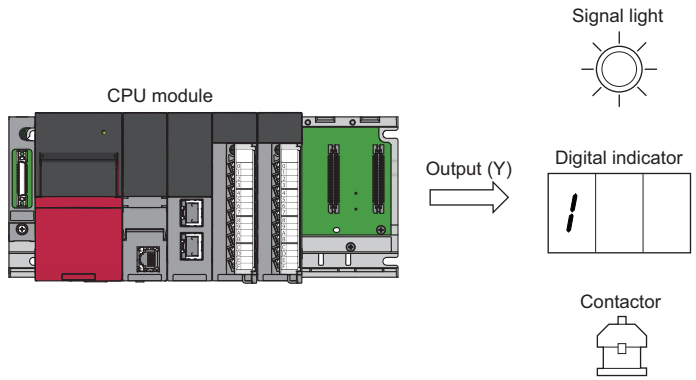
Assume that one virtual relay Xn is incorporated into the CPU module for each input point. In the program, a normally open contact and normally closed contact for the Xn are used.



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.

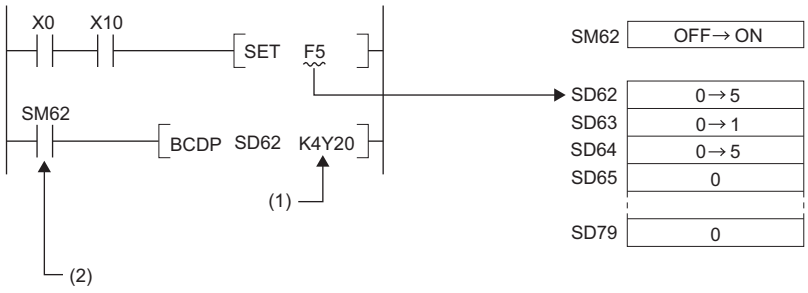
Refreshing network modules using link relay

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.

When using the SET F☐ instruction or the OUT F☐ instruction for registering annunciators in redundant mode, refer to the following as well.

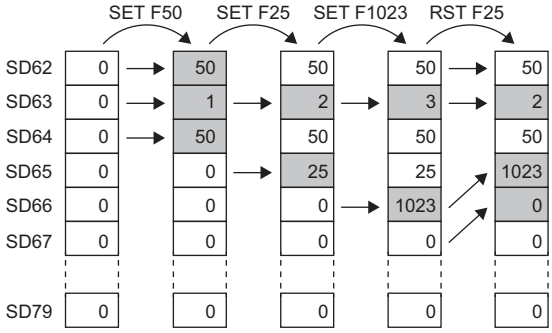
☞ Page 452 Precautions for using the annunciator (F)

■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.

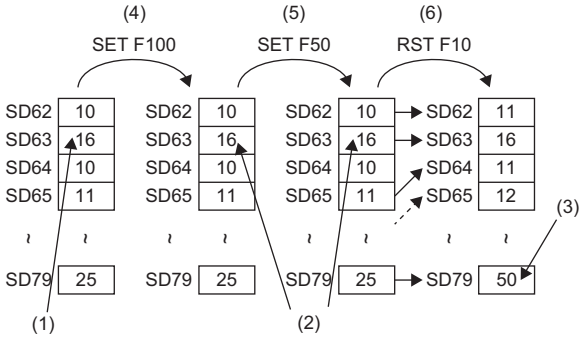
■Operations when annunciators are turned off

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.

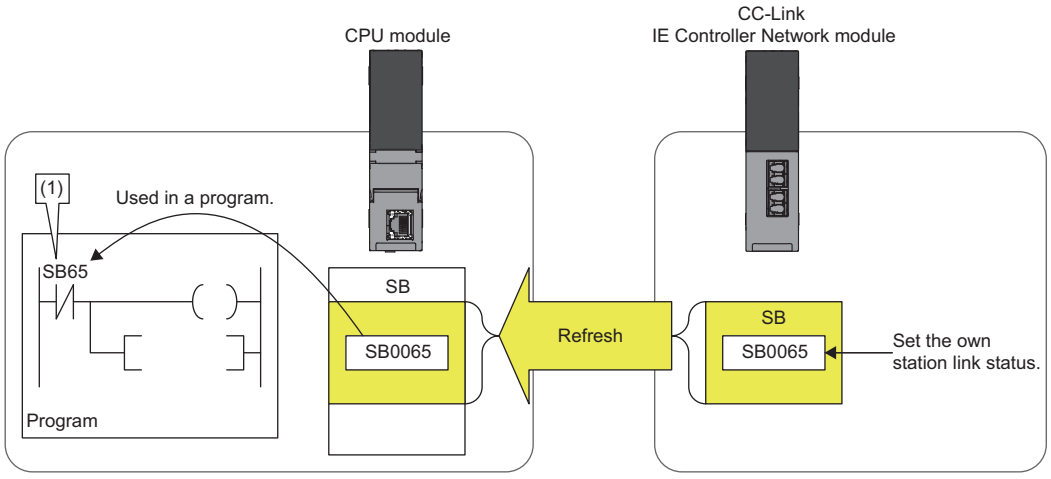
(4) Turn on the 17th device.

(5) Turn on the 18th device.

(6) Turn off the first device.

Link special relay (SB)

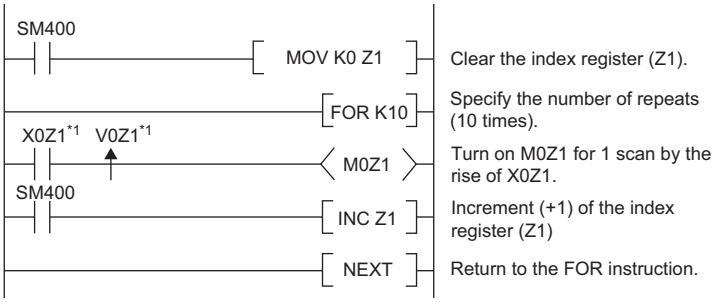
The communication status and error detection status of network modules, such as the CC-Link IE Controller Network module, are output to the link special relay (J□\SB□) on the network. The link special relay (SB) is a device for using as a refreshing target of link special relays in the network. The part which is not used for refreshing can be used for other applications.



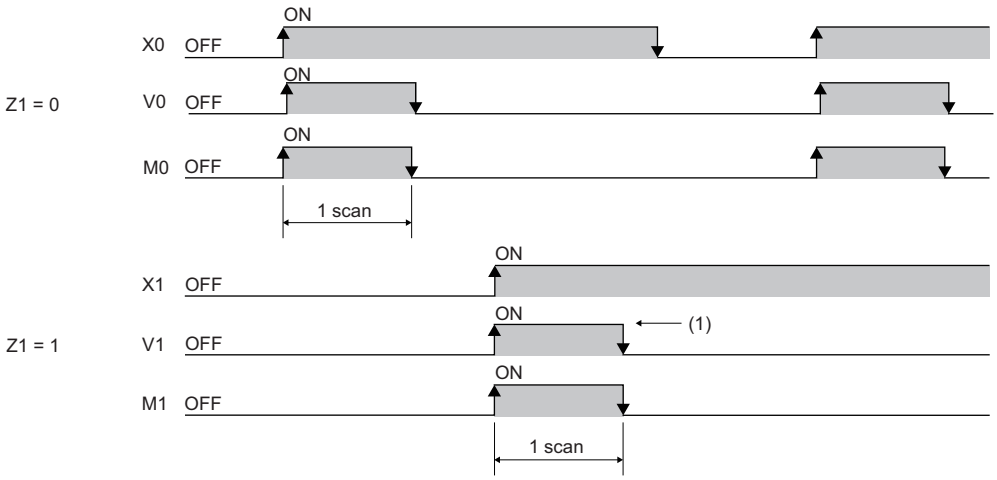
(1) The network status is checked.

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

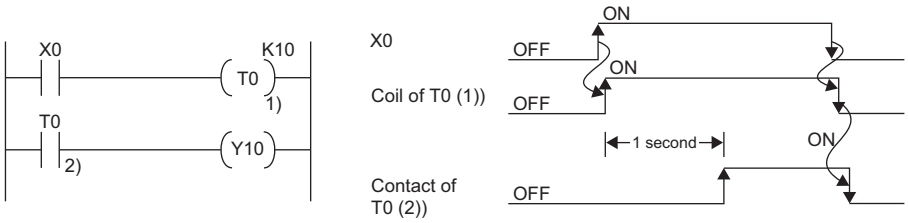
There are two types of timers: timer (T) which retains the current value in 16-bit units and long timer (LT) which retains it in 32-bit units. The timer (T) and the long timer (LT) are different devices and the number of device points can be set for each of them. In addition, there are the retentive 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.

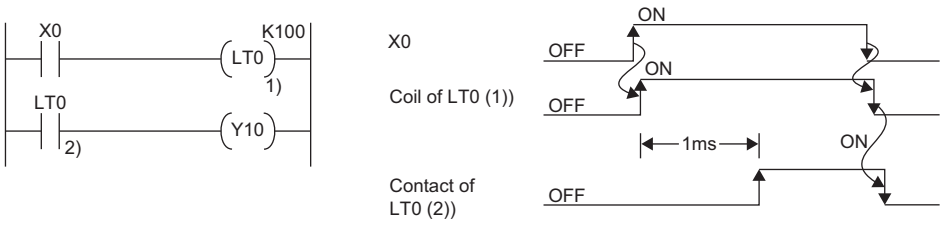
■Timer (T)

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.



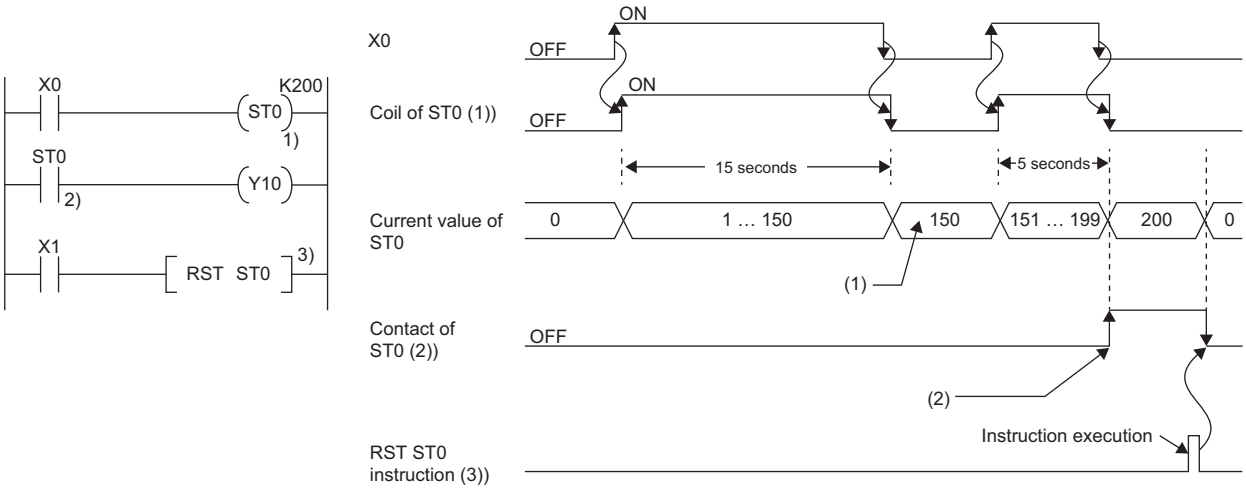
*1 This figure shows the example when the long timer time limit value setting is 0.01ms.

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 115 Interrupt Program)

■Retentive timer (ST)

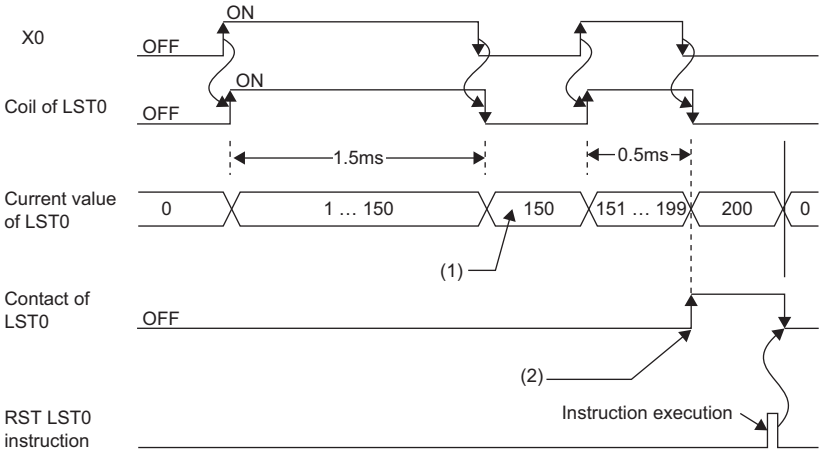
This device counts the sum of time duration in which the coil is turned on. When the retentive timer coil is turned on, measurement starts and when the timer current value matches a setting value (when time is up), the retentive timer contact is turned on. The current value and the contact on/off state 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 115 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

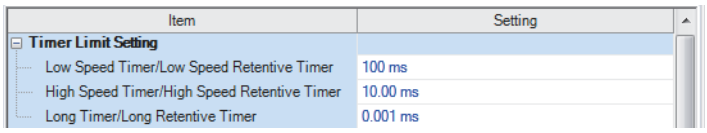
Although the low-speed timer and high-speed timer are the same device, timer limit value are different depending on how to specify the timer device (how to write the instruction). For example, specifying OUT T0 generates a low-speed timer and specifying OUT H T0 produces a high-speed timer even when 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	Set 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	Set 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	Set 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

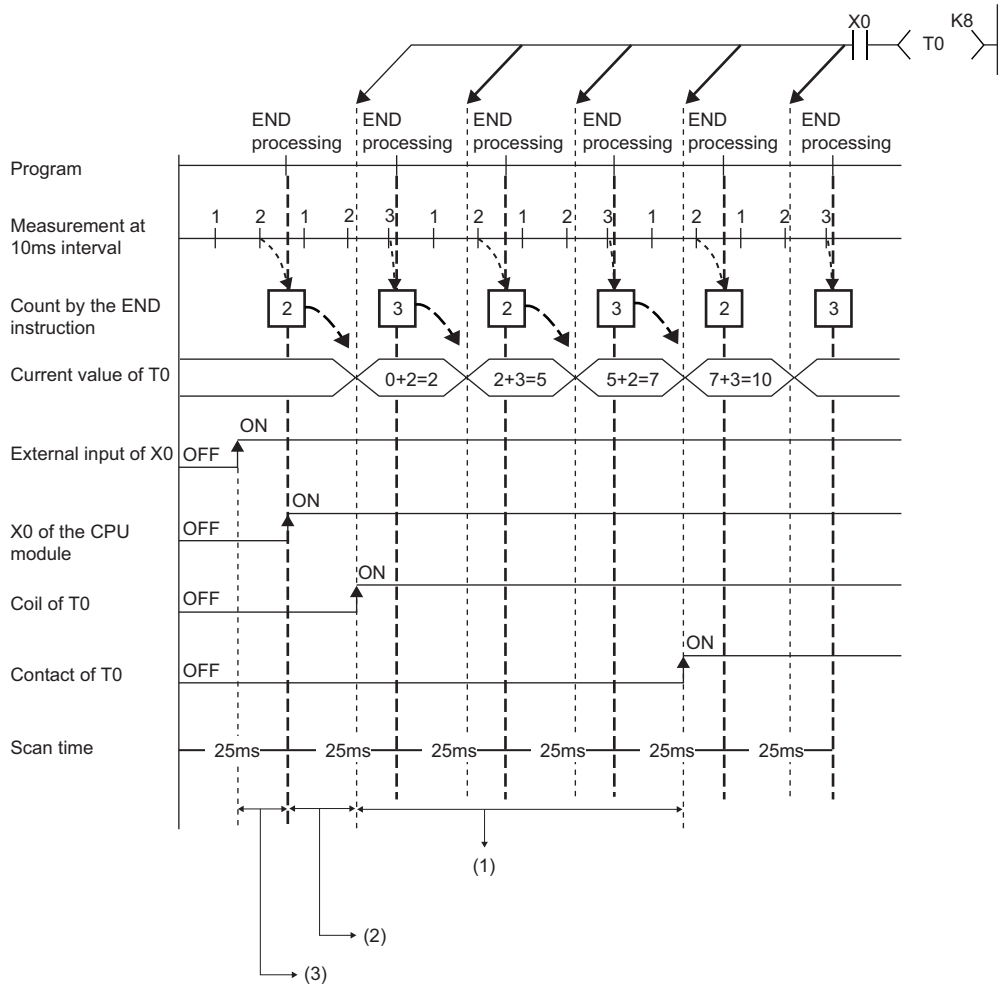
This sections describes the accuracy of timers.

■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



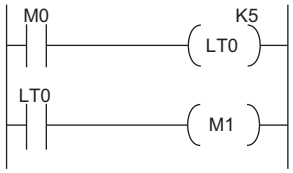
(1) Accuracy from when the coil of the timer turns on until when the contact of the timer turns on
- (1 scan time + timer limit setting) to (1 scan time)

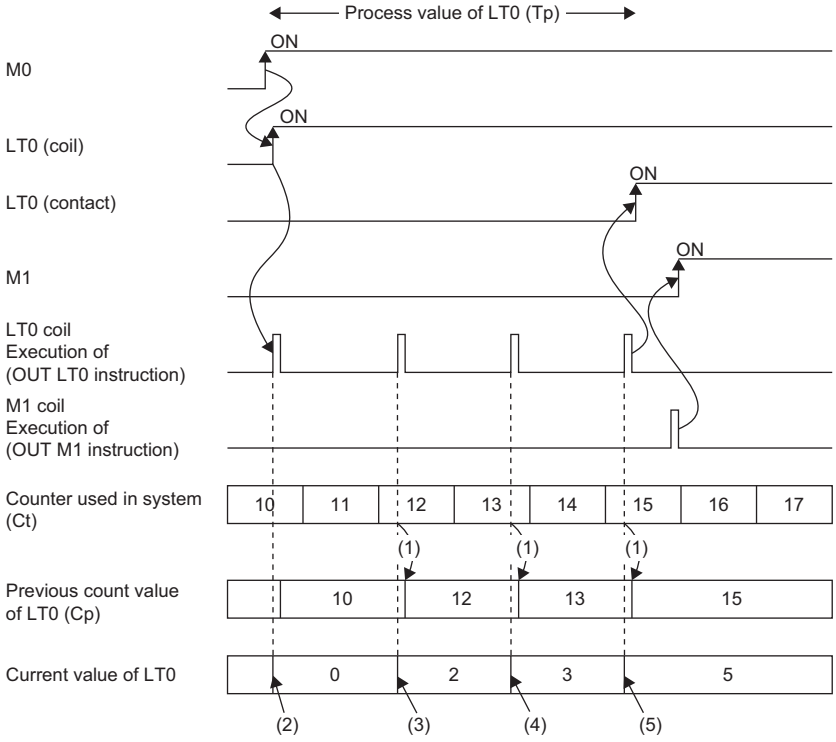
(2) Timing when the coil of the timer turns on

(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).

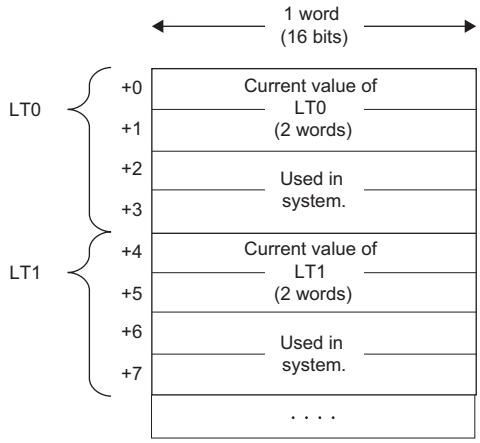




(1) Cp is updated.
(2) LT0 is initialized to 0.
(3) The result of (Ct - Cp) is added.: 0 + (12 - 10) = 2
(4) The result of (Ct - Cp) is added.: 2 + (13 - 12) = 3
(5) The result of (Ct - Cp) is added.: 3 + (15 - 13) = 5
Tp: time duration from the long timer coil activation to long timer contact activation
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.

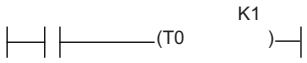
■Timer setting value and timer limit setting

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)

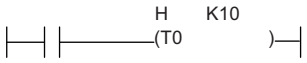


■Timer limit setting

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

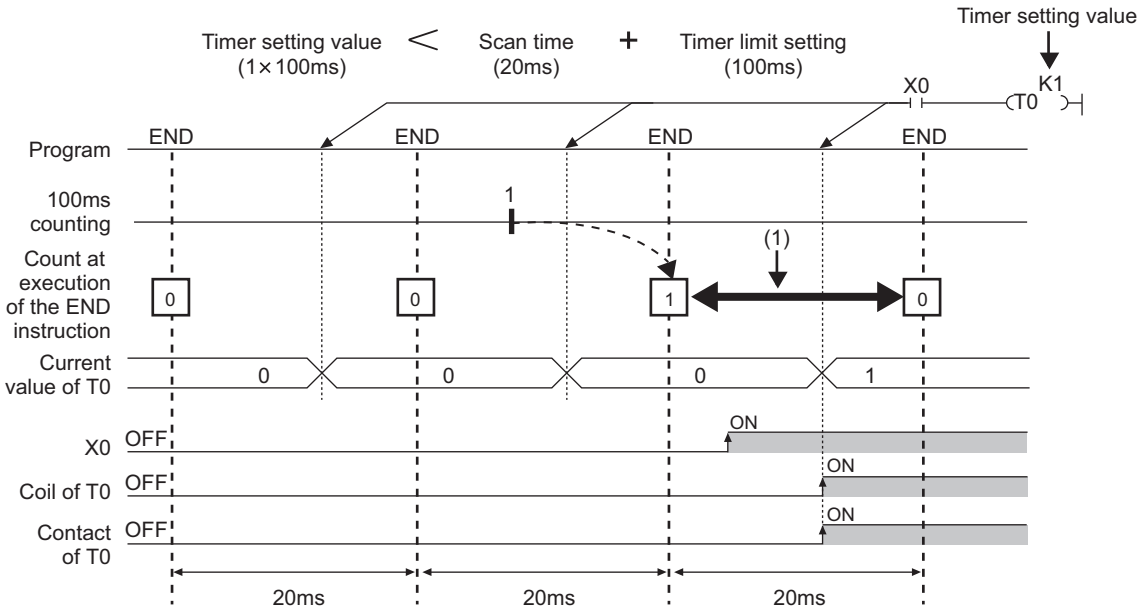
High Speed Timer/High Speed Retentive Timer: 10.00ms

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.

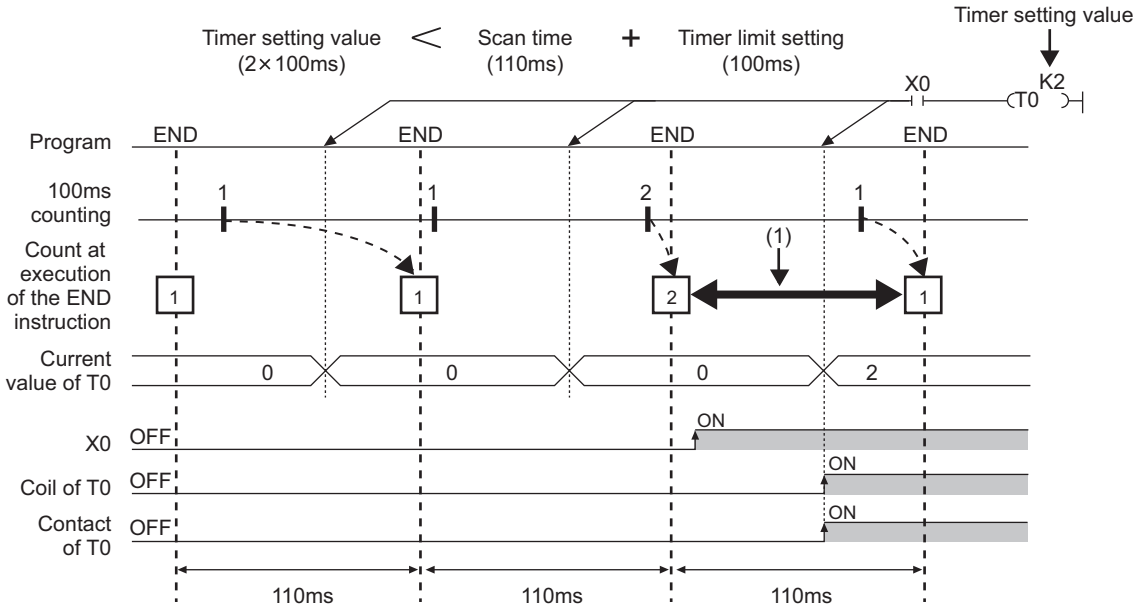
In the case where the values are set as follows (timer setting value: 1 (1 × 100ms), scan time: 20ms, and timer limit setting: 100ms), when the coil of the timer (T0) turns on in the next scan after the coefficient of the END instruction becomes equal to or higher than the timer setting value, the coil 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.

■Precautions in redundant mode

For precautions on timers and timer function blocks at system switching, refer to the following.

☞ Page 453 Precautions on timers and timer function blocks

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

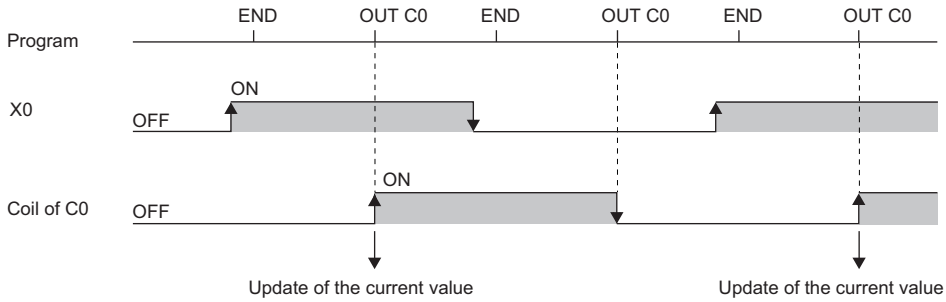
The long counter (LC) can be used in interrupt programs. (☞ Page 115 Interrupt Program)

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]

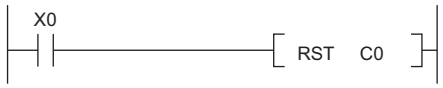
[Update timing of the current value]



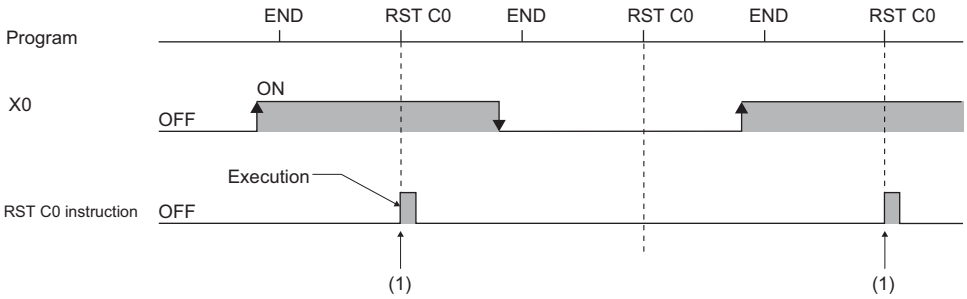
Resetting counters

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]



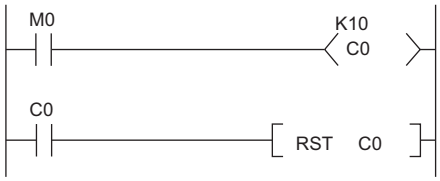
[Reset timing of the counter]



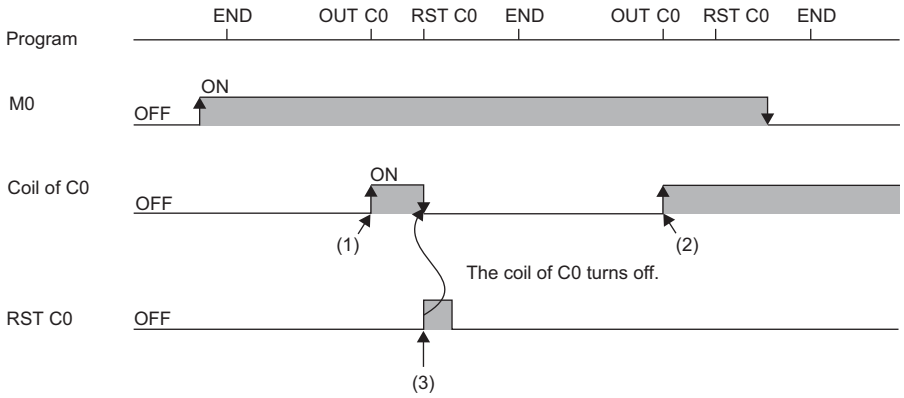
(1) The count value is cleared, and the contact turns off.

■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.

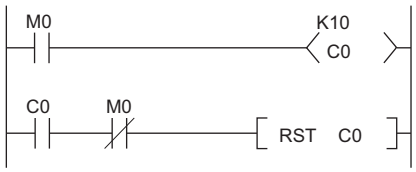


In the above ladder example, the coil of C0 is turned on by turning on M0 and as a result the current value is updated. When C0 count reaches its upper limit, C0 contact is turned on and C0 current value is cleared by execution of the RST C0 instruction. At this time C0 coil is also turned off. When 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.

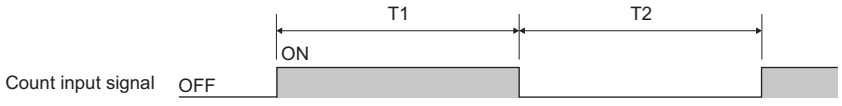


Maximum counting speed for counters

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 $T1 \geq T2$, $n = \frac{T2}{T1+T2} \times 100\%$
- When $T1 < T2$, $n = \frac{T1}{T1+T2} \times 100\%$

Data register (D)

This device can store numerical values.

Link register (W)

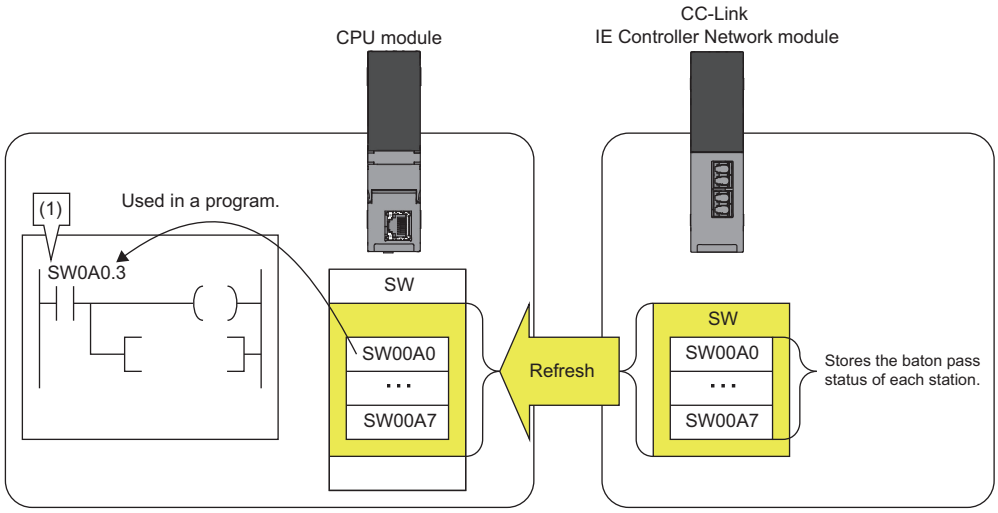
This device is used as a CPU module side device when refreshing word data between the network module, such as the CC-Link IE Controller Network module and the CPU module.

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.

27.4 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.

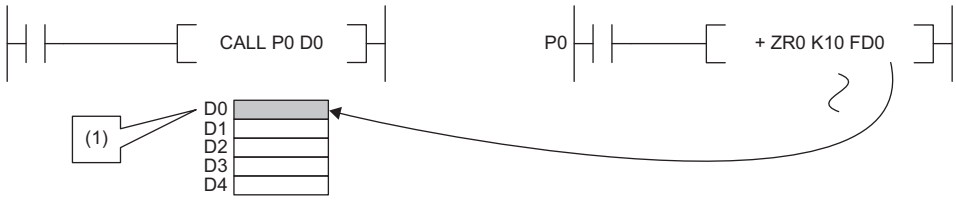
Function output (FY)

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)

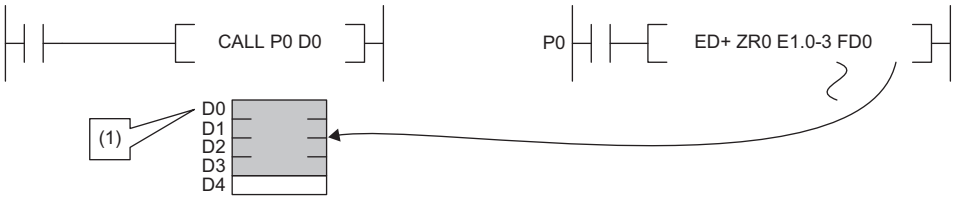
This device is used for writing/reading between the subroutine call source and the subroutine program. The input/output condition of the function register is automatically identified by the CPU module. "Source data" in a subroutine program means data inputted into the subroutine program. "Destination 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.

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 659 List of Special Relay Areas)

Special register (SD)

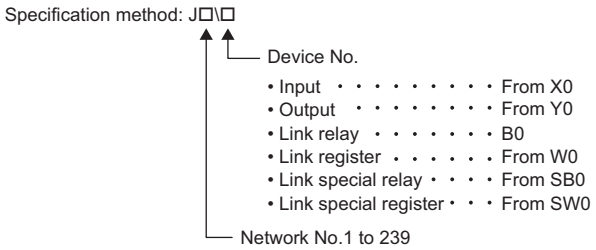
This is the internal register for which the specification is defined in the CPU module, where the status (diagnostics information, system information, etc) of the CPU module is stored. (☞ Page 682 List of Special Register Areas)

27.5 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".



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 Controller Network-equipped module to which the link points extension is set.*2

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Link Direct Device Setting]

Window

Item	Setting
Link Direct Device Setting	
Link Direct Device Setting	Extended Mode (iQ-R Series Mode)

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 Controller Network-equipped module to which the link points extension is set.*2

Other modules operate in the both mode.

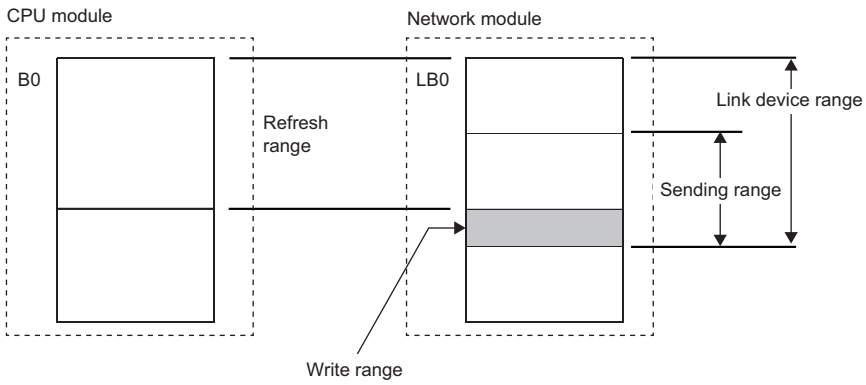
*1 Before using, check the versions of the CPU module and engineering tool used. (☞ Page 762 Added and Enhanced Functions)

*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 send 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

27.6 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.

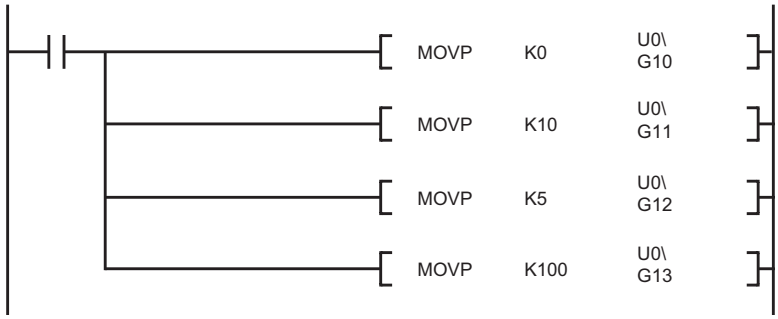
Specify this device with 'Un\Gn'. (Example: U5\G11)

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)

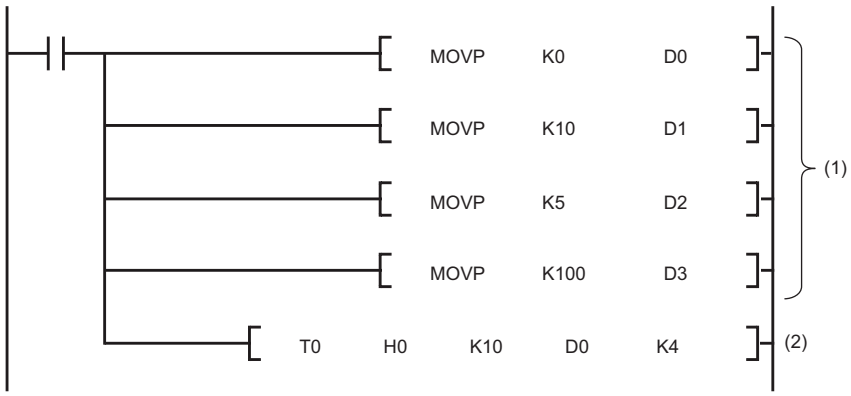
Point

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.

27.7 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 359 Specification method thorough CPU buffer memory access device)

Specification method

Specify this device with 'Un\Gn'. (Example: U3E1\G4095, U3E2\HG1024)

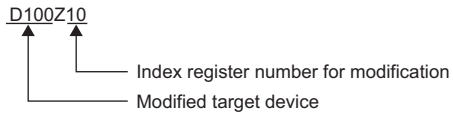
Specified item		Value to be specified
Un (I/O number of the CPU module)	CPU No.1	3E0
	CPU No.2	3E1
	CPU No.3	3E2
	CPU No.4	3E3
G (CPU buffer memory area)	CPU buffer memory	G
	Fixed scan communication area	HG
n (CPU buffer memory address)		0 to 268435455 (decimal)

27.8 Index Register (Z/LZ)

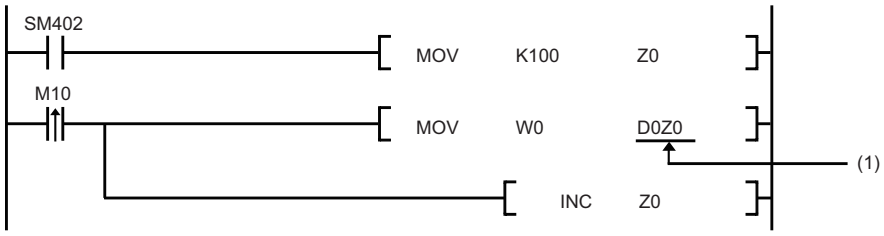
This device is used for the index modification of the device. The index modification is the indirect specification using the index register.
Specify the device with the number obtained from "Device number of device targeted for modification" + "Contents of index register".

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.



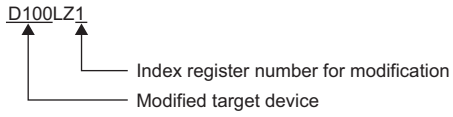
Ex.
Modifying D0 with Z0



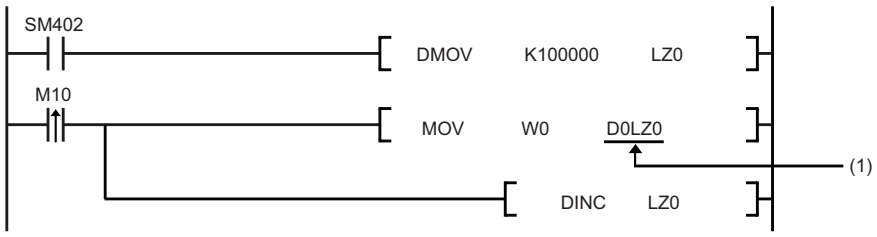
(1) Access D0Z0 = D100.

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

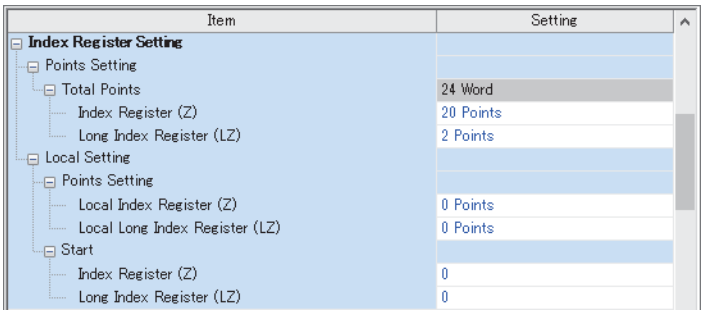
*1 The devices can be used for the contact, coil and current value.
*2 For network numbers and the specification source of I/O numbers, 32-bit index modification cannot be performed.
*3 When the devices are used as an interrupt pointer, index modification cannot be performed.
*4 Before specifying, check the versions of the CPU module and engineering tool used. (☞ Page 762 Added and Enhanced Functions)

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
Number of points setting	Total Points		Check the total number of points for index register and long index register.	—	—
	Index register (Z)		Set the number of points for the index registers.	0 to 24 points (in two-point increments)	20 points
	Long index register (LZ)		Set the number of points for the long index registers.	0 to 12 points (in one-point increments)	2 points
Local setting	Number of points setting	Local Index register (Z)	Set 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)	Set 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
	Head	Index register (Z)	Set the start number for the local index registers. Set within the range of the index register.	0 to 23	0
		Long index register (LZ)	Set 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 468 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 468 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:

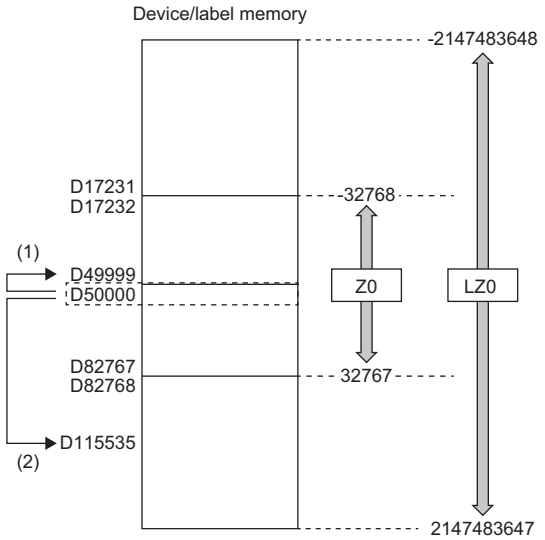
- Review the index modification block(s) within the program.
- To perform the 32-bit index modification specification with ZZ expression, review the range of the index register (Z). Note that the range within the LZ cannot be specified.
- For 32-bit index modification with ZZ expression, because the 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 489 Index register setting)

When values are stored in the index registers

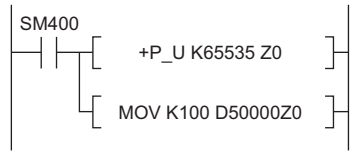
For 16-bit index modification using the index register (Z), the range is -32768 to 32767. Therefore, when values within the range from 32768 to 65535 are stored in the index register (Z) for an instruction which processes unsigned data, the instruction does not work in design because the range of the 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.

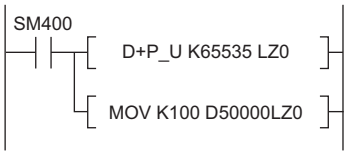
Operation for Index modification



(1) When unexpected operation is executed



(2) When proper operation is executed



(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.

27.9 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.

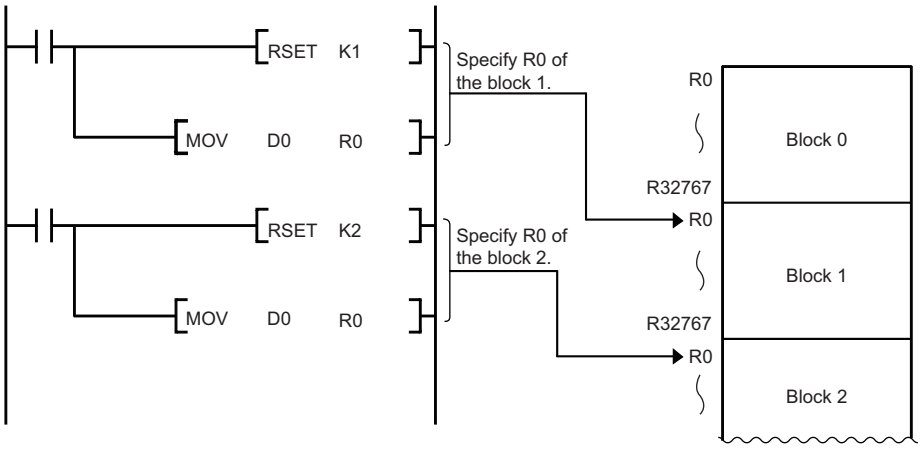
Specification method

There are two types of the specification methods for the file register: block switching and serial number methods.

Block switching method

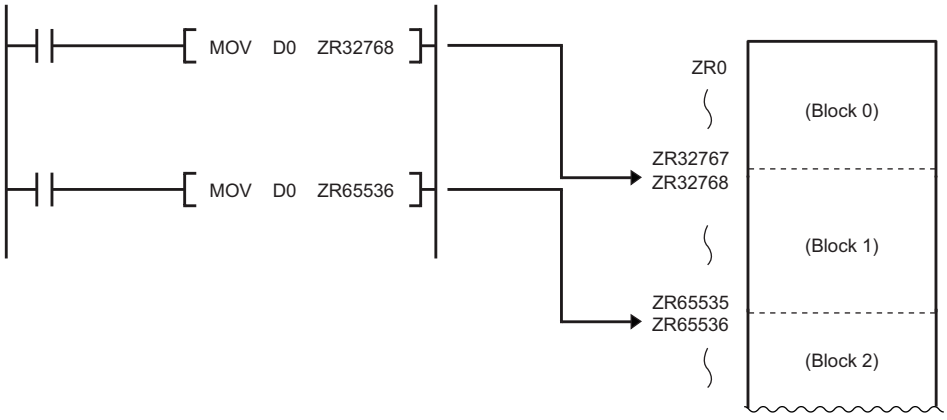
In this method the number of points of file register being used is specified by being divided in increments of 32K point (R0 to R32767). When using more than one block, specification is conducted by switching to the block number used by the RSET instruction. "R" is used as the device symbol. The range 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).



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
4. Write parameters and file register files into the CPU module.

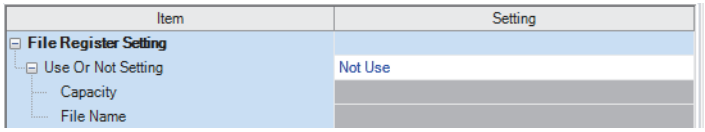
*1 If the capacity is not set, it must be set when creating a file register file and writing it to the programmable controller in the same manner as the procedure 2.

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	Specify 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	Specify 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 143 The setting range of the capacity of each area)	—
File name	Assign a file name to the file register when "Use Common File Register in All Programs" is selected.	1 to 60 characters	—

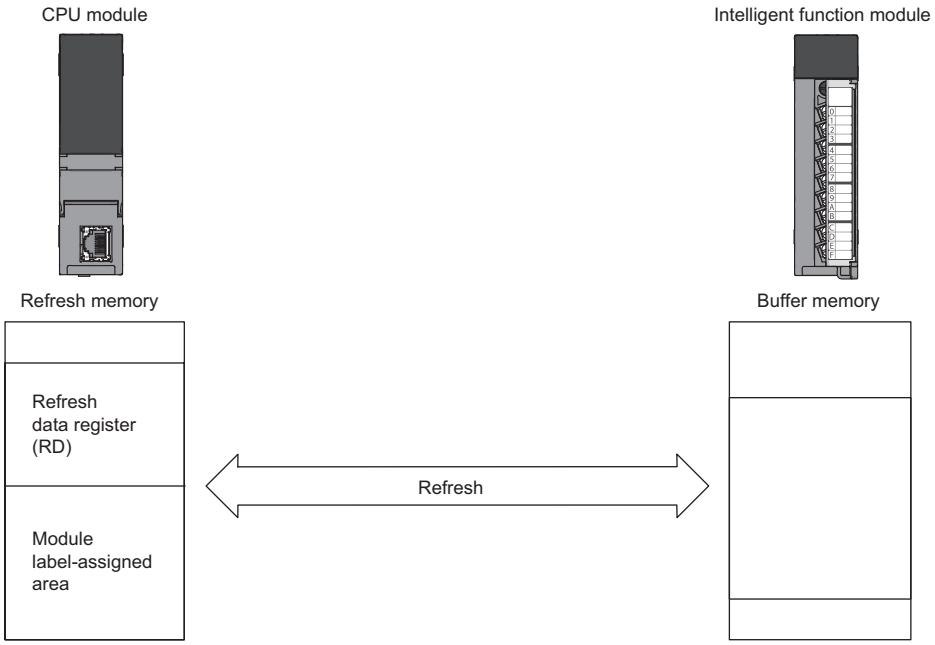
Clearing file registers

To clear the file registers, use the following methods (☞ Page 150 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)

27.10 Refresh Data Register (RD)

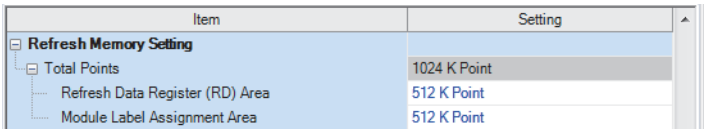
This device is provided for using as a refreshing target of buffer memory on the various devices, such as an intelligent function module. Refresh Data Register (RD) is assigned into the refresh memory area. (☞ Page 145 Refresh memory)



Refresh memory setting

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Refresh Memory Setting]

Window

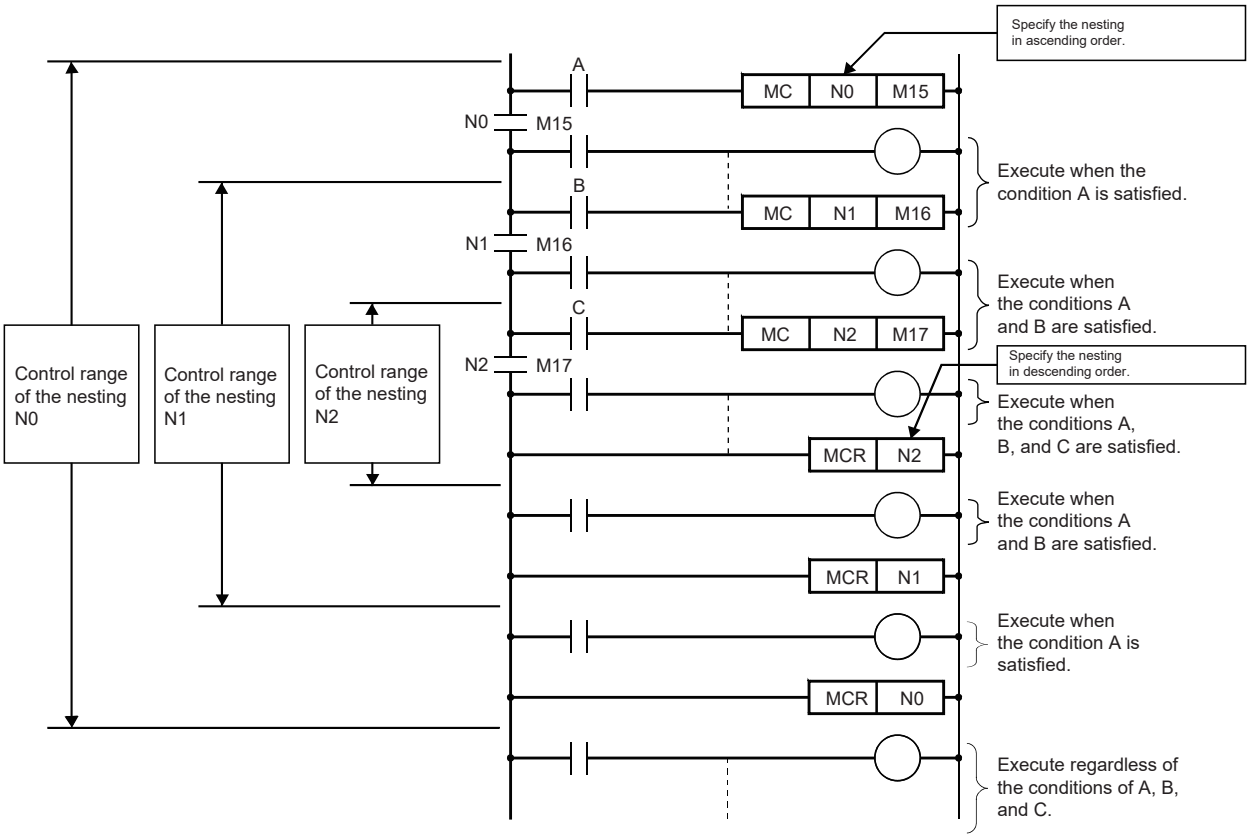


Displayed items

Item	Description	Setting range	Default
Total Points	Check the total number of points for the refresh data register and the assigned area.	—	1024K Point
Refresh Data Register (RD) area	Set the number of points of refresh data register.	0 to 1024K points (in one-point increments)	512K Point
Module Label Assignment Area	Set the number of points for the module label assigned area.	0 to 1024K points (in one-point increments)	512K Point

27.11 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.

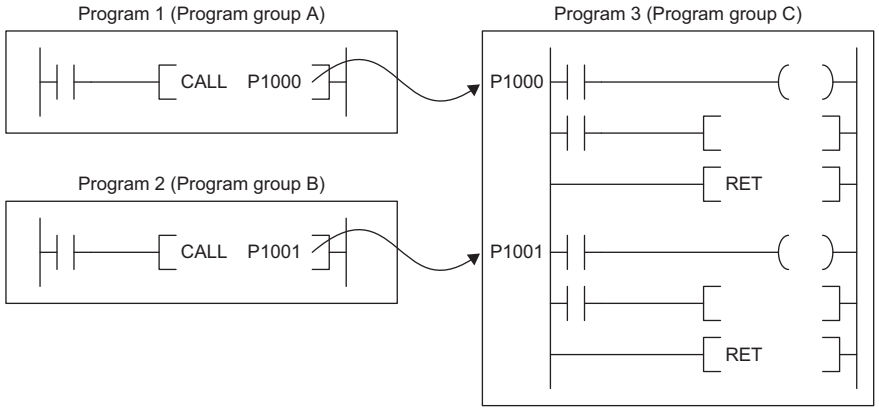
27.12 Pointer (P)

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".

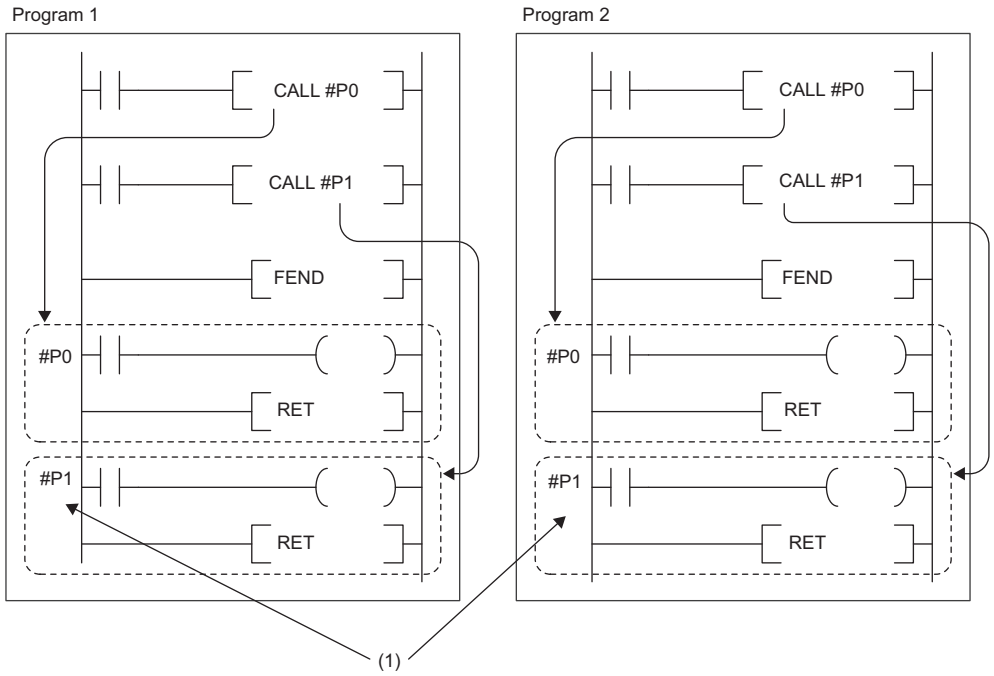
Precautions

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

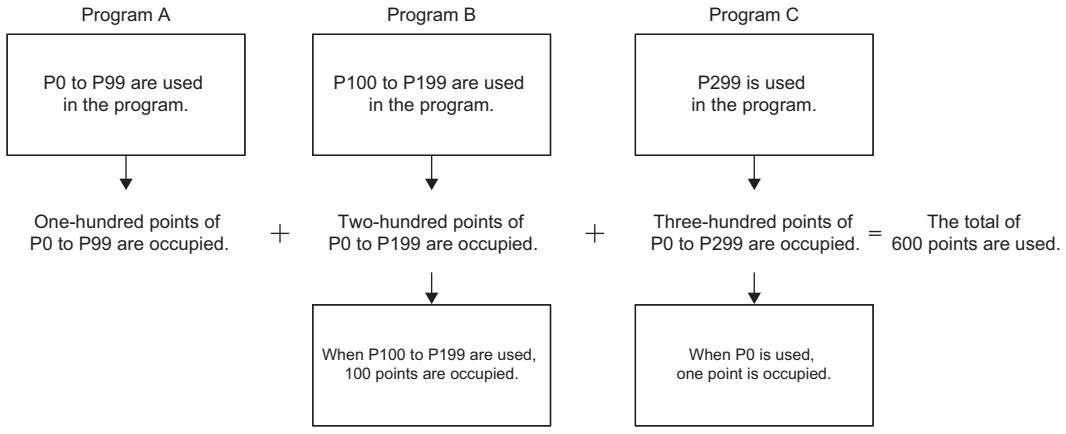
This is the pointer to be independently used in each program where the same pointer number can be used. This pointer is specified in the following format: # (pointer number) (Example: #P0) (☞ Page 505 Specification method for the local devices).



(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.

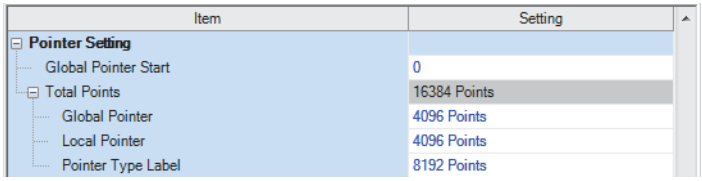


Pointer setting

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	Set the start number of the global pointer.	P0 and over*1	0
Total Points	Check the total number of points for the pointer.	—	16384 points*2
Global Pointer	Set the number of points of the global pointer.	• R120PCPU: 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	Set the number of points of the local pointer.		4096 points*3
Pointer Type Label	Set the number of points for the pointer type label assignment area.		8192 points*4

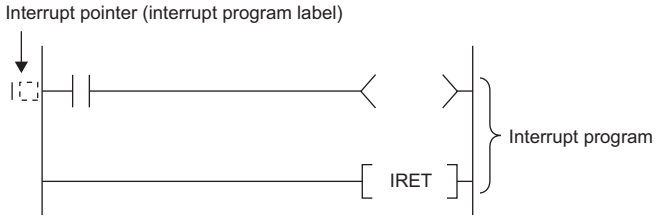
*1 Up to the number of "Total points of pointer device area" - "Number of points of global pointer".
*2 The default number of points is 32768 for the R120PCPU.
*3 The default number of points is 8192 for the R120PCPU.
*4 The default number of points is 16384 for the R120PCPU.

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)".

27.13 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 107 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 129 Interrupt priority)
- The interrupt priority order is the order which is executed when the interrupt factor with the same interrupt priority is generated. (☞ Page 132 Multiple interrupt execution sequence)

27.14 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))

27.15 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))

27.16 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))

27.17 SFC Transition Device (TR)

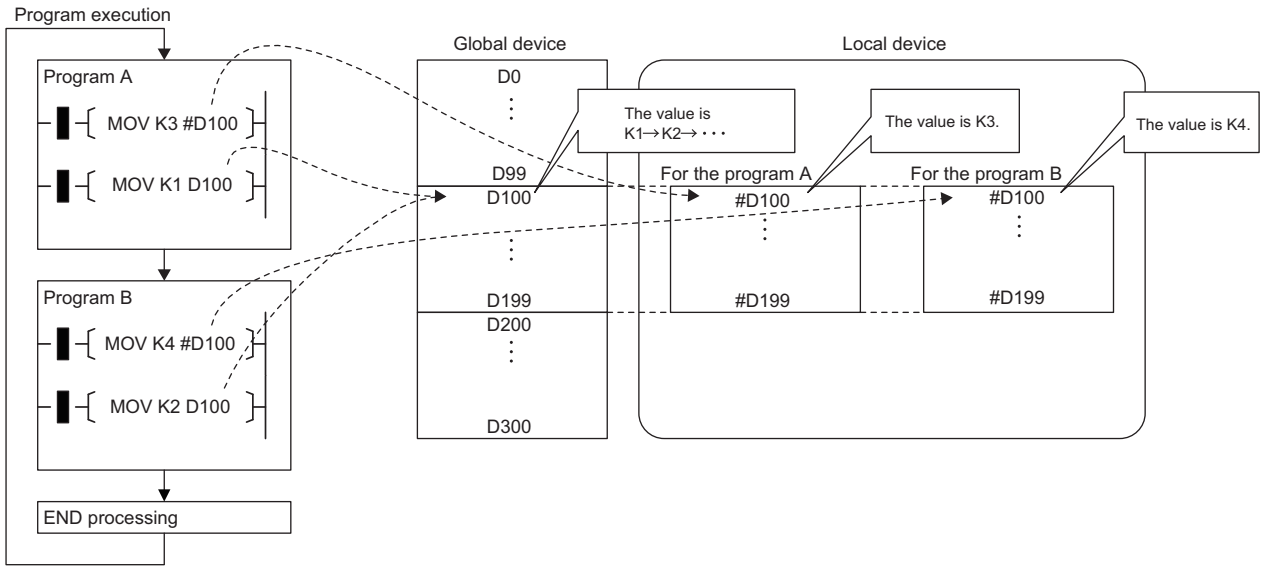
This device is used when specifying SFC program transition conditions. This device can only be used for device comments for transition conditions. (📖 MELSEC iQ-R Programming Manual (Program Design))

27.18 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.

27.19 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 488 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.

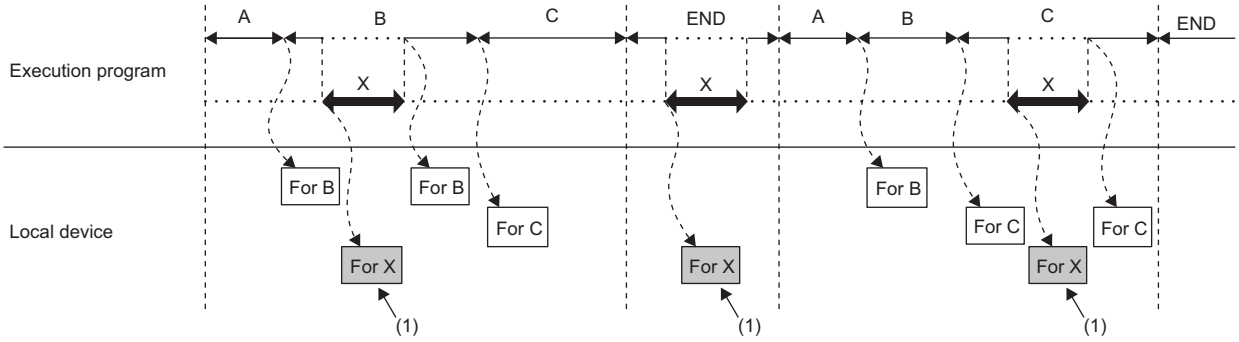
When local device is used in interrupt and other programs

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.

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

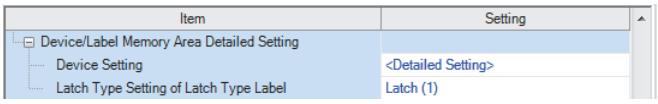
Set the range where each device will be used as a local device and also set whether or not it should be used.

■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



"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 Timer	LST			
Counter	C			
Long Counter	LC			
Data Register	D			
Link Register	W			
Link Special Register	SW			
Latch Relay	L			
File Register	ZR			
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 ÷ 16) + B + (C × 2) + (D × 4) + ((E × 2) ÷ 16)) × 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 463 Range of use of device points)

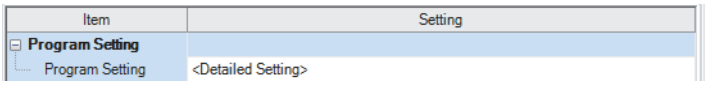
Setting whether or not it should be used

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]

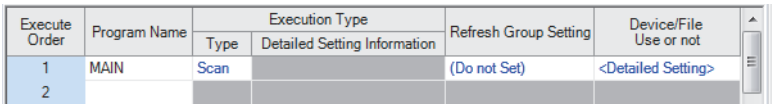
Operating procedure

"Program Setting" window



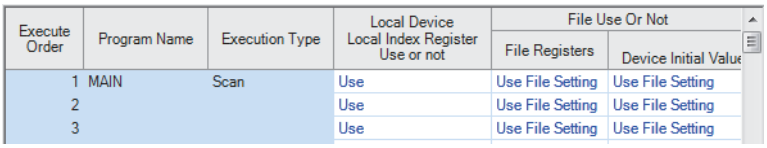
Item	Setting
Program Setting	
Program Setting	<Detailed Setting>

"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.

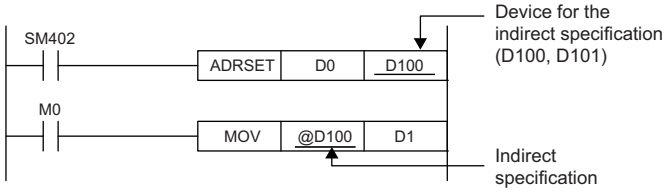
Precautions

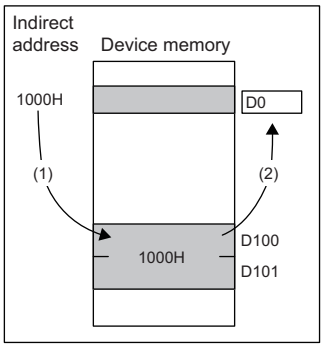
This section describes the precautions when using local devices.

- 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.

27.20 Indirect Specification

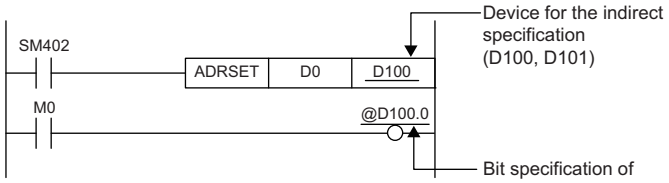
Specify the device using the indirect address of device. Store the indirect address of device to be specified into the device for indirect specification, and write as "@ + Device for indirect specification".

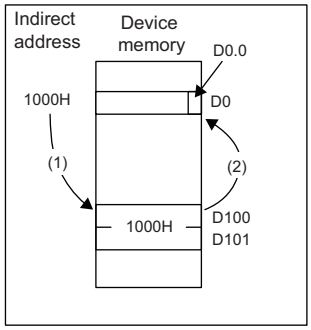




(1) The indirect address of D0 is read into D100, D101.
(2) The indirect address is used to indirectly specify D0.

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.

28 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 521 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

28.1 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.

28.2 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.

28.3 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	×	○	×

28.4 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 510 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 496 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 541 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 460 DEVICES

The specification method of each member is the same as that of the structure data type. (Page 516 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.

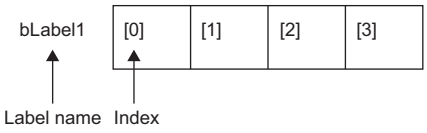
28.5 Arrays

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

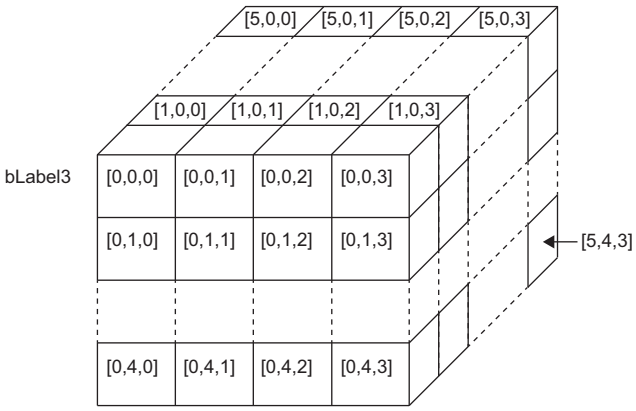
- One-dimensional array (The number of elements is 4.)



- Two-dimensional array (The number of elements is 5 × 4.)

bLabel2				
	[0,0]	[0,1]	[0,2]	[0,3]
	[1,0]	[1,1]	[1,2]	[1,3]
	[4,0]	[4,1]	[4,2]	[4,3]

- 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 515 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 509 Primitive data type • For the structure name: ☞ Page 516 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

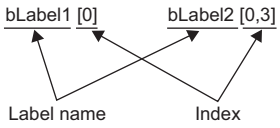
One initial value can be set for a single array definition. (Different initial values cannot be set for each element.)

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 '[ ]'.

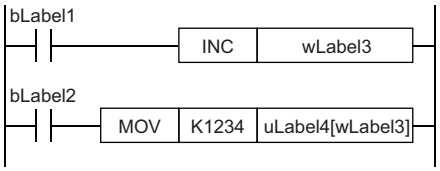


The following table lists the types of indexes that can be specified for arrays.

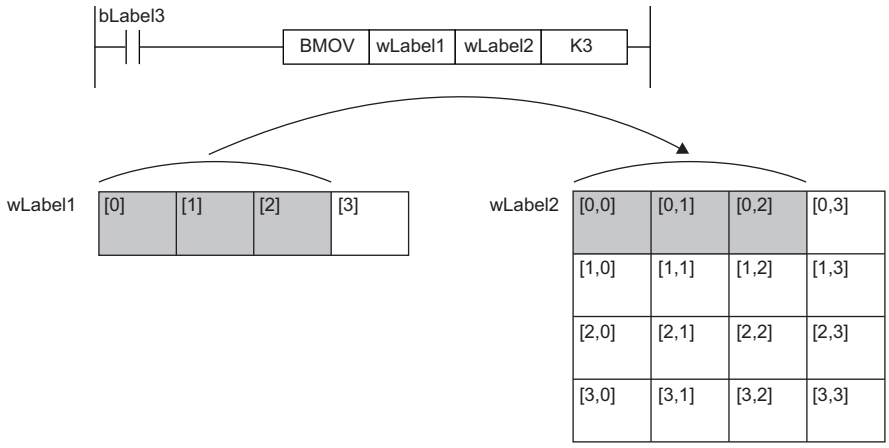
Type	Specification example	Remarks
Constant	bLabel1[0]	An integer can be specified.
Device	bLabel1[D0]	A word device, double-word device, decimal constant, or hexadecimal constant can be specified. (ST, LST, G, and HG cannot be specified.)
Label	bLabel1[uLabel2]	The following data types can be specified. • 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.

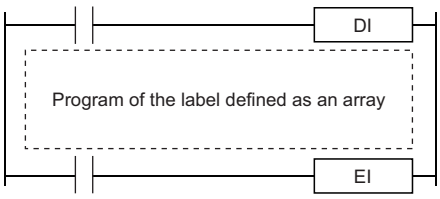
Data type	Setting range
Bit Word [unsigned]/bit string [16 bits] Word [signed]	1 to 2147483648
Double word [unsigned]/bit string [32 bits] Double word [signed] Single-precision real number Time	1 to 1073741824
Timer Counter Retentive timer Long counter Long retentive timer Long timer	1 to 32768
Double-precision 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.

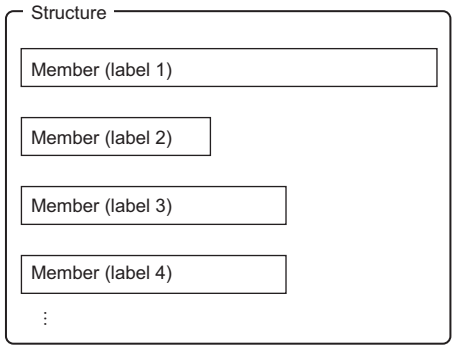
If the array index is specified with data other than a constant, a compile error will not occur. The processing will be performed by accessing another label area or latch label area.

28.6 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

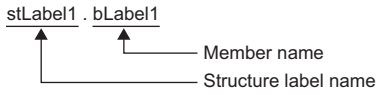
To create a structure, first define the structure, and then define members in the structure.



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.

Ex.
Specifying a member in the structure



Point

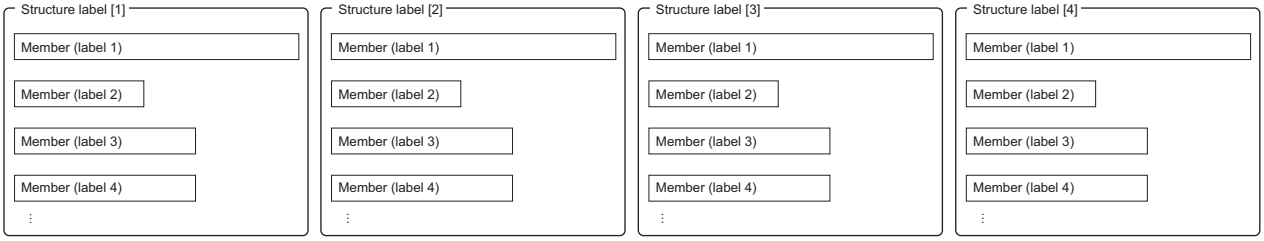
- When labels are registered by defining multiple data types in a structure and used in a program, the order the data is stored after operation is not the order the data types were defined. When programs are converted using the engineering tool, labels are classified into type and data type, and then 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.

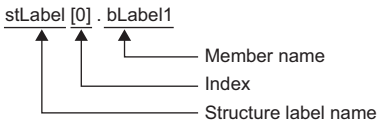


When a structure is declared as an array, add an index enclosed by '[ ]' after the structure label name.

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

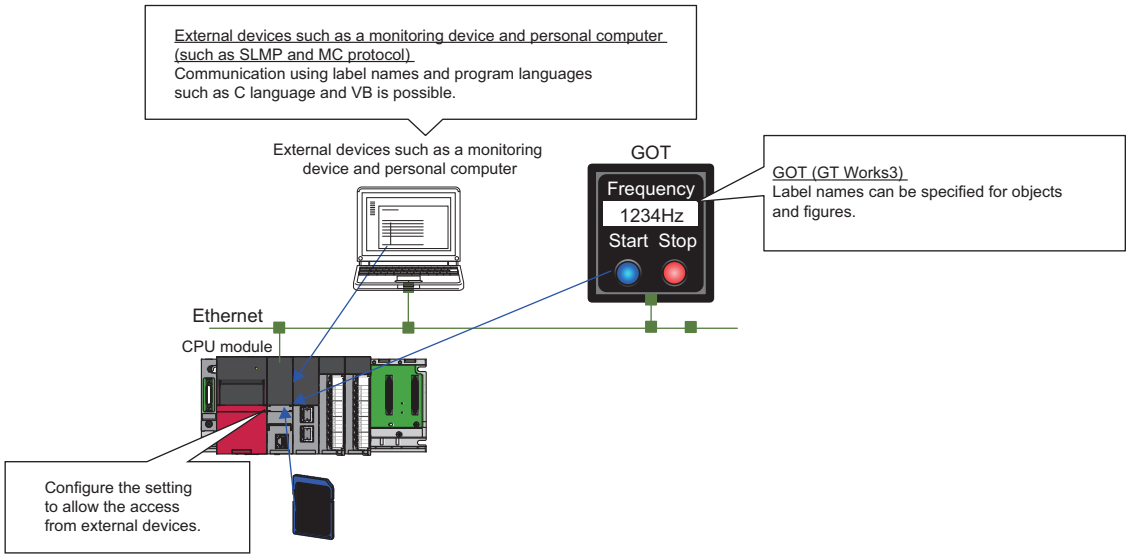
Types of structures

Each of the following labels is predefined as a structure.

Type	Reference
Module label	📖 Function Block Reference for the module used
Timer type	☞ Page 509 Data Types
Retentive timer type	
Counter type	
Long timer type	
Long retentive timer type	
Long counter type	

28.7 Label Access Setting from External Device

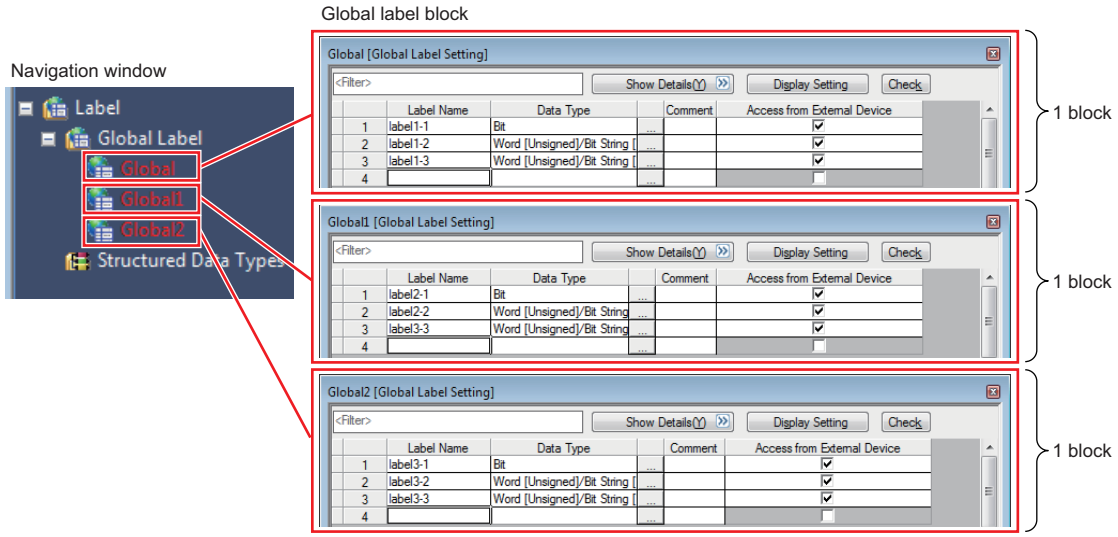
Set the parameters to enable external devices, such as GOT and SLMP-compatible devices, to specify global label names.



Point

The CPU module with the firmware version "28" or later controls global labels in the global label settings by block.

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.

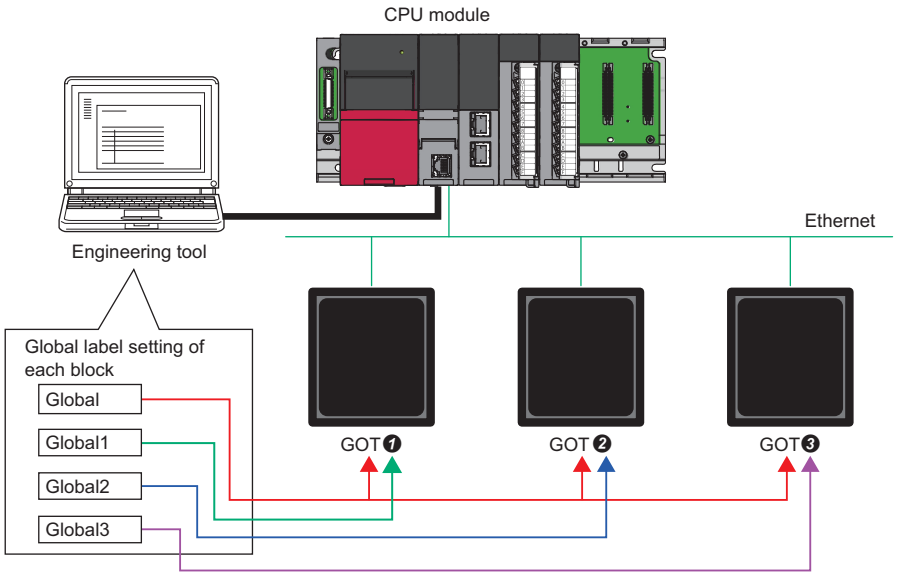
For the GT Designer3 setting, refer to the following.

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.



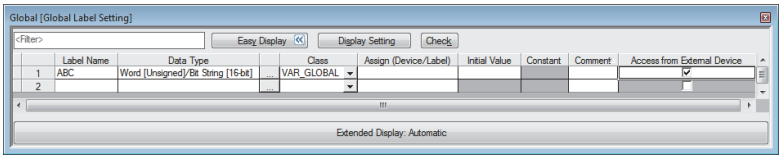
In redundant mode, select "Control System/Standby System", "System A/System B", or "Not Specified" to start communications with a global label name specified.

Configuration procedure

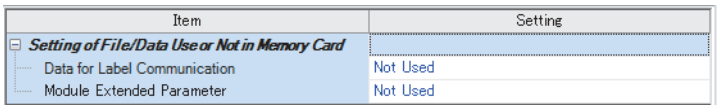
This section describes the configuration procedure to enable access by specifying the global label from external devices.

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.

28.8 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		Use devices if there is a possibility for using these functions because global labels nor local labels cannot be specified for these functions.*1 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.

■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

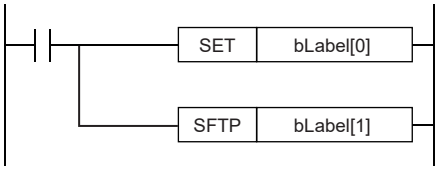
- The number of steps increases because of the transfer instruction. (The scan time increases.)
- Decide the transfer instruction position considering the timing of writing data to the label and executing the function.

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.

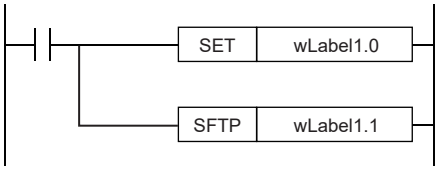
Ex.

SFT(P) instruction



Specify the array of the bit type label to execute the SFT(P) instruction.

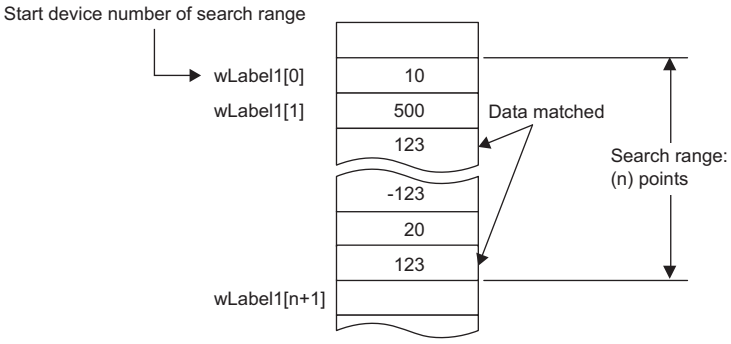
or



Specify the bit number of the word type label to execute the 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.

29 LATCH FUNCTION

29.1 Latch with Battery

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.

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

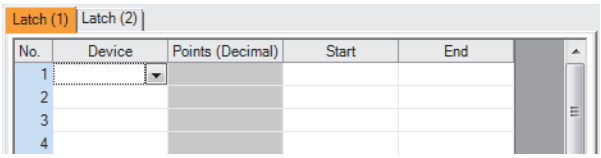
Set the device to latch, its range, and the latch type.

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 Timer	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 Register	SW		
Latch Relay	L		No Setting
File Register	ZR		No Setting
Total Device			
Total Word Device			
Total Bit Device			

"Latch Range Setting" window



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.

3. Check the tab for the latch type, select the device to set and set the latch range (Start, End).

Setting the latch interval

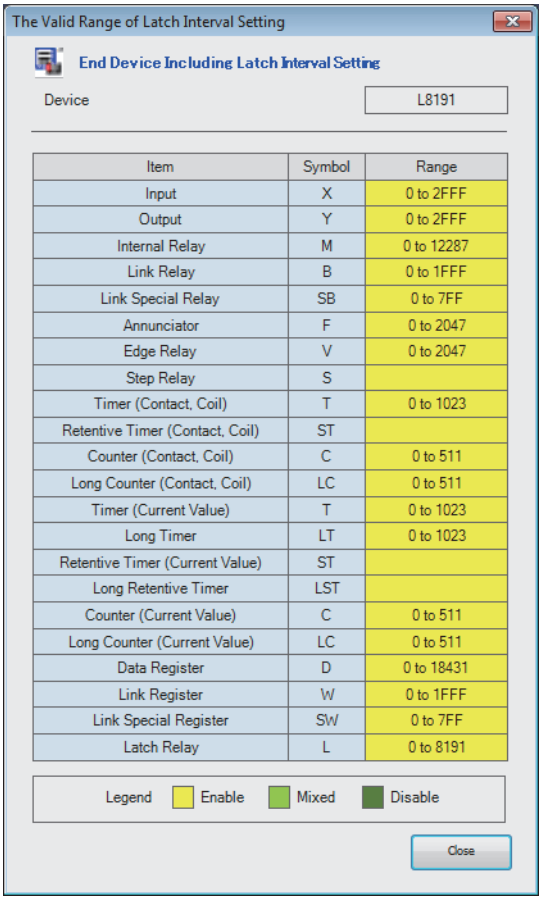
The user can specify the operation which should be performed at a latch interval (☞ Page 528 Setting the latch interval) within the effective range of the latch interval setting*1.

*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]

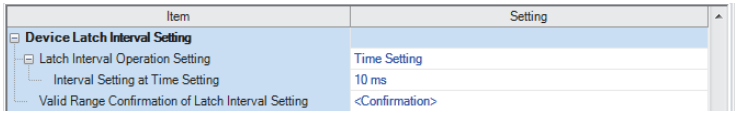
Operating procedure

"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.

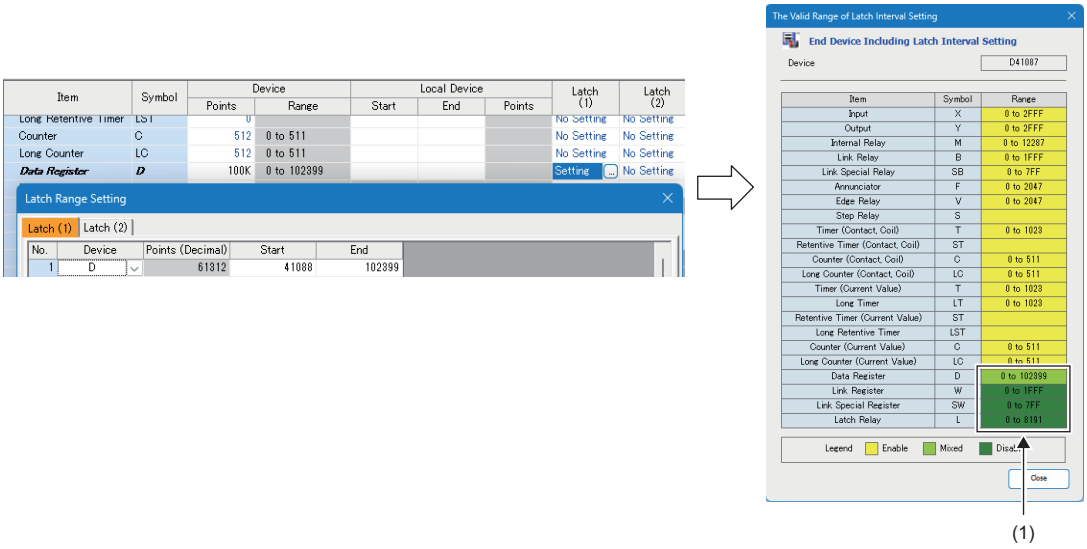
Displayed items

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	1 to 2000ms (in units of 1ms)	10ms
Valid Range Confirmation of Latch Interval Setting	Shows the effective range for the latch interval setting.	—	—

*1 In redundant mode, when the systems are switched, the measurement of the latch interval set in "Interval Setting at Time Setting" in "Device Latch Interval Setting" starts from zero in both the new control system and the new standby system.

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 R08PCPU: (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) D41088 and later are specified as the latch target devices. In this configuration, D41088 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.



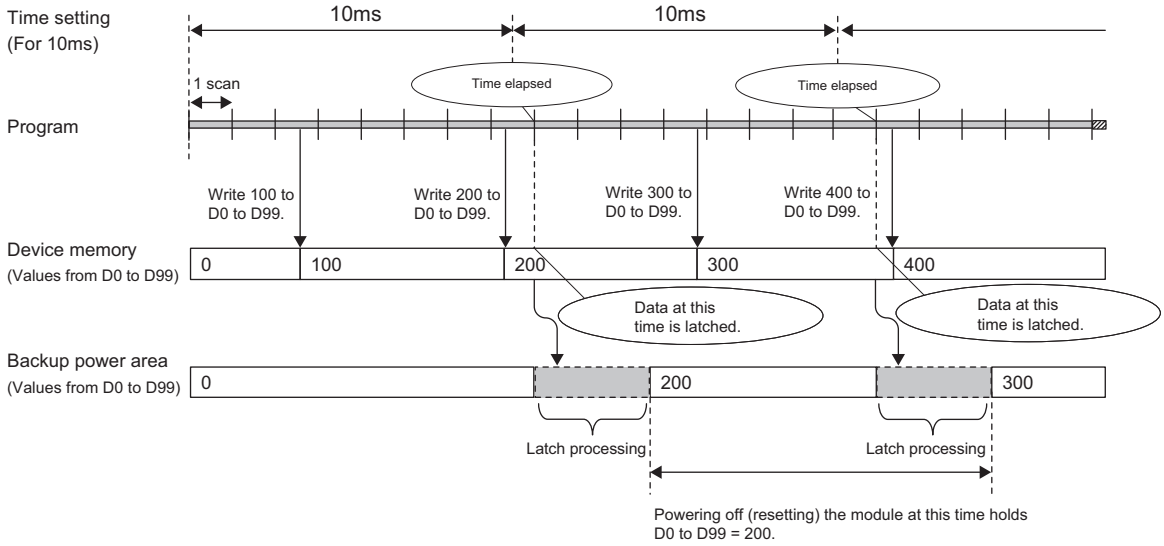
(1) D41088 or later areas of the data register (D) and areas of the devices (W, SW, and L) in disabled ranges can be latched in real time.

■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 526 Setting the latch interval)

• When set to "Time Setting"

Latch processing is started in the END processing executed after the set time.

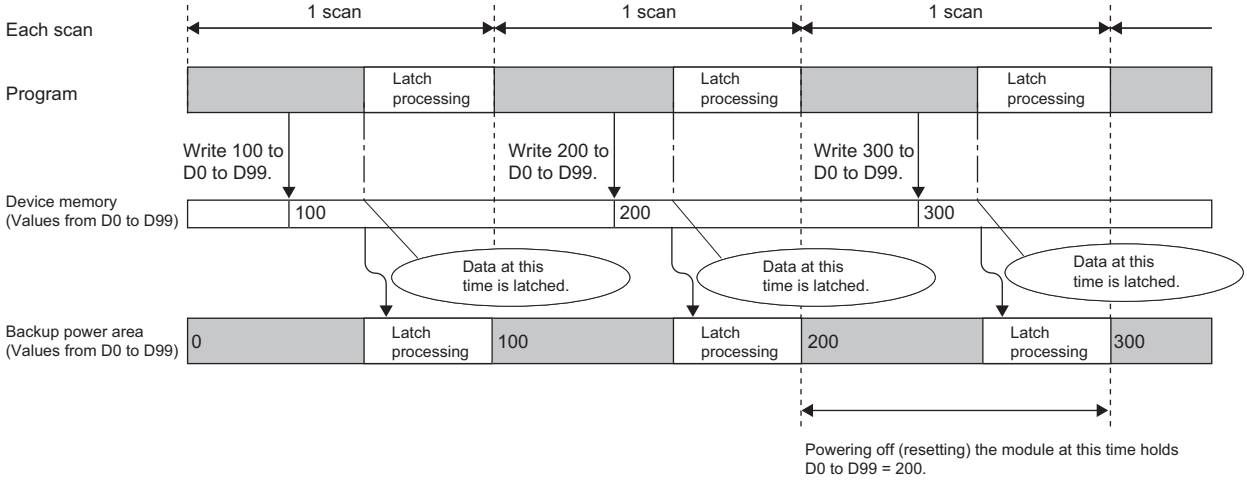


Point

- 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.

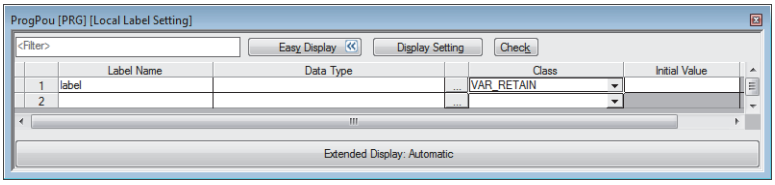


Setting latch on labels

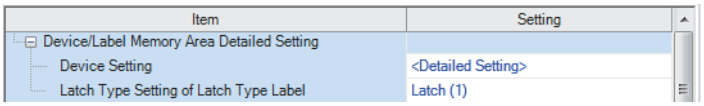
This section describes latch setting on labels.

Operating procedure

Label edit window



"Device/Label Memory Area Detailed Setting" window



1. In the label edit window, specify "RETAIN" for label attribute.

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.

[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 150 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.

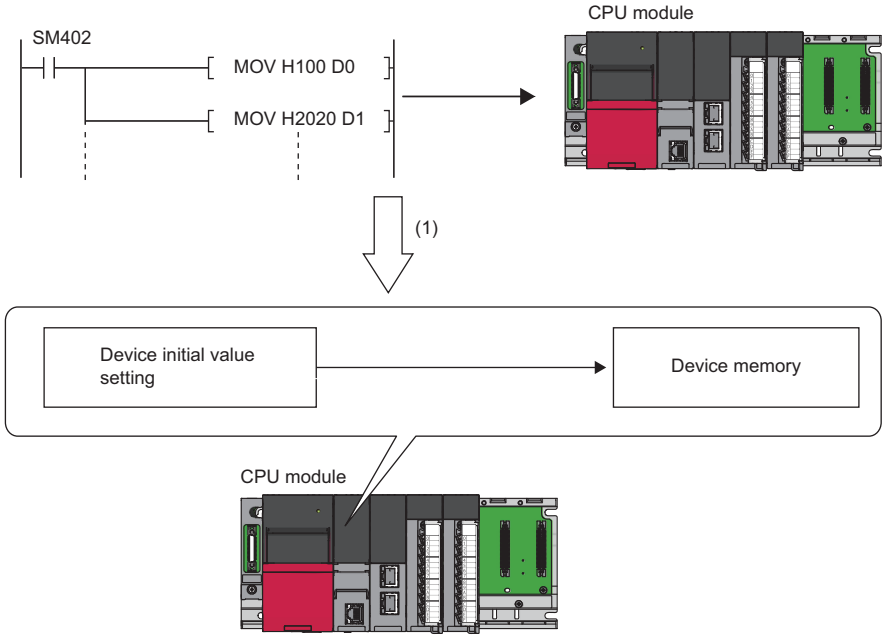
Precautions

This section describes the precautions when using the latch function.

- 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.

30 DEVICE/LABEL INITIAL VALUE SETTINGS

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.

30.1 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 531 Initial value setting)
5. Write the set initial device value file and the CPU parameters to the CPU module.

(📖 GX Works3 Operating Manual)

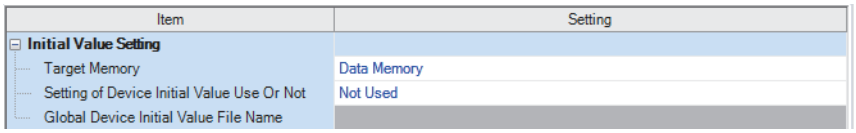
6. The data in the specified initial device value file is automatically set to the specified device or buffer memory of the intelligent function module when the CPU module is powered off and on, reset, or the status changes from STOP to RUN.

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.*3	• 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 If "Target Memory" is set to "Memory card", boot a target file from the SD memory card with the boot operation. (☞ Page 225 BOOT OPERATION)

*3 If "Setting of Device Initial Value Use Or Not" is set to "Use" and the specified initial global device value file does not exist, an error occurs. If the other file (initial local device value file, initial global label value file, or initial local label value file) does not exist, an initial value is not applied, but an error does not occur.

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 531 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.

30.2 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

30.3 Precautions

This section describes the precautions when using the initial device/label value setting.

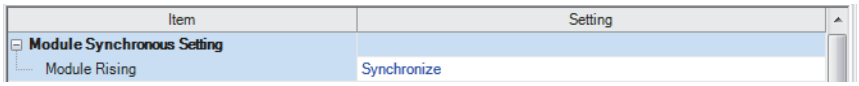
- When initial device values or initial label values are overlapped with the latch range, these initial values take precedence over the latch range. Therefore, even if a device or label is already latched, its values are changed to the specified initial values.
- The initial device value and initial 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, refer to the following.
Page 534 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



- In redundant mode, 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 powering off and on the CPU module of the control system or switching the operating status of the CPU module from STOP to RUN.

31 LABEL INITIALIZATION FUNCTION

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.

31.1 Initialization of Labels After Rebuilt All (Reassignment)

All the labels assigned to the label area 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

When using this function, check the versions of the CPU module and engineering tool used. (☞ Page 762 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
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

*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 536 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 type 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. However, when "Label Initialization Setting During Booting" is set to "Initialize", the latch type label can be initialized (by reflecting the initial value, or clearing the label value to zero if no initial value is set).
- 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.

31.2 Label Initial Value Reflection Setting

With the default settings, 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

When using this function, check the versions of the CPU module and engineering tool used. (☞ Page 762 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
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.

*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 537 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)

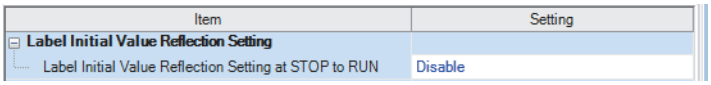
*1 Initial values are set when the CPU module is powered on or the operating status of the CPU module is switched from STOP to RUN for the first time after all data is rebuilt (reassigned) and written to the programmable controller.

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	Set 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".

32 CONSTANTS

This section describes constants.

32.1 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

32.2 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.

32.3 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 ±∞.

32.4 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)

32.5 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 541 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#-24d20h31m23s648.123ms (The time is treated as T#-24d20h31m23s648ms.)
  T#1.2345ms (The time is treated as T#1ms.)
- 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 8 TROUBLESHOOTING

This part consists of the following chapters.

33 TROUBLESHOOTING PROCEDURE

34 TROUBLESHOOTING BY SYMPTOM

35 MAINTENANCE AND INSPECTION OF A REDUNDANT SYSTEM

36 ERROR CODES

37 EVENT LIST

33 TROUBLESHOOTING PROCEDURE

This section describes errors that may occur during system operation, error causes, and actions to be taken.
For the troubleshooting specific to each module, refer to the manual for the module used.

Point

Saving the program and devices at the time of an error helps to analyze the error cause.
For reading data from the programmable controller, refer to the following.
GX Works3 Operating Manual

When the system has any trouble, perform troubleshooting in the following order.

1. Check the LED status of the power supply module.
(MELSEC iQ-R Module Configuration Manual)
2. Check the LED status of the CPU module. (Page 545 LED status of the CPU module)
3. Check the LED status of each I/O module and intelligent function module. (User's Manual (Application) for the module used)
4. Connect an engineering tool, and execute the system monitor function. The error module can be identified. (Page 546 System monitor)
5. Select the error module, and execute the module diagnostics function. The error cause and the action to be taken can be displayed. (Page 546 Module diagnostics)
6. If the error cause cannot be identified by the module diagnostics, check the executed operations and detected error logs on the event history window. (Page 547 Event history)
7. If the error cause cannot be identified in steps 1 to 6, perform troubleshooting by symptom. (User's Manual (Application) for the module used)

33.1 Troubleshooting with LED Indicators

Check the LED status of each module as primary diagnostics.

LED status of the CPU module

The following table lists the check points on the LED status of the CPU module.
The error status can be checked visually using the READY LED and ERROR LED.

LED indicator		Error status	Program execution status	Description	Action
READY	ERROR				
On	Off	No error	Continued	Normal operation	—
	On	Minor error		A minor error or warning has been detected. Program execution and data communications with network stations are continued.	Identify the error cause using the engineering tool, and take an action.
	Flashing	Moderate error	Stopped	A programming error, parameter setting error, or temporal noise has been detected. Program execution and data communications with network stations are stopped.	Identify the error cause using the engineering tool, and take an action.
Off	On/ flashing	Major error		A hardware failure has been detected. No operation cannot be performed.	Perform troubleshooting for the error module. If the problem still exists after troubleshooting, replace the error module. (☞ Page 548 When the READY LED of the CPU Module Turns Off)
	Off	Hardware failure		A hardware failure has been detected. No operation cannot be performed.	Check that the power is supplied to the system. If the power is supplied properly, the possible cause is a hardware failure. Perform troubleshooting for the error module. If the problem still exists after troubleshooting, replace the error module. (☞ Page 548 When the READY LED of the CPU Module Turns Off)
		No error		Power is not supplied or power went out.	—
Flashing	On	Minor error		A minor error has been detected during online module change.	Identify the error cause using the engineering tool, and take an action.
	Off	No error		• The READY LED is flashing every 400ms: Online module change is in progress. • The READY LED is flashing every 2s: Initial processing is being performed.	—
Flashing	Off	No error		The READY LED is flashing every 2s: Initial processing is being performed.	—

Point

The LED status can also be checked on the module diagnostics window using the engineering tool. (GX Works3 Operating Manual)

33.2 Troubleshooting Using the Engineering Tool

Check the error or history using the engineering tool, and identify the error cause. More detailed information on the error as well as the error cause and action to be taken can be checked by using the engineering tool.
The engineering tool has the following functions for troubleshooting.

Function	Description
System monitor	Displays the module configuration, and detailed information and error status of each module. (☞ Page 546 System monitor)
Module diagnostics	Diagnoses the module. (The current error and its details can be checked.) (☞ Page 546 Module diagnostics)
Event history	Displays the event information collected by the CPU module, such as errors occurred in each module, executed operations, and network errors. (☞ Page 547 Event history)

For details on each function, refer to the following.
📖 GX Works3 Operating Manual

System monitor

This function displays the module configuration, and detailed information and error status of each module.
For the error module identified, the module diagnostics function can be executed from this window.

[Diagnostics] ⇨ [System Monitor]

Point

In a redundant system, systems A and B are both displayed on the main base information part (on the left side of the window). The display of system A and system B can be switched in the main base information.

Module diagnostics

This function diagnoses the module. (The current error and its details can be checked.)
Information required for troubleshooting, such as the current error, details and cause of the error, and action to be taken, are displayed. The incorrect parameter setting and program error location can also be identified by selecting the error and clicking the [Error Jump] button.
On the [Module Information List] tab, the LED status and the switch status of the target module can be checked.

[Diagnostics] ⇨ [System Monitor] ⇨ Double-click the error module.

Restriction

In a multiple CPU system, when the I/O assignment setting in a system parameter differs from the configuration of the modules actually mounted, correct information may not be displayed on "Module Information List".

Event history

This function displays the event information, such as errors occurred in the module(s) mounted, executed operations, and network errors.
Since information collected before the CPU module is powered off or reset can also be displayed, the error cause can be identified based on the past operations and errors.
The displayed information can also be saved in CSV file format.

[Diagnostics] ⇨ [System Monitor] ⇨ [Event History] button

Point

Use this function for the following purposes:

- To check the error status and identify the cause of error occurred in machinery or equipment
- To check when and how the program and parameters of the programmable controller have been changed
- To check for unauthorized access

For details on this function and collected information, refer to the following.
☞ Page 211 Event History Function

34 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.

34.1 When the POWER LED of the Power Supply Module Turns Off

For troubleshooting, refer to the following:

MELSEC iQ-R Module Configuration Manual

34.2 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.

34.3 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 434 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. (☞ Page 78 Redundant Function Module) • Check that the length of each tracking cable is within the range of the specifications. (☞ Page 78 Redundant Function Module) • 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 434 Redundant Function Module Communication Test)

34.4 When the Specific Extension Base Unit Cannot Be Recognized

For troubleshooting, refer to the following:

MELSEC iQ-R Module Configuration Manual

34.5 When the Specific Q Series Extension Base Unit Cannot Be Recognized

For troubleshooting, refer to the following:

MELSEC iQ-R Module Configuration Manual

34.6 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

34.7 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.

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.

34.8 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 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.

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.

34.9 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.

34.10 When the Ethernet Function Cannot Be Used

For troubleshooting, refer to the following:

MELSEC iQ-R Ethernet User's Manual (Application)

34.11 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 398 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 432 Setting to wait cyclic data receive after system switching)

35 MAINTENANCE AND INSPECTION OF A REDUNDANT SYSTEM

This chapter describes the maintenance and inspection of a redundant system.

35.1 Module Replacement in a 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 change function 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 an 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 table lists modules and whether each module can be replaced.

○: Replaceable, ×: Not replaceable, —: Inapplicable

Replacement target	Availability							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 an extension base unit	
	While power is on	After powered off	While power is on	While power is on	After powered off	While power is on	While power is on	
CPU module	×*2	○	×	×*2	○	×	—	Page 554 Replacing the CPU module
Power supply module	×*2	○	×	×*2	○	×	×	Page 556 Replacing the power supply module
Redundant power supply module	○*1	○	○*1	○*1	○	○*1	○*1	Page 556 Replacing the redundant power supply module
Redundant function module	○	○	○	○	○	×	—	Page 557 Replacing the redundant function module
I/O module	○	○	○	○	○	×	○	Page 557 Replacing the I/O module
Intelligent function module	○*3*4	○	○*3	○*3*4	○	×	○*3	Page 558 Replacing the 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 in the CPU module of the control system. After replacing a module, clear the continuation error that has occurred in 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 those functions. (☞ Page 762 Added and Enhanced Functions)

■Replacement/addition of a base unit and extension cable

The following table lists base units and extension cables, and whether they can be replaced/added.

Replacement target		Availability	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 559 Replacing the 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 560 Replacing the extension cable between the 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 disconnected, system switching occurs, and the cable cannot be replaced.	Page 560 Replacing the extension cable between extension base units

Replacing the 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 and switch the system to the standby system.
The replacement procedure differs depending on whether the automatic memory copy function is enabled or disabled. (☞ Page 411 Auto memory copy)

When the auto memory copy setting is enabled

1. Checking the operation mode

Check the BACKUP LED of the redundant function module in the control system, and confirm that the CPU module of the control system is in backup mode.

- The BACKUP LED is on or flashing.

2. Checking the system of the replacement target CPU module

Check the CTRL LED and SBY LED of the redundant function module in the system to which the replacement target CPU module belongs, and confirm that the replacement target CPU module is set as the standby system.

- The CTRL LED is off and the SBY LED is on.

3. Powering 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. Replacing 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 762 Added and Enhanced Functions)

5. Changing the switch position 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. Powering on the standby system

Power on the standby system.

7. Executing the memory copy from the control system to the standby system

The memory copy function automatically starts. Upon completion of the memory copy function, the MEMORY COPY LED of the redundant function module turns off in both systems. Then, the CPU module in the standby system is automatically reset and restarted. (☞ Page 411 Auto memory copy)
The BACKUP LED of the redundant function module in the control system turns on. (The control system no longer waits for the start-up of the other system.)

When the auto memory copy setting is disabled

1. Checking the system of the replacement target CPU module

Check the CTRL LED and SBY LED of the redundant function module in the system to which the replacement target CPU module belongs, and confirm that the replacement target CPU module is set as the standby system.

- The CTRL LED is off and the SBY LED is on.

2. Powering 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 enters the wait state for the start-up of the other system.)

3. Replacing 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 762 Added and Enhanced Functions)

4. Changing the switch position 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. Powering on the standby system

Power on the standby system.

6. Executing the memory copy from the control system to the standby system

Execute the memory copy using the engineering tool or the special relay and special register. (☞ Page 412 Memory copy using the engineering tool, ☞ Page 413 Memory copy using the special relay and special register)

Upon completion of the memory copy function, the MEMORY COPY LED of the redundant function module in the control system turns off and that in the standby system turns on.

7. Restarting 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 no longer waits for the start-up of the other system.)

Replacing the power supply module

The following describes the procedure for replacing the power supply module on the main base unit 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 and switch the system to the standby system.

1. Checking the system of the replacement target module

Check the CTRL LED and SBY LED of the redundant function module in the system to which the replacement target module belongs, and confirm that the replacement target module is in the standby system.

- The CTRL LED is off and the SBY LED is on.

2. Powering 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 enters the wait state for the start-up of the other system.)

3. Replacing the power supply module in the standby system

Replace the power supply module in the standby system.

4. Powering on the standby system

Check that the cables are correctly connected to the power supply module and the power supply voltage is normal, and then 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 no longer waits for the start-up of the other system.)

Replacing the redundant power supply module

To replace a redundant power supply module while the redundant system is operating, if the power supply system is redundant, replace one power supply module at a time after 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 the redundant function module

The following describes the procedure for replacing the redundant function 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 and switch the system 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. Checking the system of the replacement target module

Check the CTRL LED and SBY LED of the redundant function module in the system to which the replacement target module belongs, and confirm that the replacement target module is in the standby system.

- The CTRL LED is off and the SBY LED is on.

2. Powering 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 enters the wait state for the start-up of the other system.)

3. Replacing the redundant function module in the standby system

Replace the redundant function module in the standby system.

4. Powering 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 no longer waits for the start-up of the other system.)

5. Checking the modules for errors

Check that no error has occurred in the CPU module or the redundant function module of the standby system. If the ERROR LED of either of the modules is on or flashing, identify the error and eliminate the error cause.

Replacing the I/O module

The following describes the procedure for replacing the I/O module on the main base unit 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 and switch the system 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. Checking the system of the replacement target module

Check the CTRL LED and SBY LED of the redundant function module in the system to which the replacement target module belongs, and confirm that the replacement target module is in the standby system.

- The CTRL LED is off and the SBY LED is on.

2. Powering 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 enters the wait state for the start-up of the other system.)

3. Replacing the I/O module in the standby system

Replace the I/O module in the standby system with a new I/O module whose model is the same as that of the I/O module in the control system.

4. Powering 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 no longer waits for the start-up of the other system.)

Replacing the intelligent function module

The following describes the procedure for replacing the intelligent function module on the main base unit 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 and switch the system 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. Checking the system of the replacement target module

Check the CTRL LED and SBY LED of the redundant function module in the system to which the replacement target module belongs, and confirm that the replacement target module is in the standby system.

- The CTRL LED is off and the SBY LED is on.

2. Powering 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 enters the wait state for the start-up of the other system.)

3. Replace the intelligent function module in the standby system

Disconnect the cables such as communication cables and wires connected to the intelligent function module in the standby system. Replace the intelligent function module with a new intelligent function module whose model is the same as that of the intelligent function module in the control system.

4. Powering 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 no longer waits for the start-up of the other system.)

5. Executing the module diagnostics for the standby system

Check the error information of the module diagnostics and confirm that no error has occurred. In addition, when replacing the network module, check the network status with the network diagnostics.

Procedure for remote head module replacement

For details on the procedure to replace the remote head modules, refer to the following.
📖 MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Application)

Replacing the main base unit

The following describes the procedure for replacing the main base unit 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 and switch the system to the standby system.

1. Checking the system of the replacement target base unit

Check the CTRL LED and SBY LED of the redundant function module in the system to which the replacement target base unit belongs, and confirm that the replacement target module is in the standby system.

- The CTRL LED is off and the SBY LED is on.

2. Powering 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 enters the wait state for the start-up of the other system.)

3. Replacing 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. Powering 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 no longer waits for the start-up of the other system.)

5. Checking the modules for errors

Check that no error has occurred in the CPU module or the redundant function module of the standby system. If the ERROR LED of either of the modules is on or flashing, identify the error and eliminate the error cause.

Replacing the extension cable between the main base unit and extension base unit

The following describes the procedure for replacing the extension cable between the main base unit of the standby system and the extension base unit after turning the power off while the redundant system is operating. When replacing the extension cable between the main base unit of the control system and the extension base unit, turn on SM1646 (System switching by a user) in advance and switch the system to the standby system.

1. Checking the system of the replacement target cable

Check the CTRL LED and SBY LED of the redundant function module in the system to which the replacement target cable belongs, and confirm that the replacement target module is in the standby system.

- The CTRL LED is off and the SBY LED is on.

2. Powering 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. Replacing 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. Powering 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. Checking the modules for errors

Check that no error has occurred in the CPU module or the redundant function module of the standby system. If the ERROR LED of either of the modules is on or flashing, identify the error and eliminate the error cause.

Replacing the extension cable between extension base units

Only the extension cable on the inactive side can be replaced. (☞ Page 561 Replacement/Addition of Extension Cables (Online))

If the extension cable on the active side is disconnected, system switching occurs, and the cable cannot be replaced.

35.2 Replacement/Addition of Extension Cables (Online)

Extension cables 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 operation mode of the redundant system.
- Extension cables can be replaced or added when the main base unit of the control system and the redundant extension base unit are connected properly.

Procedures

This section describes the procedures for replacing or adding extension cables.

■Replacing an extension cable

1. Check that the target extension cable is inactive based on SD1761 (extension cable route information) or the ACTIVE LED of the redundant extension base unit.
2. Disconnect the target extension cable. (CONNECT LED of the redundant extension base unit (on the side of the removed extension cable) turns off.)

Point

Due to the disconnection 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 after the replacement had been completed successfully, this extension cable error is not cleared and the ERROR LED remains on. After replacement, clear the error using the engineering tool.
The date and time when the extension cable was disconnected or failed can be checked in the event history.

3. Connect a new extension cable. (CONNECT LED of the redundant extension base unit (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 redundant extension base unit.

■Adding an extension cable

1. Check the place to add an extension cable.
2. Connect the extension cable to be added. (CONNECT LED of the redundant extension base unit (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 redundant extension base unit.

Point

The date and time when the extension cable was connected properly can be checked in the event history.

Precautions

This section describes the precautions for replacement/addition of extension cables (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 two main base units and a redundant extension base unit and for the two extension cables connecting two redundant extension base units, 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 disconnected 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.

36 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 of the engineering tool (GX Works3 Operating Manual)
- Special register (SD0 (Latest self-diagnostics error code), SD10 to SD25 (Self-diagnostic error number)) (Page 682 List of Special Register Areas)

This section describes errors that may occur in the CPU module and actions to be taken for the errors.

36.1 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
		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

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 682 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
	Frequency information	Information on the frequency, such as the number of writes to memory
	Time information	Information on time
	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
	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. 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.

36.2 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 203 CPU module operation upon error detection setting) Besides, if the stop error occurs when the multiple CPU system is configured, the stop error target CPU module (all CPU modules or only the corresponding CPU module) can be set in parameter. (☞ Page 341 Stop setting)

Continuation error

If a continuation error occurs, the CPU module continues its operation. (The operating status remains the same.)

36.3 How to Clear Errors

Continuation errors can be cleared. For the error clear method, refer to the following.

☞ Page 208 Error Clear

- CPU module: ☞ Page 208 Error Clear

When a redundant system 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 210 Clearing errors in the CPU module of the standby system from the CPU module of the control system)

36.4 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.	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.	System configuration information	Always
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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	• 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.	Continue	• Replace the CPU 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.	—	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
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.	—	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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.	—	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.	—	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.	—	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.	—	Always
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.	—	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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.	—	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
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
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			
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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.	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.	—	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.	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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2101H	Memory error	• An extended SRAM cassette is inserted or removed while the programmable controller is powered on.	Stop	• Do not insert or remove an extended SRAM cassette during operation. • Check the mounting status of the CPU module of the extended SRAM cassette. If the same error code is displayed again, the possible cause is a hardware failure of the extended SRAM cassette. Please consult your local Mitsubishi representative.	Drive/file information	Always
2102H	Memory error	• An error has been detected in the inserted extended SRAM cassette.	Stop	• Check the mounting status of the CPU module of the extended SRAM cassette. If the same error code is displayed again, the possible cause is a hardware failure of the extended SRAM cassette or CPU module. Please consult your local Mitsubishi representative.	Drive/file information	Always
2103H	Memory error	• An unsupported extended SRAM cassette is mounted.	Stop	• Replace the extended SRAM cassette with the one supported by the CPU module.	Drive/file information	Always
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
2160H	IP address duplication error	• Overlapping IP addresses have been detected.	Stop	• Check and correct the IP addresses.	—	Always
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. Please consult your local Mitsubishi representative.	Drive/file information	At power-on, at RESET, at STOP → RUN state
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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 device/ label memory in the CPU module or the memory card 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
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 data memory in the CPU module, memory card, or the 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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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 data memory in the CPU module, memory card, or the 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
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. • Although the module of local station is set as a synchronization target in "Select Inter-module Synchronization Target Module", "Inter-module Synchronization Master Setting" is not set.	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 "Inter-module Synchronization Master Setting". • 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 data memory in the CPU module, memory card, or the 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
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 CPU module model set to the project using the engineering tool differs from that of the CPU module actually mounted. • The operation set in the memory card parameters cannot be performed. (The boot function cannot be executed.)	Stop	• Correct the CPU module model set to the project in accordance with the CPU module actually mounted. • Delete the memory card parameter settings. • Remove the memory card so that the operation set in the memory card parameters will not be performed. (Do not execute the boot operation.)	Parameter 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
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
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 data memory in the CPU module or the 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
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 data memory in the CPU module or the 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 data memory in the CPU module or the intelligent function module connected. Please consult your local Mitsubishi representative.	System configuration 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
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 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
2262H	Parameter error (network)	• 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, "RJ71GP21-SX", "RJ71GP21S-SX", "RJ71LP21-25", "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	• 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 "RJ71GP21-SX(R)", "RJ71GP21S-SX(R)", "RJ71LP21-25(R)", "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
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 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
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 2282H	Parameter error (refresh)	• A device that cannot be used as a refresh device is specified. • The number of specified refresh points is invalid.	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
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 2301H	Security key authentication error	• The security key set to the program does not match the one registered to the CPU module (or cassette). • The security key is set to the program, but it is not 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
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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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.	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.	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.	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 • 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. ■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 • 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 (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 • 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. ■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	• The scan time exceeded the execution monitoring time set in parameter. • The initial (1st) scan time exceeded the execution monitoring time set in the CPU parameters. • The execution time of the fixed scan interrupt program exceeded the interrupt execution interval.	Stop	• 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 execution monitoring time set in parameter. (2) Change the execution monitoring time setting to an appropriate value. • 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.	Time information	Always
2501H		• The scan time exceeded the execution monitoring time set in parameter. • The 2nd or later scan time exceeded the execution monitoring time set in the CPU parameters. • The execution time of the fixed scan interrupt program exceeded the interrupt execution interval.				

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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, base unit, or module (I/O module or intelligent function module) connected. 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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
2820H	Device, label, or buffer memory specification error	• 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).	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. • Set the file register file in parameter, and access the file.	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.				

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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. • Clear the CPU built-in memory by selecting "Clear" to "Operation Setting at CPU Built-in Memory Boot" in the memory card parameters, and execute the boot function.	Drive/file information	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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. Please consult your local Mitsubishi representative.	Error location information	At power-on, at RESET, at STOP → RUN state, at instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3101H	Program error	• The program contains a dedicated SFC program instruction even although 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. 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
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. Please consult your local Mitsubishi representative.	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. Please consult your local Mitsubishi representative.	Error location 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
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. 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. 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						

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
3200H	Program execution error	• 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.)	Stop	• 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.	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
3202H	Program execution error	• 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".	Stop	• 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".	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	• 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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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
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).				

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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. 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

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		• Data that cannot be converted by using the data conversion instruction was input.				
3402H		• The operation was performed with the invalid data (-0, denormalized number, NaN (not a number), or ±∞).				
3403H		• An overflow has occurred during the operation.				
3404H		• A string that is not supported in the instruction was specified.				
3405H		• The input data was out of range.				
3406H		• 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.)				
3420H		• 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.			Error location information	
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
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
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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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/ continue	• 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 system switching execution, at tracking cable connection
3701H	System consistency check error (CPU module)	• The CPU module model names differ between the systems A and B. • The following settings of the CPU parameters differ between both systems, or an unsupported CPU module is mounted. (1) Link Direct Device Setting (2) Control System Start-up Setting at One System Error (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 following settings of the CPU parameters in both systems, or replace the CPU module with the one with a supported firmware version. Then, restart the system. (1) Link Direct Device Setting (2) Control System Start-up Setting at One System Error	—	At power-on, at RESET, at system switching execution, at tracking cable connection
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/ continue	• 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/ continue	• 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 the same file as that stored in the control system 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

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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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
3761H	Redundant system configuration error	• In a redundant system with redundant extension base unit, when "Control System Start-up Setting at One System Error" is set to "Enable" in the CPU parameters ("Redundant System Settings"), the following values differ between both systems. (1) Number of slots on the main base unit (2) Module mounting status of each slot of the main base unit, model name of the mounted module, or start I/O number (3) I/O assignment setting of the system parameters or module parameters (The error is detected in 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 of the main base unit, module model name, and start I/O number between the systems, and restart the CPU module. • Set the system parameters ("I/O Assignment Setting") and module parameters so that the parameters in both systems are the same.	—	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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 of the main base unit, model name of the mounted module, or start I/O number differs between system A and system B. • In a redundant system with redundant extension base unit, the system parameters ("I/O Assignment Setting") or the module parameters are different between system A and system B. (The error is detected in the control system or the standby system.)	Stop/ continue	• Use the main base units with the same number of slots in both systems. • Match the module mounting status of each slot of the main base unit, module model name, and start I/O number between the systems, and restart the CPU module. • Set the system parameters ("I/O Assignment Setting") and module parameters 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

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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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
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
37AAH	Parameter error (redundant function)	• When "Control System Start-up Setting at One System Error" is set to "Enable" in the CPU parameters ("Redundant System Settings"), the module parameters do not exist or a module which is not set in the system parameters ("I/O Assignment Setting") is mounted.	Stop	• Write the module parameters. • Re-set the system parameters ("I/O Assignment Setting") in accordance with the modules actually mounted.	Parameter information, system configuration information	At power-on, at RESET, at write
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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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

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 • 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. 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 • 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 extended SRAM cassette. 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 • 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. ■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

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3C20H	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
3C21H						At END instruction execution, at power-on, at RESET
3C22H						
3C2FH				■CPU module • 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. ■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. • 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 instruction execution
3C31H	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	Always
3C32H	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	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3E00H	Operation circuit error	■CPU module • An error has been detected in the CPU module. ■Redundant function module • An error has been detected in the redundant function module.	Stop	■CPU module • 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. ■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
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. • 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 END instruction execution
3E22H	Program execution error	• The FB/FUN program did not complete successfully.	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 instruction execution

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.

Error code	Error name	Error details and cause	Action
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.
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.
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.
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.

Error code	Error name	Error details and cause	Action
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 online change) or data logging 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 setting data of the online debug function (such as online change) and data logging 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.
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 is mounted. • 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 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.

Error code	Error name	Error details and cause	Action
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 program change again. • Write the program including the program restoration information to the programmable controller (including online change), or execute online change.
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).
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 program 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.

Error code	Error name	Error details and cause	Action
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	• 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	• A step No. that does not exist in the 0 to 511 range was specified.	• Check the setting and correct it.
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 program 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.

Error code	Error name	Error details and cause	Action
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 program 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.
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.

Error code	Error name	Error details and cause	Action
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.
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 setting and correct it.
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.

Error code	Error name	Error details and cause	Action
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.
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.

Error code	Error name	Error details and cause	Action
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.
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.

Error code	Error name	Error details and cause	Action
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.	• Execute the operation again after checking that the SD memory card has been inserted. • 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.
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.

Error code	Error name	Error details and cause	Action
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.
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.
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.

Error code	Error name	Error details and cause	Action
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 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.
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.

Error code	Error name	Error details and cause	Action
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.
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.

Error code	Error name	Error details and cause	Action
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.
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.
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.

Error code	Error name	Error details and cause	Action
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.
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
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.
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.
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.

Error code	Error name	Error details and cause	Action
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, redundant function module, or tracking cable. Please consult your local Mitsubishi representative.
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	Security function error	• Password authentication failed when access was required.	• Set a correct password and perform password authentication again 60 minutes later.
440DH			
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.
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.

Error code	Error name	Error details and cause	Action
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 inserted in the CPU module.	• Check that an extended SRAM 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).	• 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).	• Clear the security key set to the file, and then execute the online change function.
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.
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 in the simple CPU communication function, an error occurred in another simple CPU communication setting No. for which the same communication destination is specified.	• 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.
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.
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. • The 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.
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 change · File transfer function (FTP server)	• Execute the function again after a while.

Error code	Error name	Error details and cause	Action
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.
4C1CH	CPU module data backup/ restoration function error	• An SD memory card is not inserted. • The 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.

Error code	Error name	Error details and cause	Action
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.
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.

Error code	Error name	Error details and cause	Action
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.
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.

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 newly-mounted to the system 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 CPU module and increase the frequency of transient processing. 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	Module error	• An error has been detected in the redundant function module.	• Please consult your local Mitsubishi representative.
6F23H			
6F24H			
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	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.
6F4AH			
6F4BH			
6F4CH			
6F4DH			
6F4EH			
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 detected errors related to the Ethernet-equipped module.

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)		

37 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 211 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.

37.1 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 3	Details of a detected event

Detailed information

The following table lists the details of information displayed in the detailed information 1 to 3.

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
	Event history file information	Information on the event history file
	Module information	Information on the target module (I/O number)
	Firmware update information	Information (firmware versions before and after update) on the firmware update using an SD memory card
	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
	Control system start-up information at one system error	Information on the control system start-up when an error occurs in one system

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
	File access control information	Information on file access control: access level
	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)
	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	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

37.2 Event List

The following table lists events related to the CPU module.

Event code	Event type	Event category	Detected event	Description	Detailed information		
					1	2	3
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		
00401			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.	—	—	
00460			Label initialization	In the label initialization and boot functions 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		

Event code	Event type	Event category	Detected event	Description	Detailed information		
					1	2	3
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			Socket communication send error	Sending a message over socket communication failed.		—	
00906			Alive check error	The alive status of an external device could not be checked within the period specified by the response monitoring timer.			
00907			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.			
00908			IP composition timeout error	An IP composition timeout error occurred. (The specified period of time expired without receiving all of the divided data.)			
00909			TCP-specified port number error	A port number being used for opened connection was set (for TCP/IP).			
0090A			UDP-specified port number error	A port number being used for opened connection was set (for UDP/IP).			

Event code	Event type	Event category	Detected event	Description	Detailed information		
					1	2	3
00A00	System	Warning	Error detection in the other system	An error was detected in the other system.	—	Error description (the other system)	—
00C02			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	
00C27			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.			
00F03			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.	—		
00F10			Reset with the control system start-up at one system error	When "Control System Start-up Setting at One System Error" was set to "Enable", a stop error was detected in both systems and the systems were reset.	—		
00F11			Failure of the reset with the control system start-up at one system error	When "Control System Start-up Setting at One System Error" was set to "Enable", a stop error was detected in both systems but the reset of the systems failed.	Control system start-up information at one system error		
1000 and after		Error	When a self-diagnostic error occurs, the error is stored as an event.				

Event code	Event type	Event category	Detected event	Description	Detailed information		
					1	2	3
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.			

Event code	Event type	Event category	Detected event	Description	Detailed information		
					1	2	3
20100	Operation	Info	Error clear	The error was cleared.	Operation source information	Operation target information	—
20200			Event history clear	The event history 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			Firmware update successful via SD memory card	CPU module firmware update using the SD memory card was performed and completed successfully.	Firmware update information		
20401			Firmware update failed via SD memory card	CPU module 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.			
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	—	
24000			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			Module communication test	A module communication test was executed.		—	—

Event code	Event type	Event category	Detected event	Description	Detailed information		
					1	2	3
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	
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)

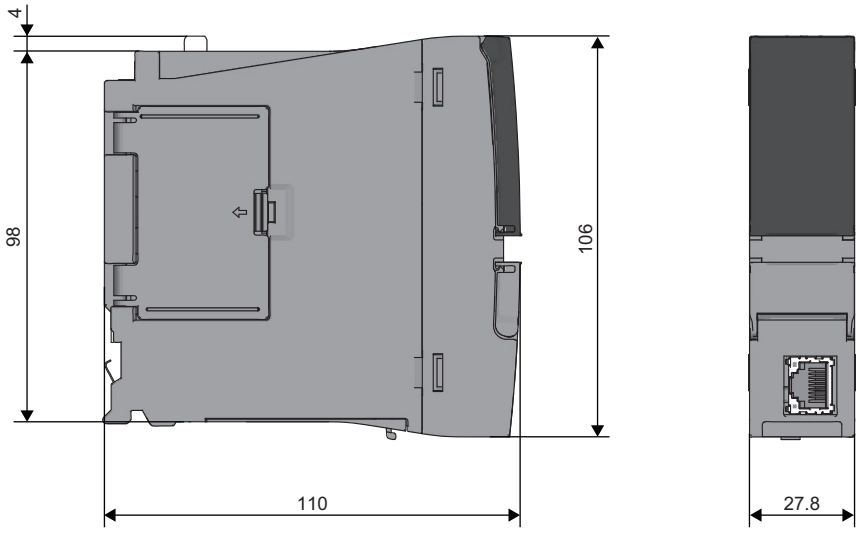
*2 The engineering tool displays "Start-up information" in detailed information 1 only when the CPU module with firmware version "06" or later is used.

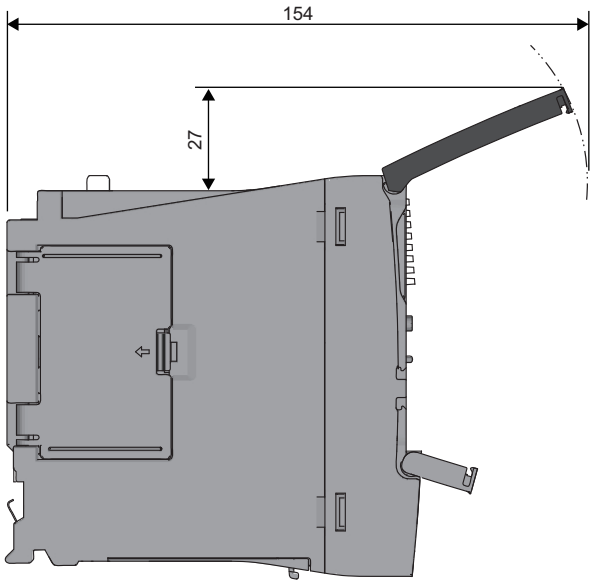
*3 Detailed information 2 is stored only when detailed information 1 is CPU module (3E00H).

APPENDICES

Appendix 1 External Dimensions

CPU module

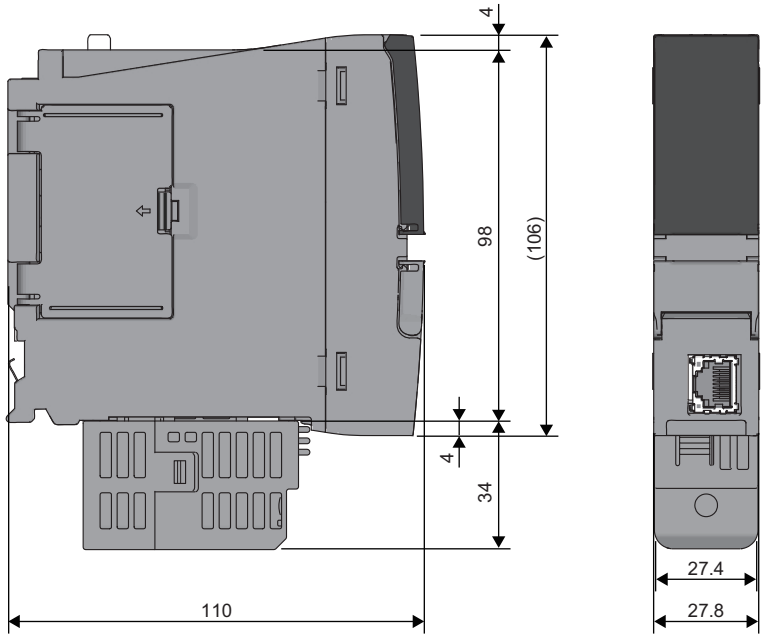




(Unit: mm)

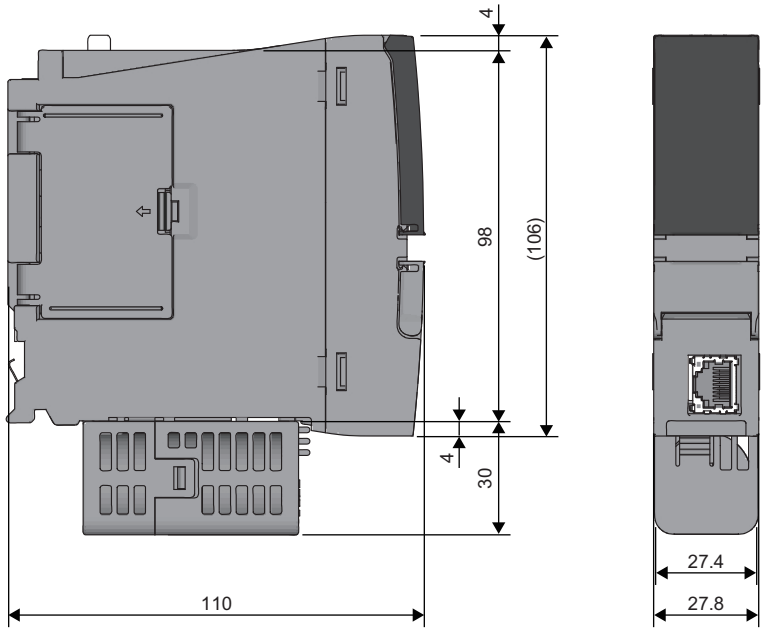
A

• With the Q7BATN-SET



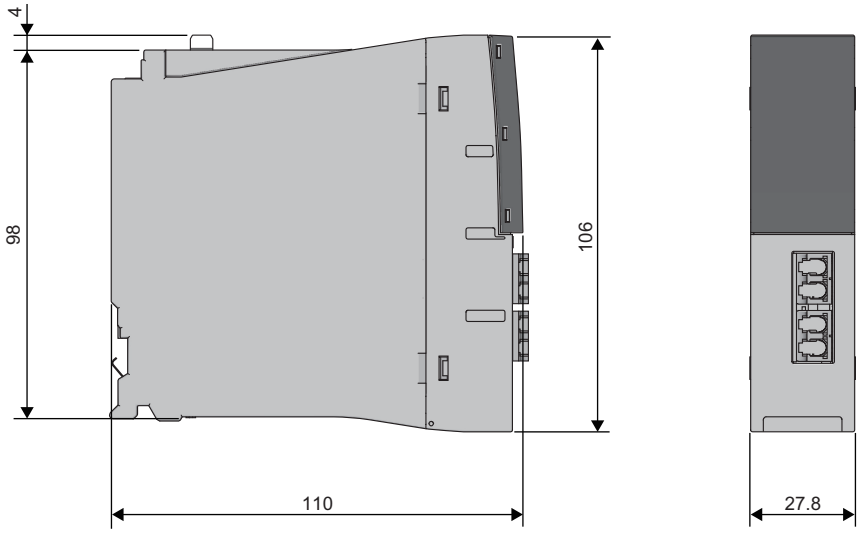
(Unit: mm)

• With the Q7BAT-SET



(Unit: mm)

Redundant function module



(Unit: mm)

A

Appendix 2 Compliance with EMC and Low Voltage Directives

Method of ensuring compliance

To ensure that Mitsubishi Electric programmable controllers maintain the EMC and Low Voltage Directives or other regulations when incorporated into other machinery or equipment, certain measures may be necessary. Please refer to one of the following manuals.

- MELSEC iQ-R Module Configuration Manual (SH-081262ENG)
- Safety Guidelines (IB-0800525)

Certification marks on the side of the programmable controller indicate compliance with the relevant regulations.

Additional measures

To ensure that this product maintains the EMC and Low Voltage Directives or other regulations, please refer to the following.

- MELSEC iQ-R Module Configuration Manual (SH-081262ENG)
- Safety Guidelines (IB-0800525)

Appendix 3 Functional Availability by CPU Module Model

The following table lists the availability of functions by CPU module model.

Rn: RnCPU, RnEN: RnENCPU, RnP(P): Process CPU (process mode), RnP(R): Process CPU (redundant mode), RnPSF: SIL2 Process CPU, RnSF: Safety CPU

○: Supported, ×: Not supported

Function		Availability*1					
		Rn	RnEN	RnP(P)	RnP(R)	RnPSF	RnSF
Constant scan		○	○	○	○	○	○
Device/label access service processing setting		○	○	○	○	○	○
Device/label access service processing constant wait function		×	×	○	○	×	×
Interrupt function	Multiple interrupt function	○	○	○	○	○	○
Output mode setting when the status changed from STOP to RUN		○	○	○	○	○	○
Device/label memory area setting		○	○	○	○	○	○
Internal buffer capacity setting		○*3	○	○	○	×	○
SD memory card forced disable		○*3	○	○	○	○	○
Clock function		○	○	○	○	○	○
Writing data to the CPU module	Writing data to the programmable controller	○	○	○	○	○	○
	Online change (ladder block)	○	○	○	○	○	○
	Online change (files)	○	○	○	○	○	○
RAS function	Scan monitoring function	○	○	○	○	○	○
	Self-diagnostics function	○	○	○	○	○	○
	FB hierarchy information	○	○	○	○	×	×
	Error clear	○	○	○	○	○	○
	Event history function	○	○	○	○	○	○
	Online module change	×	×	○	○	○	×
	Program cache memory auto recovery function	○	○	○	○	○	○
Remote operation		○	○	○	○	○	○
Boot operation		○*3	○	○	○	×	×
Monitor function	Circuit monitor	○	○	○	○	○	○
	Device/buffer memory batch monitor	○	○	○	○	○	○
	Watch	○	○	○	○	○	○
	Program monitor list	○	○	○	○	○	○
	Interrupt program monitor list	○	○	○	○	○	○
	Real-time monitor function	○	○	×	×	×	○
	Scan time measurement	○	○	×	×	×	×
	Scan time clear	○	○	×	×	×	○
	Specified program monitor	○	○	×	×	×	×
Test function	External input/output forced on/off function	○	○	○	○	×	○
	Device test with execution condition	○	○	○	○	×	○
Data logging function		○*3	○	○	○	×	○
Recording function		○*2	○	×	×	×	○
Debug function	Memory dump function	○*3	○	×	×	×	○
SFC function		○	○	○	○	×	○
Database function	Database access instruction	○*2	○	×	×	×	×
	CPU module database access (from external device) function	○*2	○	×	×	×	×
PID control function		○	○	○	○	○	○
Process control function		×	×	○	○	○	×
CPU module data backup/restoration function		○*2	○	○	○	×	○

A

Function		Availability[*1]					
		Rn	RnEN	RnP(P)	RnP(R)	RnPSF	RnSF
Multiple CPU system function	Out-of-group I/O fetch	○	○	○	×	×	○
	Multiple CPU synchronized startup	○	○	○	×	×	○
	Data communications between CPU modules	○	○	○	×	×	○
	Multiple CPU synchronous interrupt	○	○	○	×	×	○
Security function	User authentication function	×	×	×	×	○	○
	Block password function	○	○	○	○	○	○
	Security key authentication function	○	○	○	○	○	○
	File password function	○	○	○	○	○	○
	IP filter function	○	○	○	○	○	○
	Remote password function	○	○	○	○	○	○
	Write-protect function for device data (from outside the CPU module)	○	○	×	×	×	×
Sequence scan synchronization sampling function		○	○	×	×	×	○
Label initialization function	Label initialization after converting all programs (reassignment)	×	×	○	○	○	×
	Label initialization with label initial values	×	×	○	○	○	×
Routing setting		○	○	○	○	○	○
Label access setting from external device		○[*2]	○	○	○	○	○
Latch function	Latch with a battery	○	○	○	○	○	○
	Latch with a battery-less option cassette	○[*2]	○	×	×	×	×
Initial device/label value setting		○	○	○	○	○	○
Redundant function	Operation mode change	×	×	×	○	×	×
	System switching	×	×	×	○	○	×
	Tracking transfer	×	×	×	○	○	×
	Memory copy from control system to standby system	×	×	×	○	○	×
	System consistency check	×	×	×	○	○	×
	Program execution in both systems	×	×	×	○	○	×
	Redundant system operation setting	×	×	×	○	○	×
	Redundant function module communication test	×	×	×	○	○	×
	Setting for redundant system with redundant extension base unit	×	×	×	○	×	×
	Automatic standby system recovery	×	×	×	○	×	×
	Replacement/addition of an extension cable (online)	×	×	×	○	×	×
Safety operation mode		×	×	×	×	○	○
Continuous RUN prevention in TEST MODE		×	×	×	×	×	○
Safety diagnostic function		×	×	×	×	○	○
Safety data identify check		×	×	×	×	○	○
Safety communication function		×	×	×	×	○	○
Ethernet function		📖 MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)					
SLMP communication function		○	○	○	○	○	○
Inter-module synchronization function		○	○	○	×	×	○
CC-Link IE Field Network Basic function		📖 CC-Link IE Field Network Basic Reference Manual					
iQ Sensor Solution function		📖 iQ Sensor Solution Reference Manual					
Firmware update function	Update using the engineering tool	○	○	○	○	×	×
	Update using an SD memory card	○[*3]	○	○	○	×	×

*1 Some functions have restrictions on the firmware version and the production information of the CPU module used or the version of the engineering tool used. Refer to the following.
☞ Page 762 Added and Enhanced Functions

*2 The R00CPU, R01CPU, and R02CPU does not support the use of these functions.

*3 The R00CPU does not support the use of these functions.

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)

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

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)
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)
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)
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)
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)
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)
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)
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)
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)

No.	Name	Data stored	Details	Set by (setting timing)
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)
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)
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)
SM112	Detailed information 2: Flag in use			
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)
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)

No.	Name	Data stored	Details	Set by (setting timing)
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)
SM153	Momentary power failure detection (power supply module 2)			S (Status change)
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)

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)
SM203	STOP contact	Off: Other than STOP state On: STOP state	This relay is on in STOP state.	S (Status change)
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)
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)
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)
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
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)
SM221	CPU No.2 preparation completed			
SM222	CPU No.3 preparation completed			
SM223	CPU No.4 preparation completed			
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)
SM231	No.2 CPU error flag			
SM232	No.3 CPU error flag			
SM233	No.4 CPU error flag			
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)
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)
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)
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)

No.	Name	Data stored	Details	Set by (setting timing)
SM315	Service processing constant wait setting flag*1	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)
SM384	System operation setting request*1	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)
SM385	System operation setting error*1	Off: No error On: Error	• This relay turns on when writing into the setting storage area (system area) fails.	S (Status change)
SM386	Program restoration information write status LED control setting mode*1	Off: LED flashing On: Without LED flashing	• This relay indicates the LED control setting status of program restoration information.	S (Initial)
SM387	Program restoration information write status*1	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)
SM388	File batch online change operation setting status*1	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)

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

SFC information

The following is a list of special relay areas relating to SFC information.

No.	Name	Data stored	Details	Set by (setting timing)
SM320	Presence/absence of SFC program*1	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)
SM321	Start/stop SFC program*1	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
SM322	SFC program start status*1	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
SM323	Presence/absence of continuous transition for entire block*1	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
SM324	Continuous transition prevention flag*1	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)
SM325	Output mode at block stop*1	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
SM326	SFC device/label clear mode*1	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. • Only labels are cleared at program writing to the programmable controller after data are rebuilt even though this relay is on.	U
SM327	Output mode at execution of the end step*1	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
SM328	Clear processing mode when the sequence reaches the end step*1	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
SM329	Online change (SFC block) status flag*1	Off: Not being executed On: Being executed	This relay turns on while the online change (SFC block) is being executed.	S (Status change)

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

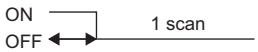
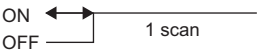
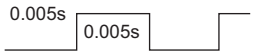
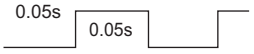
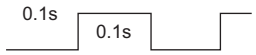
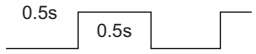
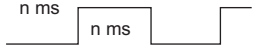
Point

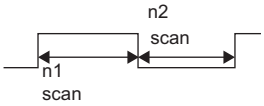
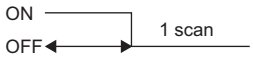
For details on the SFC program, refer to the following.

MELSEC iQ-R Programming Manual (Program Design)

System clock

The following is the list of special relay areas relating to the system clock.

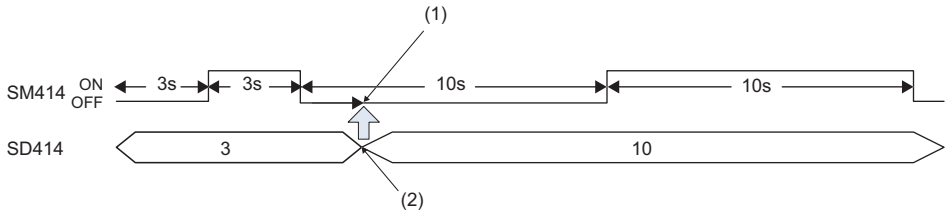
No.	Name	Data stored	Details	Set by (setting timing)
SM400	Always On		Always On	S (Power-on to RUN/ STOP to RUN/every END)
SM401	Always Off		Always Off	S (Power-on to RUN/ STOP to RUN/every END)
SM402	After RUN, ON for 1 scan only		• This relay is on during only one scan after RUN mode starts. • This relay is enabled only for the scan execution type program.	S (Status change/every END)
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)
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)
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)
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)
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)
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)
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)
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)

No.	Name	Data stored	Details	Set by (setting timing)
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)
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		• 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)
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)

*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:

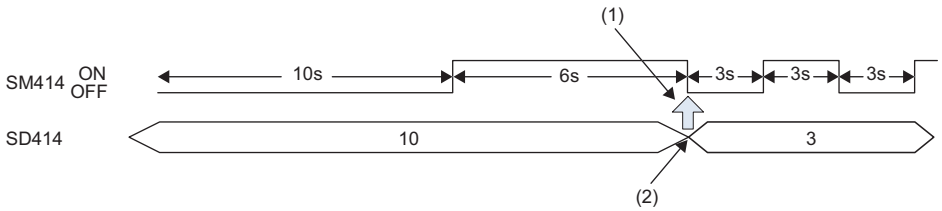


(1) The elapsed time after the ON/OFF state of SM414 changes remains.

(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

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)
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)
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)
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)
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)

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)
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)
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)
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)
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)
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
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)
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)
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)
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)
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)
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)
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)
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)
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)

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)
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)
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)
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
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
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
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)
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
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
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 during access to the SD memory card and data memory.	S (Status change)
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
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

No.	Name	Data stored	Details	Set by (setting timing)
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
SM775	Selection of refresh processing during the COM instruction execution	Off: Executes all the refresh processing 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
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
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.	U
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
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
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)
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)
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)
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)
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
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

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)
SM922	Firmware update completion with/without an error[*1]	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)
SM940	Operation setting of the device test with execution conditions[*1]	Off: Registration disabled On: Registration not disabled	Set the operation when files relevant to the device test with execution conditions are changed.	U
SM953	CPU module data backup error check flag[*1]	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)
SM959	CPU module data restoration error check flag[*1]	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)
SM960	Upper limit setting flag for the number of CPU module backup data[*1]	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
SM961	Automatic backup retry failure flag[*1]	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)

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

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)
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)
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)
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)
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)
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)
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)
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)
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)
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)

No.	Name	Data stored	Details	Set by (setting timing)
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)
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 with the data logging.	S (Status change)
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)
SM1220 to SM1228	Data logging setting No.2	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1230 to SM1238	Data logging setting No.3	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1240 to SM1248	Data logging setting No.4	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1250 to SM1258	Data logging setting No.5	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1260 to SM1268	Data logging setting No.6	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1270 to SM1278	Data logging setting No.7	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1280 to SM1288	Data logging setting No.8	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1290 to SM1298	Data logging setting No.9	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1300 to SM1308	Data logging setting No.10	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
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

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)
SM1350	CPU module data backup status flag*1	Off: Not being executed On: Being executed	This relay turns on during the backup of the CPU module.	S (Status change)
SM1351	CPU module data backup execution request*1	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
SM1356	Retry status flag for CPU module data automatic backup*1	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)

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

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)
SM1392	FTP client connection status*1*2	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)

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

*2 The CPU module with firmware version "13" or later supports this special relay area.

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)
SM1464	Event history logging restriction status*1	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)
SM1466	Event history logging restriction status of the CPU module (minor error)*1	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)
SM1467	Event history logging restriction status of the CPU module (information, warning)*1	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)

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

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)
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
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)
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) 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
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)
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)
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)

*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.)

Online module change function

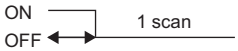
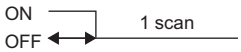
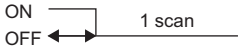
The following is the list of special relay areas relating to the online module change function.

No.	Name	Data stored	Details	Set by (setting timing)
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)
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)
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)
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)
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)
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)
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)
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)
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)
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)
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)
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)
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)

No.	Name	Data stored	Details	Set by (setting timing)
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)
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)

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)
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)
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)
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)
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)
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)
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)
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)
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)
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)
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)
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
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

No.	Name	Data stored	Details	Set by (setting timing)
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)
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
SM1656	Auto memory copy	Off: Disabled On: Enabled	This relay turns on when the automatic memory copy is set to enable.	S (Initial)
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)
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)
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)
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)
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)
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)
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)
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)
SM1756	Wait timeout for receiving cyclic data after system switching[*1]	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)

No.	Name	Data stored	Details	Set by (setting timing)
SM1762	Operation setting for access from the standby system to the extension base unit*1	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

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

*2 ☞ Page 746 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".)

Module error information collection function

The following table lists the special relay areas for the module error information collection function.

No.	Name	Data stored	Details	Set by (setting timing)
SM3696	Module error information update setting*1	Off: Do not update the information. On: Update the information (default).	Whether to update the error information (SD3696 to SD3825) of the mounted module is specified. • On: The error information is updated every 5 seconds during END processing. The error information is updated regardless of the CPU operating status. (The information is updated even when a stop error occurs.) • Off: The error information is not updated for each module. • When the status is changed from on to off, the error information (SD3696 to SD3825) is not cleared to zero.	U

*1 There are restrictions on the firmware version of the CPU module. (☞ Page 762 Added and Enhanced Functions)

A

Appendix 5 List of Special Register Areas

The following table lists items in the list.

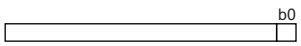
Item	Description
No.	Special register number
Name	Special register name
Data stored	Data stored in the special register
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (program, 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)

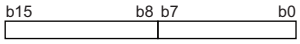
Point

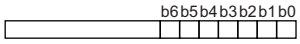
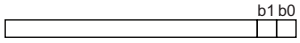
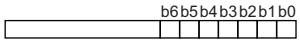
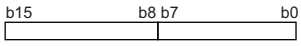
Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

Diagnostic information

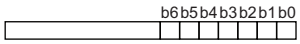
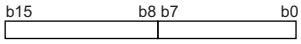
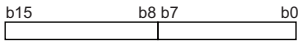
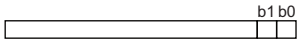
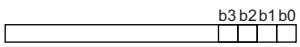
The following is the list of special register areas relating to the diagnostic information.

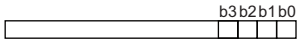
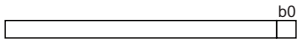
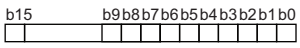
No.	Name	Data stored	Details	Set by (setting timing)
SD0	Latest self diagnostics error code	Latest self diagnostics error code	Error codes are stored in a hexadecimal value when the diagnostics detects an error.	S (Error)
SD1	Latest self-diagnostics error time	Latest self-diagnostics error time	The year value (four digits) of the date/time when SD0 data was updated is stored as a BIN code.	S (Error)
SD2			The month value of the date/time when SD0 data was updated is stored as a BIN code.	
SD3			The day value of the date/time when SD0 data was updated is stored as a BIN code.	
SD4			The hour value of the date/time when SD0 data was updated is stored as a BIN code.	
SD5			The minute value of the date/time when SD0 data was updated is stored as a BIN code.	
SD6			The second value of the date/time when SD0 data was updated is stored as a BIN code.	
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)	
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)
SD11		Self-diagnostic error number 2		
SD12		Self-diagnostic error number 3		
SD13		Self-diagnostic error number 4		
SD14		Self-diagnostic error number 5		
SD15		Self-diagnostic error number 6		
SD16		Self-diagnostic error number 7		
SD17		Self-diagnostic error number 8		
SD18		Self-diagnostic error number 9		
SD19		Self-diagnostic error number 10		
SD20		Self-diagnostic error number 11		
SD21		Self-diagnostic error number 12		
SD22		Self-diagnostic error number 13		
SD23		Self-diagnostic error number 14		
SD24		Self-diagnostic error number 15		
SD25		Self-diagnostic error number 16		
SD49	Error detection invalidation setting[*1]	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: Built-in Ethernet port communication error	U
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)

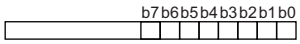
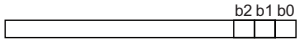
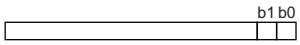
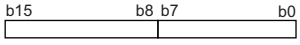
No.	Name	Data stored	Details	Set by (setting timing)
SD60	Number of module with blown fuse	Number of module with blown fuse	• The lowest I/O number of module in which a fuse blew is stored. • The fuse blown state check is also done for output modules on the remote I/O station. • In a redundant system 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, the CPU module of the system where the error is cleared clears the value.	S (Error)
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, the CPU module of the system where the error is cleared clears the value.	S (Error)
SD62	Annunciator number	Annunciator number	The annunciator number which is detected first is stored.	S (Instruction execution)
SD63	Number of annunciators	Number of annunciators	The number of detected annunciator is stored.	S (Instruction execution)
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 465 Annunciator (F))	S (Instruction execution)
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 number and file name 4: Parameter information 5: System configuration information 6: Number of times information 7: Time information 24: Failure information 27: System switching information 37: Data type (tracking transfer) information 38: Tracking transfer trigger information 46: CPU module data backup/restoration folder information 58: Extension cable information	S (Error)

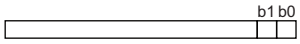
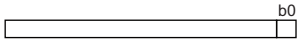
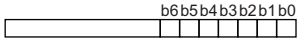
No.	Name	Data stored	Details	Set by (setting timing)
SD81 to SD111	Detailed information 1	Detailed information 1	• Detailed information 1 corresponding to the error code (SD0) is stored. • The type of the detailed information 1 can be obtained using SD80 (the value of the "Detailed information 1 information 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 number and file name ■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)
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 • 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)

A

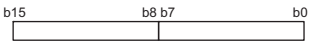
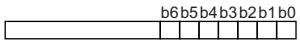
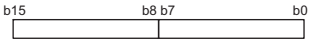
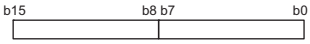
No.	Name	Data stored	Details	Set by (setting timing)
SD81 to SD111	Detailed information 1	Detailed information 1	(5) System configuration information ■SD81: With or without specification  b0: I/O number b1: Slot number b2: Base number b3: Power supply number b4: CPU number b5: Network number b6: Station number ■SD82: I/O number ■SD83  • 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  • 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 0 to 120 (0 is stored for the master station.) (6) Number of times information When there is no specification, 0 is set to each SD. ■SD81: With or without specification  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  b0: Time (value set) (ms) b1: Time (value set) (μs) b2: Time (actual measurement value) (ms) b3: Time (actual measurement value) (μs)	S (Error)

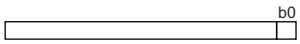
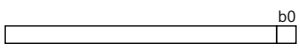
No.	Name	Data stored	Details	Set by (setting timing)
SD81 to SD111	Detailed information 1	Detailed information 1	■SD82: Time (value set) (ms) ■SD83: Time (value set) (μs) ■SD84: Time (actual measurement value) (ms) ■SD85: Time (actual measurement value) (μs) (24) Failure information Failure information is system information. (27) System switching information ■SD81: With or without specification  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 ■SD85: Control system/standby system transition 1: Control system → Standby system 2: Standby system → Control system (37) Data type (tracking transfer) information ■SD81: With or without specification  b0: Data type ■SD82: Data type  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)

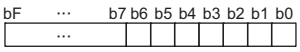
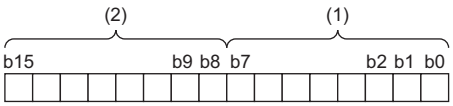
No.	Name	Data stored	Details	Set by (setting timing)
SD81 to SD111	Detailed information 1	Detailed information 1	(38) Tracking transfer trigger information ■SD81: With or without specification  b0: Block No.1 to 8 b1: Block No.9 to 16 b2: Block No.17 to 24 b3: Block No.25 to 32 b4: Block No.33 to 40 b5: Block No.41 to 48 b6: Block No.49 to 56 b7: Block No.57 to 64 ■SD82 to SD85: Block number b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SD82: 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 SD83: 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 SD84: 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 SD85: 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 (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.) (58) Extension cable information ■SD81: With or without specification  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)
SD112	Detailed information 2 information category	Detailed information 2 information category code	Detailed information 2 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 • 2: Drive number and file name • 3: Annunciator number • 4: Parameter information • 5: System configuration information • 25: Process control instruction processing information • 28: Program error information	S (Error)

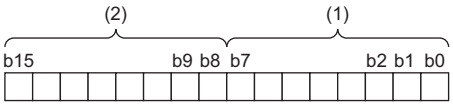
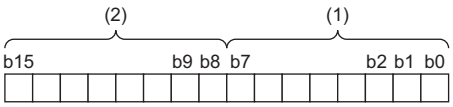
No.	Name	Data stored	Details	Set by (setting timing)
SD113 to SD143	Detailed information 2	Detailed information 2	• Detailed information 2 corresponding to the error code (SD0) is stored. • The type of information can be checked in SD112. (The value of the "Detailed information 2 information category code" stored in SD112 corresponds to the following (2) to (5), (25), (28), (29) and (57).) (2) Drive number and file name ■SD113: With or without specification  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: Annunciator number ■SD114: Annunciator number (4) Parameter information ■SD113: 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	S (Error)

A

No.	Name	Data stored	Details	Set by (setting timing)
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 • 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 0 to 120 (0 is stored for the master station.) ■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 0 to 120 (0 is stored for the master station.)	S (Error)

No.	Name	Data stored	Details	Set by (setting timing)
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)

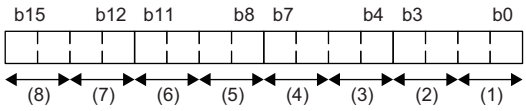
No.	Name	Data stored	Details	Set by (setting timing)
SD113 to SD143	Detailed information 2	Detailed information 2	(29) Other station error information (CC-Link IE Field) ■SD113: With or without specification  b0: Error classification b1: Error code b2: Date (yyyymmdd) b3: Time (hhmmss) b4: Day of the 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	S (Error)
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.  (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)

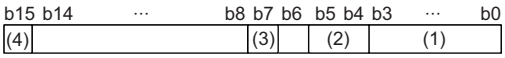
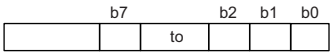
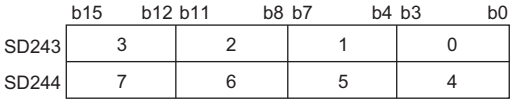
No.	Name	Data stored	Details	Set by (setting timing)
SD151	Power supply failure detection status	Power supply failure detection status (bit pattern) 0: Not detected/power-off/no power supply module 1: Detected	• For the redundant power supply base unit or redundant extension base unit, failure detection 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.  (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)
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)
SD153	Momentary power failure detection count (power supply module 2)	Momentary power failure detection count for power supply module 2		S (Status change)
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.  (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)

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

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)
SD160	Firmware version*2	Firmware version	The firmware version is stored.	S (Initial)
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)
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.  (1) READY (2) ERROR (3) PROGRAM RUN (4) USER (5) BATTERY (6) CARD READY (7) CARD ACCESS (8) FUNCTION	S (Status change)
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)
SD210	Clock data	Clock data (year)	The year value (four digits) of the clock data is stored as a BIN code.	S/U (Request)
SD211		Clock data (month)	The month value of the clock data is stored as a BIN code.	S/U (Request)
SD212		Clock data (day)	The day value of the clock data is stored as a BIN code.	S/U (Request)
SD213		Clock data (hour)	The hour value of the clock data is stored as a BIN code.	S/U (Request)
SD214		Clock data (minute)	The minute value of the clock data is stored as a BIN code.	S/U (Request)
SD215		Clock data (second)	The second value of the clock data is stored as a BIN code.	S/U (Request)
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)
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)

No.	Name	Data stored	Details	Set by (setting timing)
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)
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)
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).  (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)
SD231		CPU No.2 operating status		
SD232		CPU No.3 operating status		
SD233		CPU No.4 operating status		
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)
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.  • 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)
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.)  0: Main 1 to 7: Extension 1 to 7	S (Initial)
SD244				
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)

No.	Name	Data stored	Details	Set by (setting timing)
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)
SD261		X (H)		
SD262		Y (L)	The number of points assigned to the Y device is stored in 32 bits.	S (Initial)
SD263		Y (H)		
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)
SD265		M (H)		
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)
SD267		B (H)		
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)
SD269		SB (H)		
SD270		F (L)	The number of points assigned to the F device is stored in 32 bits.	S (Initial)
SD271		F (H)		
SD272		V (L)	The number of points assigned to the V device is stored in 32 bits.	S (Initial)
SD273		V (H)		
SD274		L (L)	The number of points assigned to the L device is stored in 32 bits.	S (Initial)
SD275		L (H)		
SD276		S (L)*1	The number of points assigned to the S device is stored in 32 bits.	S (Initial)
SD277		S (H)*1		
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)
SD281		D (H)		
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)
SD283		W (H)		
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)
SD285		SW (H)		
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)
SD289		T (H)		
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)
SD291		ST (H)		
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)
SD293		C (H)		
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)
SD295		LT (H)		
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)
SD297		LST (H)		
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)
SD299		LC (H)		
SD300	Number of points assigned to the index register	Z	The number of points assigned to the Z device is stored.	S (Initial)
SD302	Number of points assigned to the long index register	LZ	The number of points assigned to the LZ device is stored.	S (Initial)
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)
SD307		ZR (H)		
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)
SD309		RD (H)		

No.	Name	Data stored	Details	Set by (setting timing)
SD312	File register block number	File register block number	The block number of the file register currently selected is stored.	S (Status change)
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)
SD384	System operation setting*1	• 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 • AFB0H: Program file only • AFBFH: Program file/FB file/global label setting file	U
SD385	System operation setting error cause*1	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: Value in SD384 out of range • 200H: Setting value failed to be written	S (Status change)

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

*2 The CPU module with firmware version "14" or later supports this special register area.

SFC information

The following is the special register area relating to SFC information.

No.	Name	Data stored	Details	Set by (setting timing)
SD329	Online change (SFC block) target block number*1	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)

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 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)
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)
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
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
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)

Fixed scan function information

The following is the list of special register areas relating to the fixed scan function information.

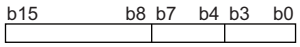
No.	Name	Data stored	Details	Set by (setting timing)
SD480	Number of cycle overrun events for inter-module synchronization cycle program (I44)	0: No cycle overrun event 1 to 65535: Accumulated number of cycle overrun events	The number of events in which the inter-module synchronous interrupt program (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)
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)
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)
SD500	Execution program number	Execution program number	The program number which is currently executed is stored as a BIN value.	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SD518	Initial scan time	Initial scan time (unit: ms)	• The initial scan time is stored into SD518 and SD519 (it is measured in increments of 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)
SD519		Initial scan time (unit: μs)		
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 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)
SD521		Current scan time (unit: μs)		
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)
SD523		Minimum scan time (unit: μs)		
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)
SD525		Maximum scan time (unit: μs)		
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)
SD527		END processing time (unit: μs)		
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)
SD529		Constant scan wait time (unit: μs)		

No.	Name	Data stored	Details	Set by (setting timing)
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)
SD531		Scan program execution time (unit: μs)		

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)
SD600	Memory card mounting status	SD memory card type	This register indicates the type of mounted SD memory cards.  • 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)
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)
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)
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)
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)
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)
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)
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)
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)
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)
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)
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)
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)
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)

A

No.	Name	Data stored	Details	Set by (setting timing)
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)
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)
SD630 SD631	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)
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)
SD634 SD635	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)

*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".)

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)
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)
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. 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)
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
SD775	Selection of refresh processing during the COM instruction execution	Selection of refresh processing during the COM instruction execution	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. ■ 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 ■ 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)	U
SD792 and SD793	PID limit setting (for complete derivative)	0: Limit restriction applied 1: No limit restriction	The limit restriction for each PID loop is specified as follows: (for the PIDCONT instruction) b15 to b1 b0 SD792: 16 to 2 1 SD793: 32 to 18 17 1 to 32: Loop 1 to 32	U
SD794 and SD795	PID limit setting (for incomplete derivative)	0: Limit restriction applied 1: No limit restriction	The limit restriction for each PID loop is specified as follows: (for the S.PIDCONT instruction) b15 to b1 b0 SD794: 16 to 2 1 SD795: 32 to 18 17 1 to 32: Loop 1 to 32	U

No.	Name	Data stored	Details	Set by (setting timing)
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
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
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
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
SD816 SD817	Basic period	Execution cycle	An execution cycle (unit: second) of process control instructions is set in real number.	U
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
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
SD820 SD821	Dummy device	Dummy device	A dummy device used in process control instructions is set.	U

*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.

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).

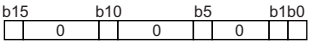
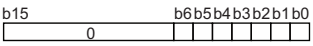
No.	Name	Data stored		Details	Set by (setting timing)
SD904	Latest firmware update information (network)*1	History information	Version after the update (network)	The firmware version after the update execution is stored. When the update is completed with an error, 0 is stored.	S (Initial)
SD905			Version before the update (network)	The firmware version before the update execution is stored.	S (Initial)
SD906	Previous firmware update information (network)*1		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)
SD907			Version before the update (network)	The firmware version before the update execution is stored.	S (Initial)
SD912	Latest firmware update information (CPU)*1		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)
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)
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)
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)
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)
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)
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)
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)
SD920			Version before the update (CPU)	The firmware version before the update execution is stored.	S (Initial)
SD921	Latest firmware update result*1	Target		The start I/O number of the module where the firmware update was executed is stored. • CPU module: 3FFH	S (Initial)
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)

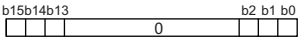
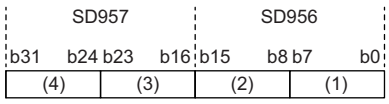
No.	Name	Data stored		Details	Set by (setting timing)
SD923	Previous firmware update information (CPU)*1	History information	Execution time (year)	The year value (four digits) of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
SD924			Execution time (month)	The month value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
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)
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)
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)
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)
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)
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)
SD931			Version before the update (CPU)	The firmware version before the update execution is stored.	S (Initial)
SD932	Previous firmware update result*1	Target		The start I/O number of the module where the firmware update was executed is stored. • CPU module: 3FFH	S (Initial)
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)

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

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)
SD944	CPU module data backup/restoration function Backup function*1	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
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
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
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
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
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
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
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 565 List of Error Codes) • "0" is set at the start of the CPU module data backup.	S (Error)

No.	Name	Data stored	Details	Set by (setting timing)
SD954	CPU module data backup/restoration function Restoration function*1	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
SD955		Restoration function setting	Set the CPU module data restoration function using the following bit pattern. (Off: Disabled, On: Enabled)  b0: Automatic restoration 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
SD956		Restoration target date folder setting	Store the target folder (date folder) of the CPU module data restoration using BCD code.  (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
SD957				U
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
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 565 List of Error Codes) • "0" is set at the start of the CPU module data backup.	S (Error)
SD960	CPU module data backup/restoration function Backup function*1	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)
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)

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

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)
SD1210 SD1211	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)
SD1212 SD1213	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)
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)
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)
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 565 List of Error Codes)	S (Error)
SD1220 to SD1226	Data logging setting No.2	Same configuration as the setting No.1	Data configuration is the same as setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1230 to SD1236	Data logging setting No.3	Same configuration as the setting No.1	Data configuration is the same as setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1240 to SD1246	Data logging setting No.4	Same configuration as the setting No.1	Data configuration is the same as setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1250 to SD1256	Data logging setting No.5	Same configuration as the setting No.1	Data configuration is the same as setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1260 to SD1266	Data logging setting No.6	Same configuration as the setting No.1	Data configuration is the same as setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1270 to SD1276	Data logging setting No.7	Same configuration as the setting No.1	Data configuration is the same as setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1280 to SD1286	Data logging setting No.8	Same configuration as the setting No.1	Data configuration is the same as setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1290 to SD1296	Data logging setting No.9	Same configuration as the setting No.1	Data configuration is the same as setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1300 to SD1306	Data logging setting No.10	Same configuration as the setting No.1	Data configuration is the same as setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1

CPU module data backup/restoration function

The following is the list of special register areas relating to the CPU module data backup/restoration function.

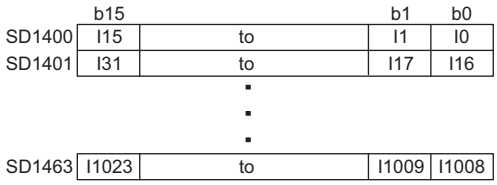
No.	Name	Data stored	Details	Set by (setting timing)
SD1350	Number of uncompleted folders/files of CPU module data backup/ restoration*2	Number of uncompleted folders/files of CPU module data backup/ restoration	This register indicates the number of folders/files where the backup/ restoration of the 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)
SD1351	Progression status of CPU module data backup/restoration*2	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)
SD1353	Upper limit value setting for the number of CPU module backup data*2	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

*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 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

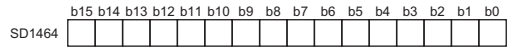
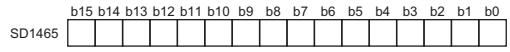
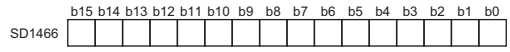
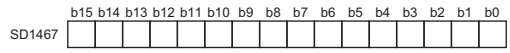
Interrupt pointer mask pattern

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)
SD1400 to SD1463	Interrupt pointer mask pattern	Mask pattern	The mask pattern for interrupt pointers is stored as shown in the following figure: 	S (at execution)

Event history function

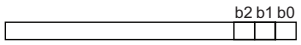
The following is the list of special relay areas relating to the event history function.

No.	Name	Data stored	Details	Set by (setting timing)
SD1464 to SD1467	Module information on event history logging restriction*1	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)  b0: Slot No.0 b1: Slot No.1 ⋮ b15: Slot No.15  b0: Slot No.16 b1: Slot No.17 ⋮ b15: Slot No.31  b0: Slot No.32 b1: Slot No.33 ⋮ b15: Slot No.47  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)

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

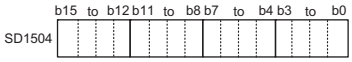
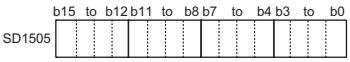
Debug function

The following is the special register area relating to the debug function.

No.	Name	Data stored	Details	Set by (setting timing)
SD1488	Debug function usage status	Debug function usage status	The usage status of the debug function is stored using the following bit pattern.  b0: External input/output forced on/off Off: Not used On: In use b1: Program restoration information write status 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 Off: Not registered On: Registered b3 to b15: Empty (fixed to 0)	S (Status change)

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)
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)
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)
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)

No.	Name	Data stored	Details	Set by (setting timing)
SD1520	IP address setting	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
SD1521		IP Address (upper)		
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
SD1523		Subnet mask pattern (upper)		
SD1524		Default gateway IP address (lower)	• 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. b15 to b8 / b7 to b0 SD1524: 3 / 4 SD1525: 1 / 2 1 to 4: First to fourth byte	S (Status change)/ U
SD1525		Default gateway IP address (upper)		
SD1526	IP address storage area writing error cause	The error cause for the case when failing in writing into IP address storage area is stored.	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.	S (Status change)
SD1527	IP address storage area clearing error cause	The cause of the error for the case when failing in clearing IP address storage area is stored.	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.	S (Status change)

*1 This is the memory used by the system when the CPU module is executing functions.

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)
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 stored 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)
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 specified 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)
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 stored value returns to its default (FFFFH) upon completion of the online module change processing.	S (Status change)
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)
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 641 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)
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 641 Error codes related to the online module change function)	S (Status change)

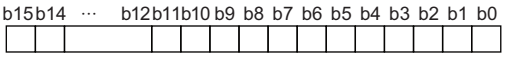
System information

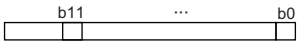
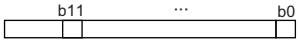
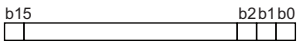
The following is the list of special register areas relating to the system information.

No.	Name	Data stored	Details	Set by (setting timing)
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)

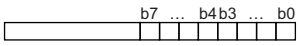
Redundant function

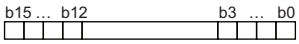
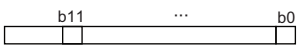
The following is the list of special register areas relating to the redundant function.

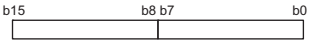
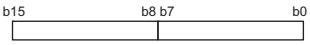
No.	Name	Data stored	Details	Set by (setting timing)
SD1642	BACKUP/SEPARATE LED flashing cause	BACKUP/SEPARATE LED flashing cause	Flashing cause of BACKUP/SEPARATE LED is stored using the following bit pattern.  Corresponding bits to each cause turns on. b1: Tracking communications disabled b3: Stop 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 • 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 384 Execution availability of system switching)	S (Status change)
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)

No.	Name	Data stored	Details	Set by (setting timing)
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 • 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)
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.)  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)
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.)  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)
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.  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

No.	Name	Data stored	Details	Set by (setting timing)
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)
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)
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
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 565 List of Error Codes)	S (Status change)
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
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)
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

No.	Name	Data stored	Details	Set by (setting timing)
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)
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.  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
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)
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)
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)
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)
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)
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)

No.	Name	Data stored	Details	Set by (setting timing)
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.  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)
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)
SD1756	Module information on wait timeout for receiving cyclic data after system switching[*1]	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.  b0 to b11: Slot No.0 to slot No.11	S (At system switching)

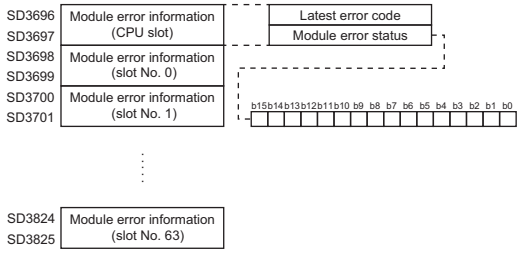
No.	Name	Data stored	Details	Set by (setting timing)
SD1760	Extension cable connection status*1	Extension cable connection status (bit pattern) Off: Not connected/ extension cable error/connected improperly On: Connected properly	• In a redundant configuration of extension cables, the connection status 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.  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)
SD1761	Extension cable route information*1	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.  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)

*1 There are restrictions on the firmware version of the CPU module and software version of the engineering tool. (☞ Page 762 Added and Enhanced Functions)

*2 ☞ Page 746 Waiting time for cyclic data receive after system switching (Twcyc)

Module error information collection function

The following table lists the special register areas for the module error information collection function.

No.	Name	Data stored	Details	Set by (setting timing)
SD3696 to SD3825	Module error information*1	Error information for modules mounted on each slot	• 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. • When SM3696 turns on, the error information of each module is updated every 5 seconds during END processing.  b0, b1:Error judgment 00: Normal 01: Minor error 10: Moderate error 11: Major error b2: Module ready status 0: Not ready 1: Ready b6, b7:Inter-module synchronous status 00: No synchronization 01: Preparing for synchronization 10: Synchronizing 11: Synchronization error b11:Fuse blown or not 0: Normal operation 1: Fuse blown b13:Module access restriction status 0: No module access restrictions 1: Module access restricted (when b15 is 0 and any of the following cases applies) ·During online module change ·During module restart ·When the standby system of a redundant system is trying to read module information on an extension base unit b15:Module accessibility 0: Access not permitted (applicable to an empty slot, a module missing, a module verification error, during online module change, or access to a module on an extension base unit from the standby system of a redundant system) 1: Access permitted b3 to b5, b8 to b10, b12, b14: Not used	S (Initial/During END)

*1 There are restrictions on the firmware version of the CPU module. (☞ Page 762 Added and Enhanced Functions)

Appendix 6 Buffer Memory

The buffer memory is memory used with the following applications.

Module	Application
CPU module	Stores values such as Ethernet function setting values. (📖 MELSEC iQ-R Ethernet User's Manual (Application))

The buffer memory content returns to its default status (initial value) when the CPU module is powered off or is reset.

Appendix 7 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.

A

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

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
- 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
- Tracking transfer time (☞ Page 739 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	
	Process mode	Redundant mode
One unit of CPU module (module is not mounted)	100μs	190μs

The common processing time by the system is 85μs for one unit of CPU module (module is not mounted).

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 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.

$\alpha T, \alpha R$ [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 ($\times 10^{-3}$)		40.00
KM2 ($\times 10^{-3}$)	Main base unit*4	0.01
	Extension base unit*4	0.12
KM3 ($\times 10^{-3}$)	Main base unit*4	0.03
	Extension base unit*4	0.15

*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.

■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.

$\alpha T, \alpha R$ [ms] = KM1 + KM2 × ((RX + RY + SB) ÷ 16 + RWr + RWw + 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
- 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 ($\times 10^{-3}$)		40.00
KM2 ($\times 10^{-3}$)	Main base unit*4	0.01
	Extension base unit*4	0.12
KM3 ($\times 10^{-3}$)	Main base unit*4	0.03
	Extension base unit*4	0.15

*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)

$\alpha T, \alpha R$ [ms] = KM1 + KM2 × ((RX + RY + SB) ÷ 16 + RWr + RWw + 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
- 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 ($\times 10^{-3}$)		40.00
KM2 ($\times 10^{-3}$)	Main base unit*4	0.01
	Extension base unit*4	0.12
KM3 ($\times 10^{-3}$)	Main base unit*4	0.03
	Extension base unit*4	0.15

*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.

A

■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 ($\times10^{-3}$)		40
KM2 ($\times10^{-3}$)	Main base unit*4	0.01
	Extension base unit*4	0.12
KM3 ($\times10^{-3}$)	Main base unit*4	0.03
	Extension base unit*4	0.15

*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 ($\times10^{-3}$)		65
KM2 ($\times10^{-3}$)	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.

Intelligent function module refresh processing time

The following calculation formula shows the refresh processing time for the intelligent function module 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

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	0.09
	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	
		Process mode	Redundant mode
When all the columns are outputted under the condition of file format of Unicode text file	KM1	35	
	KM2	37	
	KM3	0.35	

■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.

■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	Calculation formula	Constant value
When SM388 is off (Program file only)	(KM1 × Number of program files[*1]) + (KM2 × Number of steps in the largest program[*2]) + 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[*3]) + (KM5 × Total number of steps in programs[*4])	KM4: 2.34 KM5: 0.06×10^{-3}

*1 This indicates the number of program files that are written to the CPU module.
*2 This indicates the number of steps in the largest program file among the program files to which the file batch online change is performed.
*3 This indicates the number of program files to which the file batch online change is performed.
*4 This indicates the total number of steps in the 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 + KM3 × (number of bit points to be communicated ÷ 16 + number of word points to be communicated) + αE_n
- αE_n [μs] = KM4 + KM5 × (number of bit points to be communicated ÷ 16 + number of word points to be communicated)
  - αT_n: Simple CPU communication processing time per setting
  - αE_n: Additional time when the file register (R, ZR) is used (It is added when the file register (R, ZR) is used.)
  - n: Number of settings

Condition		Constant value	
		Process mode	Redundant mode
KM1		85.00	127.00
KM2		5.50	5.50
KM3		0.024	0.024
KM4		1.60	1.60
KM5	When an extended SRAM cassette is not used	0.10	0.10
	When an extended SRAM cassette is used	0.13	0.13

■Module error information collection function

When the module error information collection function is executed (when SM3696 turns on), the scan time will be extended by up to 4μs in one scan after the module error information is collected (at five second intervals).[*1*2]

*1 Regardless of whether process mode or redundant mode is selected, the scan time will be extended.

*2 This value is the maximum value when 1 CPU module and 64 other modules are mounted.

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.

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.)

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 = Data register (D) (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: 10000 records, operation when the number of saved files exceeded: overwriting
- Functions that require access to files in the SD memory card except for the data logging function must be not active.
- For the parameters, use the default setting values.

*1 3ms (to 4.5ms) for redundant mode

Number of points		Collection interval at which data can be collected	
		NZ1MEM-2GBSD	NZ1MEM-4/8/16GBSD
8 points	(8 points × 1 setting)	1.0ms	1.0ms
16 points	(16 points × 1 setting)	1.0ms	1.0ms
64 points	(64 points × 1 setting)	1.0ms	2.0ms
128 points	(128 points × 1 setting)	1.0ms	4.0ms
256 points	(128 points × 2 setting)	4.0ms	7.0ms
1280 points	(128 points × 10 setting)	18.0ms	20.0ms

When the file format is Unicode text file

■When a global device or global label is specified

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 = Data register (D) (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: 10000 records, operation when the number of saved files exceeded: overwriting
- Functions that require access to files in the SD memory card except for the data logging function must be not active.
- For the parameters, use the default setting values.

*1 3ms (to 4.5ms) for redundant mode

Number of points		Collection interval at which data can be collected	
		NZ1MEM-2GBSD	NZ1MEM-4/8/16GBSD
8 points	(8 points × 1 setting)	1.0ms	1.0ms
16 points	(16 points × 1 setting)	1.0ms	2.0ms
64 points	(64 points × 1 setting)	2.0ms	5.0ms
128 points	(128 points × 1 setting)	4.0ms	7.0ms
256 points	(128 points × 2 setting)	7.0ms	13.0ms
1280 points	(128 points × 10 setting)	24.0ms	26.0ms

When the file format is binary file

■When a global device is specified

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 = Data register (D) (Data-type: word-signed (decimal number type))
- Output setting = Day/time column (output format is default) and index column are outputted.
- Save setting = 10000 records, operation when the number of saved files exceeded: overwriting
- Functions that require access to files in the SD memory card except for the data logging function must be not active.
- For the parameters, use the default setting values.

*1 3ms (to 4.5ms) for redundant mode

Number of points		Collection interval at which data can be collected	
		NZ1MEM-2GBSD	NZ1MEM-4/8/16GBSD
8 points	(8 points × 1 setting)	1.0ms	1.0ms
16 points	(16 points × 1 setting)	1.0ms	2.0ms
64 points	(64 points × 1 setting)	2.0ms	5.0ms
128 points	(128 points × 1 setting)	4.0ms	7.0ms
256 points	(128 points × 2 setting)	7.0ms	13.0ms
1280 points	(128 points × 10 setting)	24.0ms	26.0ms

Process control function processing time

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

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
Active block processing time coefficient		4.4
Inactive block processing time coefficient		2.7
Nonexistent block processing time coefficient		0.23
Active step processing time coefficient		6.8
Active transition processing time coefficient		0.12
Transition establishment step processing time coefficient	Hold step	22.5
	Normal step	31.0
SFC END processing time		56.5
Operation output processing time coefficient		0.94

SFC program switching

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
Km	7.90*1
Kn	1.55*1
Kp	0.21*1
Kq	2500

*1 For the firmware version "12" or earlier, the values are as follows.
· R08PCPU, R16PCPU, R32PCPU: Km = 3.97, Kn = 0.41, Kp = 0.39
· 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)
R08PCPU, R16PCPU, R32PCPU	28K steps or less	14ms
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.

Redundant function processing time

This section describes the processing time in the redundant function.

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.

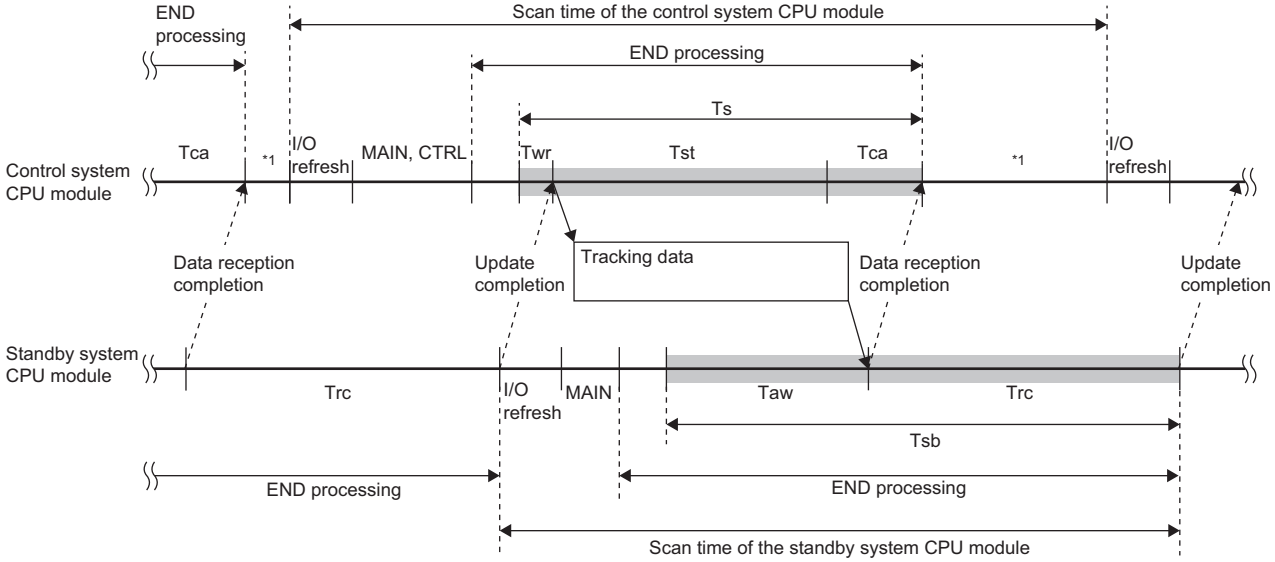
Point

The scan time can be checked in SD520 and SD521 (Current scan time). (☞ Page 698 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 740 Waiting time for completion of the previous tracking data reflection (Twr) Tst: Page 740 Tracking data send time (Tst) Tca: Page 741 Waiting time for completion of tracking data reception (Tca) α: Page 742 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 742 Waiting time for tracking data reception (Taw) Trc: Page 743 Tracking data reflection time (Trc) α: Page 743 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 81 In redundant mode), but the operation is the same as in a configuration with the main base unit 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 × 300.0 × 10^{-6}) + (Total size of tracking data [word] × 45.0 × 10^{-6})
- When an extended SRAM cassette is inserted: 1 + (No. of tracking blocks × 300.0 × 10^{-6}) + (Total size of tracking data [word] × 106.0 × 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.

Extended SRAM cassette inserting status	Tracking data send time
Not inserted	0.5 + (26.7 × 10^{-6}) × D1 + (43.5 × 10^{-6}) × D2 + (1.5 × 10^{-3}) × D3 + E1 + F1[ms]
Inserted	0.5 + (26.7 × 10^{-6}) × D1 + (113.5 × 10^{-6}) × D2 + (1.5 × 10^{-3}) × D3 + E1 + 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, and refresh data register (RD)
- D5: Size [word] of tracking data of global devices, local devices, global labels, and local labels
- D6: Number of tracking transfer settings of global devices
- D7: Total size [word] of tracking data in the system data, signal flow memory, local devices, global labels, local labels, and global devices
- 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)

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 × (No. of tracking settings of T, ST, C, and LC) + 8 × (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*1 3131 × (Number of CC-Link modules mounted on the extension base unit) ■Data that varies depending on the simple CPU communication setting • When it is set: 68 • When it is not set: 0
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 398 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*2 Number of units to be refreshed to the module label of modules on the extension base unit × 1.7KW (fixed value) • When not transferring the module label of modules on the extension base unit*3 0

*1 The CC-Link module that was set with the parameter is the target.

*2 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".

*3 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*4
Global device (other than the refresh data register (RD))	Follows the tracking transfer settings. (☞ Page 398 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*6
Global label*5	Follows the global label settings.
Local label*5	Follows the local label settings.

*4 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 401 Detailed setting)

*5 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.

*6 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)
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 398 Tracking transfer setting)

D5 is determined as follows.

Item	Size*7
Global device (other than the refresh data register (RD))	Follows the tracking transfer settings. (☞ Page 398 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*9
Global label*8	Follows the global label settings.
Local label*8	Follows the local label settings of each program.

*7 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 401 Detailed setting)

*8 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.

*9 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 extended times (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] × 26.7 × 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 85 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 208 Error Clear)

■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.

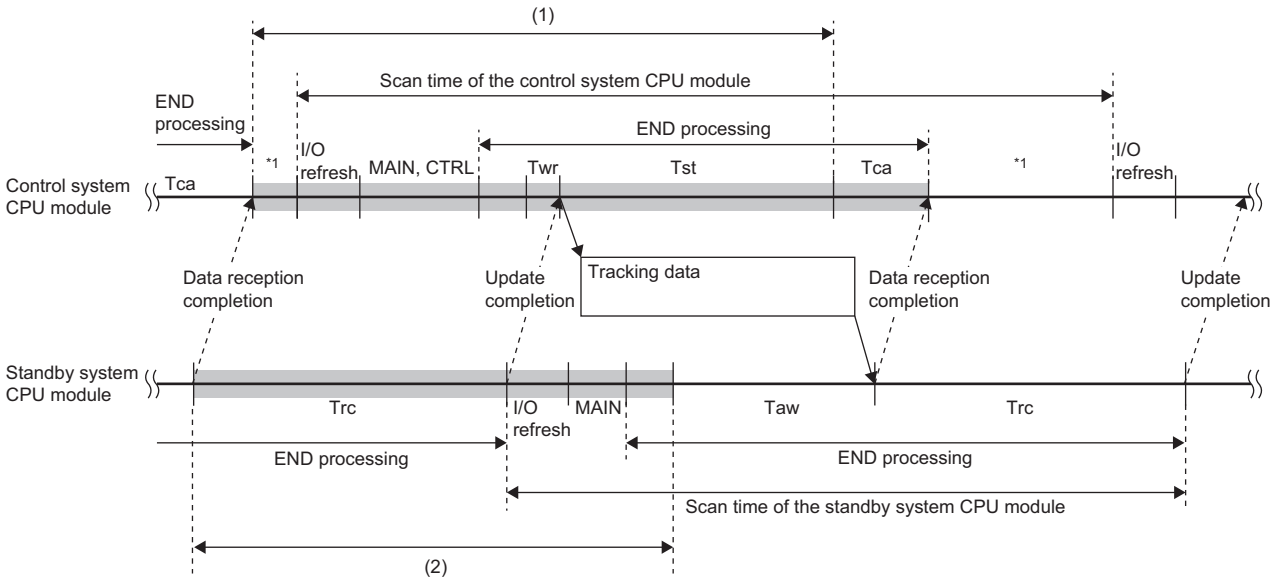
The waiting time for tracking data reception is determined by two types of period; (1) the period from when the control system CPU module receives a receive completion notification until next time it receives the notification and (2) the period when the standby system CPU module sends a receive completion 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 81 In redundant mode), but the operation is the same as in a configuration with the main base unit only.

- For the I/O refresh processing time for (1) and (2) and END processing time for others, refer to the following.

☞ Page 725 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.

Extended SRAM cassette inserting status	Tracking data reflection time
Not inserted	$1+(20.0\times10^{-6})\times D1+(37.2\times10^{-6})\times D2+E1+F1+(300.0\times10^{-6})\times G1$[ms]
Inserted	$1+(20.0\times10^{-6})\times D1+(106.0\times10^{-6})\times D2+E1+F1+(300.0\times10^{-6})\times G1$[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, and refresh data register (RD)
- D5: Size [word] of tracking data of global devices, local devices, global labels, and local labels
- 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 740 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 the SP.CONTSW instruction		0ms
System switching request from the engineering tool		0ms

Time required for system switching

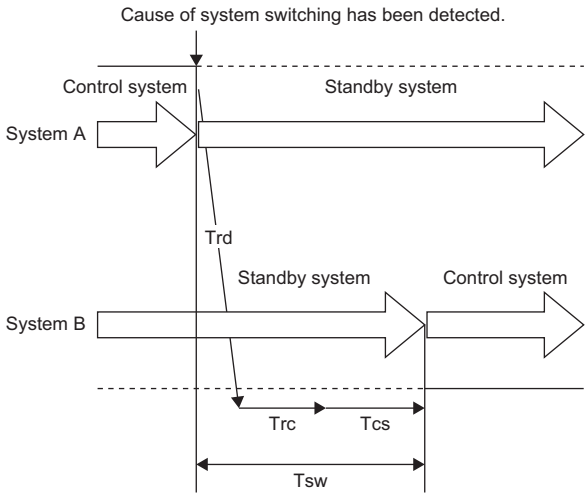
The following describes the time required for the CPU module in the new control system to start operating as the new control system after the detection of a system switching cause in the control system.

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	1ms	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 745 Tracking data reflection time (Trc)*4	5ms	
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 745 Tracking data reflection time (Trc)		
System switching request by 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 When a CC-Link IE Field Network module or CC-Link IE Controller Network module is not mounted or the redundant extension base unit is not used, the system switching may not occur depending on states of the power supply module failure. To perform the system switching regardless of the states of the power supply module failure, mount the CC-Link IE Field Network module or CC-Link IE Controller Network module, or use the redundant extension base unit. Note that Trd is as follows in this case:
· Via the CC-Link IE Field Network module: Trd = 400ms (maximum) + Standby system scan time
· Via the CC-Link IE Controller Network module: Trd = 400ms (maximum) + Standby system scan time
· Via the redundant extension base unit: Trd = 57.5ms (maximum)

*3 When a CC-Link IE Field Network module or CC-Link IE Controller Network module is not mounted or the redundant extension base unit is not used, the system switching may not occur even though a communication error has been detected between the redundant function module and the CPU module. To perform the system switching when the redundant function module has been removed from the base unit or the base unit has failed, mount the CC-Link IE Field Network module or CC-Link IE Controller Network module, or use the redundant extension base unit. Note that Trd is as follows in this case:
Trd = Standby system scan time
· Via the CC-Link IE Field Network module: Trd = Standby system scan time
· Via the CC-Link IE Controller 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.

Extended SRAM cassette inserting status	Tracking data reflection time
Not inserted	$1+(20.0\times10^{-6})\times D1+(37.2\times10^{-6})\times D2+(300.0\times10^{-6})\times F1$[ms]
Inserted	$1+(20.0\times10^{-6})\times D1+(106.0\times10^{-6})\times D2+(300.0\times10^{-6})\times F1$[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, and refresh data register (RD)
- D5: Size [word] of tracking data of global devices, local devices, global labels, and local labels
- F1: Number of transferred blocks

For the calculation formula for D1 to D5, refer to the following.

☞ Page 740 Tracking data send time (Tst)

Extended time until initial output after system switching (Tjo)

This section describes the extended 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.

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 746 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)

■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 746 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 743 Extended time required for detecting a system switching cause)
- Tsw: Time required for system switching (☞ Page 744 Time required for system switching)
- Tjo: Extended time from completion of a system switching to the first output (☞ Page 745 Extended 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:

[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.

Appendix 8 Parameter List

This section lists parameters.

Point

For the parameters that are not described in this chapter, refer to the manuals for each module used.

System parameters

The following is the list of system parameters.

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
	Points Occupied by Empty Slot Batch Setting		0100H
	Redundant Module Group Setting		0400H
Multiple CPU Setting	Communication Setting between CPUs	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 Function	—
		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	0306H
	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	Use Inter-module Synchronization Function in System		—
	Select Synchronous Target Unit between Unit		0101H
	Fixed Scan Interval Setting of Inter-module Synchronization		0101H
	Inter-module Synchronous Master Setting		0102H

A

CPU parameters

The following is the list of CPU parameters.

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		3A00H
	Interrupt Enable Setting in Executing Instruction		3A00H
	Block No. Save/Recovery Setting		3A00H
	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
	Device/Label Memory Area Setting	Device/Label Memory Area Capacity Setting	3400H
		Device Points	3401H
		Local Device	3405H
		Latch Range Setting	3407H
		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
RAS Setting	Scan Time Monitoring Time (WDT) Setting		3500H
	Constant Scan Setting		3503H
	Error Detections Setting		3501H
	CPU Module Operation Setting at Error Detected		3501H
	LED Display Setting		3502H
	Event History Setting	Save Destination	3507H[*1]
		Set Save Volume of Per File	3507H[*1]
	Online Module Change Function Setting	Direct Change Setting	3505H
Program Setting	Program Setting	Program Name	3700H
		Execution Type	3700H
		Type (Fixed Scan)	3700H
		Type (Event)	3701H
		Detail Setting Information	—
		Refresh Group Setting	3700H
		Device/File Use or not	3700H
		Both Systems Program Executions Setting	3700H
	FB/FUN File Setting		3702H
Refresh Setting between Multiple CPU	Refresh Setting (At the END)		3901H
	Refresh Setting (At I45 Exe.)		3902H

Item			Parameter No.
Routing Setting	Routing Setting		3800H
SFC Setting	SFC Program Start Mode Setting		3C00H
	Start Condition Setting		3C00H
	Output Mode at Block Stop Setting		3C00H
Redundant System Settings	Redundant System Behavior Setting		5000H
	Tracking Setting	Signal Flow Memory Tracking Setting	5001H
		Tracking Device/Label Setting	5001H
	Redundant System with Extension Base Unit	Extension Cable Redundant Error Detection Setting at Startup	5002H
		Automatic Standby System Recovery Function	5002H

*1 When the device/label operation save setting is set to "Do not save", the value is 3504H.

Module parameters

The following are the lists of module parameters.

Ethernet function

For the module parameters relating to the Ethernet function, refer to the following.

MELSEC iQ-R Ethernet User's Manual (Application)

Memory card parameters

The following is the list of memory card parameters.

Item		Parameter No.
Boot Setting	Operation Setting at CPU Built-in Memory Boot	2000H
	Label Initialization Setting During Booting	
	Boot File Setting	
Setting of File/Data Use or Not in Memory Card	Data for Label Communication	2010H
	Module Extended Parameter	

A

Appendix 9 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.)

A

Operation details

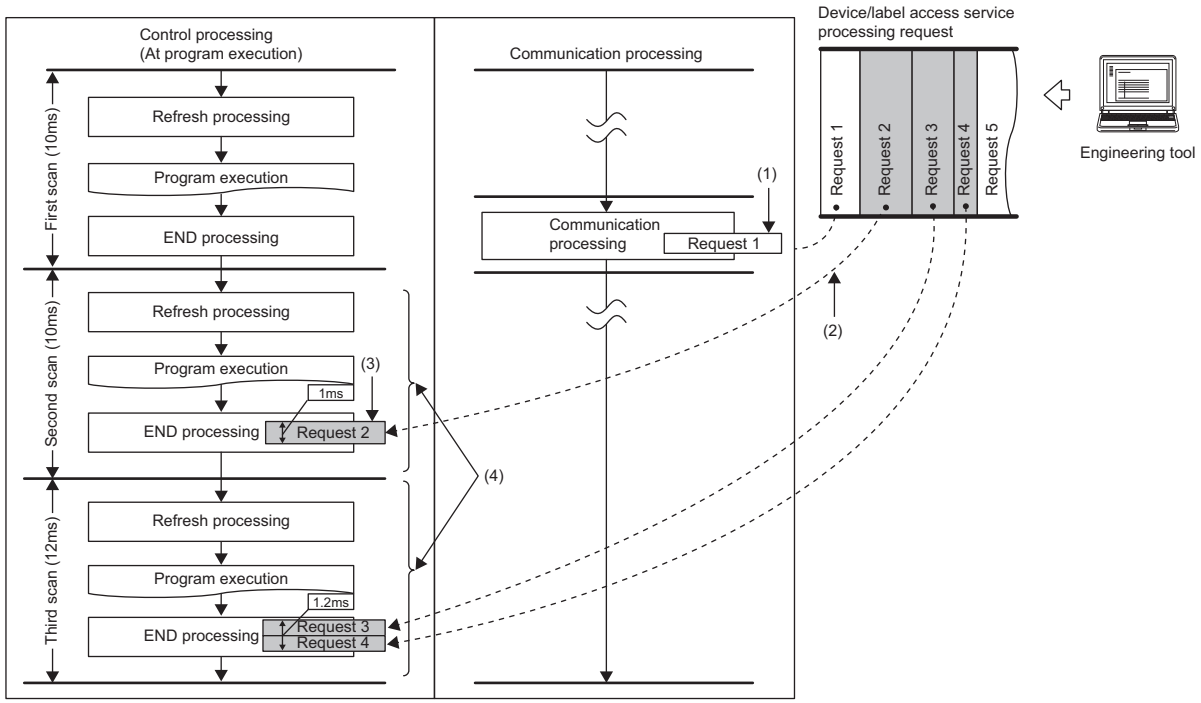
Operations enabled by setting details of the device/label access service processing setting are as follows.

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 versions of the compatible CPU modules, refer to the following.

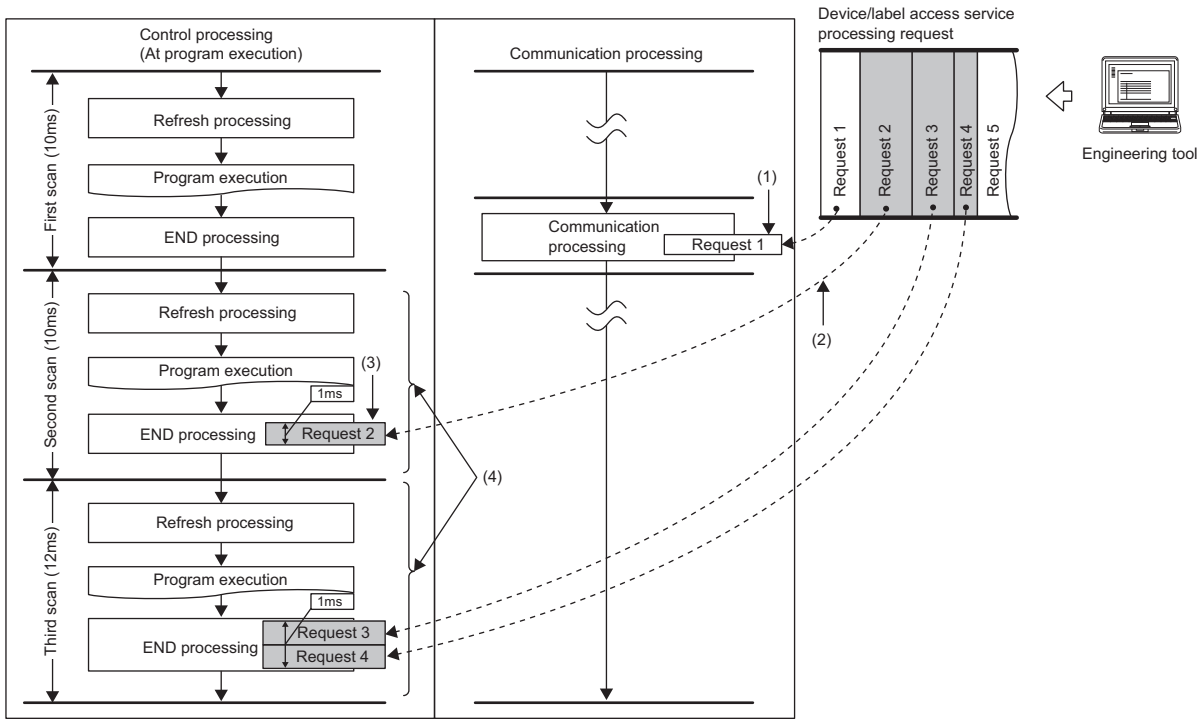
☞ Page 762 Added and Enhanced Functions

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 versions of the compatible CPU modules, refer to the following.

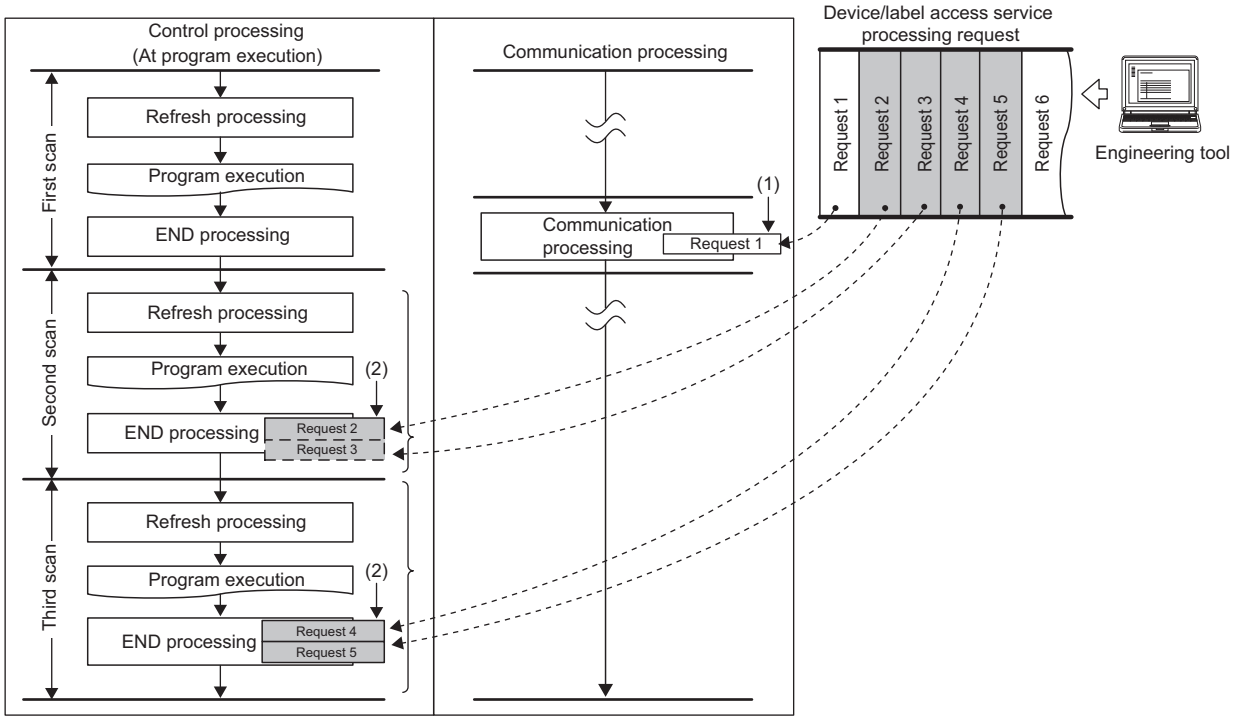
☞ Page 762 Added and Enhanced Functions

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

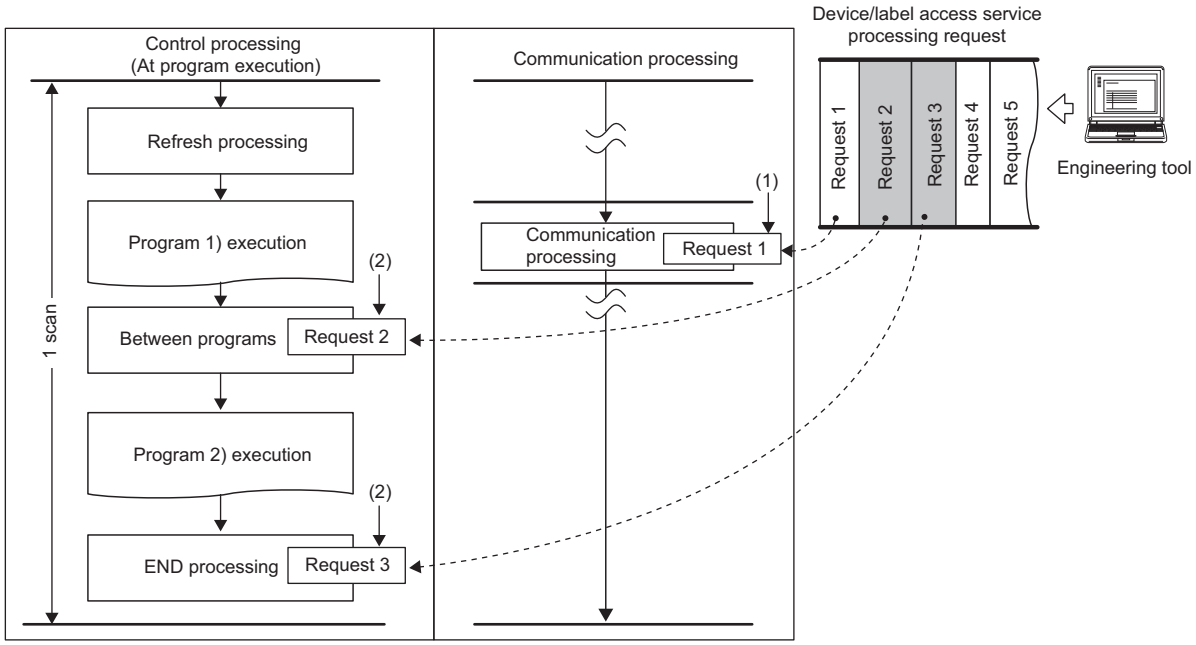
For operations such as access to devices, which are synchronized with the program, adjust the number of executions by this setting because they are processed during the END processing. If no request data for the device/label access service processing exists, the CPU module proceeds to the next scan without 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.

A

Appendix 10 Program Restoration Information Write Selection

Program restoration information includes the information required to read a program from the programmable controller with the engineering tool. (☞ Page 140 Data allocation and procedure of read/write operations)
Generally, use the CPU module with program restoration information written.
With this setting, a format in which the program restoration information is not written or a format in which only the program restoration information is written during writing to the programmable controller and online change can be selected.*1
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.

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 663 System information
SD1488 (Debug function usage status)	b1: Program restoration information write status Off: All written*1 On: Not all written*2		Page 711 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.

FUNCTION LED

The FUNCTION LED indication follows the priority order shown below.

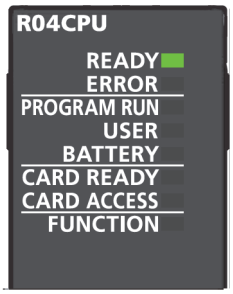
Priority	Description	Remarks
High	When program restoration information is not written, when the external input/output forced on/off function is executed (in registration), when the device tests with execution conditions are registered	Same priority
Low	Functions set in "Function to use FUNCTION LED" of "LED Display Setting" (such as the data logging function)	☞ Page 206 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.

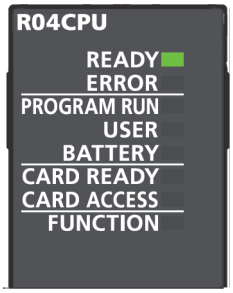
When other functions relating to the FUNCTION LED indication are used, the LED indication follows the execution status.

Point

When using 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.

In redundant mode

Only when the firmware version of the CPU modules in both systems supports the program restoration information write selection, "Not to Write" can be selected.

Appendix 11 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

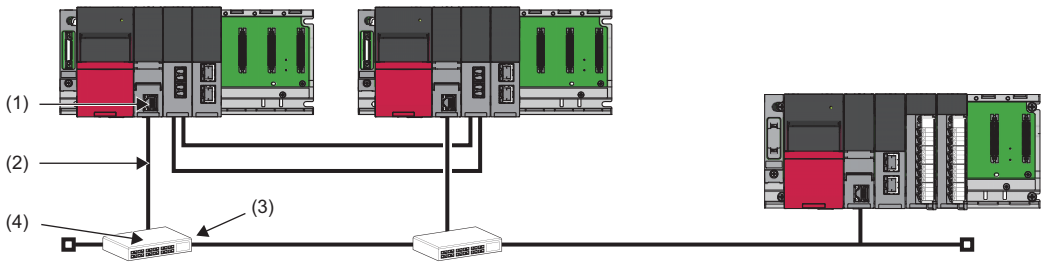
*1 The program consistency check ("Check the consistency between the editing target program file in GX Works3 and the one in PLC.") cannot be executed. (GX Works3 Operating Manual)

*2 When "Specify Redundant CPU" is set to "Not Specified", "Control System", or "Standby System", the event history cannot be displayed or cleared.

Appendix 12 Program for System Switching at Built-in Ethernet Communication Error

In the redundant system, the CPU module (built-in Ethernet port part) does not issue a system switching request even though a communication error has been detected at the own station side. Therefore, use a program shown in this section to execute the system switching instruction so that communications continue even at the error.

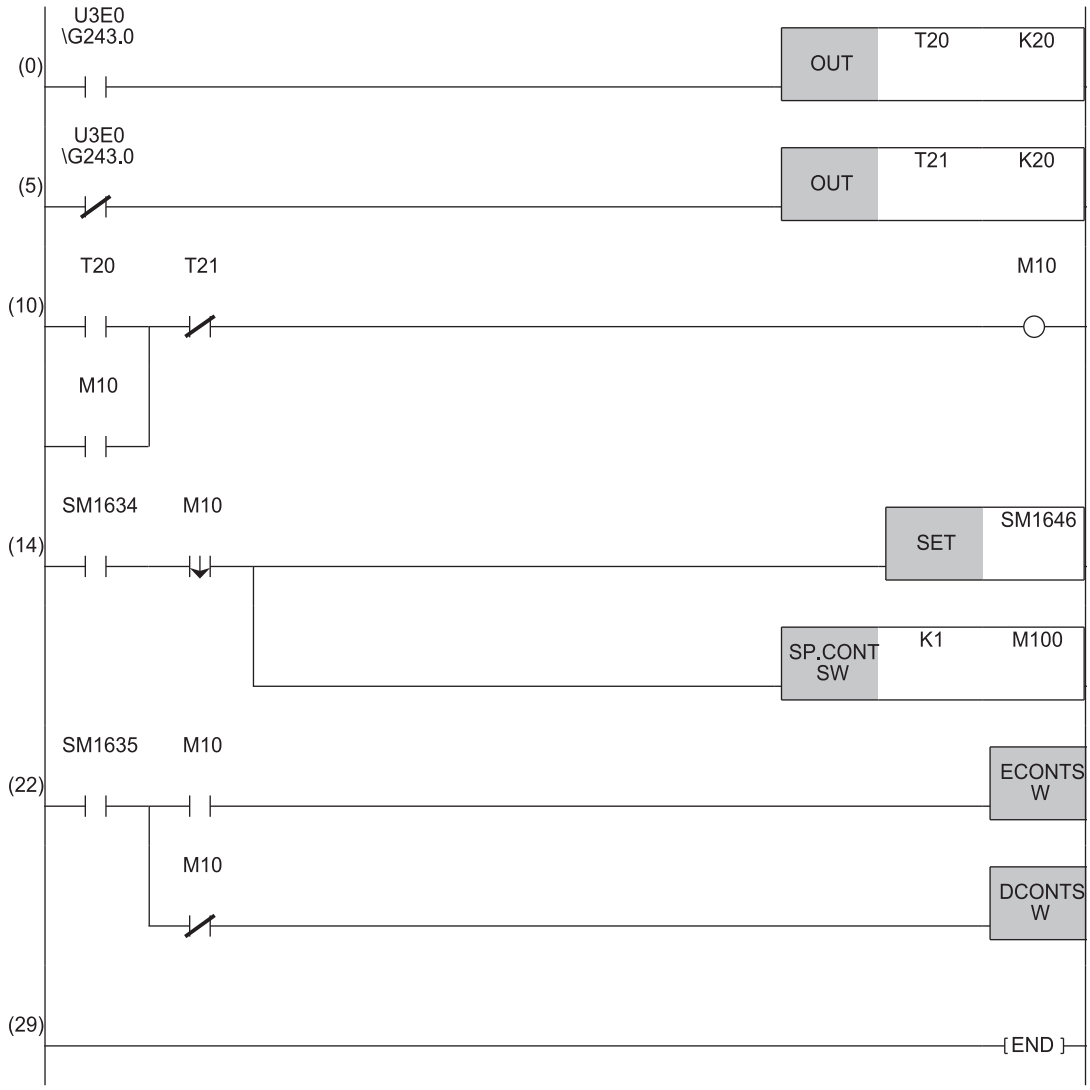
The figure below shows built-in Ethernet communication errors that can be detected by the program.



(1) Cable disconnection from the connector of the built-in Ethernet port
(2) A break in the cable connecting the built-in Ethernet port and the hub
(3) Cable disconnection from the connector of the hub
(4) Power-off of the hub

The following is the program example.

Create this program as a program executed in both systems, and exclude the devices used for the program from the tracking transfer targets.



(0) A hub connection has been confirmed.
(5) A hub disconnection has been detected.
(10)The hub connection is being attempted.
(14)Enables the system switching by a user and executes the system switching in the control system.
(22)Enables/disables the system switching in the standby system.

Point

- According to the environment in which the systems are used, change conditions or a timer value used for the system switching.
- For a communication error monitoring time, set a value in the range of a few seconds to a few tens of seconds. Setting the monitoring time too short may cause a system switching due to noise.

Appendix 13 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
	Process mode	Redundant mode	
Mounting of MELSEC iQ-R series modules occupying two slots	F: ○ S: 1.007H	F: ○ S: 1.007H	For the applicable modules, refer to the following. MELSEC iQ-R Module Configuration Manual
SFC support	F: 03 S: 1.020W O: 1.49B[*1*2*3]	F: 18 S: 1.050C	MELSEC iQ-R Programming Manual (Program Design)
Duplication check for POU	F: 04 S: 1.025B	F: 04 S: 1.025B	GX Works3 Operating Manual
Background processing of program transfer (writing the program restoration information) during changing a program online	F: 10 S: 1.022Y	F: 15 S: 1.045X	GX Works3 Operating Manual
CPU module data backup/restoration function	F: 20 S: —	F: 20 S: —	Page 311 CPU MODULE DATA BACKUP/ RESTORATION FUNCTION
Redundant function	×	F: 04 S: 1.025B	Page 373 REDUNDANT FUNCTION
Redundant power supply system	F: 04 S: 1.025B	F: 04 S: 1.025B	MELSEC iQ-R Module Configuration Manual
Label initialization function	F: 04 S: 1.025B	F: 04 S: 1.025B	Page 534 LABEL INITIALIZATION FUNCTION
Firmware update function (Firmware update using the engineering tool)	F: 24 S: 1.065T	F: 24 S: 1.065T	MELSEC iQ-R Module Configuration Manual
Firmware update function (Firmware update using an SD memory card)	F: 14 S: —	F: 14 S: —	MELSEC iQ-R Module Configuration Manual
External input/output forced on/off function	F: 15 S: 1.045X	F: 15 S: 1.045X	Page 235 External Input/Output Forced On/ Off Function
Multiple CPU system	F: ○ S: 1.007H	×	MELSEC iQ-R Module Configuration Manual
Program restoration information write selection	F: 13 S: 1.040S	F: 15[*4] S: 1.045X[*4]	Page 756 Program Restoration Information Write Selection
File batch online change of FB files and the global label setting file	F: 13 S: 1.040S	F: 13 S: 1.040S	Page 193 File batch online change
File batch online change when the program memory does not have enough free space	F: 13 S: 1.040S	F: 13 S: 1.040S	Page 193 File batch online change
Error detection invalidation setting	F: 14 S: —	F: 14 S: —	Page 207 Error detection invalidation setting
'Communication load status' (Un\G100 to Un\G103)	F: 22 S: —	F: 22 S: —	MELSEC iQ-R Ethernet User's Manual (Application)
Online change (SFC block)	F: 20 S: 1.055H	F: 20 S: 1.055H	MELSEC iQ-R Programming Manual (Program Design)
Omission of writing files that are not changed	F: 18 S: 1.050C	F: 18 S: 1.050C	GX Works3 Operating Manual
Device test with execution conditions	F: 20 S: 1.055H	F: 20 S: 1.055H	Page 244 Device Test with Execution Conditions

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
	Process mode	Redundant mode	
Setting to wait cyclic data receive after system switching[*5]	F: 18 S: 1.050C	F: 18 S: 1.050C	Page 432 Setting to wait cyclic data receive after system switching
Event history logging restriction	F: 20 S: —	F: 20 S: —	Page 215 Event history logging restriction
Extension of points for CC-Link IE Controller Network	F: 27 S: 1.075D	F: 27 S: 1.075D	• Page 460 Device List • Page 483 Link Direct Device • MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)
Online Change	F: ○ S: 1.007H	F: 04 S: 1.025B	Page 189 Online Change
Data communications between CPU modules	F: ○ S: 1.007H	×	• Page 337 MULTIPLE CPU SYSTEM FUNCTION • Page 347 Data Communication Between CPU Modules
Multiple CPU synchronous interrupt (I45)	F: ○ S: 1.007H	×	Page 337 MULTIPLE CPU SYSTEM FUNCTION
Inter-module synchronous interrupt (I44)	F: ○ S: 1.007H	×	MELSEC iQ-R Inter-Module Synchronization Function Reference Manual
Writing of program restoration information	F: 20 S: 1.055H	F: 20 S: 1.055H	Page 756 Program Restoration Information Write Selection
FB hierarchy information	F: 24 S: 1.060N	F: 24 S: 1.060N	GX Works3 Operating Manual
MELSEC iQ-R series MELSECNET/H network module	F: 23 S: 1.063R	F: 23 S: 1.063R	• 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: 1.007H	• Page 78 Extended SRAM Cassette • Page 142 Device/label memory area setting
Redundant system with redundant extension base unit	×	■For R68WRB F: 25 S: 1.070Y ■For R66WRB-HT F: 25 S: 1.072A	• Page 373 REDUNDANT FUNCTION • MELSEC iQ-R Module Configuration Manual
Laser displacement sensor control module supporting the redundant system with redundant extension base unit	×	F: 26 S: —	MELSEC iQ-R Module Configuration Manual
Program start/stop	F: 27 S: 1.075D	F: 27 S: 1.075D	GX Works3 Operating Manual
Label memory read/write	F: 27 S: 1.075D	F: 27 S: 1.075D	GX Works3 Operating Manual
Device/label access service processing constant wait function	F: 27 S: 1.075D	F: 27 S: 1.075D	Page 91 Device/label access service processing constant wait function
Copying/cutting across data logging settings for a data setting item	F: — S: — O: 1.118X[*3]	×	CPU Module Logging Configuration Tool Version 1 Operating Manual
System memory supports the memory copy from control system to standby system	×	F: 33 S: —	Page 409 Memory Copy from Control System to Standby System
Simple CPU communication function	F: 34 S: 1.090U	F: 34 S: 1.090U	MELSEC iQ-R Ethernet User's Manual (Application)
Control system start-up at one system error	×	F: 44 S: 1.110Q	☞ Page 438 Control System Start-up at One System Error
Module error information collection function	F: 46 S: —	F: 46 S: —	☞ Page 232 Module Error Information Collection Function
Label Initialization Setting During Booting	F: 46 S: 1.115V	F: 46 S: 1.115V	☞ Page 227 Configuring the Boot Setting

*1 This indicates the compatible software version of GX LogViewer.
*2 Step relays (BLn\Sn) can be specified with CPU Module Logging Configuration Tool or GX LogViewer.
*3 This indicates the compatible software version of CPU Module Logging Configuration Tool.
*4 Support of the program restoration information write selection of writing to the programmable controller while the CPU module is stopped and of the file batch online change is as follows: F: 13, S: 1.040S.
*5 The firmware version of the usable RJ71GF11-T2 is "35" or later.

INDEX

Symbols

"$MELPRJ$" folder 145

A

Accuracy of constant scan 86
Accuracy of timers 472
Annunciator (F) 76,465
ASCII 509

B

BACKUP LED 33
Backup mode 25
Battery 30,38
BATTERY LED 29
Behavior at the time of a transition to RUN 292
Bit 509
Block number 492
Block switching method 492
BOOL 509
Boot operation 225
Buffer memory 25

C

CARD ACCESS LED 29
CARD READY LED 29
CC-Link IE Controller Network-equipped module 26
CC-Link IE Field Network-equipped master/local module 26
Character string constant 539
Class 511
Clearing file registers 493
Constant 538
Constant scan 76,85
Continuous logging 267
Control system 25
Control system execution program 25
Controlled module 338
COUNTER 509,510
Counter 509,510
Counter (C) 76,477
CPU buffer memory 74,145
CPU buffer memory access device 77,487
CPU module 74,653
CPU parameter 50
CPU parameters 748
CTRL LED 33

D

Data logging 258
Data memory 74,145
Data register (D) 76,480
Data type 509,510,511
Decimal 538
Decimal constant (K) 538
Device area 74
Device assignment 507
Device initial value 530
Device supporting iQSS 26
Device/label access service processing 87
Device/label memory 74,141
DINT 509
Double word [signed] 509
Double word [unsigned]/bit string [32 bits] 509
Double-precision real number 509,539
DWORD 509

E

Edge relay (V) 76,468
END processing 83
ERR LED 33
Error clear 208
Error code 563
ERROR LED 29,51,545
Ethernet interface module with built-in CC-Link IE 26
Ethernet port 30,74
Ethernet-equipped module 26
Event execution type program 107
Event history 211
Event history file 213
Extended SRAM cassette 39,78
External dimensions 74,78,653

F

File 151
File password 368
File register (R/ZR) 77,492
File register setting 493
File registers common to all programs 493
File storage area 74
Fixed scan execution mode 105
Fixed scan execution type program 102
Fixed scan interval 103
Function (FUN) 508
Function block 48
Function block (FB) 508
Function device 481
Function input (FX) 77,481
FUNCTION LED 29
Function output (FY) 77,481
Function register (FD) 77,481

G

Generic data type (ANY type) 511
Global device 501
Global label 507,508,511
Global pointer 496

H

Hexadecimal 538
Hexadecimal constant (H) 538
High-speed internal timer interrupt 1 (I49) 76
High-speed internal timer interrupt 2 (I48) 76
High-speed timer 470

High-speed timer (T/ST) 470

I

I/O No. specification device (U) 501
I/O refresh 82
Index modification 488
Index register (Z) 77
Index register setting 489
Indirect specification 506
Initial execution type program 101
Initial processing (when powered on) 82
Initial processing (when switched to RUN) 82
Initial scan time 84
Initial scan time monitoring time 84
Initialization of labels after rebuilt all (reassignment) 534
Initializing the CPU module 43
Input (X) 76,464
INT 509
Inter-module synchronous interrupt (I44) 76
Internal buffer 286
Internal current consumption (5VDC) 74
Internal relay (M) 76,465
Interrupt factor 499
Interrupt pointer (I) 77,499
Interrupt pointer number 499
Interrupt priority 129
Interrupt using the internal timer (I28 to I31) 76
IP filter 368

J

Jump instructions 496

L

L ERR LED 34
Label 46
Label area 74
Label initial value 530
Label Initial value reflection setting 536
Latch (1) 523
Latch (2) 523
Latch clear 529
Latch label area 74
Latch relay (L) 76,465
LCOUNTER 509,510
Librarization of programs 111
Link direct device 77,483
LINK LED 34
Link refresh 485
Link register (W) 76,480
Link relay (B) 76,465
Link special register (SW) 76,480
Link special relay (SB) 76,467
Local label 508,511
Local pointer 497
LOGTRG instruction 270
LOGTRGR instruction 304
Long counter 509,510
Long counter (LC) 76,477
Long index register (LZ) 77
Long retentive timer 509,510
Long retentive timer (LST) 76,470
Long timer 509,510
Long timer (LT) 76,469
Low-speed timer 470
Low-speed timer (T/ST) 470
LREAL 509
LRETENTIVETIMER 509,510
LTIMER 509,510

M

Master control instructions 495
Maximum counting speed for counters 479
Memory capacity 74
Memory card parameters 749
Memory configuration 139
Memory copy 409
MEMORY COPY LED 33
Module access device 77,486
Module communication test 434
Module extension parameter 50
Module label 507
Module label assignment area 494
Module parameter 50
Momentary power failure 138
Monitor function 231
Multiple CPU synchronous interrupt 366
Multiple CPU synchronous interrupt (I45) 76
Multiple CPU synchronous interrupt program 366
Multiple CPU system 337
Multiple interrupt 129
Multiple interrupt function 129

N

Nesting (N) 77,495
Nesting structure 495
Network No. specification device (J) 501
New control system 25
New standby system 25
Non-controlled module 338
Number of array elements 515
Number of FB files 76
Number of I/O points 76
NZ2MC-2MBSE 78
NZ2MC-8MBSE 78

O

Online change 189
Online change (ladder block) 189
Operation mode 159,374
Optical connector (IN) 34
Optical connector (OUT) 34
Output (Y) 76,464
Output mode at operating status change (STOP to RUN) 137
Overflow 539
Own system 25

P

Parameter 43
Parameter No. 747
PAUSE state 135
PID control 308
POINTER 509
Pointer 509

Pointer (P) 77,496
Pointer setting 498
Priority for interrupt factors 500
Process control function 309
Process control instruction 310
Process CPU 26
Process mode 25
Processing time 723
Production information marking 30,34
Program arrangement change 111
Program block 508
Program cache memory 140
Program capacity 74
Program executed in both systems 25
Program execution type 101
Program file 50
Program list monitor 84
Program memory 74,140
PROGRAM RUN LED 29,52
Programming language 41,76

Q

Q6BAT 38

R

READY LED 29,545
REAL 509
Real constant (E) 539
Real number 539
Redundant extension base unit 26
Redundant function module 25
Redundant mode 25
Redundant system 25
Redundant system with redundant extension base unit 25
Refresh 355
Refresh data register (RD) 77,494
Refresh memory 74,145
Refresh memory setting 494
Remote head module 26
Remote operation 219
Remote password 368
Remote RESET 223
Remote RUN 219
Remote STOP 219
Reserved area for online change 191
Resetting counters 478
Resetting the CPU module 51
Retentive timer 509,510
Retentive timer (ST) 76,470
RETENTIVETIMER 509,510
RUN LED 33
RUN state 135
RUN/STOP/RESET switch 29,51,52
RUN-PAUSE contacts 222

S

Saving/restoring of the file register (R) block number 126
Saving/restoring of the index register (Z, LZ) . . . 127
SBY LED 33
Scan execution type program 102
Scan monitoring function 200
Scan time 84
Scan time monitoring time 200
Scan time monitoring time setting 200
SD CARD OFF button 29
SD memory card 40,74,147
SD/RD LED 29
Security function 368
Security key authentication 368
SEPARATE LED 33
Separate mode 25
Serial number method 492
Setting RUN-PAUSE contacts 222
Setting time zone 187
SFC block device (BL) 77,501
SFC transition device (TR) 77,501
Shift JIS 509
Signal flow memory 74,146,399,447
Single-precision real number 509,539
Special register (SD) 77,482
Special relay (SM) 77,482
SPEED LED 29
Stack file 287
Standby system 25
Standby type program 111
Station sub ID number 26
Step relay (S) 76,468
STOP state 135
Storage file 289
STRING 509
String 509
String [Unicode] 509
Structure 510,516
Structure array 517
Subroutine call instruction 114
SYS A LED 33
SYS B LED 33
System A 26
System B 26
System clock 188
System consistency check 417
System device 481
System label 507
System parameter 50
System parameters 747
System switching 183,377,658

T

TIME 509
Time 509
Time setting 185
Time zone 187
TIMER 509,510
Timer 509,510
Timer (T) 76,469
Timer current value 471
Timer limit setting 471
Timer time limit value 471
Tracking cable 26
Tracking transfer 183,391,658
Trigger condition 269
Trigger instruction 270
Trigger logging 267
Troubleshooting by symptom 548

I

U

Underflow 539
Unicode 509
USB port 29,74
User device 464
USER LED 29

W

Watchdog timer 200
Weight 75,78
WORD 509
Word [signed] 509
Word [unsigned]/bit string [16 bits] 509
WSTRING 509

REVISIONS

*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
July 2022	SH(NA)-082493ENG-A	First edition
October 2022	SH(NA)-082493ENG-B	■Added or modified parts Chapter 24, Section 26.4, 36.4, Appendix 7, 12, 13
June 2024	SH(NA)-082493ENG-C	■Added or modified parts Section 26.4, Appendix 7
October 2024	SH(NA)-082493ENG-D	■Added or modified parts RELEVANT MANUALS, Chapter 1, Section 10.4, Chapter 12, Section 15.3, Chapter 26, Section 26.2, 26.5, 26.7, 26.10, 36.4, 37.1, 37.2, Appendix 13
April 2025	SH(NA)-082493ENG-E	■Added or modified parts RELEVANT MANUALS, Chapter 12, Section 17.4, 18.1, 31.1, 37.2, Appendix 4, 5, 7, 8, 13

Japanese manual number: SH-082492-E

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2022 MITSUBISHI ELECTRIC CORPORATION

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.

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.

INFORMATION AND SERVICES

For further information and services, please contact your local Mitsubishi Electric sales office or representative.
Visit our website to find our locations worldwide.

MITSUBISHI ELECTRIC Factory Automation Global Website
Locations Worldwide
www.MitsubishiElectric.com/fa/about-us/overseas/

TRADEMARKS

Microsoft, Excel, and Windows 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.

COPYRIGHTS

The screens (screenshots) are used in accordance with the Microsoft Corporation guideline.

SH(NA)-082493ENG-E(2504)MEE
MODEL: RNPCPU-U-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

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.