MITSUBISHI ELECTRIC

Programmable Controller

MELSEC iQ-R series

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

-RJ71CN91

SAFETY PRECAUTIONS

(Read these precautions before using this product.)

Before using this product, please read this manual and the relevant manuals carefully and pay full attention to safety to handle the product correctly.

The precautions given in this manual are concerned with this product only. For the safety precautions of the programmable controller system, refer to the MELSEC iQ-R Module Configuration Manual.

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.
- For the operating status of each station after a communication failure, refer to manuals relevant to the network. Incorrect output or malfunction due to a communication failure may result in an accident.
- 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. Failure to do so may result in an accident due to an incorrect output or malfunction.
- To maintain the safety of the programmable controller system against unauthorized access from external devices via the network, take appropriate measures. To maintain the safety against unauthorized access via the Internet, take measures such as installing a firewall.

[Design Precautions]

⚠CAUTION

- Do not install the control lines or communication cables together with the main circuit lines or power cables. Keep a distance of 100mm or more between them. Failure to do so may result in malfunction due to noise.
- 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.

[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 included with the base unit. 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 screw, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
- When using an extension cable, connect it to the extension cable connector of the base unit securely. Check the connection for looseness. Poor contact may cause malfunction.
- When using an SD memory card, fully insert it into the SD memory card slot. Check that it is inserted completely. Poor contact may cause malfunction.
- Securely insert an extended SRAM cassette or a battery-less option cassette into the cassette connector of the CPU module. After insertion, close the cassette cover and check that the cassette is inserted completely. Poor contact may cause malfunction.
- Do not directly touch any conductive parts and electronic components of the module, SD memory card, extended SRAM cassette, battery-less option cassette, or connector. Doing so can cause malfunction or failure of the module.

[Wiring Precautions]

⚠WARNING

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

[Wiring Precautions]

⚠CAUTION

- Individually ground the FG and LG terminals of the programmable controller with a ground resistance of 100 ohms or less. Failure to do so may result in electric shock or malfunction.
- Use applicable solderless terminals and tighten them within the specified torque range. If any spade solderless terminal is used, it may be disconnected when the terminal screw comes loose, resulting in failure.
- Check the rated voltage and signal layout before wiring to the module, and connect the cables correctly. Connecting a power supply with a different voltage rating or incorrect wiring may cause fire or failure.
- Connectors for external devices must be crimped or pressed with the tool specified by the manufacturer, or must be correctly soldered. Incomplete connections may cause short circuit, fire, or malfunction.
- 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. Keep a distance of 100mm or more between them. Failure to do so may result in malfunction due to noise.
- 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, fire, 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.
- Prevent foreign matter such as dust or wire chips from entering the module. Such foreign matter can cause a fire, failure, or malfunction.
- A protective film is attached to the top of the module to prevent foreign matter, such as wire chips, from entering the module during wiring. Do not remove the film during wiring. Remove it for heat dissipation before system operation.
- 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.

[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 or battery-less option cassette to/from the CPU module
  - Mounting/removing the terminal block to/from the module
- After the first use of the product, do not insert/remove the SD memory card to/from the CPU module more than 500 times. Exceeding the limit may cause malfunction.
- Do not touch the metal terminals on the back side of the SD memory card. Doing so may cause malfunction or failure of the module.
- Do not touch the integrated circuits on the circuit board of an extended SRAM cassette or a battery-less option cassette. Doing so may cause malfunction or failure of the module.
- Do not drop or apply shock to the battery to be installed in the module. Doing so may damage the battery, causing the battery fluid to leak inside the battery. If the battery is dropped or any shock is applied to it, dispose of it without using.

[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. 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 can cause malfunction or failure of the module.

[Disposal Precautions]

⚠CAUTION

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

[Transportation Precautions]

⚠CAUTION

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

CONDITIONS OF USE FOR THE PRODUCT

(1) Mitsubishi 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 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'S USER, 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 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 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 representative in your region.

INTRODUCTION

Thank you for purchasing the Mitsubishi Electric MELSEC iQ-R series programmable controllers.
This manual describes the functions, parameter settings, programming, and troubleshooting of the relevant product 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 product

RJ71CN91

CONTENTS

RELEVANT MANUALS

Manual name [manual number]	Description	Available form
MELSEC iQ-R CANopen Module User's Manual (Application) [SH-081736ENG] (this manual)	Functions, parameter settings, programming, troubleshooting, I/O signals, buffer memory, and object dictionary of the CANopen module	Print book e-Manual PDF
MELSEC iQ-R CANopen Module User's Manual (Startup) [SH-081734ENG]	Performance specifications, procedures before operation, system configuration, wiring, and communication examples of the CANopen module	Print book e-Manual PDF

Point

e-Manual refers to the Mitsubishi Electric FA electronic book manuals that can be browsed using a dedicated tool.

e-Manual has the following features:

- Required information can be cross-searched in multiple manuals.
- Other manuals can be accessed from the links in the manual.
- The hardware specifications of each part can be found from the product figures.
- Pages that users often browse can be bookmarked.
- Sample programs can be copied to an engineering tool.

TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
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.
Device	A device (X, Y, M, D, or others) in a CPU module
Engineering tool	A tool used for setting up programmable controllers, programming, debugging, and maintenance
Global label	A label that is valid for all the program data when multiple program data are created in the project. There are two types of global label: a module specific label (module label), which is generated automatically by GX Works3, and an optional label, which can be created for any specified device.
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 label that represents a device in a given character string
Module label	A label that represents one of memory areas (I/O signals and buffer memory areas) specific to each module in a given character string. For the module used, GX Works3 automatically generates this label, which can be used as a global label.

GENERIC TERMS AND ABBREVIATIONS

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

Generic term/abbreviation	Description
CDCF	An abbreviation for the concise device configuration file.
CIF	An abbreviation for the command interface.
CPU module	A generic term for the MELSEC iQ-R series CPU module
RTR	An abbreviation for the remote transmission request.

1 FUNCTIONS

1.1 Overview of CANopen and CAN

This section describes overview of CANopen and CAN.

Function modes

The RJ71CN91 has three function modes. Available functions and buffer memory assignment vary depending on the function mode. (☞ Page 159 Buffer Memory)

The following table lists and describes the overview of the function modes.

Function mode	Overview
CANopen 405 mode	This mode is used to communicate with CANopen nodes. CAN nodes cannot be used.
11-bit CAN-ID Layer 2 message mode	This mode is used to communicate with CAN nodes. CANopen nodes cannot be used. The standard format (11-bit CAN-ID) is used for communications.
29-bit CAN-ID Layer 2 message mode	This mode is used to communicate with CAN nodes. CANopen nodes cannot be used. The extended format (29-bit CAN-ID) is used for communications.

The following table shows availability of functions in each function mode.

○: Available, ×: Not available

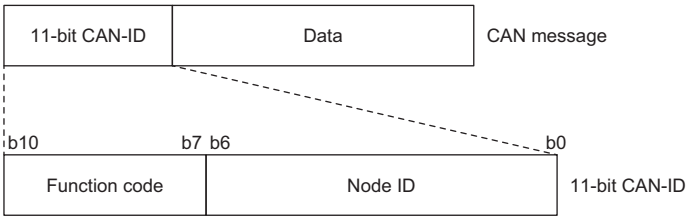
Function	CANopen 405 mode	11-bit CAN-ID Layer 2 message mode, 29-bit CAN-ID Layer 2 message mode
NMT	○	×
SDO	○	×
PDO	○	×
SYNC	○	×
TIME	○	×
EMCY	○	×
Node guarding	○	×
Heartbeat	○	×
Operation setting at error occurrence	○	×
Layer 2 message transmission and receive	×	○
CPU module STOP transition message	×	○

CAN-ID

CAN-ID is an identifier used for bus arbitration and identification of CAN messages.
In bus arbitration, a CAN message with a smaller ID has higher priority to use the bus. CAN-IDs with lower priority will wait until the bus is free.
This section describes CAN-IDs used in each function mode.

■CANopen 405 mode

11-bit CAN-IDs are used. As shown below, in CANopen, a CAN-ID is composed of a node ID and function code.



In CANopen, each function comes with a function code. However, for some functions, a CAN-ID can be set as desired. The following table shows CAN-IDs used for the functions of this module and whether CAN-IDs can be set as desired.
○: Can be set, ×: Cannot be set

Function	Optional setting	Function code (binary)	Node ID	CAN-ID
NMT	×	0000	00H*1	0000H
SYNC	○*2	0001	00H*1	0080H
EMCY	×*3	0001	01H to 7FH	0081H to 00FFH
TIME	○*2	0010	00H*1	0100H
TPDO	○	Optional*4	Optional*4	Optional*4
RPDO	○	Optional*4	Optional*4	Optional*4
SDO (Request)	×	1011	01H to 7FH	0581H to 05FFH
SDO (Response)	×	1100	01H to 7FH	0601H to 067FH
NMT error control	×	1110	01H to 7FH	0701H to 077FH

*1 Broadcast message; Node ID is fixed as 0.
*2 Cannot be set with the parameter of CANopen Configuration Tool. Specify the value with the object dictionary.
For details on the setting method, refer to the following.
☞ Page 228 How to Set Parameters When Not Using CANopen Configuration Tool
*3 The CAN-ID of EMCY to be sent by the own node cannot be changed. However, the CAN-ID of a message to be received as EMCY can be set as desired for the ID of each target node.
*4 To set the CAN-ID as desired, it is not possible to use the same CAN-ID as the CAN-ID of another function used in the same network, or the CAN-ID that is use-prohibited (reserved) in CANopen. The following table shows whether or not each CAN-ID (range) is available for use as desired in this module.

○: Available, ×: Not available

CAN-ID	Availability	Remarks
0000H	×	Used in the NMT function.
0001H to 007FH	×	Use prohibited in CANopen.
0080H	×	Used in the SYNC function.
0081H to 00FFH	×	Used in the EMCY function.
0100H	×	Used in the TIME function.
0101H to 0180H	×	Use prohibited in CANopen.
0181H to 0580H	○*5	—
0581H to 5FFH	×	Used in the SDO function.
0600H	○*5	—
0601H to 067FH	×	Used in the SDO function.
0680H to 06DFH	○*5	—
06E0H to 06FFH	×	Use prohibited in CANopen.
0701H to 077FH	×	Used in NMT error control.
0780H to 07FFH	×	Use prohibited in CANopen.

*5 Do not use the CAN-ID used in the network by another function. The duplication of the CAN-ID with another function may cause malfunction.

■11-bit CAN-ID Layer 2 message mode

11-bit CAN-IDs are used. Set any value for each node and each CAN message.

■29-bit CAN-ID Layer 2 message mode

29-bit CAN-IDs are used. Set any value for each node and each CAN message.

Object dictionary

Object dictionary is a structure with specifications defined in CANopen. It is used in the CANopen 405 mode.

The data within the object dictionary is used to set CAN bus parameters, initialize special functions, control data flow, store data in various formats and send EMCY messages. (☞ Page 191 Object Dictionary)

To read and write data from/to the object dictionary, use the SDO function. (☞ Page 29 SDO)

1.2 NMT

NMT is a function to manage the CANopen network status. The NMT master controls all NMT slaves within the CANopen network.

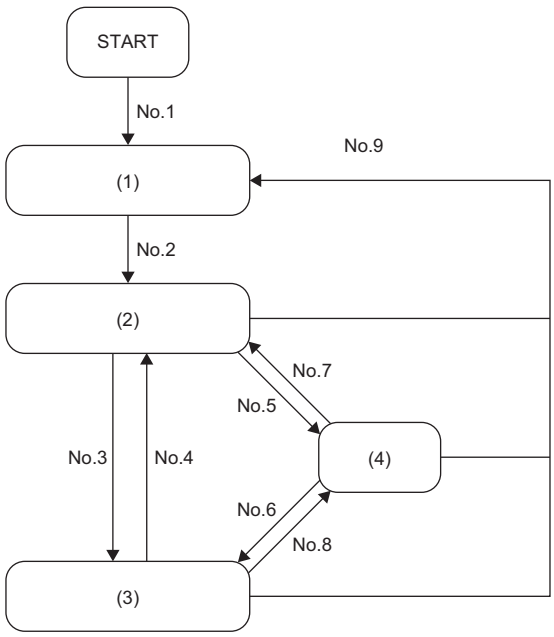
NMT state

NMT state is the operating status of a CANopen node. When the NMT state is Operational, the CANopen node is capable of all communications.

The following table lists the NMT states for the RJ71CN91.

NMT state	Description
Initialization	The CANopen node is being initialized.
Pre-operational	Communications other than PDO are possible. (SDO, SYNC, EMCY, TIME, Node control, and Heartbeat/Node-guarding communications are possible.) Changing the object dictionary regarding PDO, setting and changing CANopen node parameters are possible. CANopen nodes that are powered off and on or reset will enter this state automatically.
Operational	All communications are possible. (PDO, SDO, SYNC, EMCY, TIME, Node control, and Heartbeat/Node-guarding communications are possible.) A CANopen node transitions to this state automatically or requires node control by the NMT master. When all nodes enter this state, start-up of the CANopen network is completed.
Stop	All communications are stopped. (Only Node control and Heartbeat/Node-guarding communications are possible.) A CANopen node transitions to this state when it has an error or is executing a specific application.

The following figure shows the NMT state transition paths (No.1 to 9).



(1) Initialization
(2) Pre-operational
(3) Operational
(4) Stop

Path Number	Condition for the NMT state to change	
	NMT master	NMT slaves
1	Power off and on or reset	Power off and on or reset
2	CANopen node initialization is completed.	CANopen node initialization is completed.
3	• NMT start-up is enabled.*1 • Remote node start is accepted.*3	• Remote node start is accepted.*3
4	• A communication error is detected.*2 • Pre-operational transition is accepted.*3	• A communication error is detected.*2 • Pre-operational transition is accepted.*3
5	• Remote node stop is accepted.*3	• Remote node stop is accepted.*3

Path Number	Condition for the NMT state to change	
	NMT master	NMT slaves
6	• Remote node start is accepted.*3	• Remote node start is accepted.*3
7	• Pre-operational transition is accepted.*3	• Pre-operational transition is accepted.*3
8	• A communication error is detected.*2 • Remote node stop is accepted.*3	• A communication error is detected.*2 • Remote node stop is accepted.*3
9	• Node reset is accepted.*3 • Communication reset is accepted.*3	• Node reset is accepted.*3 • Communication reset is accepted.*3

*1 For details, refer to NMT master/slave setting. (☞ Page 96 "NMT master / slave" window)

*2 For details, refer to Operation setting at error occurrence. (☞ Page 61 Operation Setting at Error Occurrence)

*3 For details, refer to Node control. (☞ Page 21 Node control)

Node control

Node control is a function to control the NMT state of a CANopen node via the NMT master.
It can also control the NMT state of other NMT master from the own NMT master. However, the NMT state of an active NMT master can be controlled only by the active NMT master itself.

Function details

The node control function provides the following controls over any CANopen node or all CANopen nodes.

Node control	Description	Remarks
Remote node stop	Changes the NMT state of the specified CANopen node to "Stop".	—
Remote node start	Changes the NMT state of the specified CANopen node to "Operational".	—
Pre-operational transition	Changes the NMT state of the specified CANopen node to "Pre-operational".	—
Node reset	Resets the specified CANopen node. (Changes the NMT state of the specified CANopen node to "Initialization".)	The CANopen node and communications settings are reset to those at the time of power-on.
Communication reset	Resets communications of the specified CANopen node. (Changes the NMT state of the specified CANopen node to "Initialization".)	The communications setting is reset to the setting at the time of power-on.
Remote node stop excluding NMT master	Changes the NMT state of the specified CANopen node to "Stop". Note that if the specified CANopen node is the NMT master, the NMT state cannot be changed.	—
Remote node start excluding NMT master	Changes the NMT state of the specified CANopen node to "Operational". Note that if the specified CANopen node is the NMT master, the NMT state cannot be changed.	—
Pre-operational transition excluding NMT master	Changes the NMT state of the specified CANopen node to "Pre-operational". Note that if the specified CANopen node is the NMT master, the NMT state cannot be changed.	—
Node reset excluding NMT master	Resets the specified CANopen node. (Changes the NMT state of the specified CANopen node to "Initialization".) Note that if the specified CANopen node is the NMT master, the NMT state cannot be changed.	The CANopen node and communications settings are reset to those at the time of power-on.
Communication reset excluding NMT master	Resets communications of the specified CANopen node. (Changes the NMT state of the specified CANopen node to "Initialization".) Note that if the specified CANopen node is the NMT master, the NMT state cannot be changed.	The communications setting is reset to the setting at the time of power-on.

Point

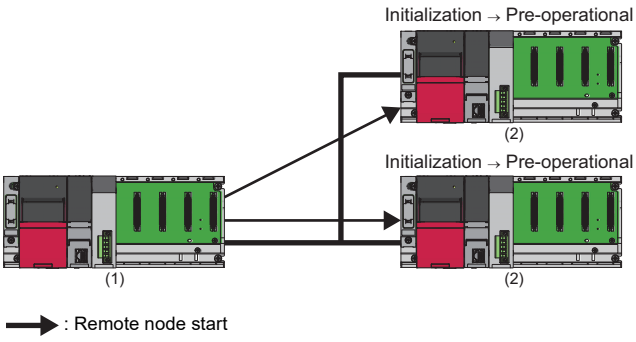
- Check 'NMT state' (Un\G601 to Un\G727) to confirm the NMT state of the CANopen node. (☞ Page 174 NMT state (Un\G601 to Un\G727))
- The node control can be executed by writing a request to RequestNMT of the active NMT master in SDO communication.
- The control over a node other than the active NMT master can be executed with RequestNMT of a node other than active NMT master as well. To change any node to Stop or Pre-operational state, however, use RequestNMT of the active NMT master only. If RequestNMT of the active NMT master is not used, the target node may not be changed to specified NMT state because an error event control of NMT master operates.

■Setting method

Use the object dictionary to set the node control function. For details, refer to the following.
☞ Page 222 Request NMT

NMT start-up

NMT start-up is a function to start up the network by setting parameters in NMT slaves (2) that were detected in the network when the NMT master (1) starts.



Function details

The NMT state of a CANopen node automatically transitions to "Pre-operational" when the node is powered off and on or reset.

Note that "Remote node start" from the NMT master is required for the NMT state of the NMT slaves to transition to "Operational".

The NMT start-up function can change the NMT state of each NMT slave in the network to "Operational".

In the NMT start-up function, when the module is powered off and on or reset, the NMT master starts the NMT slaves and makes itself and the NMT slaves "Operational" to allow the network to start up automatically.

The processing when the RJ71CN91 module executes the NMT startup function is explained below.

1. When the power is turned off and on, a reset is performed, or the CPU module switches from STOP state to RUN state, the NMT startup function is executed.

2. Checks whether the own node is the NMT master.

Start up as the NMT master when the NMT master is enabled.

Start up as an NMT slave when the NMT master is disabled. (The process ends.)

3. Check if the flying master function is enabled.

Start up as the NMT master when the flying master function is disabled.

Compare the priority between the flying masters when the flying master function is enabled.

Start up as the NMT master (active NMT master) when the own node is the highest priority.

Start up as an NMT slave (standby master) when another node is the highest priority. (The process ends.)

4. For the NMT slaves, check the existence of an NMT slave where the communication reset condition is disabled.

When an invalid NMT slave for the communication reset condition exists, execute a communication reset for each NMT slave other than the nodes where the communication reset condition is disabled.

When the communication reset condition for all NMT slaves is enabled, execute a communication reset for all nodes.

5. Start a boot start process for each NMT slave.

Operations listed in the table below are performed in boot processing.

If any operation fails, it is regarded as a boot failure.

Item	Description
NMT boot slave	Execute the following operations when NMT boot slave is enabled. • NMT slave verification • Configuration manager
Node guarding/heartbeat	Start monitoring when node guarding or heartbeat is enabled.
Remote node start (To NMT slave)	Execute Remote node start (To NMT slave) when NMT boot slave and Start node are enabled and Start all nodes is disabled.
LSS	Execute LSS if Identity Object is set when the boot time is timeout without detection of a mandatory slave and a slave. NMT start up processing fails after the LSS execution.

6. Check if all mandatory slaves have been booted successfully.

When the boot start process fails, to start up each NMT slave again, the NMT master must be reset. (The process ends.)

7. Check if the state of the NMT master should be automatically transitioned to "Operational".

When the NMT master start is enabled, the state of the NMT master automatically transitions to "Operational" at startup.
When the NMT master start is disabled, the state of the NMT master does not automatically transition to "Operational" at startup. Processing is stopped until the NMT master transitions to "Operational".

8. Execute Remote node start for all nodes through node control when Start node and Start all nodes are enable. If a slave that failed boot process exists, execute Remote node start to each NMT slave that succeeded in boot process.

9. Check if all NMT slaves should start up at the same time.

Execute the Remote node start to all nodes through node control when both Start node and Start all nodes are enable.
If a slave that failed boot process exists, execute Remote node start to each NMT slave that succeeded in boot process.

■Setting method

Use parameters to set the NMT start-up function. For details, refer to the following.

☞ Page 96 "NMT master / slave" window

Error event control

When the NMT master detects an error in an NMT slave, this function controls the status of the NMT slave or network.
An error in an NMT slave is detected by node guarding or heartbeat.
For details on node guarding and heartbeat, refer to the following.
☞ Page 57 Node Guarding
☞ Page 59 Heartbeat
The following table shows the actions of the NMT master when detecting an error in an NMT slave.

Parameter condition*1			Action
The target node is not assigned as an NMT slave.	—	—	No action
The target node is assigned as an NMT slave.	The target node is a mandatory slave.	"Stop all nodes" is enabled.	Execute remote node stop for all NMT slaves.
		"Stop all nodes" is disabled, and "Reset all nodes" is enabled.	Execute communication reset for all NMT slaves.
		"Stop all nodes" is disabled, and "Reset all nodes" is disabled.	Execute communication reset for the target node. Then, restart (boot) the target node.
	The target node is not a mandatory slave.	—	

*1 For details on the parameter setting operation, refer to the following.
☞ Page 96 "NMT master / slave" window,Page 98 "NMT slave assignment" window

Boot-up control

After NMT startup, if the NMT master detects a new NMT slave, the detected NMT slave is started up.

Function details

When the power is turned off and on or a reset is performed, if the status of an NMT slave changes from Initialization to Pre-operational, a boot-up message is sent.
When the NMT master detects a boot-up message for an NMT slave, the detected NMT slave is started up.
For details on the status of an NMT slave, refer to the following.
☞ Page 19 NMT state

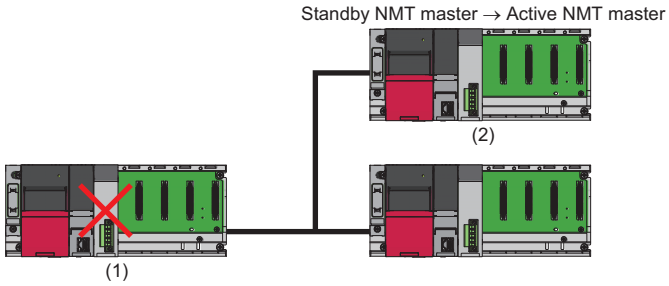
Point

The NMT slave will not start up in either of the following cases:
- The NMT state of the NMT master is Stop.
- The target NMT slave is not registered for slave assignment in the parameter setting.

For details on the parameter setting operation, refer to the following.
☞ Page 98 "NMT slave assignment" window

Flying master

Flying master is a function to continue network control by making one of its own nodes to become the NMT master (2) when the NMT master in the same network is in failure (1).
Among those flying masters, the CANopen node that works as the actual NMT master is called the active NMT master, and the other CANopen nodes are called the hot standby NMT masters.



Function details

Flying masters monitor errors of each other via heartbeat. (☞ Page 59 Heartbeat)
When an error of the active NMT master is detected via heartbeat, NMT flying master negotiation is executed so that the flying master with the highest priority becomes the active NMT master.

■Flying master priority

The flying master priority is determined in the following way.

- A flying master with a higher NMT master priority level has a higher flying master priority. (High > Medium > Low)
- If flying masters have the same NMT master priority level, a flying master with a smaller node ID has higher priority.

■Setting method

Use parameters to set flying masters. For details, refer to the following.
☞ Page 96 "NMT master / slave" window

■Precautions

When using the flying master function, pay attention to the following.

- When the active NMT master is in failure, network communications will be reset and applications will be interrupted.
- The flying master function does not provide data synchronization with applications. Data synchronization requires appropriate CANopen setting and system design.
- Configure the flying master setting properly. NMT flying master negotiation may not work properly if the setting is incorrect. Also, test the system configuration before using the flying master function.

Point

- When using the flying master function, set all NMT masters in the same network as flying masters.
- When the flying master setting of the RJ71CN91 is enabled, and the "Producer heartbeat time" is not set, the "Producer heartbeat time" will automatically set to 1000ms. (☞ Page 102 "Heartbeat" window)
- When the RJ71CN91 is a hot standby NMT master and the flying master setting is enabled, if the Consumer setting for the active NMT master is not configured, the "Heartbeat time" for the active NMT master will be automatically set to (1500 + 10 × RJ71CN91 node ID) [ms]. (☞ Page 102 "Heartbeat" window)
- When manually setting the Consumer heartbeat for each flying master, set a different heartbeat time for the same node ID. (☞ Page 102 "Heartbeat" window)
- When a flying master other than the RJ71CN91 exists in the same network, enable heartbeat message transmission from that flying master. If it is not enabled, the RJ71CN91, when it becomes the active NMT master, will determine that this flying master is faulty and continue to reset communications with this flying master.
- All flying masters must have the same configuration for the NMT slaves. (☞ Page 96 "NMT master / slave" window)
- Set parameters so that a flying master with higher priority has a shorter "NMT flying master response waiting time". (☞ Page 102 "Heartbeat" window)

Use the following formula to calculate the "NMT flying master response waiting time".

"NMT flying master response waiting time" = "NMT master priority level" setting value × "Priority time slot" setting value + Node ID x "Node time slot" setting value

- Communications with all CANopen nodes will be reset during NMT flying master negotiation.

LSS

LSS is a function to configure a CANopen node via the network in case the node ID and baud rate cannot be set via switches.

Function details

When the LSS master starts, an LSS slave is detected based on NMT slave identification information, and its baud rate and node ID are set.

Only the active NMT master can run the LSS master function.

The RJ71CN91 supports only the LSS master function. The LSS slave function is not supported.

To enable LSS of the RJ71CN91, configure all the following CANopen settings.

Category		Item	Setting value	Index - sub-index
NMT setting	NMT master/slave	Boot time	A longer time than the target node boot time	1F89H-00H
	Slave	NMT slaves	Check	1F81H-n*1.bit 0
		NMT boot slave	Check	1F81H-n*1.bit 2
		Mandatory slave	Check	1F81H-n*1.bit 3
		Device type	Identification information of the target node	1F84H-n*1
		Vendor ID	Identification information of the target node	1F85H-n*1
		Product code	Identification information of the target node	1F86H-n*1
		Revision number	Identification information of the target node	1F87H-n*1
		Serial number	Identification information of the target node	1F88H-n*1

*1 n corresponds to the node ID.

If no CANopen node is detected in communications at the baud rate set in the RJ71CN91, the RJ71CN91 will change the baud rate automatically to detect a CANopen node. Note that because communications are executed at a different baud rate, the CANopen node in the network may be in the bus-off state. If the CANopen node does not support automatic recovery from the bus-off state or takes some time to recover, the LSS master cannot set baud rate and node ID for the CANopen node.

To prevent unexpected start of the LSS master function, connect a CANopen node and the LSS master on a one-to-one bases at the time of configuration. When the configuration is complete, remove the serial number from the LSS master.

Point

- Some LSS slaves may have a built-in terminating resistor. In that case, enable or disable the built-in terminating resistor according to the wiring condition.

Configuration manager

The configuration manager sets parameters of an NMT slave when the NMT master starts the NMT slave.

Function details

The NMT master configures an NMT slave according to the CDCF when starting the NMT slave.
Available setting items vary depending on the target slave. For details, refer to the manual and EDS file of the target slave.
To set parameters of any NMT slave at any timing, write 666E6F63H (ISO8859 string code fnoc(conf)) in the Configuration request (index 1F25H subindex 01H to 80H) of the NMT master via SDO.

■Setting method

Register a CDCF to the RJ71CN91 to enable the configuration manager. The following two methods are available to register a CDCF to the RJ71CN91.

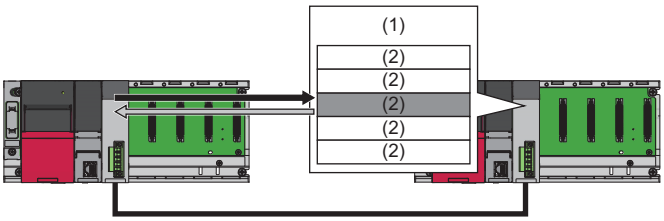
- Use the CANopen configuration software to register a CDCF via the CANopen network. (☞ Page 218 Concise DCF)
- Create a CDCF with CANopen Configuration Tool and register it. (☞ Page 85 CANopen node list)

Point

- The configuration manager can only be used on the active NMT master.
- The RJ71CN91 cannot set its parameters via the configuration manager.
- A maximum of 60 CDCFs can be registered with the RJ71CN91. The maximum size of a file is 65531 bytes.
- Use a configuration tool for CANopen to create a CDCF based on the EDS file of the target slave.
- An EDS file defines CANopen device information.

1.3 SDO

SDO is a function to directly access an object entry in the object dictionary of any CANopen node. The RJ71CN91 can read and write data in the object dictionary of other CANopen nodes or own nodes.



(1) Object dictionary
(2) Object entry
: Request (Read/Write)
: Response

Use 'Command interface (CIF)' (Un\G1000 to Un\G1066) to run the SDO.
The following table lists the commands that can run with the SDO function. Each command is detailed below using examples of the RJ71CN91 accessing other CANopen nodes.

Command	Description	Data size
SDO read	Reads data from any CANopen node.	124 bytes maximum
SDO multi read	Executes multiple SDO read operations in one command. SDO read can be executed a maximum of eight times.	8 bytes maximum for each SDO read
SDO write	Writes data to any CANopen node.	124 bytes maximum
SDO multi write	Executes multiple SDO write operations in one command. SDO write can be executed a maximum of eight times.	8 bytes maximum for each SDO write

SDO abort codes

An SDO abort code is an error code of the communication error that occurred in SDO.
When an SDO execution source or access destination detects an error, the error code is issued to the execution source.
The SDO abort code is stored in the receive message as an additional information of the SDO error by using the command interface.
(☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066))
The following table lists the common SDO abort codes in CANopen.

Abort code	Description
05030000H	The toggle bit was not changed.
05040000H	SDO protocol timeout (RJ71CN91 initial value: 500ms)
05040001H	The client/server command specifier is invalid or unknown.*1
05040002H	Invalid block size (Block mode only)
05040003H	Invalid sequence number (Block mode only)
05040004H	CRC error (Block mode only)
05040005H	Memory shortage
06010000H	Access to an unsupported object
06010001H	Attempted to read a write-only object.
06010002H	Attempted to write to a read-only object.
06020000H	No object exists in the object dictionary.
06040041H	Cannot map an object to PDO.
06040042H	The number and data length of objects to be mapped exceed the data length of PDO.
06040043H	Incompatible general parameters
06040047H	General internal incompatibility of the device
06060000H	Access failure due to an hardware error
06070010H	Data type mismatch, service parameter data length mismatch
06070012H	Data type mismatch, the service parameter data length is too long.
06070013H	Data type mismatch, the service parameter data length is too short.
06090011H	No sub-index exists.
06090030H	Invalid parameter value (write access only)
06090031H	The value of the written parameter is too large.
06090032H	The value of the written parameter is too small.
06090036H	The maximum value is smaller than the minimum value.
060A0023H	Resource not available for the SDO connection
08000000H	Generic error
08000020H	Cannot send or save data in the application.
08000021H	Cannot send or save data in the application. (due to local control)
08000022H	Cannot send or save data in the application. (due to the current device status)
08000023H	Object dictionary dynamic generation failed or the object dictionary does not exist.
08000024H	Data not available
50000000H	Timeout, identifier assignment to SDO transmission not available, or protocol mismatch
60600000H	The buffer for the received SDO data is too small (generated at initialization of transmission).

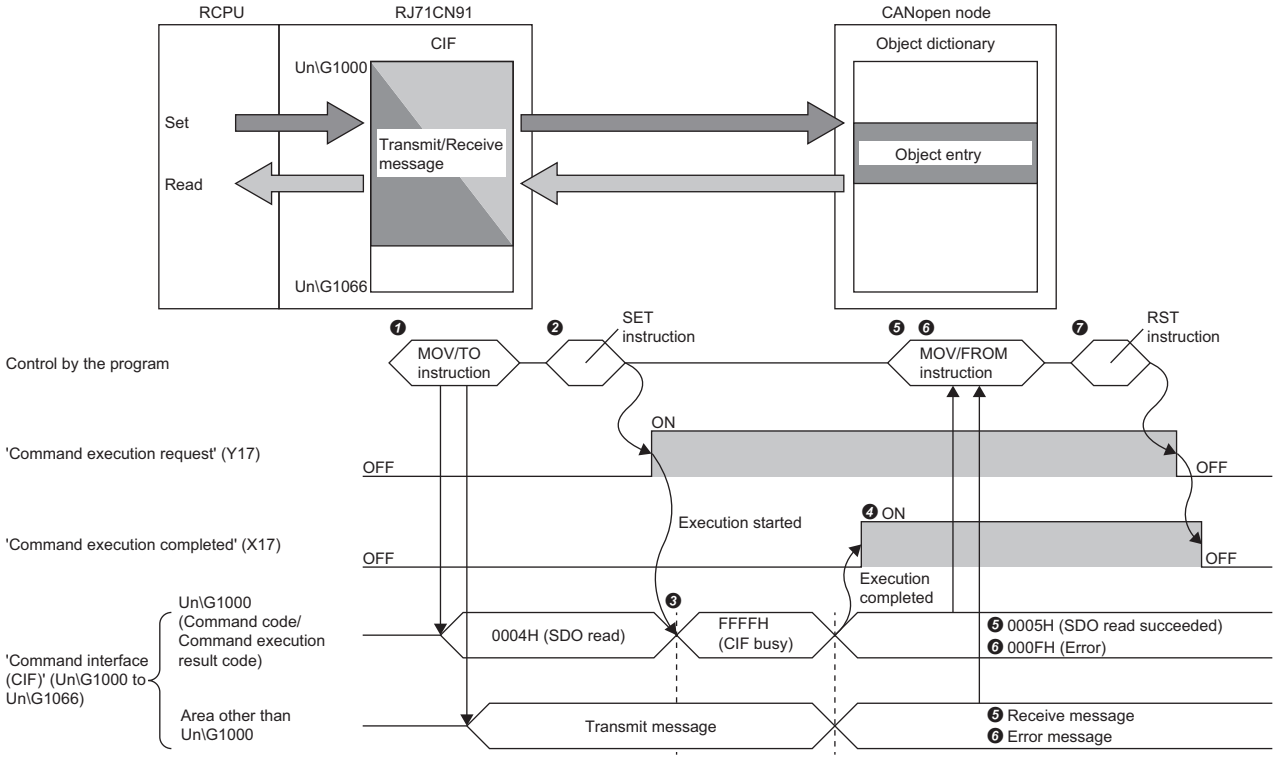
*1 For the RJ71CN91, this code is generated also when the node ID is outside the specified range.

Point

Other than the above, various SDO abort codes are defined in various CiA devices and application profiles and by each device manufacturer. For SDO abort codes not listed above, refer to the manual of the CANopen node that returned the SDO abort code.

SDO read

The following is the procedure for executing the SDO read.



❶ Set the transmit message in 'Command interface (CIF)' (Un\G1000 to Un\G1066) via the program. (☞ Page 32 Buffer memory area assignment)
❷ Turn on 'Command execution request' (Y17) via the program.
❸ FFFFH (CIF busy) is stored in Un\G1000 while the command is being processed.
❹ When the command execution is finished, 'Command execution completed' (X17) turns on.
❺ When the process is completed successfully, 0005H (SDO read succeeded) is stored in Un\G1000.
The receive message is stored in an area other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066). (☞ Page 32 Buffer memory area assignment)
Check Un\G1004, and Un\G1005 to Un\G1066 via the program, and retrieve data read by the SDO read.
❻ When the process is completed with an error, 000FH (Error) is stored in Un\G1000.
The error message is stored in an area other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066). (☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066))
❼ After the SDO read is completed, turn off 'Command execution request' (Y17) via the program. When 'Command execution request' (Y17) is turned off, 'Command execution completed' (X17) is also turned off.

Buffer memory area assignment

This section describes 'Command interface (CIF)' (Un\G1000 to Un\G1066) assignment for SDO read.

The following table shows the transmit message assignment.

Address	Description
	Transmit message
Un\G1000	Command code • 0004H: SDO read
Un\G1001	Set the node ID. • 0000H: Access the own node. • 0001H to 007FH: Access the specified node.
Un\G1002	Set the index. • 0000H to FFFFH: Configurable indexes of the object dictionary (☞ Page 191 Object Dictionary)
Un\G1003	Upper byte: Reserved Lower byte: Set the subindex. • 00H to FFH: Configurable subindexes of the object dictionary (☞ Page 191 Object Dictionary)
Un\G1004 to Un\G1066	System area

The following table shows memory assignment for a receive message.

Address	Description	
	Receive message	
	Completed successfully	**Completed with an error**
Un\G1000	Command execution result code • 0005H: SDO read succeeded	Command execution result code • 000FH: Error
Un\G1001	Node ID (value set by the transmit message)	☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066)
Un\G1002	Index (value set by the transmit message)	
Un\G1003	Sub-index (value set by the transmit message)	
Un\G1004	The data length (bytes) is stored. • 1 to 124	
Un\G1005 to Un\G1066	Result data is stored. • Un\G1005: The lower byte in first, the upper byte in second ⋮ • Un\G1066: The lower byte in 123rd, the upper byte in 124th	

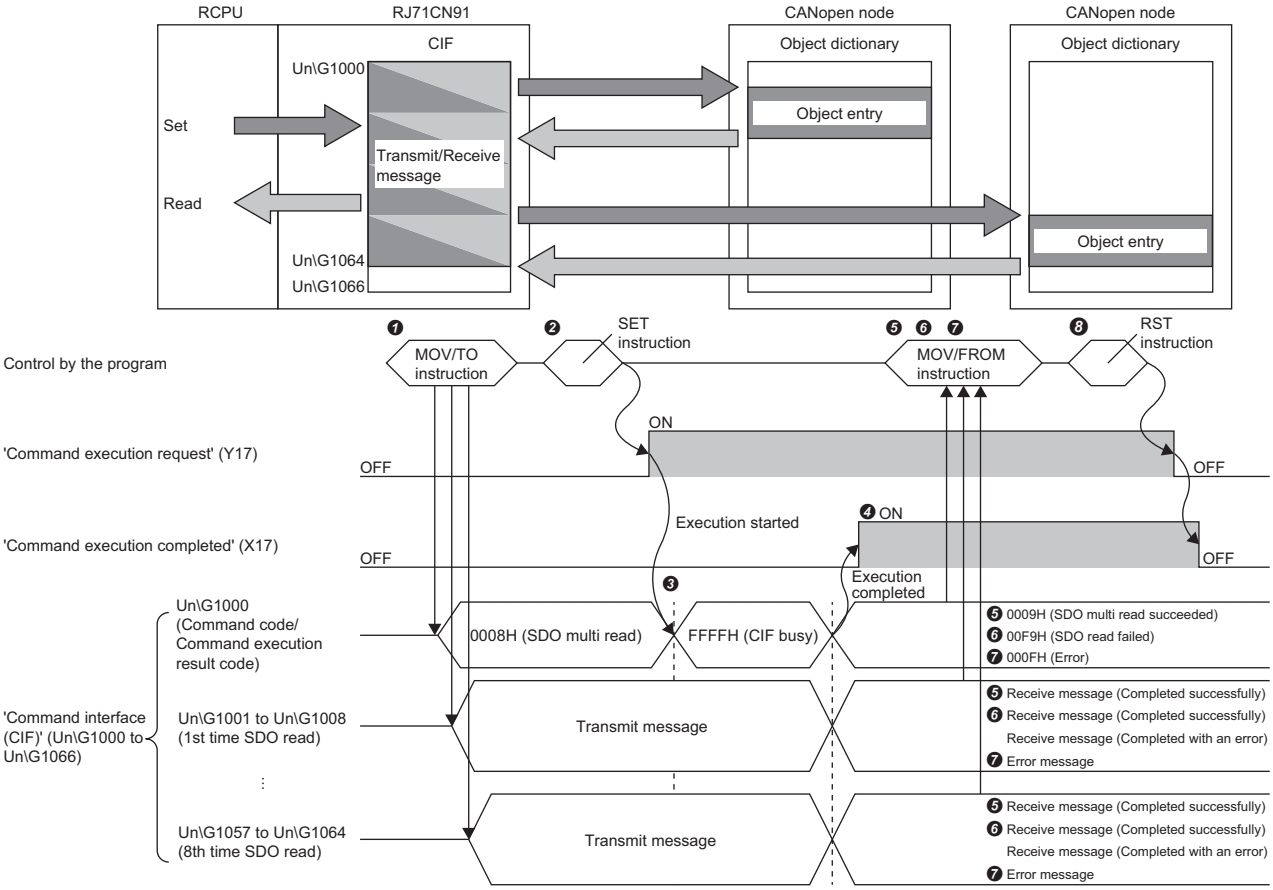
SDO multi read

The following is the procedure for executing the SDO multi read.

The following figure shows the first SDO multi read.

For buffer memory areas for the second and subsequent SDO multi read operations, refer to the following.

☞ Page 34 Buffer memory area assignment



❶ Set the transmit message in 'Command interface (CIF)' (Un\G1000 to Un\G1066) via the program. Set any transmit message in Un\G1001 to Un\G1008. (☞ Page 34 Buffer memory area assignment)

❷ Turn on 'Command execution request' (Y17) via the program.

❸ FFFFH (CIF busy) is stored in Un\G1000 while the command is being processed.

❹ When the command execution is finished, 'Command execution completed' (X17) turns on.

❺ When the process is completed successfully, 0009H (SDO multi read succeeded) is stored in Un\G1000.

1st time SDO read received messages are stored in Un\G1001 to Un\G1008. (☞ Page 34 Buffer memory area assignment)

Every time an SDO read is completed, Un\G1004, and Un\G1005 to Un\G1008 are checked via the program, and data read is retrieved by the SDO read.

❻ When any of SDO read operations is completed with an error, 00F9H (SDO read failed) is stored in Un\G1000.

1st time SDO read received messages are stored in Un\G1001 to Un\G1008. (☞ Page 34 Buffer memory area assignment)

Every time an SDO read is completed, when it is completed successfully, Un\G1004, and Un\G1005 to Un\G1008 are checked via the program, and data read is retrieved by the SDO read. Retrieve the error message when the process is completed with an error.

❼ When the process is completed with an error, 000FH (Error) is stored in Un\G1000.

The error message is stored in an area other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066). (☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066))

❽ After the SDO multi read is completed, turn off 'Command execution request' (Y17) via the program. When 'Command execution request' (Y17) is turned off, 'Command execution completed' (X17) is also turned off.

Buffer memory area assignment

This section describes 'Command interface (CIF)' (Un\G1000 to Un\G1066) assignment for SDO multi read.

The following table shows the transmit message assignment.

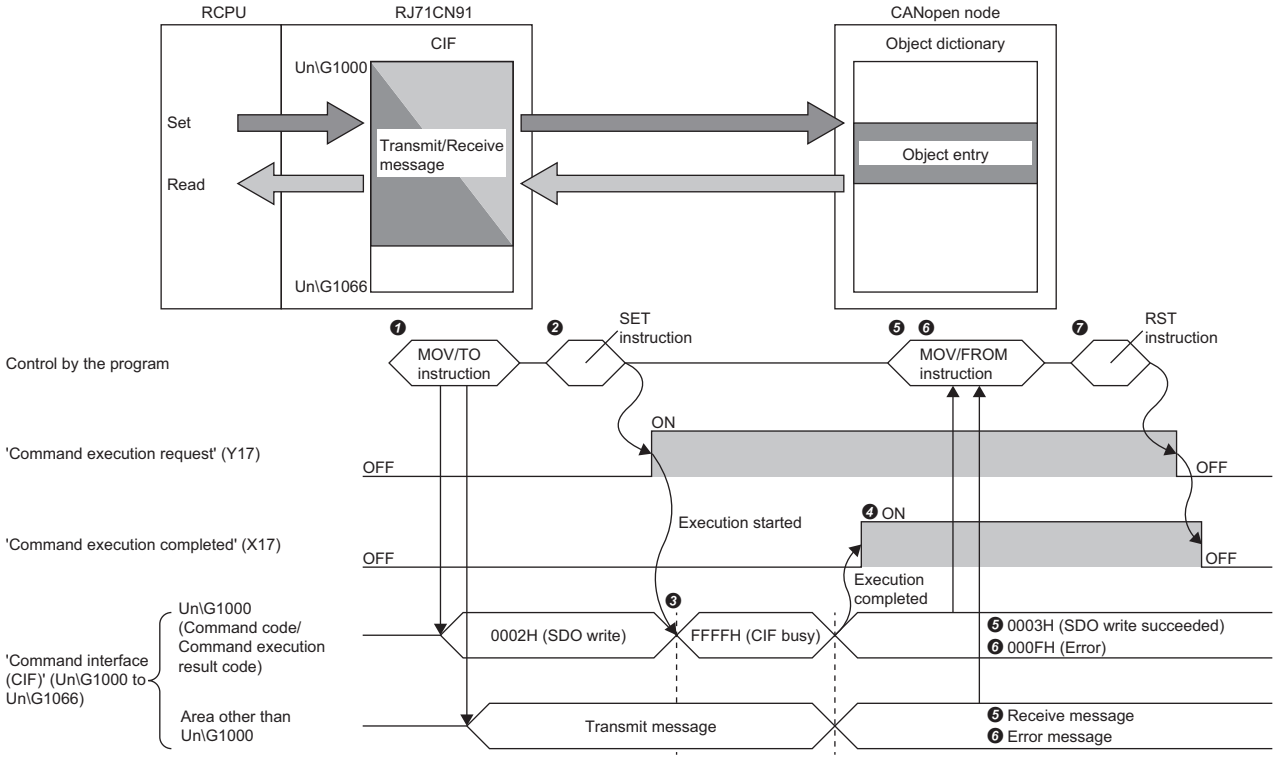
Address	Description
	Transmit message
Un\G1000	Command code • 0008H: SDO multi read
Un\G1001	■First SDO multi read Set the node ID. • 0000H: Access the own node. • 0001H to 007FH: Access the specified node. • FFFFH: No access
Un\G1002	■First SDO multi read Set the index. • 0000H to FFFFH: Configurable indexes of the object dictionary (☞ Page 191 Object Dictionary)
Un\G1003	■First SDO multi read Upper byte: Reserved Lower byte: Set the subindex. • 00H to FFH: Configurable subindexes of the object dictionary (☞ Page 191 Object Dictionary)
Un\G1004 to Un\G1008	■First SDO multi read System area
⋮	⋮
Un\G1057	■Eighth SDO multi read Set the node ID. • 0000H: Access the own node. • 0001H to 007FH: Access the specified node. • FFFFH: No access
Un\G1058	■Eighth SDO multi read Set the index. • 0000H to FFFFH: Configurable indexes of the object dictionary (☞ Page 191 Object Dictionary)
Un\G1059	■Eighth SDO multi read Upper byte: Reserved Lower byte: Set the subindex. • 00H to FFH: Configurable subindexes of the object dictionary (☞ Page 191 Object Dictionary)
Un\G1060	■Eighth SDO multi read System area
Un\G1061 to Un\G1066	■Eighth SDO multi read System area

The following table shows memory assignment for a receive message.

Address	Description		
	Receive message		
	Completed successfully	SDO read completed with an error	Completed with an error
Un\G1000	Command execution result code • 0009H: SDO multi read succeeded	Command execution result code • 00F9H: SDO read failed	Command execution result code • 000FH: Error
Un\G1001	■First SDO multi read Node ID (value set by the transmit message)	■First SDO multi read Lower byte • Node ID (value set by the transmit message) Upper byte • 0FH: Completed with an error	■First SDO multi read ☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066)
Un\G1002	■First SDO multi read Index (value set by the transmit message)	■First SDO multi read Index (value set by the transmit message)	
Un\G1003	■First SDO multi read Sub-index (value set by the transmit message)	■First SDO multi read Sub-index (value set by the transmit message)	
Un\G1004	■First SDO multi read The data length (bytes) is stored. • 1 to 8	■First SDO multi read Fixed to 0	
Un\G1005 to Un\G1008	■First SDO multi read Result data is stored. • Un\G1005: The lower byte in first, the upper byte in second ⋮ • Un\G1008: The lower byte in seventh, the upper byte in eighth	■First SDO multi read The SDO abort code is stored. (☞ Page 30 SDO abort codes)	
⋮	⋮	⋮	⋮
Un\G1057	■Eighth SDO multi read Node ID (value set by the transmit message)	■Eighth SDO multi read Lower byte • Node ID (value set by the transmit message) Upper byte • 0FH: Completed with an error	■Eighth SDO multi read ☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066)
Un\G1058	■Eighth SDO multi read Index (value set by the transmit message)	■Eighth SDO multi read Index (value set by the transmit message)	
Un\G1059	■Eighth SDO multi read Sub-index (value set by the transmit message)	■Eighth SDO multi read Sub-index (value set by the transmit message)	
Un\G1060	■Eighth SDO multi read The data length (bytes) is stored. • 1 to 8	■Eighth SDO multi read Fixed to 0	
Un\G1061 to Un\G1064	■Eighth SDO multi read Result data is stored. • Un\G1061: The lower byte in first, the upper byte in second ⋮ • Un\G1064: The lower byte in seventh, the upper byte in eighth	■Eighth SDO multi read The SDO abort code is stored. (☞ Page 30 SDO abort codes)	
Un\G1065 to Un\G1066	System area		

SDO write

The following is the procedure for executing the SDO write.



❶ Set the transmit message in 'Command interface (CIF)' (Un\G1000 to Un\G1066) via the program. (☞ Page 37 Buffer memory area assignment)

❷ Turn on 'Command execution request' (Y17) via the program.

❸ FFFFH (CIF busy) is stored in Un\G1000 while the command is being processed.

❹ When the command execution is finished, 'Command execution completed' (X17) turns on.

❺ When the process is completed successfully, 0003H (SDO write succeeded) is stored in Un\G1000.

The receive message is stored in an area other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066). (☞ Page 37 Buffer memory area assignment)

❻ When the process is completed with an error, 000FH (Error) is stored in Un\G1000.

The error message is stored in an area other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066). (☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066))

❼ After the SDO write is completed, turn off 'Command execution request' (Y17) via the program. When 'Command execution request' (Y17) is turned off, 'Command execution completed' (X17) is also turned off.

Buffer memory area assignment

This section describes 'Command interface (CIF)' (Un\G1000 to Un\G1066) assignment for SDO write.

The following table shows the transmit message assignment.

Address	Description
	Transmit message
Un\G1000	Command code • 0002H: SDO write
Un\G1001	Set the node ID. • 0000H: Access the own node. • 0001H to 007FH: Access the specified node.
Un\G1002	Set the index. • 0000H to FFFFH: Configurable indexes of the object dictionary (☞ Page 191 Object Dictionary)
Un\G1003	Upper byte: Reserved Lower byte: Set the subindex. • 00H to FFH: Configurable subindexes of the object dictionary (☞ Page 191 Object Dictionary)
Un\G1004	Set the data length (bytes). • 1 to 124
Un\G1005 to Un\G1066	Set data to transmit. • Un\G1005: The lower byte in first, the upper byte in second ⋮ • Un\G1066: The lower byte in 123rd, the upper byte in 124th

The following table shows memory assignment for a receive message.

Address	Description	
	Receive message	
	Completed successfully	**Completed with an error**
Un\G1000	Command execution result code • 0003H: SDO write succeeded	Command execution result code • 000FH: Error
Un\G1001	Node ID (value set by the transmit message)	☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066)
Un\G1002	Index (value set by the transmit message)	
Un\G1003	Sub-index (value set by the transmit message)	
Un\G1004 to Un\G1066	System area	

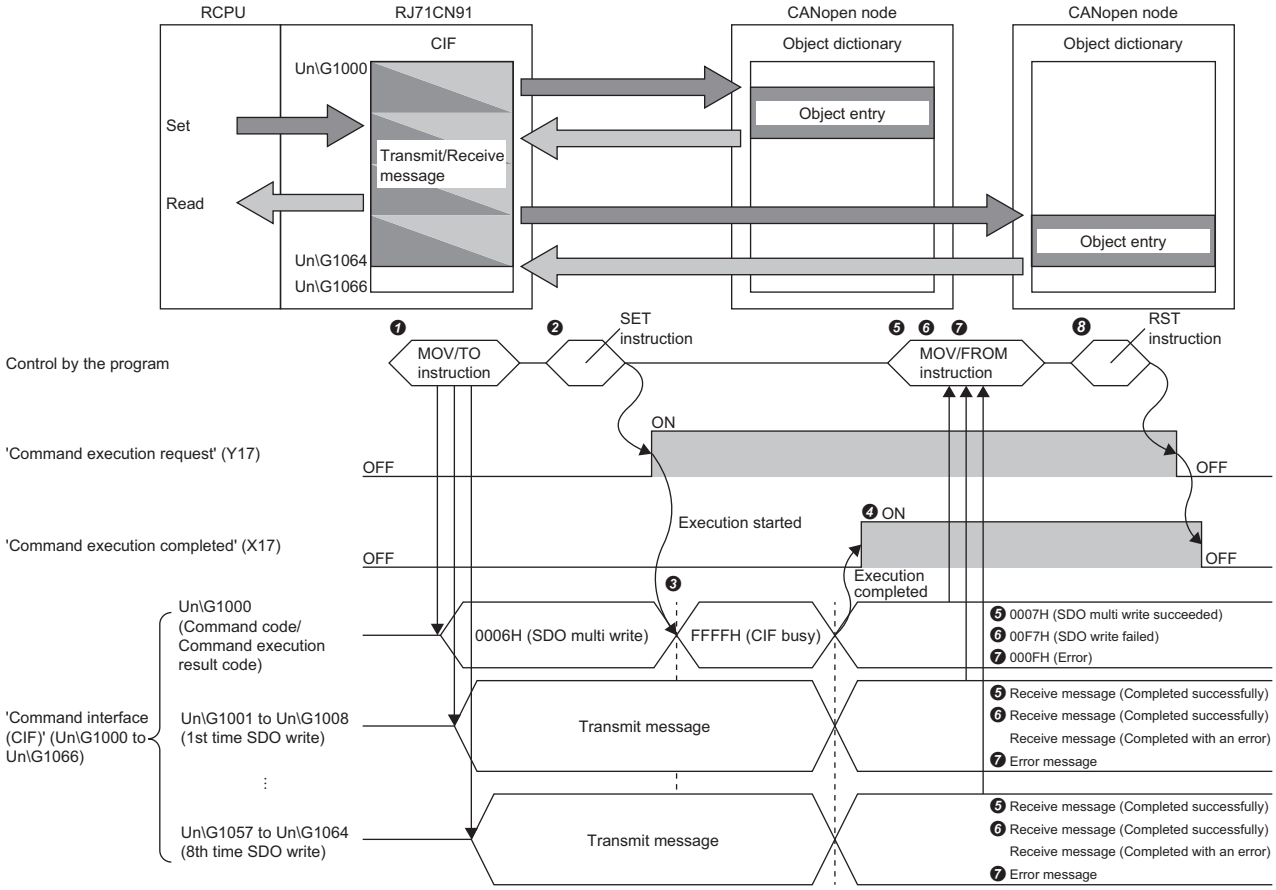
SDO multi write

The following is the procedure for executing the SDO multi write.

The following figure shows the first SDO multi write.

For buffer memory areas for the second and subsequent SDO multi write operations, refer to the following.

☞ Page 39 Buffer memory area assignment



❶ Set the transmit message in 'Command interface (CIF)' (Un\G1000 to Un\G1066) via the program. Set any transmit message in Un\G1001 to Un\G1008. (☞ Page 39 Buffer memory area assignment)

❷ Turn on 'Command execution request' (Y17) via the program.

❸ FFFFH (CIF busy) is stored in Un\G1000 while the command is being processed.

❹ When the command execution is finished, 'Command execution completed' (X17) turns on.

❺ When the process is completed successfully, 0007H (SDO multi write succeeded) is stored in Un\G1000.
1st time SDO multi write received messages are stored in Un\G1001 to Un\G1008. (☞ Page 39 Buffer memory area assignment)

❻ When any of SDO write operations is completed with an error, 00F7H (SDO write failed) is stored in Un\G1000.
1st time SDO multi write received messages are stored in Un\G1001 to Un\G1008. (☞ Page 39 Buffer memory area assignment)

❼ When the process is completed with an error, 000FH (Error) is stored in Un\G1000.
The error message is stored in an area other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066). (☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066))

❽ After the SDO multi write is completed, turn off 'Command execution request' (Y17) via the program. When 'Command execution request' (Y17) is turned off, 'Command execution completed' (X17) is also turned off.

Buffer memory area assignment

This section describes 'Command interface (CIF)' (Un\G1000 to Un\G1066) assignment for SDO multi write.

The following table shows the transmit message assignment.

Address	Description
	Transmit message
Un\G1000	Command code • 0006H: SDO multi write
Un\G1001	■First SDO multi write Set the node ID. • 0000H: Access the own node. • 0001H to 007FH: Access the specified node. • FFFFH: No access
Un\G1002	■First SDO multi write Set the index. • 0000H to FFFFH: Configurable indexes of the object dictionary (☞ Page 191 Object Dictionary)
Un\G1003	■First SDO multi write Upper byte: Reserved Lower byte: Set the subindex. • 00H to FFH: Configurable subindexes of the object dictionary (☞ Page 191 Object Dictionary)
Un\G1004	■First SDO multi write Set the data length (bytes). • 1 to 8
Un\G1005 to Un\G1008	■First SDO multi write Set data to transmit. • Un\G1005: The lower byte in first, the upper byte in second ⋮ • Un\G1008: The lower byte in seventh, the upper byte in eighth
⋮	⋮
Un\G1057	■Eighth SDO multi write Set the node ID. • 0000H: Access the own node. • 0001H to 007FH: Access the specified node. • FFFFH: No access
Un\G1058	■Eighth SDO multi write Set the index. • 0000H to FFFFH: Configurable indexes of the object dictionary (☞ Page 191 Object Dictionary)
Un\G1059	■Eighth SDO multi write Upper byte: Reserved Lower byte: Set the subindex. • 00H to FFH: Configurable subindexes of the object dictionary (☞ Page 191 Object Dictionary)
Un\G1060	■Eighth SDO multi write Set the data length (bytes). • 1 to 8
Un\G1061 to Un\G1064	■Eighth SDO multi write Set data to transmit. • Un\G1061: The lower byte in first, the upper byte in second ⋮ • Un\G1064: The lower byte in seventh, the upper byte in eighth
Un\G1065 to Un\G1066	System area

The following table shows memory assignment for a receive message.

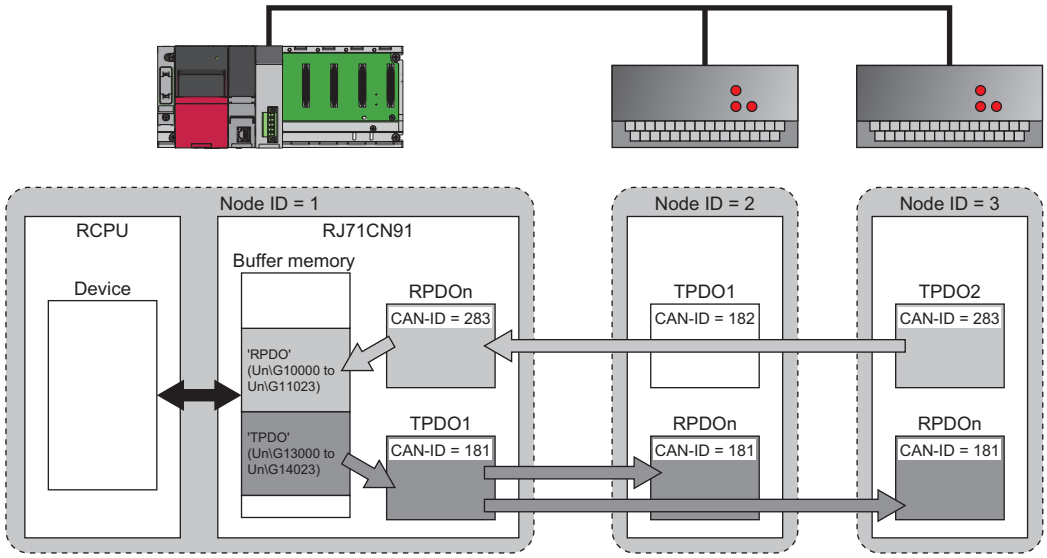
Address	Description		
	Receive message		
	Completed successfully	SDO write completed with an error	Completed with an error
Un\G1000	Command execution result code • 0007H: SDO multi write succeeded	Command execution result code • 00F7H: SDO write failed	Command execution result code • 000FH: Error
Un\G1001	■First SDO multi write Node ID (value set by the transmit message)	■First SDO multi write Lower byte • Node ID (value set by the transmit message) Upper byte • 0FH: Completed with an error	■First SDO multi write ☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066)
Un\G1002	■First SDO multi write Index (value set by the transmit message)	■First SDO multi write Index (value set by the transmit message)	
Un\G1003	■First SDO multi write Sub-index (value set by the transmit message)	■First SDO multi write Sub-index (value set by the transmit message)	
Un\G1004	■First SDO multi write System area	■First SDO multi write System area	
Un\G1005 to Un\G1008	■First SDO multi write System area	■First SDO multi write The SDO abort code is stored. (☞ Page 30 SDO abort codes)	
⋮	⋮	⋮	⋮
Un\G1057	■Eighth SDO multi write Node ID (value set by the transmit message)	■Eighth SDO multi write Lower byte • Node ID (value set by the transmit message) Upper byte • 0FH: Completed with an error	■Eighth SDO multi write ☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066)
Un\G1058	■Eighth SDO multi write Index (value set by the transmit message)	■Eighth SDO multi write Index (value set by the transmit message)	
Un\G1059	■Eighth SDO multi write Sub-index (value set by the transmit message)	■Eighth SDO multi write Sub-index (value set by the transmit message)	
Un\G1060	■Eighth SDO multi write System area	■Eighth SDO multi write System area	
Un\G1061 to Un\G1064	■Eighth SDO multi write System area	■Eighth SDO multi write The SDO abort code is stored. (☞ Page 30 SDO abort codes)	
Un\G1065 to Un\G1066	System area		

1.4 PDO

The PDO function transfers data in real time between multiple CANopen nodes.
There are two types of PDO: TPDO and RPDO. TPDOs send data and RPDOs receive data from any TPDO.

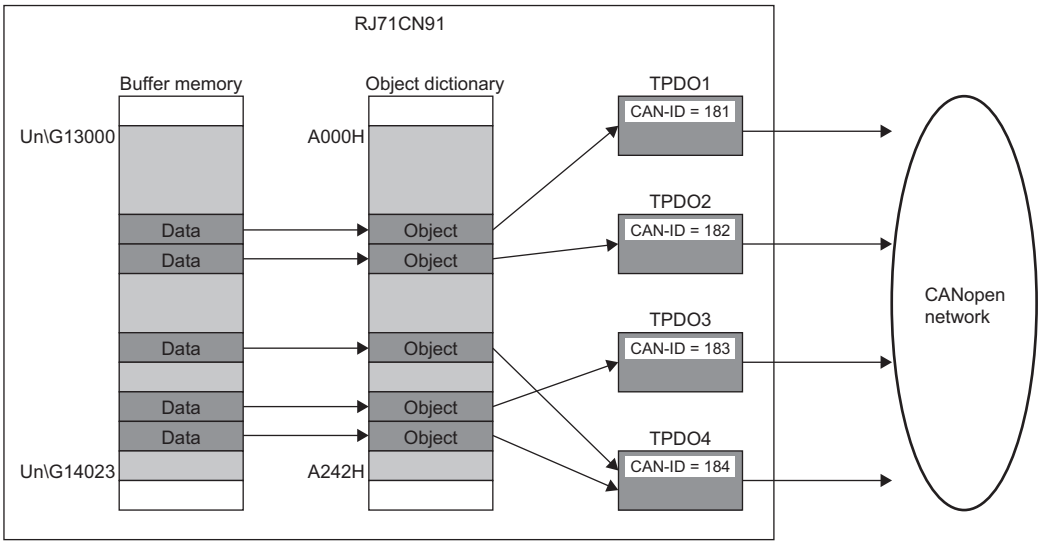
Data flow

The following figure shows the flow of sending/receiving buffer memory data using PDOs.
Any CAN-ID can be set to each TPDO and each RPDO, and RPDO receives data from the TPDO whose CAN-ID is the same as its CAN-ID.
TPDOs send data of any object in the object dictionary. RPDOs store received data in any object.
Map TPDOs and RPDOs to areas 'TPDO' (Un\G13000 to Un\G14023) and 'RPDO' (Un\G10000 to Un\G11023) to send/receive buffer memory data.



TPDO

This section describes the flow of sending data from 'TPDO' (Un\G13000 to Un\G14023) of the buffer memory.

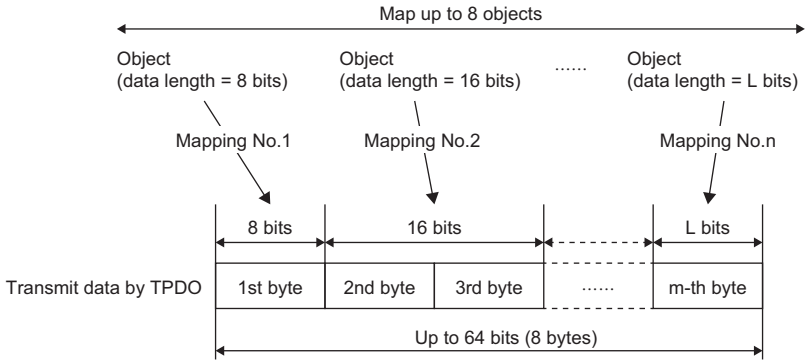


Setting method

Use the engineering tool to map each TPDO in 'TPDO' (Un\G13000 to Un\G14023) and set the CAN-ID of each TPDO. (☞ Page 91 TPDO details window)

The following is the mapping of send data when multiple objects are mapped to one TPDO.

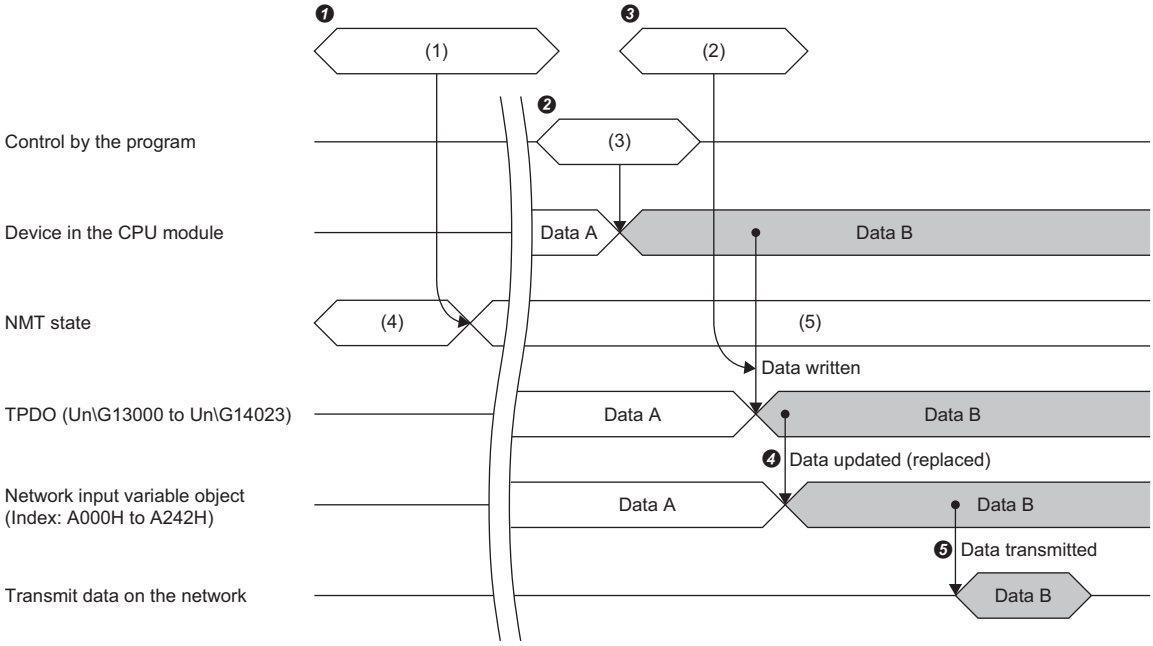
For the RJ71CN91, a maximum of eight objects (total of 64 bits) can be mapped to one TPDO.



While the RJ71CN91 is operating, each TPDO can be stopped, configured, started via the program or by other nodes by changing the object dictionary corresponding to the parameters.

Flow of data transmission

The following is the procedure up to the point of sending data. In this procedure, the communication process data is updated by refreshing.



(1) NMT master start/remote node start
(2) Refresh
(3) MOV/TO instruction
(4) Pre-operational
(5) Operational

❶ The autostart or remote node start function changes the NMT state of the RJ71CN91 to "Operational". (☞ Page 19 NMT state)
❷ The transmission data is set in the device of the CPU module via the program.
❸ When the refresh is performed, data in the device of the CPU module is written to 'TPDO' (Un\G13000 to Un\G14023).
❹ When the refresh is complete, data of Network input variable object (index A000H to A242H) corresponding to the TPDO are updated.
❺ After the object data is updated, the data will be sent under certain conditions or at a certain timing depending on the TPDO setting.

Data transmission timing

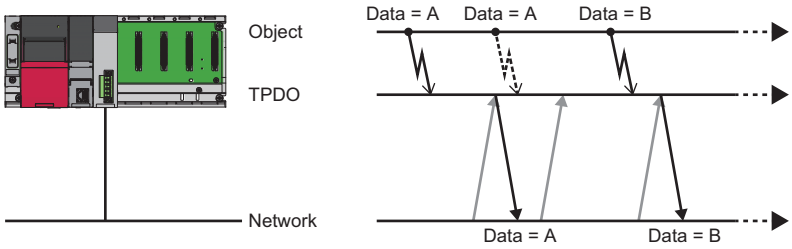
The TPDO transmission condition or timing varies depending on the PDO transmission setting. (☞ Page 91 TPDO details window)

The following describes each setting and timing of transmission.

■When the transmission type is SYNC

When TPDO transmission data is changed, the TPDO sends data when it receives the next SYNC message.

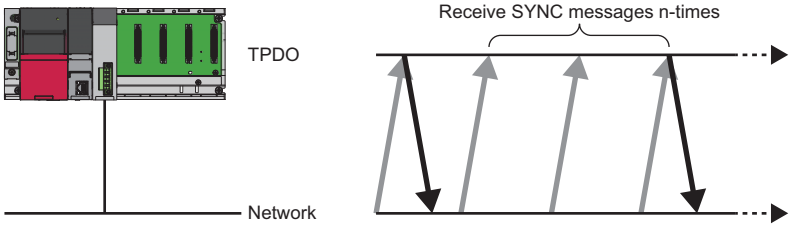
Note that the TPDO does not send data if the transmission data is not changed even when objects are refreshed or updated by 'Data exchange request' (Y1).



: Data is exchanged (with values changed).
: Data is exchanged (with values not changed).
: SYNC message
: Data transmission

■When the transmission type is SYNC (every SYNCn times)

The TPDO sends data every time it receives SYNC messages the number of times specified in the PDO transmission setting.



: SYNC message
: Data transmission

■When the transmission type is Event-driven

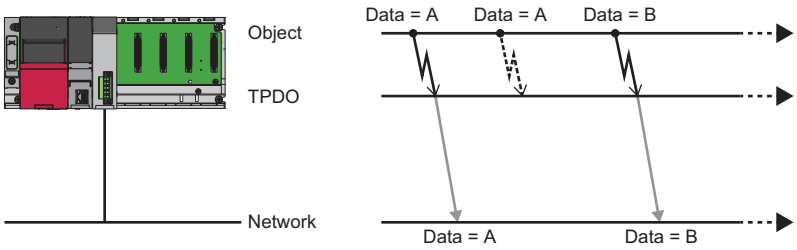
The TPDO sends data when the TPDO transmission data is changed. Note that the TPDO does not send data if the transmission data is not changed even when objects are refreshed or 'Data exchange request' (Y1) is turned on.

The transmission condition and timing vary depending on the inhibit time and event timer settings.

The following describes each setting and timing of transmission.

• When the inhibit time and event timer are not set

The TPDO sends data when the TPDO transmission data is changed.

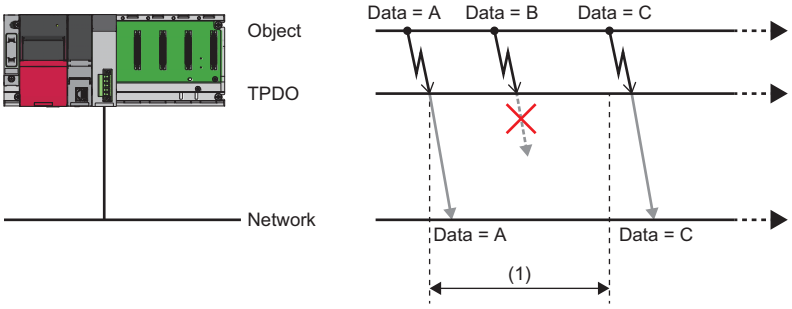


: Data is exchanged (with values changed).
: Data is exchanged (with values not changed).
: Data transmission

• When the inhibit time is set but the event timer is not set

The TPDO sends data when the TPDO transmission data is changed.

However, it does not send data when the inhibit time (1) has not passed since the previous transmission.



(1) Inhibit time

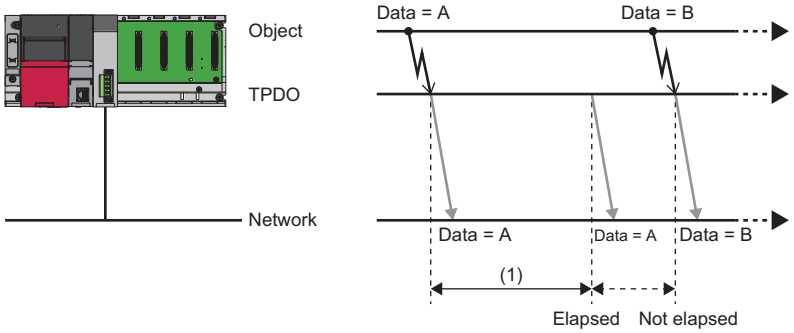
: Data is exchanged (with values changed).

: Data transmission

• When the inhibit time is not set but the event timer is set

The TPDO sends data when the TPDO transmission data is changed.

It also sends data when time set for the event timer (1) has passed since the previous transmission.



(1) Event timer

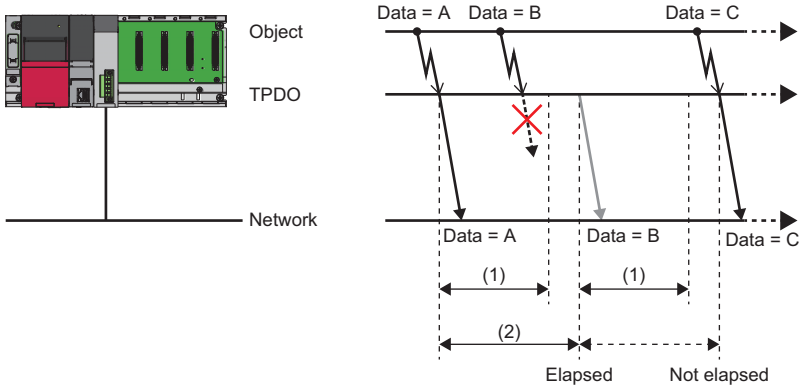
: Data is exchanged (with values changed).

: Data transmission

• When the inhibit time and event timer are set (inhibit time < event timer)

The TPDO sends data when the TPDO transmission data is changed.

However, it does not send data when the inhibit time (1) has not passed since the previous transmission. It also sends data when time set for the event timer (2) has passed since the previous transmission.



(1) Inhibit time

(2) Event timer

: Data is exchanged (with values changed).

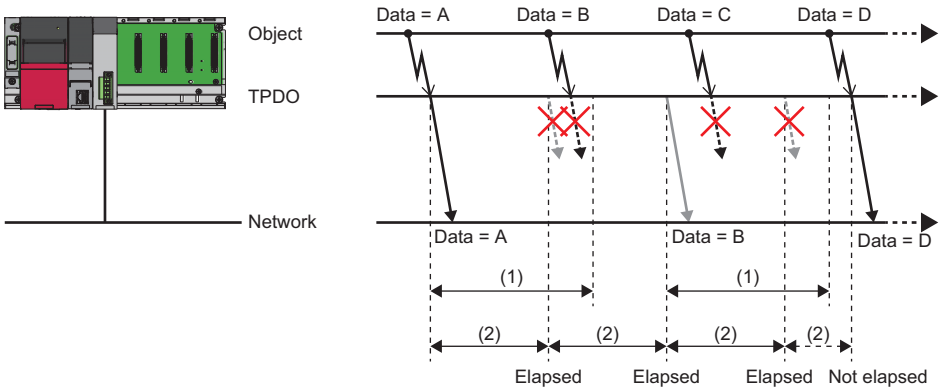
: Data is sent (event).

: Data is sent (event timer).

• When the inhibit time and event timer are set (inhibit time > event timer)

The TPDO sends data when the TPDO transmission data is changed.

However, it does not send data when the inhibit time (1) has not passed since the previous transmission. It also sends data when time set for the event timer (2) has passed since the previous transmission. (It does not send data when the time set for the event timer has passed but the time set for the inhibit time has not passed.)



(1) Inhibit time

(2) Event timer

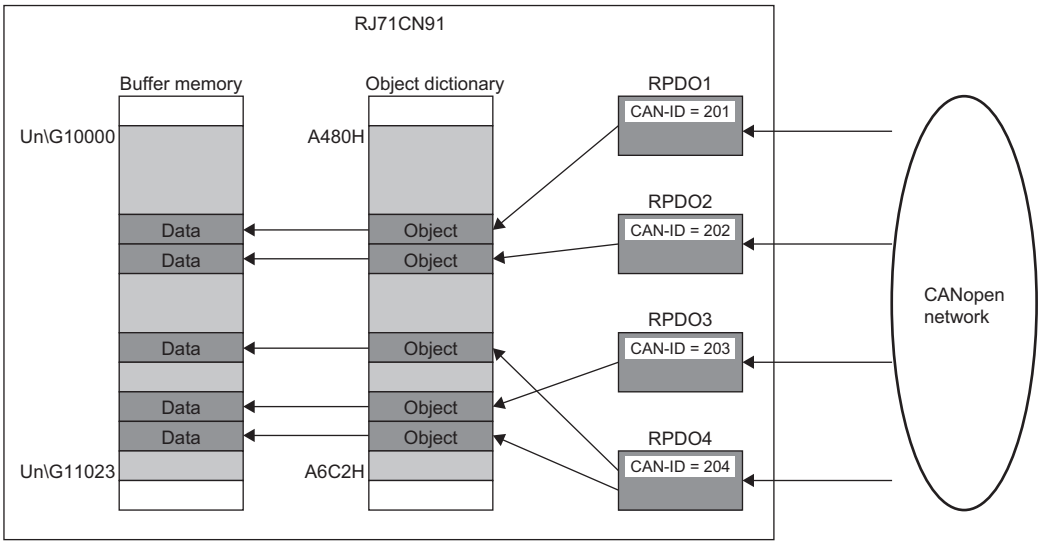
: Data is exchanged (with values changed).

: Data is sent (event).

: Data is sent (event timer).

RPDO

This section describes the flow of receiving data in 'RPDO' (Un\G10000 to Un\G11023) of the buffer memory.



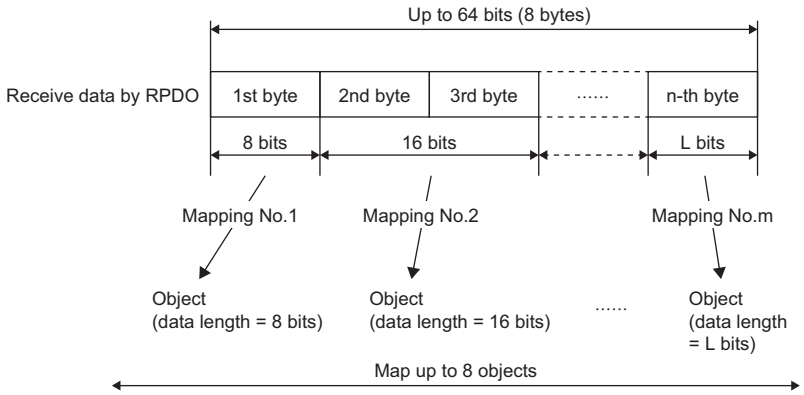
Setting method

Use the engineering tool to map each RPDO in 'RPDO' (Un\G10000 to Un\G11023) and set the CAN-ID of each RPDO. (☞ Page 93 RPDO details window)

The following is the mapping of received data to objects when multiple objects are mapped to one RPDO.

For the RJ71CN91, a maximum of eight objects (total of 64 bits) can be mapped to one RPDO.

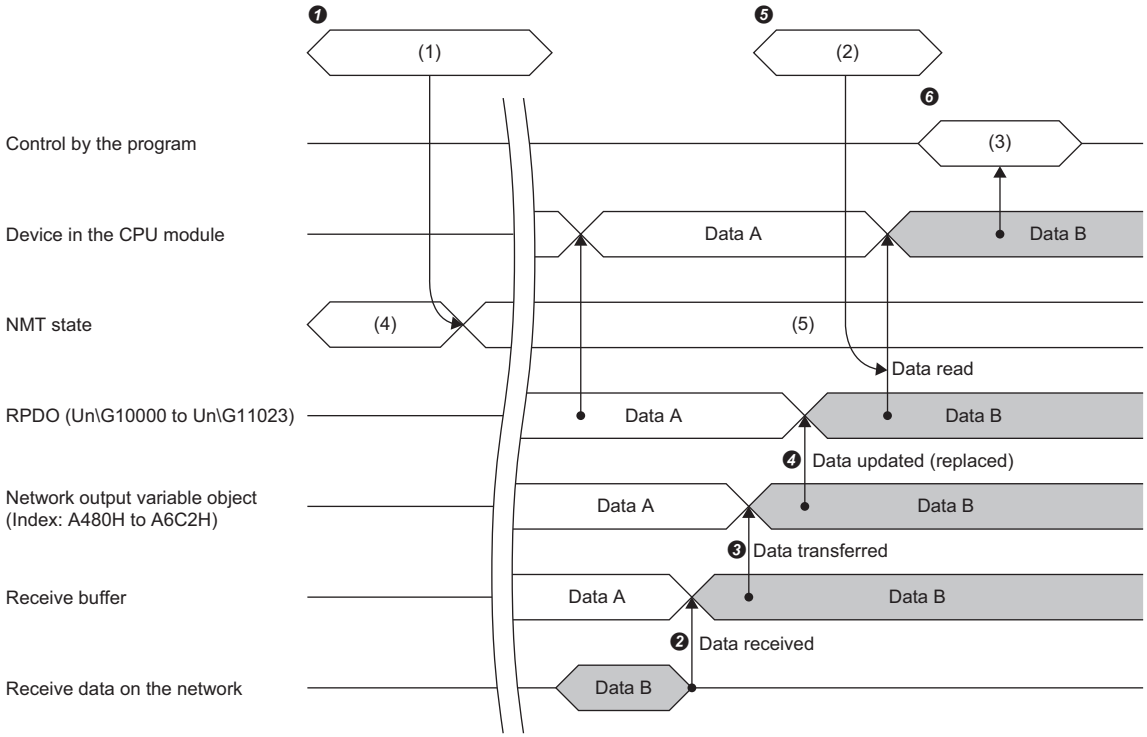
If there is no appropriate object in a mapping destination due to some reasons such as not using the own node, a data type object is set in the mapping destination.



While the RJ71CN91 is operating, each RPDO can be stopped, configured, or started via the program or by other nodes by changing the object dictionary corresponding to the parameters.

Flow of receiving data

The following is the procedure of receiving data.



(1) NMT master start/remote node start
(2) Refresh
(3) MOV/FROM instruction
(4) Pre-operational
(5) Operational

❶ The autostart or remote node start function changes the NMT state of the RJ71CN91 to "Operational". (☞ Page 19 NMT state)
❷ The RJ71CN91 stores data in an RPDO when it receives data.
❸ At the timing defined in the RPDO setting, the received data is stored in Network output variable object (index A480H to A6C2H) mapped to the object 'PRDO' (Un\G10000 to Un\G11023) of the object dictionary that is mapped to the RPDO. (☞ Page 49 Timing of receiving data)
❹ When the data is transferred to the object, the data in 'RPDO' (Un\G10000 to Un\G11023) corresponding to the object is updated.
❺ When the refresh is performed, data in the RPDO is read to the device of the CPU module.
❻ The data received in the device of the CPU module is referenced by the program.

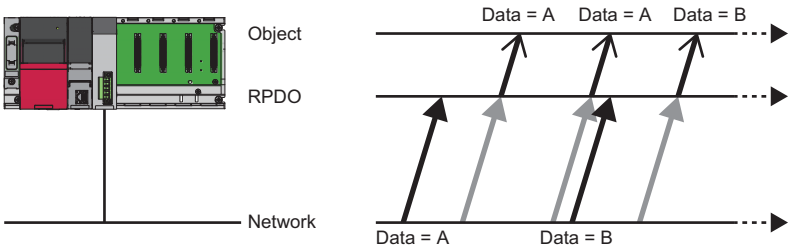
Timing of receiving data

The timing of receiving data in Network output variable object (index A480H to A6C2H) varies depending on the PDO receive setting. (☞ Page 93 RPDO details window)

The following describes each setting and receive timing.

■When the transmission type is SYNC

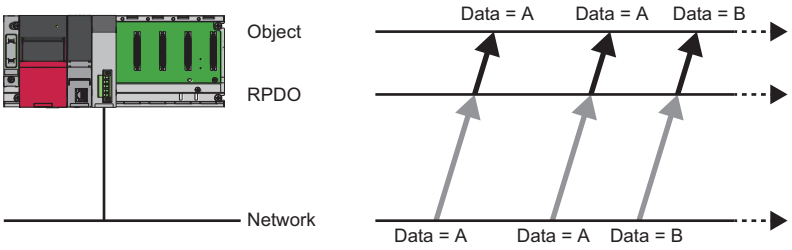
The RPDO transfers data to the network output variable object (index A480H to A6C2H) every time it receives a SYNC message.



: Data transmission
: SYNC message

■When the transmission type is Event-driven

The RPDO transfers data to the network output variable object (index: A480H to A6C2H) every time it receives data.



: Data transfer
: Data reception

1.5 SYNC

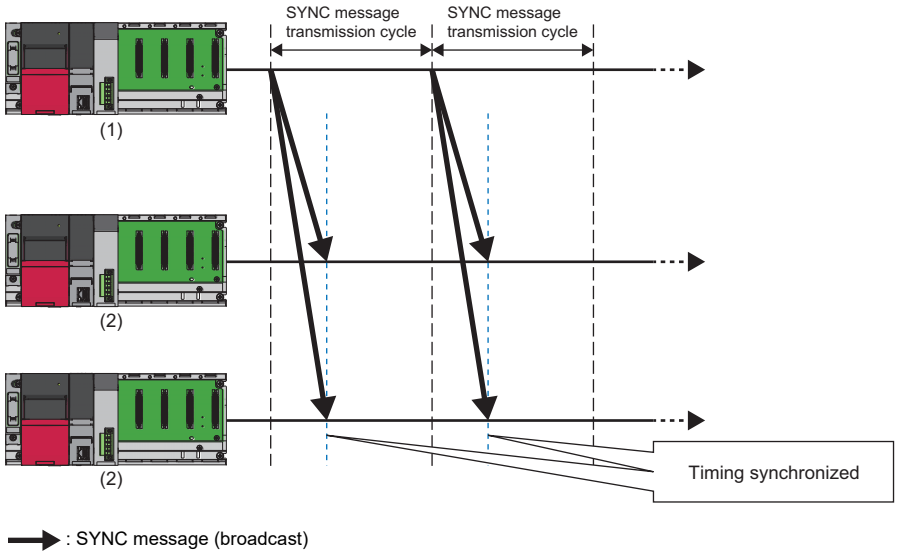
This function provides timing synchronization in the CANopen network. This function can synchronize the TPDO transmission timing and RPDO receive timing.

Function details

The SYNC producer (active NMT master) (1) periodically sends a SYNC message to synchronize the timing of the SYNC consumers (hot standby NMT master, NMT slave) (2).

For details on TPDO transmission timing and RPDO receive timing, refer to the following.

☞ Page 44 Data transmission timing,Page 49 Timing of receiving data



Point

- Only the active NMT master can be set as the SYNC producer.
- The SYNC message transmission cycle may vary due to the latency from some other message being transmitted just before the SYNC.

■Setting method

Use parameters to set the SYNC function. For details, refer to the following.

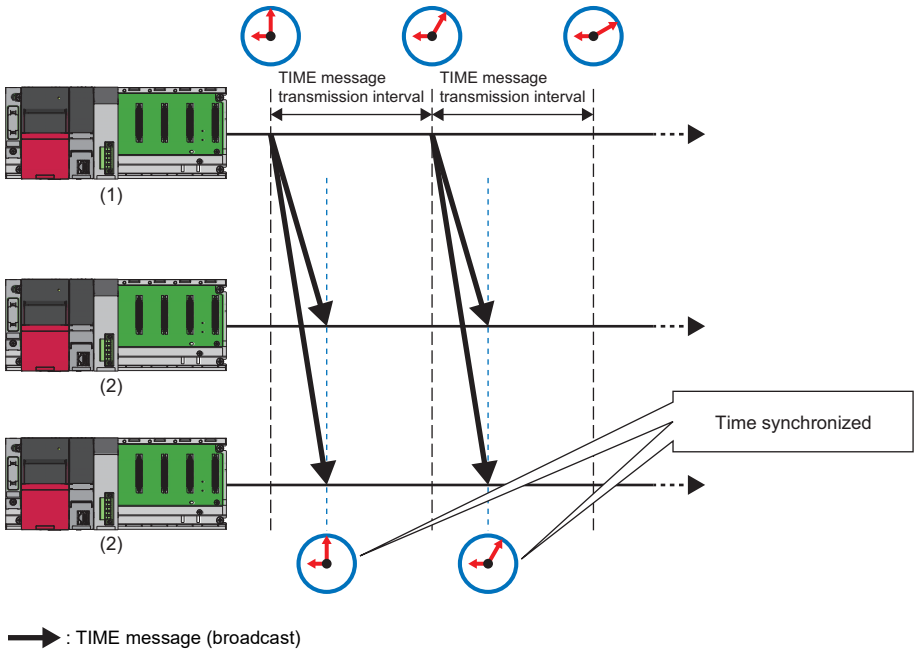
☞ Page 96 "NMT master / slave" window

1.6 TIME

This function provides time synchronization between CANopen nodes in the CANopen network.

Function details

The TIME producer (active NMT master) (1) sends a TIME message to synchronize the time of the TIME consumers (hot standby NMT master, NMT slave) (2).



Point

Only the active NMT master can be set as the TIME producer.

'Time stamp' (Un\G50 to Un\G59) can be set in the following time range.

- Minimum value: 2000 January 1st 00:00:00
- Maximum value: 2079 December 31st 23:59:59

If the time received in the TIME message is out of this range, the time is set to the minimum value.

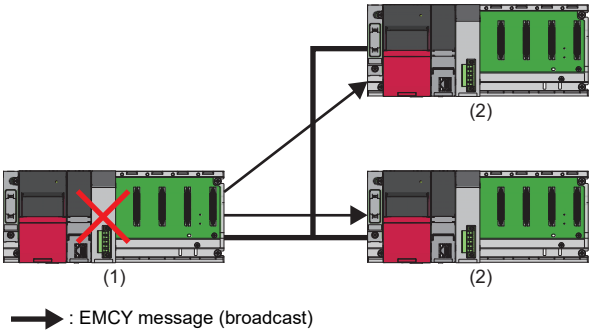
■Setting method

Use the buffer memory and parameters to set the TIME function. For details, refer to the following.

☞ Page 170 Time stamp (Un\G50 to Un\G59)
☞ Page 96 "NMT master / slave" window

1.7 EMCY

The EMCY function notifies other nodes (2) of an error that occurred in the own node (1). The own node sends an EMCY message to other nodes when it detects an error. The own node can also receive EMCY messages from other nodes.



Function details

The RJ71CN91 automatically sends an EMCY message when an error occurs in the own node. EMCY messages can be sent at any timing.

EMCY messages received from other nodes can be read manually from the buffer memory. EMCY messages can be registered as events in the event history.

An EMCY message notifies of the following three types of information.

- Emergency error code (2 bytes) (☞ Page 52 Emergency error codes)
- Error register (1 byte) (☞ Page 211 Error register)
- Manufacturer-specific error code (5 bytes) (☞ Page 53 Manufacturer-specific error code)

■Setting method

The minimum time for the interval for sending an EMCY message can be set with the inhibit time parameter.

For details, refer to the following.

☞ Page 213 Inhibit time EMCY

■Emergency error codes

The following table lists common error codes in error codes notified of as an EMCY message defined by CANopen.

○: Send, ×: Not send

Error code	Description	Transmission in the RJ71CN91
0000H	Error reset or no error	○
1000H	Generic error	×
2000H	Current	×
2100H	CANopen device input current	×
2200H	CANopen device internal current	×
2300H	CANopen device output current	×
3000H	Voltage	×
3100H	Power supply voltage	×
3200H	CANopen device internal voltage	×
3300H	Output voltage	×
4000H	Temperature	×
4100H	Ambient temperature	×
4200H	Device temperature	×
5000H	CANopen device hardware	×
6000H	CANopen device software	×
6100H	Internal software	×
6200H	User software	○
6300H	Set data	×
7000H	Additional module	×

Error code	Description	Transmission in the RJ71CN91
8000H	Monitoring	×
8100H	Communications	×
8110H	CAN overrun (object lost)	○
8120H	CAN error passive mode	○
8130H	Lifeguard error or heartbeat error	×
8140H	Recovery from the bus off state	×
8150H	CAN-ID conflict	×
8200H	Protocol error	○
8210H	PDO not processed due to a length error	○
8220H	PDO over length	○
8230H	DAM MPDO not processed, destination object not available	×
8240H	Unexpected SYNC data length	×
8250H	RPDO timeout	○
8F01H to 8F7FH	Lifeguard error or heartbeat error for node ID 1 to 127	○
9000H	External error	×
F000H	Added function	×
FF00H	Device specific	○

Point

Other than the above, various emergency error codes are defined in various CiA devices and application profiles and by each device manufacturer. For emergency error codes not listed above, refer to the manual of the CANopen node that sent the EMCY message.

■Manufacturer-specific error code

The following table lists the RJ71CN91 specific error codes notified of as an EMCY message.

Emergency error code	Manufacturer-specific error code	Description
FF00H	4D45303032H	"ME002": CPU module STOP transition
6200H	4D45303034H	"ME004": Module restart by 'Module restart request' (Y2)

When sending a message automatically

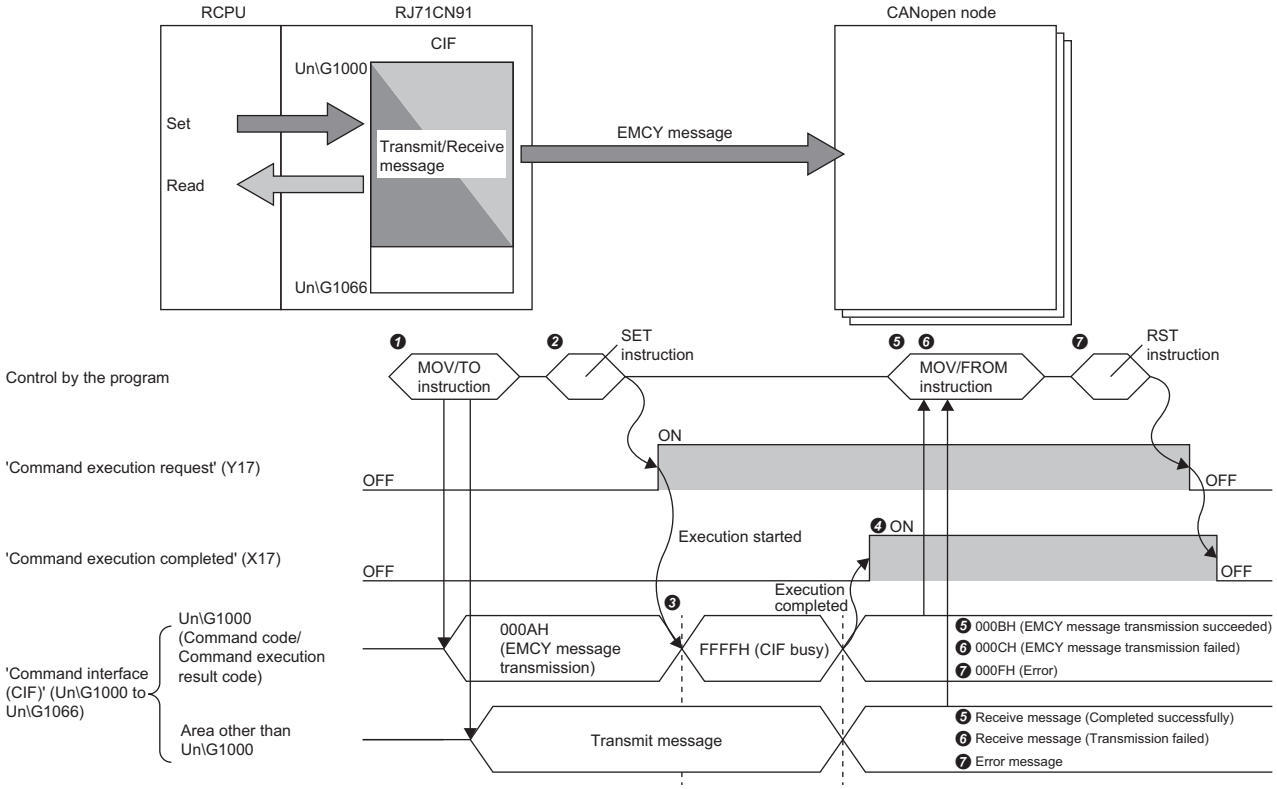
An EMCY message is sent automatically when an communication error or CPU module error occurs in the own node.
One EMCY message is sent for each error. No further EMCY message is sent until the next error occurs.

Point

- When an error occurs in the own node, the bit corresponding to the emergency error code in Error register (index 1001H) is turned on.
- When an error occurs in the own node, the emergency error code is stored in Pre-defined error field (index 1003H). A maximum of 15 latest emergency error codes are stored in Pre-defined error field (index 1003H).
- Automatic transmission of EMCY messages cannot be disabled.
- When the CPU module is switched from STOP state to RUN state, the emergency error code (FF00H: "ME002": CPU module STOP transition) in the Pre-defined error field (index 1003H) is cleared.
- When all emergency error codes in the Pre-defined error field (index 1003H) are cleared, the value in the Error register (index 1001H) is cleared, and an EMCY message with the emergency error code (0000H: Error reset or no error) is sent.

When sending a message manually

To send an EMCY message manually, use 'Command interface (CIF)' (Un\G1000 to Un\G1066).



❶ Set the transmit message in 'Command interface (CIF)' (Un\G1000 to Un\G1066) via the program. (☞ Page 55 Buffer memory area assignment)
❷ Turn on 'Command execution request' (Y17) via the program.
❸ FFFFH (CIF busy) is stored in Un\G1000 while the command is being processed.
❹ When the command execution is finished, 'Command execution completed' (X17) turns on.
❺ When the process is completed successfully, 000BH (EMCY message transmission succeeded) is stored in Un\G1000.
The receive message is stored in an area other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066). (☞ Page 55 Buffer memory area assignment)
❻ When the transmission fails, 000CH (EMCY message transmission failed) is stored in Un\G1000.
The receive message is stored in an area other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066). (☞ Page 55 Buffer memory area assignment)
❼ When the process is completed with an error, 000FH (Error) is stored in Un\G1000.
The error message is stored in an area other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066). (☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066))
❽ After EMCY message transmission is completed, turn off 'Command execution request' (Y17) via the program. When 'Command execution request' (Y17) is turned off, 'Command execution completed' (X17) is also turned off.

■Buffer memory area assignment

This section describes 'Command interface (CIF)' (Un\G1000 to Un\G1066) assignment for sending an EMCY message manually.

The following table shows the transmit message assignment.

Address	Description
	Transmit message
Un\G1000	Command code • 000AH: EMCY message transmission
Un\G1001	Emergency error code
Un\G1002	Lower byte • Error register (☞ Page 211 Error register) Upper byte • Manufacturer-specific error code (first byte)
Un\G1003	Lower byte • Manufacturer-specific error code (second byte) Upper byte • Manufacturer-specific error code (third byte)
Un\G1004	Lower byte • Manufacturer-specific error code (fourth byte) Upper byte • Manufacturer-specific error code (fifth byte)
Un\G1005 to Un\G1066	System area

The following table shows memory assignment for a receive message.

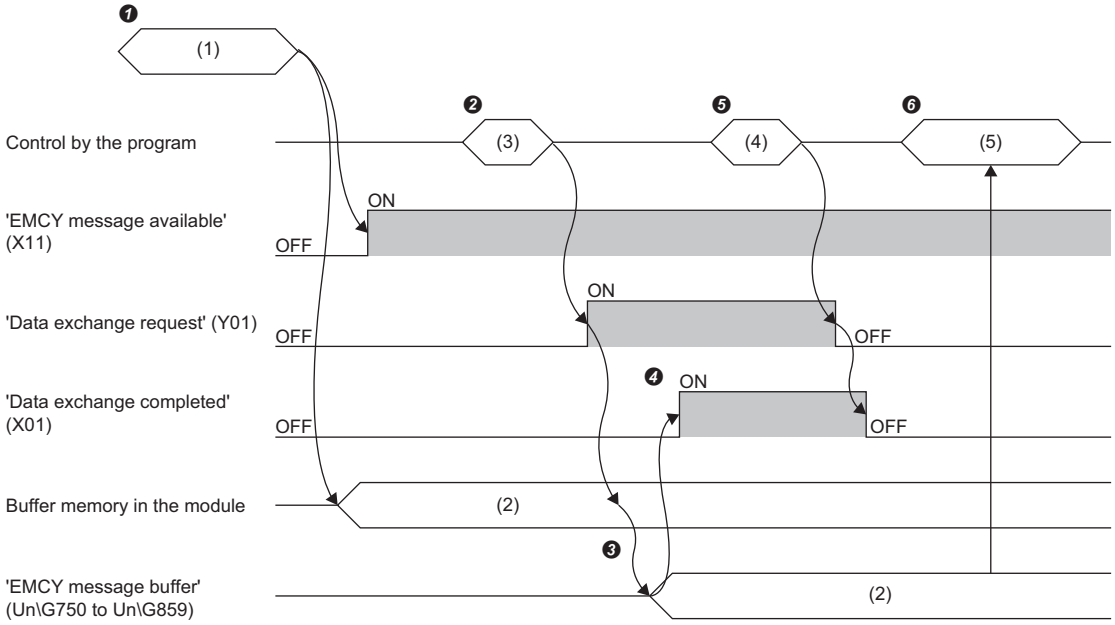
Address	Description		
	Receive message		
	Completed successfully	Transmission failed	Completed with an error
Un\G1000	Command execution result code • 000BH: EMCY message transmission succeeded	Command execution result code • 000CH: EMCY message transmission failed	Command execution result code • 000FH: Error
Un\G1001	Communication error code • 0000H: No error	Communication error code • 0001H: EMCY transmission inhibiting time has not passed. • 0002H: NMT state violation (The NMT state must be Pre-operational or Operational to send a message.)	☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066)
Un\G1002 to Un\G1066	System area		

Point

- The NMT state of the own node must be Operational or Pre-operational to send an EMCY message.
- When an EMCY message is sent manually, the bit set in the error register of the EMCY message is turned on in Error register (index 1001H). Note that the bit cannot be turned off automatically.
- When an EMCY message is sent manually, the emergency error code set in the EMCY message is stored in Pre-defined error field (index 1003H). As with when the EMCY message is sent automatically, a maximum of 15 latest emergency error codes are stored in Pre-defined error field (index 1003H).

When a message is received

Received EMCY messages are registered in the buffer memory and event history. Check the event history to confirm received EMCY messages. (☞ Page 150 Event List)



(1) EMCY message received
(2) EMCY message (new)
(3) SET instruction
(4) RST instruction
(5) MOV/FROM instruction

❶ When an EMCY message is received, 'EMCY message available' (X11) turns on.
(The EMCY message is stored inside the module but not stored in the buffer memory.)
❷ 'Data exchange request' (Y01) is turned on.
❸ 'EMCY message buffer' (Un\G750 to Un\G859) is updated to the latest state.
❹ When data exchange is completed, 'Data exchange completed' (X01) is turned on.
❺ 'Data exchange request' (Y01) is turned off.
❻ The EMCY message is read from EMCY message buffer.

For details on EMCY message buffer, refer to the following.
☞ Page 175 EMCY message buffer (Un\G750 to Un\G859)

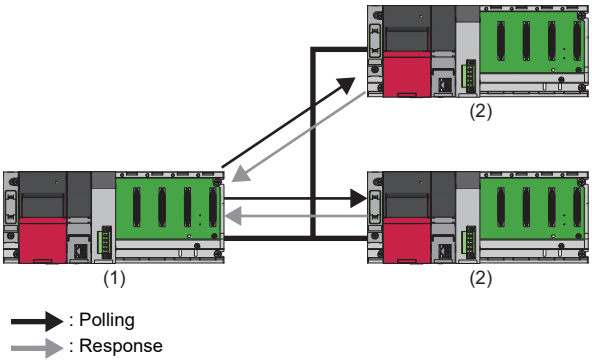
For how to clear 'EMCY message available' (X11) and EMCY message buffer, refer to the following.
☞ Page 157 EMCY message available (X11), EMCY message area clear request (Y11)

1.8 Node Guarding

The NMT master uses the node guarding function to monitor NMT slaves to detect errors in them.

Function details

The NMT master (1) polls each NMT slave (2) periodically for monitoring. And, each NMT slave monitors polling from the NMT master.

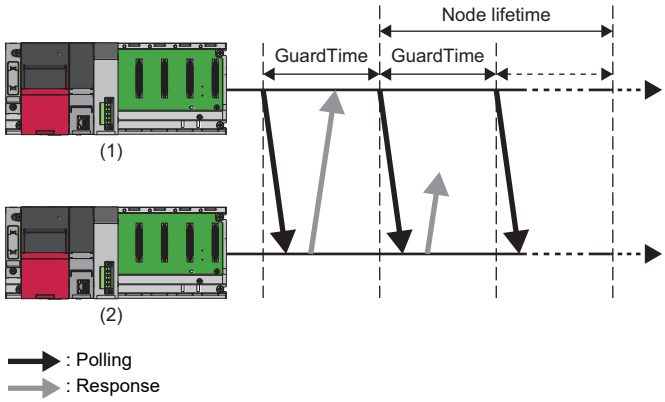


Point

- Node guarding produces a heavy load on the bus, so it is recommended to use heartbeat instead.
- When the NMT master is enabled, if node guarding and heartbeat consumer are set to the same node ID, the node guarding setting for that node ID is disabled because a conflict occurs between heartbeat and node guarding.
- When the NMT slave is enabled, if node guarding and heartbeat producer are set, the node guarding setting is disabled because a conflict occurs between heartbeat and node guarding.

■Error detection by the NMT master

The NMT master (1) monitors polling responses from each NMT slave (2) to detect the NMT state and errors of the NMT slave. If the NMT slave does not send a valid response within the node life time, the NMT master determines that an error has occurred in the NMT slave. Also, if the NMT state of the NMT slave is different from the latest NMT state instructed from the NMT master to the NMT slave through node control, the NMT master determines that an error has occurred in the NMT slave. In this case, the CAN ERR LED of the NMT master turns on or flashes.



NMT slave information detected by the NMT master is reflected in the following buffer memory areas.

Buffer area name	Details reflected
NMT error control status	For node guarding for each NMT slave, the following statuses are stored. • Node guarding is enabled/disabled (running/stopped). • There is no response from an NMT slave within the guard time. Or, an invalid response is received. • There was no response from an NMT slave within the node life time. • The NMT state of an NMT slave is different from the expected state.*1

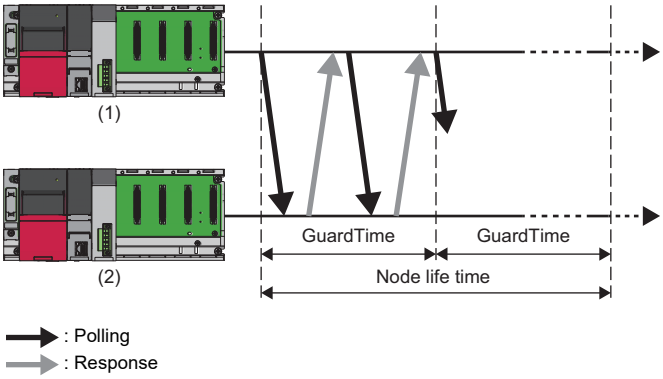
Buffer area name	Details reflected
NMT state	The NMT state obtained from each NMT slave through polling is stored. The obtainable NMT states are as follows. • Stop state • Operational state • Pre-operational state

*1 The latest NMT state in which the NMT master instructed each NMT slave through node control. If nothing was instructed, the NMT state of each NMT slave is that of the point when the NMT master started monitoring.

NMT slave information detected by the NMT master is stored in 'NMT error control status' (Un\G401 to Un\G527) and 'NMT state' (Un\G601 to Un\G727). (☞ Page 173 NMT error control status (Un\G401 to Un\G527), Page 174 NMT state (Un\G601 to Un\G727))

■Error detection by the NMT slave

The NMT slave (2) monitors polling from the NMT master (1) to detect errors of the NMT master. If the NMT slave is not polled by the NMT master within the life time, the NMT slave determines that an error has occurred in the NMT master. In this case, the CAN ERR LED of the NMT slave turns on or flashes.



NMT slave information detected by the NMT master is reflected in the following buffer memory areas.

Buffer area name	Details reflected
NMT error control status	For node guarding for each NMT slave, the following statuses are stored. • Node guarding is enabled/disabled (running/stopped). • There was no polling from the NMT master within the guard time. • There was no polling from the NMT master within the life time.

NMT master information detected by the NMT slave is stored in 'NMT error control status' (Un\G401 to Un\G527). (☞ Page 173 NMT error control status (Un\G401 to Un\G527))

■Setting method

Use parameters to set the node guarding function. For details, refer to the following.

☞ Page 100 "NMT slave" window

Point

When node guarding is executed for an NMT slave, it must also be executed for the NMT master. However, when node guarding is executed for the NMT master, whether to execute it for an NMT slave is optional.

The NMT master starts executing polling in either of the following timings.

- When an NMT slave boots
- When "Node guarding start" is requested to the guarding request of the object dictionary

An NMT slave starts monitoring polling by the NMT master in the following timing.

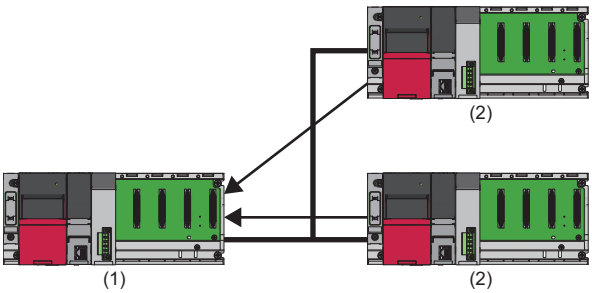
- When the first polling message is received from the NMT master

1.9 Heartbeat

Heartbeat is the function to monitor and detect errors in other nodes from any CANopen node.

Function details

The producer (2) (CANopen node monitored by other nodes) sends a heartbeat message periodically to the consumer (1) (CANopen node that monitors other nodes). The consumer checks the heartbeat message receiving interval to monitor the producer.



: Heartbeat message (broadcast)

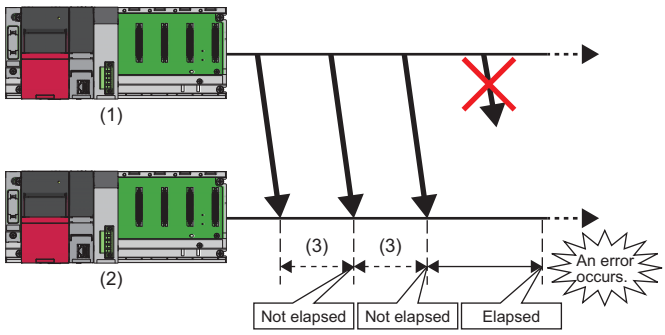
Point

- Heartbeat imposes heavy load on the bus. However, the load level is half the level of the load imposed by node guarding.
- When the flying master function is used, the heartbeat for flying master is automatically enabled. (☞ Page 25 Flying master)
- Node guarding may be disabled because a conflict occurs between heartbeat and node guarding. (☞ Page 57 Node Guarding)

■Error detection

The consumer (2) monitors the interval of receiving heartbeat message from the producer (1). If the consumer does not receive a heartbeat message from the producer within the set heartbeat time (3), the consumer determines that an error has occurred in the producer. In this case, the CAN ERR LED of the consumer turns on or flashes.

Producer information detected by the consumer is stored in 'NMT error control status' (Un\G401 to Un\G527) and 'NMT state' (Un\G601 to Un\G727). (☞ Page 173 NMT error control status (Un\G401 to Un\G527), Page 174 NMT state (Un\G601 to Un\G727))



: Heartbeat message

Producer information detected by consumers is reflected in the following buffer memory areas.

Buffer area name	Details reflected
NMT error control status	For the heartbeat function of each producer, the following statuses are stored. • Heartbeat is enabled/disabled (running/stopped). • A heartbeat message was not received within the heartbeat time. • The NMT state of a producer is different from the expected state.*1
NMT state	The NMT states obtained from heartbeat messages of each producer are stored. The obtainable NMT states are as follows. • Stop state • Operational state • Pre-operational state

*1 The latest NMT state in which instructions were given from the consumer through node control. If nothing was instructed, the NMT state of each producer is that of the point when the consumer started monitoring.

■Setting method

Use parameters to set the heartbeat function. For details, refer to the following.

☞ Page 102 "Heartbeat" window

Point

• A producer starts sending a heartbeat message in either of the following timings.

(1) After a producer started up, the producer heartbeat time was changed to enabled.

(2) The producer heartbeat had been enabled even before a producer started up, and the state of that producer transitioned from the initialization state to Pre-operational.

• A consumer starts monitoring each producer in either of the following timings.

(1) The first heartbeat message was received from a producer.

(2) The consumer is the NMT master, and started booting a producer (NMT slave).

1.10 Operation Setting at Error Occurrence

This function can set the NMT state that the RJ71CN91 transitions to if a communication error occurs when the NMT state of the RJ71CN91 is Operational.

Function details

The following three states can be set as the NMT state that the RJ71CN91 transitions to if a communication error occurs when the NMT state of the RJ71CN91 is Operational.

- Pre-operational (initial value)
- Stop
- Operational (Not change the NMT state)

The following table lists events that cause a communication error in the RJ71CN91.

Event	Remarks
The node is in the bus off state.	—
The node guarding function detected an NMT master error. (Only if the own node is an NMT slave)	—
The heartbeat function detected an error of another node. (Only if the own node is an NMT slave)	—
The CPU module state changed from RUN to STOP.*1	The NMT state will change to Pre-operational even when the operation setting at error occurrence is set to "Not to change the NMT state". Note that the NMT state can be changed to Operational even if the CPU module is in the STOP state. When the setting is set to other than "Not to change the NMT state", the NMT state cannot be changed to Operational if the CPU module is in the STOP state.

*1 For details on operation other than the NMT state when the CPU module is switched from RUN state to STOP state, refer to the following.

☞ Page 226 Operation of the RJ71CN91 When the CPU Module Operating Status Is Changed

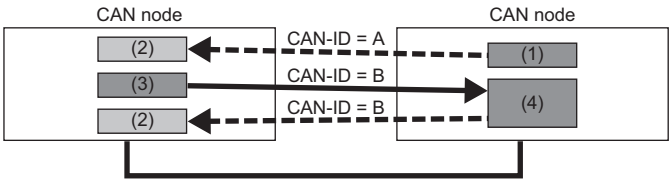
■Setting method

Use the object dictionary to set the operation at error occurrence. For details, refer to the following.

☞ Page 215 Error behavior

1.11 Layer 2 Message Transmission and Receive

Layer 2 messages are used to send/receive data between CAN nodes.
Different from CANopen nodes, CAN nodes do not have node IDs, so they send/receive data based on the CAN-ID set in each message. CAN nodes can also request other nodes to send data by RTR.
Layer 2 messages can send/receive messages when the own node is in Layer 2 online mode.



(1) Data transmission
(2) Data reception
(3) RTR
(4) RTR response
: Data frame
: Remote frame

The following table lists the messages used for data communication and their execution methods.
○: Available, ×: Not available

CAN message	Execution method		Description
	CIF	Layer 2 message	
Data transmission	○	○*1	Sends data to another node.
RTR	○	○	Requests another node to send data.
RTR response	×	○*1	Sends data in response to RTR from another node. (Select either automatic response or manual response.)
Data reception	×	○	Receives data from another node.

*1 When sending data as a Layer 2 message, the same message can be used for both data transmission and RTR response.

Data transmission

This section describes the procedure for sending data using a Layer 2 message.

Setting method

Configure the Layer 2 message data transmission setting in the buffer memory. The following table lists the settings and their locations.

Item	Location	Setting value	Remarks
Send/receive setting	Parameter A of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	FFFFH	The setting value is fixed.
CAN-ID	CAN-ID n of 'Receive/Transmit process data' (Un\G10000 to Un\G10293)	Optional	Set the CAN-ID for the message to be sent.
RTR response setting	Parameter B of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	5FFFH	To use the message also for RTR response, refer to the following. ☞ Page 67 RTR response
RTR response condition	b12 of RTR/new/DLC of 'Receive/Transmit process data' (Un\G10000 to Un\G10293)	0: Invalid	
Frame type	b15 of RTR/new/DLC of 'Receive/Transmit process data' (Un\G10000 to Un\G10293)	0: Send data frame	The setting value is fixed.
Transmission type	Parameter C of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	Optional	Set the transmission timing.
Cycle time	Parameter D of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	Optional	
Data length	Lower byte of RTR/new/DLC of 'Receive/Transmit process data' (Un\G10000 to Un\G10293)	Optional	Set the data length for data to be sent in bytes.

For details, refer to the following.

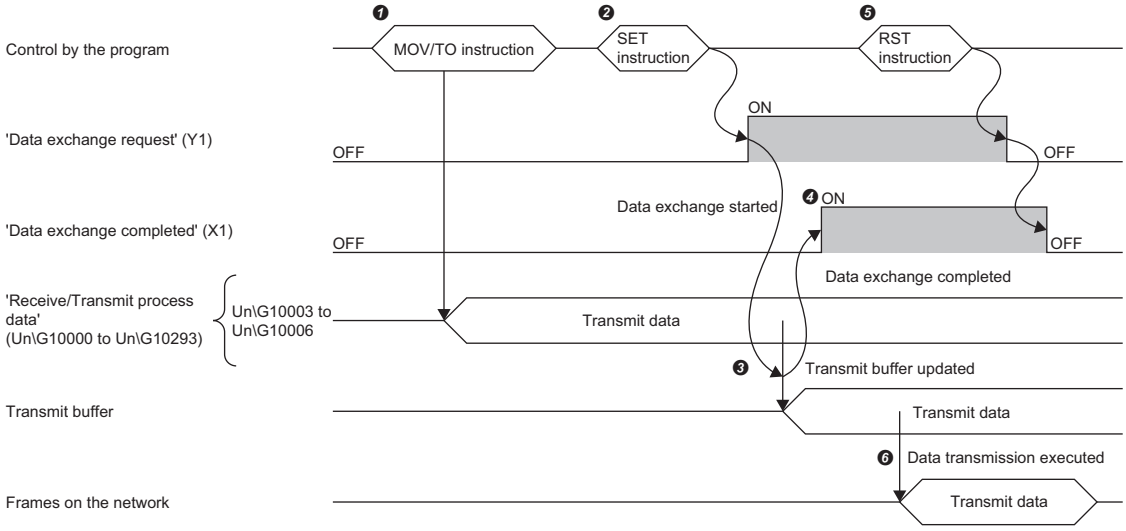
☞ Page 181 Pre-defined Layer 2 message configuration (Un\G6000 to Un\G6167)

☞ Page 189 Receive/Transmit process data (Un\G10000 to Un\G10293)

Flow of data transmission

The following is the procedure up to the point of sending data.

This section describes the procedure for sending data using message slot 1.



❶ Set the send data in Un\G10003 to Un\G10006 of 'Receive/Transmit process data' (Un\G10000 to Un\G10293) via the program. (☞ Page 189 Receive/Transmit process data (Un\G10000 to Un\G10293))

❷ Turn on 'Data exchange request' (Y1) via the program.

❸ Set data is stored in the send buffer of the RJ71CN91 by data exchange.

❹ When data exchange is completed, 'Data exchange completed' (X1) is turned on.

❺ Turn off 'Data exchange request' (Y1) via the program. When 'Data exchange request' (Y1) is turned off, 'Data exchange completed' (X1) is also turned off.

❻ After the send buffer is updated, data is sent under the transmission condition or at the timing corresponding to the setting. (☞ Page 44 Data transmission timing)

Point

To execute ❻, the RJ71CN91 module must be in online mode.

'Layer 2 online mode status' (X3) is turned on when 'Layer 2 online mode request' (Y3) is turned on before ❷ is executed. (The state must be in Layer 2 online mode during network connection except when parameter change work is performed.)

Data transmission timing

The data transmission condition or timing varies depending on the setting of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167). (☞ Page 181 Pre-defined Layer 2 message configuration (Un\G6000 to Un\G6167))

The following describes each setting and timing of transmission.

However, regardless of the transmission type, data is transmitted at the time of transition to online mode and also against a send request by 'Message transmit trigger request' (Y4).

■When the transmission type is Event

Data is sent every time 'Data exchange request' (Y1) is turned on and data exchange is completed.

■When the transmission type is Event (COS)

Data is sent every time 'Data exchange request' (Y1) is turned on and data exchange is completed. However, data is not sent if it is not changed even when 'Data exchange request' (Y1) is turned on and data exchange is completed.

■When the transmission type is Cycle

Data is sent every time the set cycle time has passed.

■When the transmission type is Cycle (COS)

Data is sent every time the set cycle time has passed.

However, regardless of the execution of 'Data exchange request' (Y1), data is not sent if it is not changed even after data exchange is completed.

■When the transmission type is Transmit trigger only

Data is sent each time a send request is made by 'Message transmit trigger request' (Y4).

For details on how to make a send request by 'Message transmit trigger request' (Y4), refer to the following.

☞ Page 155 Message transmit trigger completed (X4), Message transmit trigger request (Y4),Page 187 Message transmit trigger flags (Un\G8400 to Un\G8402)

Point

- Regardless of the transmission type setting, at the time of transition to online mode, send data is automatically updated and sent. Note that to update send data afterward, 'Data exchange request' (Y1) is required.
- When Auto RTR response is selected for RTR response setting, regardless of "Transmission type", data is sent also when a remote transmission request is received.

RTR transmission

This section describes the procedure for sending an RTR using a Layer 2 message.

Setting method

Configure the Layer 2 message RTR transmission setting in the buffer memory. The following table lists the settings and their locations.

Item	Location	Setting value	Remarks
Send/receive setting	Parameter A of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	FFFFH	The setting value is fixed.
CAN-ID	CAN-ID n of 'Receive/Transmit process data' (Un\G10000 to Un\G10293)	Optional	Set the CAN-ID for a data transmission to be requested to another node.
RTR response setting	Parameter B of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	5FFFH	The setting value is fixed.
RTR response condition	b12 of RTR/new/DLC of 'Receive/Transmit process data' (Un\G10000 to Un\G10293)	0: Invalid	The setting value is fixed.
Frame type	b15 of RTR/new/DLC of 'Receive/Transmit process data' (Un\G10000 to Un\G10293)	1: Send RTR frame	The setting value is fixed.
Transmission type	Parameter C of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	One of the following values: 0: Event 2: Cycle 4: Transmission trigger only	Set the transmission timing. However, COS cannot be used.
Cycle time	Parameter D of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	Optional	
Data length	Lower byte of RTR/new/DLC of 'Receive/Transmit process data' (Un\G10000 to Un\G10293)	Optional	Set the data length (in bytes) for a data transmission to be requested to another node.

For details, refer to the following.

☞ Page 181 Pre-defined Layer 2 message configuration (Un\G6000 to Un\G6167)

☞ Page 189 Receive/Transmit process data (Un\G10000 to Un\G10293)

Flow of RTR transmission

The procedure for sending an RTR is the same as the procedure for sending data. (☞ Page 64 Flow of data transmission)
However, for RTR transmission, Receive/Transmit process data is not sent.

RTR transmission timing

The RTR transmission timing is the same as the data transmission timing. (☞ Page 65 Data transmission timing)

Point

Since RTR transmission does not have send data, neither an event (COS) nor a cycle (COS) can be selected.

RTR response

This section describes the procedure for sending data in a Layer 2 message in response to the RTR.
An RTR response data can be sent automatically or manually. Use parameters to specify whether to send an RTR response automatically or manually.

Item	Description	Remarks
Auto	Response data is sent automatically when an RTR is received.	Regardless of RTR reception, it can be sent in any timing as data transmission.
Manual	Response data needs to be sent manually at any timing. (☞ Page 67 Setting method, Page 68 Flow of RTR response)	

Setting method

Configure the Layer 2 message RTR response setting in the buffer memory. The following table lists the settings and their locations.

Item	Location	Setting value	Remarks
Send/receive setting	Parameter A of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	FFFFH	The setting value is fixed.
CAN-ID	CAN-ID n of 'Receive/Transmit process data' (Un\G10000 to Un\G10293)	Optional	Set the CAN-ID for the message to be sent.
RTR response setting	Parameter B of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	6FFFH: Manual RTR response 7FFFH: Auto RTR response	—
RTR response condition	b12 of RTR/new/DLC of 'Receive/Transmit process data' (Un\G10000 to Un\G10293)	Optional	Set the condition for returning a response to a remote transmission request.
Frame type	b15 of RTR/new/DLC of 'Receive/Transmit process data' (Un\G10000 to Un\G10293)	0: Send data frame	The setting value is fixed.
Transmission type	Parameter C of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	Optional	Set the transmission timing for making a manual response.
Cycle time	Parameter D of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	Optional	
Data length	Lower byte of RTR/new/DLC of 'Receive/Transmit process data' (Un\G10000 to Un\G10293)	Optional	Set the data length (in bytes) for data to be sent.

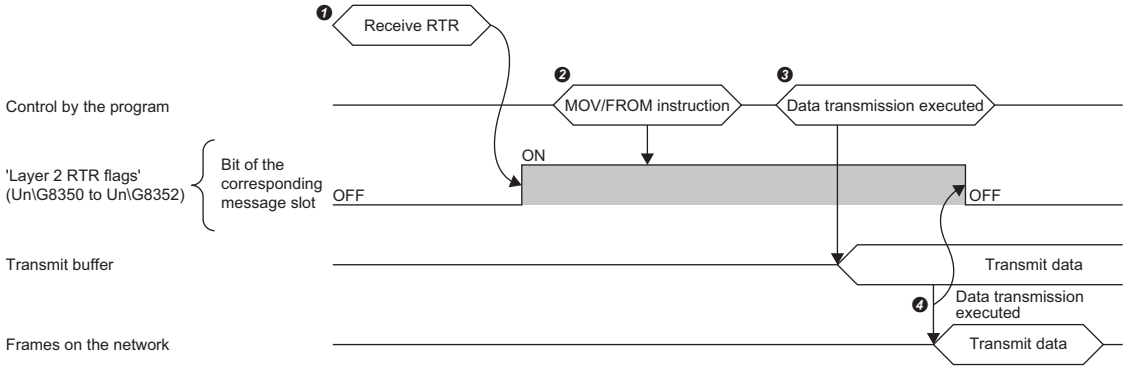
For details, refer to the following.

☞ Page 181 Pre-defined Layer 2 message configuration (Un\G6000 to Un\G6167)
☞ Page 189 Receive/Transmit process data (Un\G10000 to Un\G10293)

Flow of RTR response

The following is the procedure of receiving an RTR and sending response data.

Follow this procedure when sending response data manually.



❶ When the RJ71CN91 receives an RTR, the bit for the corresponding message slot in 'Layer 2 RTR flags' (Un\G8350 to Un\G8352) turns on.

❷ Check via the program whether there is any message slot for which the bit is turned on in 'Layer 2 RTR flags' (Un\G8350 to Un\G8352).

❸ If there is any message slot for which the bit is turned on in 'Layer 2 RTR flags' (Un\G8350 to Un\G8352), send data of the corresponding message slot via the program. (☞ Page 64 Flow of data transmission)

❹ When the data is sent from the corresponding message slot, the bit for the corresponding message slot in 'Layer 2 RTR flags' (Un\G8350 to Un\G8352) turns off.

Point

To receive ❶, the RJ71CN91 module must be in online mode.

'Layer 2 online mode status' (X3) is turns on when 'Layer 2 online mode request' (Y3) is turned on in advance. (The state must be in Layer 2 online mode during network connection except when parameter change work is performed.)

RTR response timing

For automatic data transmission, response data is sent automatically when an RTR is received.

For manual data transmission, the RTR response timing is the same as the data transmission timing. (☞ Page 65 Data transmission timing)

Data reception

This section describes the procedure for receiving data in a Layer 2 message.

Setting method

Configure the Layer 2 message data receive setting in the buffer memory. The following table lists the settings and their locations.

Item	Location	Setting value	Remarks
CAN-ID	Parameters A and B of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	Optional	Set the CAN-ID for the message to be received.
ID filter	Parameters C and D of 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167)	Optional	Set the CAN-ID filter for the message to be received.

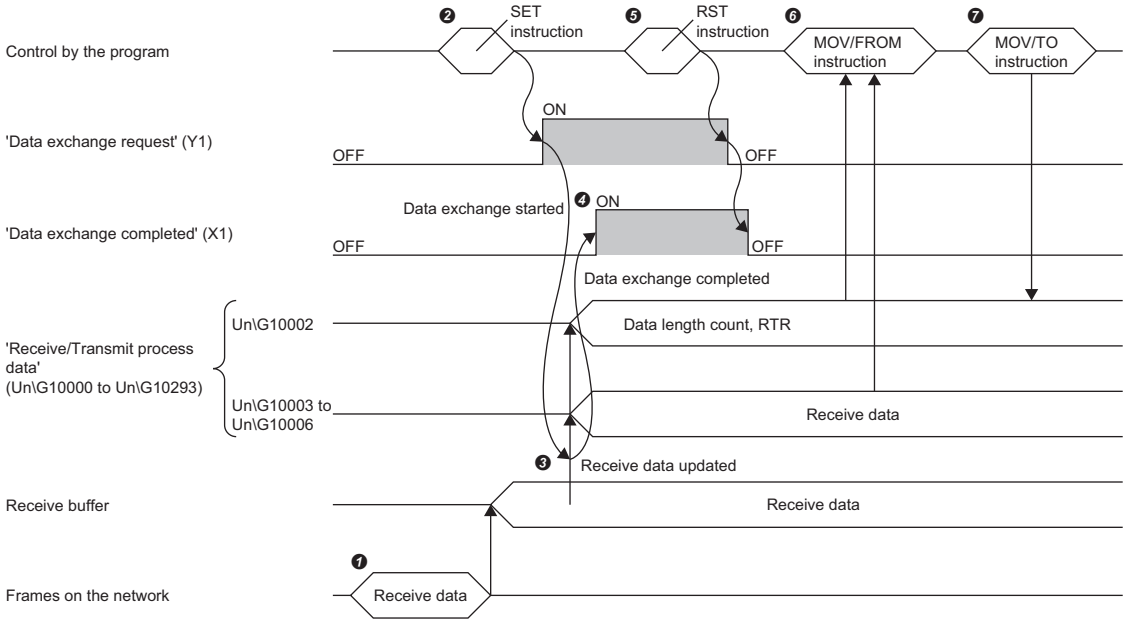
The ID filter filters CAN-IDs.
When the setting value of the ID filter is 00000000H, a message whose CAN-ID is a perfect match will be received.
When the setting value of the ID filter is other than 00000000H, a message matching the bit specified as OFF by the ID filter from the setting value of CAN-ID will be received. Comparison is not made for the bit specified as ON.

Ex.
When the CAN-ID is 40H and 0FH is specified by the ID filter, a message with a CAN-ID in the range 40H to 4FH is received.
For details, refer to the following.
☞ Page 181 Pre-defined Layer 2 message configuration (Un\G6000 to Un\G6167)

Flow of receiving data

The following is the procedure of receiving data.

This section describes the procedure for receiving data using message slot 1.



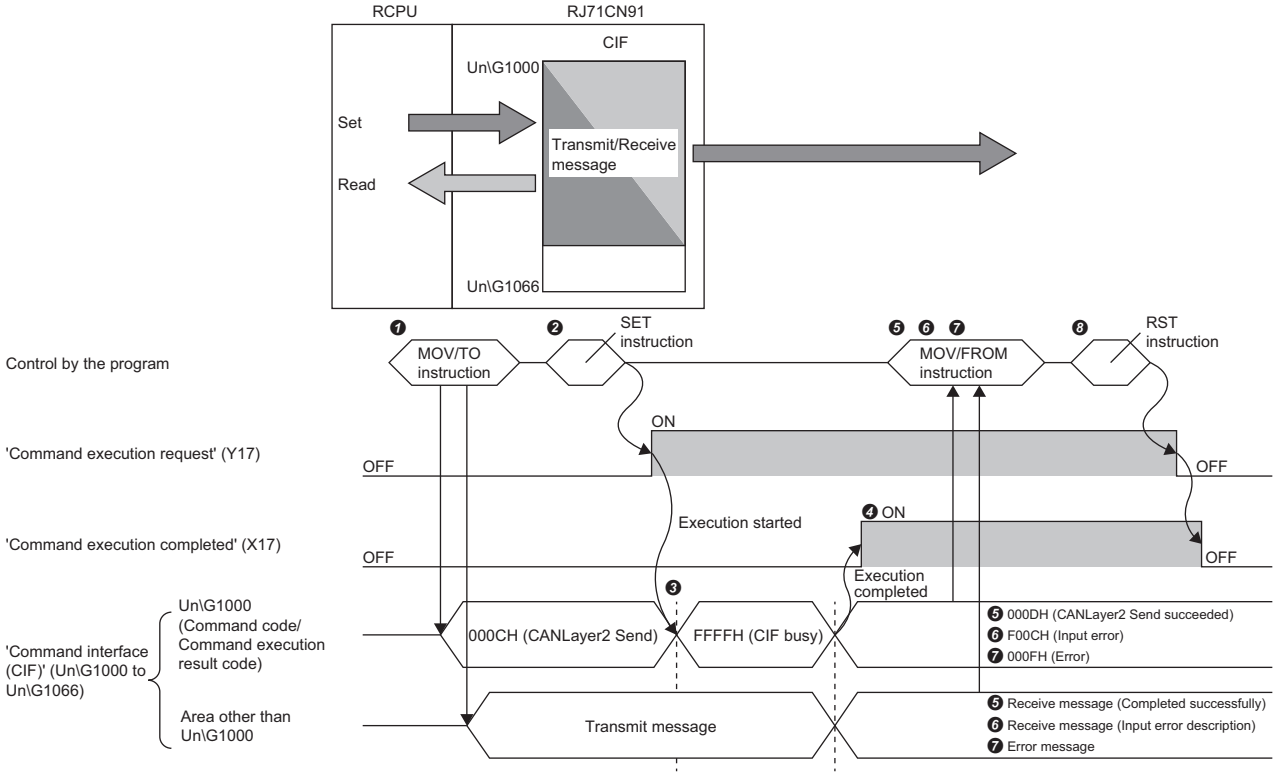
❶ When the RJ71CN91 receives a CAN message, the received data is stored in the receive buffer.
❷ Turn on 'Data exchange request' (Y1) via the program.
❸ The received data is stored in Un\G10003 to Un\G10006 of 'Receive/Transmit process data' (Un\G10000 to Un\G10293) from the receive buffer. (☞ Page 189 Receive/Transmit process data (Un\G10000 to Un\G10293))
❹ When data exchange is completed, 'Data exchange completed' (X1) is turned on.
❺ Turn off 'Data exchange request' (Y1) via the program. When 'Data exchange request' (Y1) is turned off, 'Data exchange completed' (X1) is also turned off.
❻ Read Un\G10002 of 'Receive/Transmit process data' (Un\G10000 to Un\G10293) via the program, and read data as necessary.
❼ Clear the bits 8, 9, and 10 of Un\G10002 of 'Receive/Transmit process data' (Un\G10000 to Un\G10293) via the program. (☞ Page 189 Receive/Transmit process data (Un\G10000 to Un\G10293))

Point

To execute ❸, the RJ71CN91 module must be in online mode.
'Layer 2 online mode status' (X3) is turned on when 'Layer 2 online mode request' (Y3) is turned on before ❷ is executed. (The state must be in Layer 2 online mode during network connection except when parameter change work is performed.)

Data/RTR transmission via CIF

The following is the procedure for sending the Layer 2 message data and RTR using CIF.



❶ Set the transmit message in 'Command interface (CIF)' (Un\G1000 to Un\G1066) via the program. (☞ Page 72 Buffer memory area assignment)
❷ Turn on 'Command execution request' (Y17) via the program.
❸ FFFFH (CIF busy) is stored in Un\G1000 while the command is being processed.
❹ When the command execution is finished, 'Command execution completed' (X17) turns on.
❺ When the process is completed successfully, 000DH (Layer 2 message transmission succeeded) is stored in Un\G1000.
The receive message is stored in an area other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066). (☞ Page 72 Buffer memory area assignment)
❻ When an input error occurs, F00CH (Input error) is stored in Un\G1000.
The input error description is stored in an area other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066). (☞ Page 72 Buffer memory area assignment)
❼ When the process is completed with an error, 000FH (Error) is stored in Un\G1000.
The error message is stored in an area other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066). (☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066))
❽ After Layer 2 message transmission is completed, turn off 'Command execution request' (Y17) via the program. When 'Command execution request' (Y17) is turned off, 'Command execution completed' (X17) is also turned off.

Buffer memory area assignment

This section describes 'Command interface (CIF)' (Un\G1000 to Un\G1066) assignment when sending Layer 2 message data using CIF.

The following table shows the transmit message assignment.

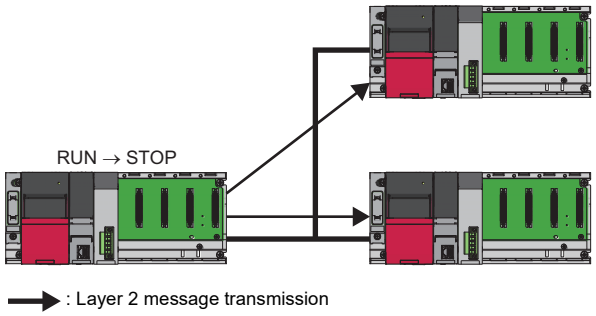
Address	Description
	Transmit message
Un\G1000	Command code • 000CH: Layer 2 message transmission
Un\G1001	CAN-ID (low word)
Un\G1002	CAN-ID (high word)
Un\G1003	Set the transmission type. • 0: Data transmission • 1: RTR transmission
Un\G1004	Set the data length (bytes). • 0 to 8
Un\G1005 to Un\G1008	Set data to transmit. • Un\G1005: The lower byte in first, the upper byte in second ⋮ • Un\G1008: The lower byte in seventh, the upper byte in eighth
Un\G1009 to Un\G1066	System area

The following table shows memory assignment for a receive message.

Address	Description		
	Receive message		
	Completed successfully	**Input error**	**Completed with an error**
Un\G1000	Command execution result code • 000DH: Layer 2 message transmission succeeded	Command execution result code • F00CH: Input error	Command execution result code • 000FH: Error
Un\G1001	System area	CAN-ID input error description • 0000H: No error • Other than 0000H: Input value that caused an error	☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066)
Un\G1002	System area	CAN-ID input error description • 0000H: No error • Other than 0000H: Input value that caused an error	
Un\G1003	System area	RTR input error description • 0000H: No error	
Un\G1004	System area	Data length error description • 0000H: No error • Other than 0000H: Input value that caused an error	
Un\G1005 to Un\G1008	System area		
Un\G1009 to Un\G1066	System area		

1.12 CPU Module STOP Transition Message

The RJ71CN91 uses the CPU module STOP transition message function to send any data using Layer 2 messages to CAN nodes when the CPU module state changes from RUN to STOP.



Setting method

Use parameters to configure the CPU module STOP transition message function. For details, refer to the following.

☞ Page 188 CPU module STOP transition message (Un\G8450 to Un\G8477)

2 PARAMETER SETTINGS

This chapter describes the parameter settings required for communications between the RJ71CN91 and other nodes.

2.1 Setting Parameters

When used in the CANopen 405 mode

The following is the procedure for setting parameters when the RJ71CN91 is used in the CANopen 405 mode.

1. Add the RJ71CN91 in the engineering tool.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ Right-click ⇨ [Add New Module]

2. Basic Setting and Refresh Setting are available for module parameter setting, which can be selected from the tree in the following window.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [RJ71CN91]

3. Start CANopen Configuration Tool to configure CANopen.

4. Write the CANopen setting to the CPU module from CANopen Configuration Tool. (☞ Page 117 Writing the settings)

5. Write the module parameters to the CPU module using the engineering tool.

[Online] ⇨ [Write to PLC]

6. The settings are reflected by resetting the CPU module or powering off and on the system.

When used in the Layer 2 message mode

The following is the procedure for setting parameters when the RJ71CN91 is used in the 11-bit CAN-ID Layer 2 message mode or 29-bit CAN-ID Layer 2 message mode.

1. Add a general-purpose intelligent module (for single slot) in the engineering tool.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ Right-click ⇨ [Add New Module]

2. Set the number of occupied points for the general-purpose intelligent module (for single slot) to 32 points.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [Gen. Intelligent Module (1 slot)] ⇨ Right-click ⇨ [Properties]

3. Set the function mode to the buffer memory.

☞ Page 161 Function mode (Un\G21)

4. Save the buffer memory settings to the flash ROM.

Set 'Save/restore configuration' (Un\G22), then turn on 'Configuration save/factory default configuration restore request' (Y1F), and write the buffer memory setting to the flash ROM.

☞ Page 158 Configuration save/factory default configuration restore completed/request (X1F)/(Y1F)
☞ Page 162 Save/restore configuration (Un\G22)

5. Restart the RJ71CN91 to apply the function mode. (Perform steps 3 to 5 only to change the mode.)

☞ Page 154 Module restart completed (X2), Module restart request (Y2)

6. Set the other parameters of the buffer memory.

☞ Page 163 Baud rate (Un\G24)
☞ Page 181 Pre-defined Layer 2 message configuration (Un\G6000 to Un\G6167)
☞ Page 188 CPU module STOP transition message (Un\G8450 to Un\G8477)
☞ Page 189 Receive/Transmit process data (Un\G10000 to Un\G10293)

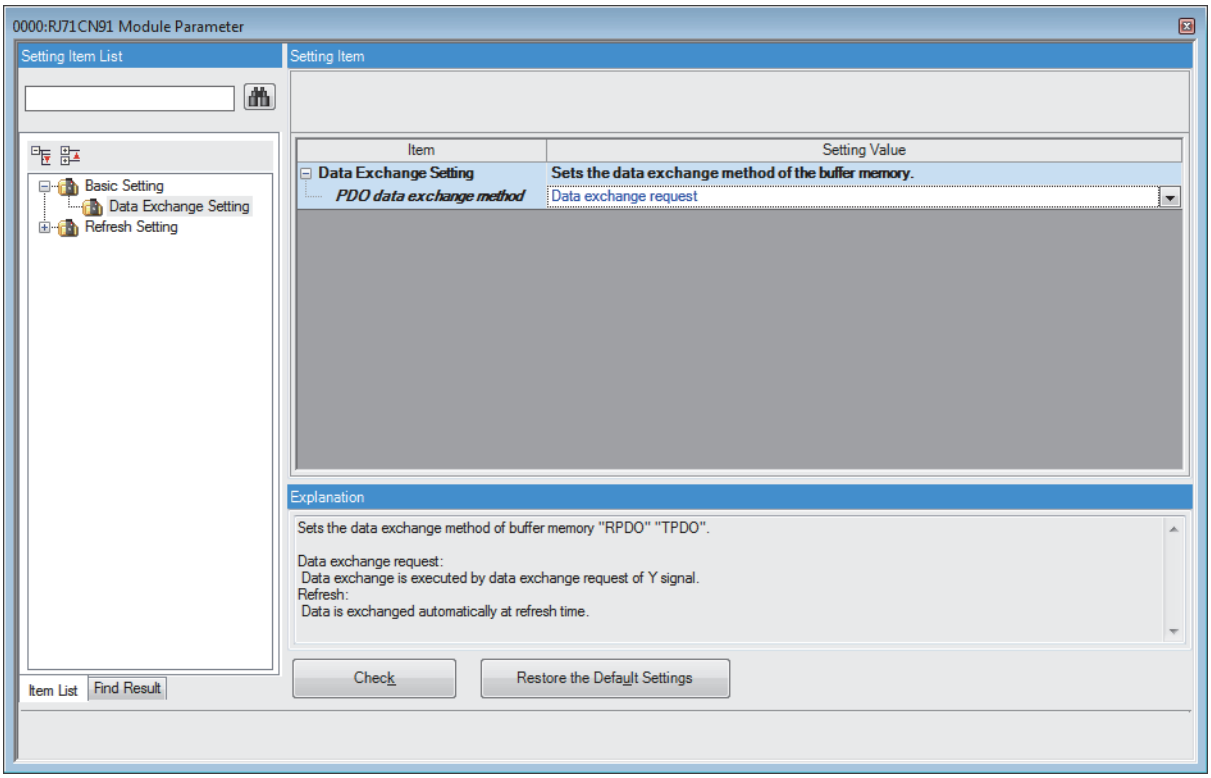
Save the above items to the flash ROM if needed. Restart of the RJ71CN91 is not required.

7. Set the RJ71CN91 to the online mode using 'Layer 2 online mode request' (Y3) and apply the parameters.

☞ Page 155 Layer 2 online mode status (X3), Layer 2 online mode request (Y3)

2.2 Basic Setting

Set the data exchange method for the RJ71CN91 buffer memory.



Data exchange setting

Item	Description	Setting range
PDO data exchange method	Set the data exchange method. • Data exchange request: Data is exchanged when 'Data exchange request' (Y1) is turned on. • Refresh: Data is exchanged at the time of refresh.	• Data exchange request • Refresh (Default: Data exchange request)

Point

Configure at least one refresh setting for RPDO and TPDO in "Refresh Setting" when "PDO data exchange method" is set to "Refresh". If not, 'TPDO' (Un\G13000 to Un\G14023) and 'RPDO' (Un\G10000 to Un\G11023) are not refreshed.

2.3 Refresh Setting

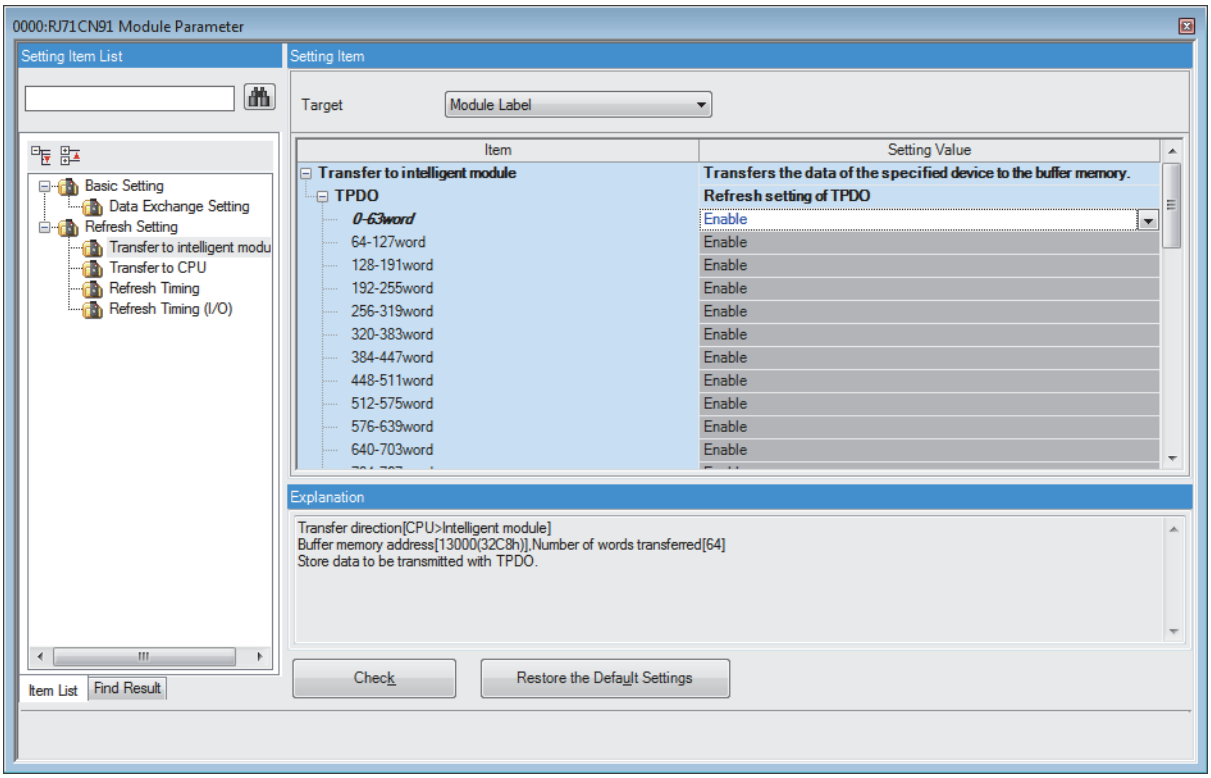
Setting method

Configure the refresh settings of the RJ71CN91 buffer memory.

Configuring the refresh settings eliminates the necessity of read and write processes by the program.

1. Open the module parameter window.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [RJ71CN91] ⇨ [Refresh Setting]



2. Click "Target" and set the refresh target.

• When "Target" is set to "Module Label"

Enable or disable "0-63word" to enable or disable the refresh.

• When "Target" is set to "Refresh Data Register (RD)"

To automatically set the transfer destinations of all items, set "Start Device Name" to the start device.

• When "Target" is set to "Device"

Double-click a relevant item and enter the refresh target device.

3. Click "Refresh Timing" and set the refresh timing.

Set "Refresh Timing" to "At the Execution Time of END Instruction" or "At the Execution Time of Specified Program".

When it is set to "At the Execution Time of Specified Program", double-click "Refresh Group [n](n: 1-64)" and set 1 to 64.

Point

When the refresh is enabled, the values in the refresh target will be enabled at the refresh timing set by the engineering tool. At this time, the buffer memory is overwritten with the values in the refresh target. To change the refreshed values in the buffer memory, change the values in the refresh target module label or device via the program.

Precautions

- The C Controller module cannot use the refresh settings.
- When refresh targets in RPDO and TPDO are set, there are restrictions depending on the firmware version of the CPU module.

Firmware version				Restrictions
RnCPU	RnENCPU	RnPCPU	RnSFCPU	
"27" or earlier	"27" or earlier	"11" or earlier	"13" or earlier	The refresh target cannot be set. If it is set, an error (error code: 2220H) occurs.
"28" to "34"	"28" to "34"	"12" to "13"	—	Data consistency can be guaranteed only one module per CPU module by the refresh.
"35" or later	"35" or later	"14" or later	"14" or later	There is no restrictions.

Refresh processing time

The refresh processing time [μs] is an element that makes up the CPU module scan time. For details on the scan time, refer to the following.

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

The refresh processing time [μs] required for refreshing is calculated as follows.

- Refresh processing time [μs] = Refresh read time (refresh transferred to the CPU) + Refresh write time (refresh transferred to the intelligent function module)

The refresh read time and refresh write time vary depending on the refresh target setting.

When "Target" is set to "Module Label" or "Refresh Data Register (RD)"

The following table shows the refresh read time and refresh write time when the CPU module is used.

Classification	Refresh processing time
Refresh read time	95.20μs
Refresh write time	28.62μs

When "Target" is set to "Device"

Calculate the refresh read time and refresh write time based on the number of items for which the refresh is set and the number of words transferred. For details on the calculation method, refer to the following.

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

2.4 CANopen Setting (Starting CANopen Configuration Tool)

Start CANopen Configuration Tool to configure CANopen.
For details, refer to the following.
☞ Page 81 CANopen Configuration Tool

Point

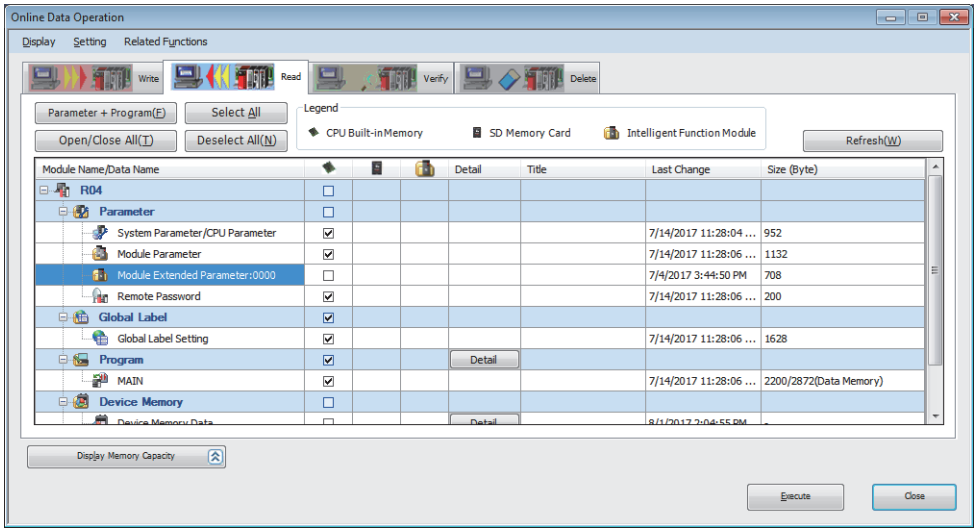
- CANopen Configuration Tool can operate standalone even if GX Works3 terminates. The user can still operate the configuration tool.
- CANopen Configuration Tool cannot be used in either 11-bit CAN-ID Layer 2 message mode or 29-bit CAN-ID Layer 2 message mode.

Parameters set with CANopen Configuration Tool

Parameters set with CANopen Configuration Tool saved into a CPU module or SD memory card can be displayed in GX Works3 as the module extension parameters.
For details on the storage of parameters set with CANopen Configuration Tool, refer to the following.
☞ Page 117 Writing the settings

Point

Module extension parameters cannot be read or edited using such a tool as GX Works3. To read parameters for the RJ71CN91 from a CPU module or SD memory card using such a tool as GX Works3, exclude module extension parameters from the targets.



Object dictionary settings

The object dictionary settings contained in the module extension parameters are treated as CDCF for each setting target node. (Including the settings for the own node)

When the power is turned on or the CPU module is reset, CDCF is saved into Concise DCF (index 1F22H, subindex 01H to 7FH) in the object dictionary of the RJ71CN91. (☞ Page 218 Concise DCF)

When the module extension parameters are saved after being changed, each node ID in the CDCF before change is compared and the following processing is performed.

CDCF before change	Action
The new node ID matches the node ID before change.	The new CDCF is written over the previous CDCF.
The new node ID does not match the node ID before change.	The CDCF before change remains, and the new CDCF is also saved.

When the own node is the NMT master, if some unnecessary CDCF still remains, the configuration manager function may operate to change the settings for other nodes. Therefore, to change the module extension parameters (node ID change or node ID deletion), delete all CDCFs using CANopen Configuration Tool.*1

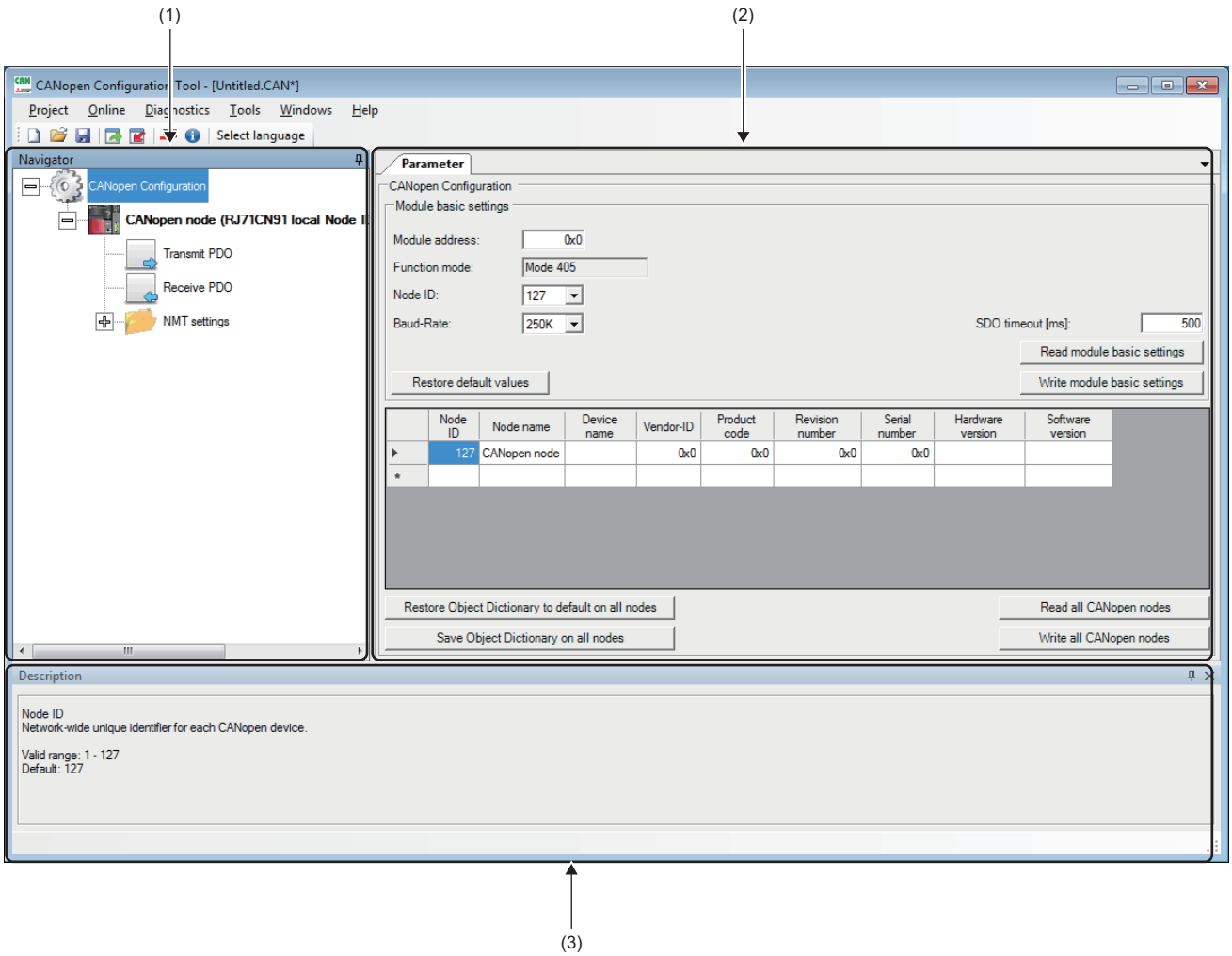
*1 All CDCFs can be deleted by using the [Restore Object Dictionary to default on all nodes] button in the "CANopen Configuration" window.
☞ Page 84 "CANopen Configuration" window

3 CANopen Configuration Tool

This chapter describes CANopen Configuration Tool.

3.1 Window Structure

The following figure shows the window structure.



No.	Name	Reference
(1)	Navigator window	—
(2)	Parameter window	Page 84 Parameter window
(3)	Description window	Page 103 Description window

Menu

The following table lists the menu items of CANopen Configuration Tool.

Item	Description	Reference
Project	Creates, saves, exports, or imports a project.	Page 82 Project
Online	For the CPU module, sets the connection destination or writes the settings configured with CANopen Configuration Tool.	Page 82 Online
Diagnostics	The status of the RJ71CN91 module can be checked or a search can be made for the connected CANopen node.	Page 82 Diagnostics
Tools	The display language can be selected.	Page 82 Tools
Windows	Whether to display/hide the description window can be selected.	Page 83 Windows
Help	Displays the version information of CANopen Configuration Tool.	Page 83 Help

Project

Creates, saves, exports, or imports a project.

Item	Description
New	Creates a new project.
Open	Opens a saved project file.*1
Close	Closes the currently opened project.
Save	Saves the currently opened project.
Save as	Saves the currently opened project with a different name.
Export	Exports the currently opened project in XML format. (☞ Page 120 Export)
Import	Imports a project file saved in XML format. (☞ Page 120 Import)
Recent Projects	Displays the names of 10 recently used projects from the saved projects. (Excluding projects opened by import operation)
Exit	Closes CANopen Configuration Tool.

*1 A project file saved with Version 1.01B cannot be opened with Version 1.00A.

Online

For the CPU module, sets the connection destination or writes the settings configured with CANopen Configuration Tool.

Item	Description
Transfer Setup	To communicate with the CPU module, the connection destination can be set and a communication test can be conducted. (☞ Page 105 Transfer setup)
Download Configuration	Writes the set project into the CPU module. (☞ Page 117 Writing the settings)
SDO send / receive	SDO read and SDO write can be executed. Also, the execution results can be checked. (☞ Page 119 SDO send/receive)
NMT master reset	Resets and restarts the connected NMT master. (☞ Page 123 NMT master reset)

Diagnostics

A search can be made for the connected CANopen node or the status of the CANopen node can be checked.

Item	Description
Network scan	A search can be made for all CANopen nodes connected to the network. (☞ Page 118 Network scan)
Module status	The status of the connected CANopen node can be checked. (☞ Page 121 Module status)

Tools

The display language can be selected.

Item	Description
Language	The display language for CANopen Configuration Tool can be selected. (Default: English) (☞ Page 122 Select language)

Windows

Whether to display/hide the description window can be selected.

Item	Description
Description	Whether to display/hide the description window can be selected.

Help

Displays the version information of CANopen Configuration Tool.

Item	Description
About	The version information of CANopen Configuration Tool can be checked. (☞ Page 124 Checking the Software Version)

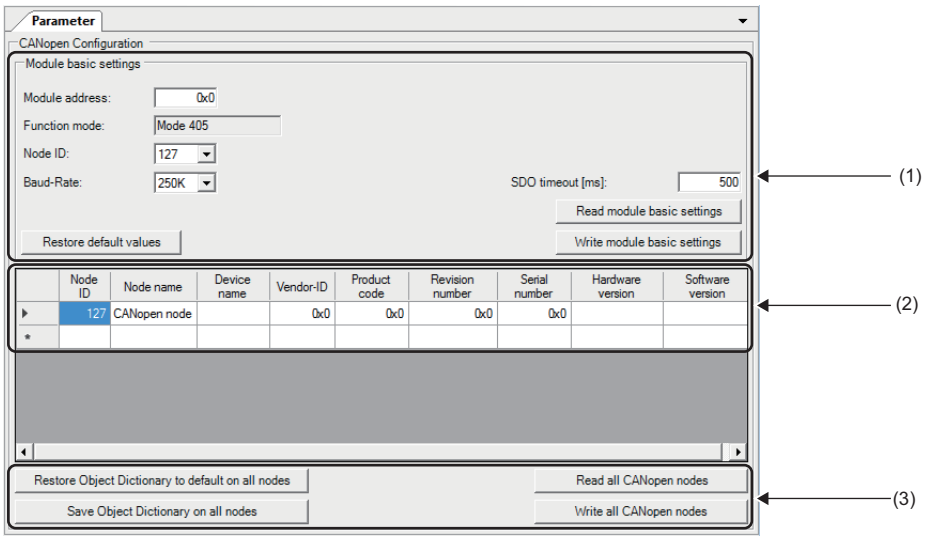
Parameter window

The following describes the windows displayed in the parameter window tab page.

"CANopen Configuration" window

This window is used for setting such as the node ID, baud rate for the CANopen node.

[Navigator] ⇨ [CANopen Configuration]



(1) Module basic settings
(2) CANopen node list
(3) Operation buttons

■Module basic settings

Item	Description	Setting range
Module address	Set the start I/O number of the RJ71CN91 module in units of 16 points. Set it as set in GX Works3.	0x0 to 0xFE0 (Default: 0x0)
Function mode	Displays the function mode of the RJ71CN91 module.	—
Node ID	Set the node ID for the RJ71CN91 module.	1 to 127 (Default: 127)
Baud-Rate	Set the baud rate for CANopen. (Unit: bps) Set the same value for all CANopen nodes connected to CANopen.	• 10K • 20K • 50K • 125K • 250K • 500K • 800K • 1000K (Default: 250K)
SDO timeout	Set the timeout time for SDO communication. (Unit: ms)	50 to 32767 (Default: 500)
[Restore default values] button	Restores the default values for the module basic settings.	—
[Read module basic settings] button	Reads the basic settings of the connected the RJ71CN91 module.	—
[Write module basic settings] button	Writes the values set for the module basic settings into the connected the RJ71CN91 module. Please note that if the module basic settings are already written in the connected the RJ71CN91 module, they will be overwritten.	—

■CANopen node list

Using the configuration manager, parameters (CDCF) to be set for other nodes can be added. By operating a button such as the "Write CANopen node" button, parameters can be read from or written to other nodes currently connected.

Item	Description	Setting range
Node ID	Set the node ID for the CANopen node.	1 to 127 (Default: 127)
Node name	Set the name for the CANopen node.	256 or less one-byte or two-byte characters can be used. (Default: CANopen node)
Device name	Displays the device name of CANopen node.*1	—
Vendor-ID	Displays the vendor ID of CANopen node.*1	—
Product code	Displays the product code of CANopen node.*1	—
Revision number	Displays the revision number of CANopen node.*1	—
Serial number	Displays the serial number of CANopen node.*1	—
Hardware version	Displays the hardware version of CANopen node.*1	—
Software version	Displays the software version of CANopen node.*1	—

*1 The name and number assigned by the CANopen node device manufacturer are displayed.

Point

- A new CANopen node can be added in the CANopen node list by setting a node ID in an empty row.
- To delete a CANopen node from the CANopen node list, select the left end of the target row, and press the Delete button.
- A maximum of 60 CANopen nodes can be added.
- When parameters set with CANopen Configuration Tool are written to a CPU module, the settings for the object dictionary of the own node are also treated as CDCF.

■Operation buttons

Item	Description
[Restore Object Dictionary to default on all nodes] button	Restores the object dictionary of all CANopen nodes connected to CANopen to default.
[Save Object Dictionary on all nodes] button	Saves the settings for the object dictionary of all CANopen nodes connected to CANopen into non-volatile memory.
[Read all CANopen nodes] button	Reads the following information from all CANopen nodes connected to CANopen. • Transmit PDO (TPDO) • Receive PDO (RPDO) • NMT settings (NMT master/slave, heartbeat)
[Write all CANopen nodes] button	Writes the following information to all CANopen nodes connected to CANopen. • Transmit PDO (TPDO) • Receive PDO (RPDO) • NMT settings (NMT master/slave, heartbeat)

Point

Stop writing data to all target nodes except this operation to write the data using [Write all CANopen nodes] button. (Writing data to a target node can be performed by using the configuration tool, via SDO communications, or by the function (such as a configuration manager) using SDO communications.)
If the writing is not stopped, a communication error may occur or the written parameters may be broken.
The following example shows how to stop writing data using by the RJ71CN91 or CANopen Configuration Tool.

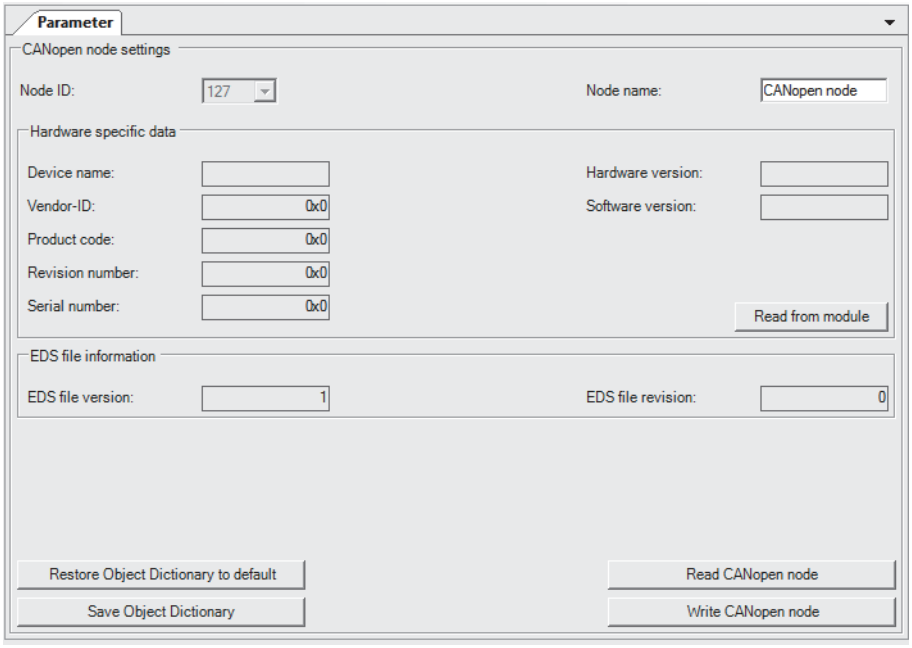
- Set the CPU module which controls the RJ71CN91 to the STOP state. (Stop the SDO communications by using the program or system.)
- Stop the operation performed by CANopen Configuration Tool.

For how to stop writing data using by nodes or the configuration tool other than above, refer to the manual for nodes or configuration tool used.

"CANopen node settings" window

This window is used to check the information of the CANopen node.

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name



Item	Description	Setting range
Node ID	Displays the number set for the node ID in the CANopen node list. (☞ Page 85 CANopen node list)	—
Node name	Set the name for the CANopen node.	256 or less one-byte or two-byte characters can be used. (Default: CANopen node)

■Hardware specific data

Item	Description	Setting range
Device name	Displays the device name of CANopen node.*1	—
Vendor-ID	Displays the vendor ID of CANopen node.*1	—
Product code	Displays the product code of CANopen node.*1	—
Revision number	Displays the revision number of CANopen node.*1	—
Serial number	Displays the serial number of CANopen node.*1	—
Hardware version	Displays the hardware version of CANopen node.*1	—
Software version	Displays the software version of CANopen node.*1	—
[Read from module] button	Reads hardware information from the connected CANopen node.	—

*1 The name and number assigned by the CANopen node device manufacturer are displayed.

■EDS file information

EDS file information is displayed only when a CANopen node that supports EDS files is selected.

Item	Description	Setting range
EDS file version	Displays the EDS file version.	—
EDS file revision	Displays the EDS file revision.	—

Point

EDS files are files defining CANopen device information.

■Operation buttons

Item	Description
[Restore Object Dictionary to default] button	Restores the object dictionary of the connected CANopen node to default.
[Save Object Dictionary] button	Saves the current settings for the object dictionary of the connected CANopen node to non-volatile memory.
[Read CANopen node] button	Reads the following information from the connected CANopen node. • Transmit PDO (TPDO) • Receive PDO (RPDO) • NMT settings (NMT master/slave, heartbeat)
[Write CANopen node] button	Writes the following information to the connected CANopen node. • Transmit PDO (TPDO) • Receive PDO (RPDO) • NMT settings (NMT master/slave, heartbeat)

Point

Stop writing data to all target nodes except this operation to write the data using [Write CANopen node] button. (Writing data to a target node can be performed by using the configuration tool, via SDO communications, or by the function (such as a configuration manager) using SDO communications.)
If the writing is not stopped, a communication error may occur or the written parameters may be broken.
The following example shows how to stop writing data using by the RJ71CN91 or CANopen Configuration Tool.

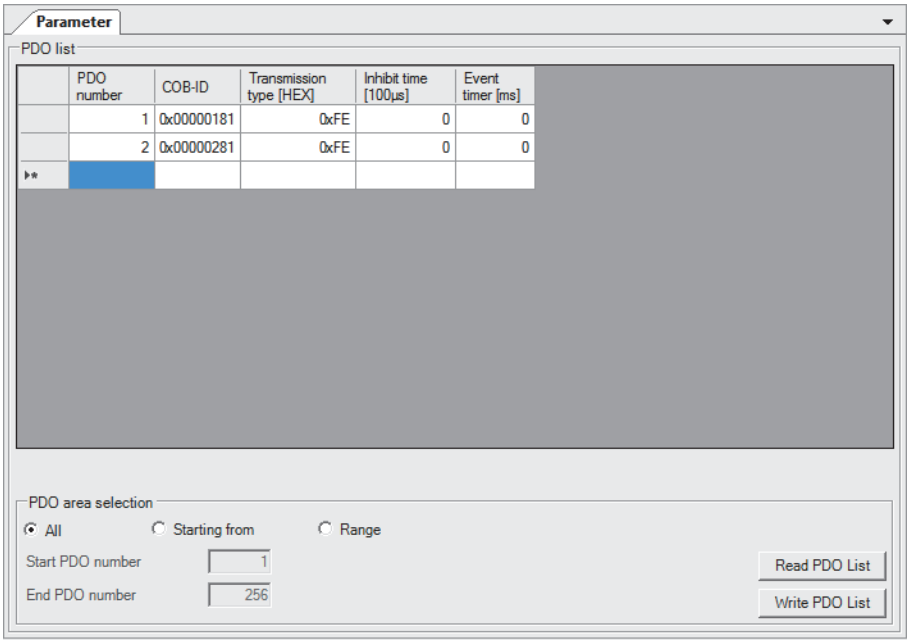
- Set the CPU module which controls the RJ71CN91 to the STOP state. (Stop the SDO communications in the program or system.)
- Stop the operation performed by CANopen Configuration Tool.

For how to stop writing data using by nodes or the configuration tool other than above, refer to the manual for nodes or configuration tool used.

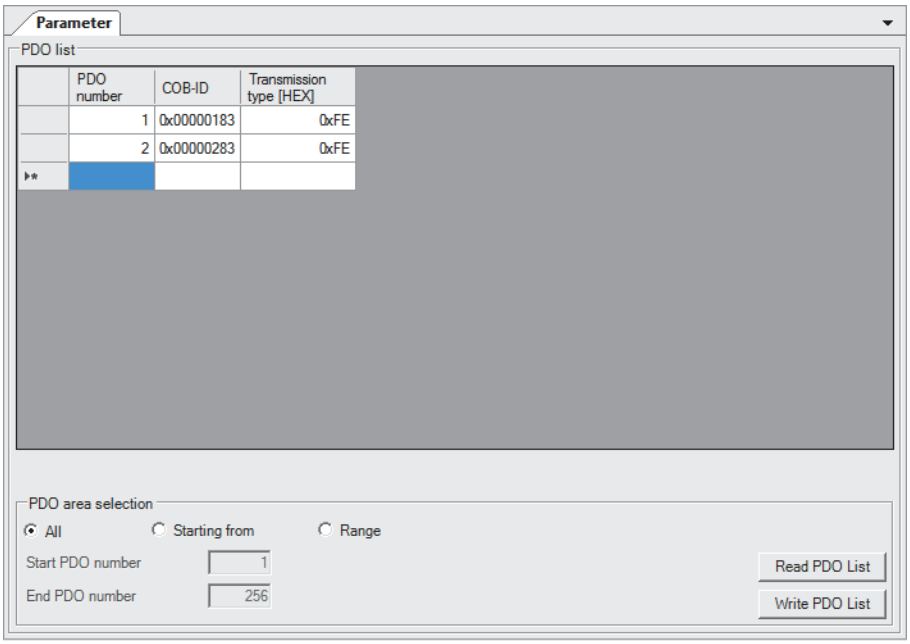
PDO list window

This window displays a list of TPDOs and RPDOs.

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [Transmit PDO]



[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [Receive PDO]



■PDO list

Item	Description	Setting range
PDO number	Set the number for identifying the PDO.	1 to 256 (Default: 1)
COB-ID	Set the COB-ID for the PDO. The COB-ID is the ID referred to in CANopen. By setting the COB-ID, the CAN-ID is determined.	—
Transmission type	■For TPDO Set the TPDO transmission type. • 0x00: Synchronous (acyclic)*1 • 0x01: Synchronous (Send data every time a SYNC message is received.) • 0x02: Synchronous (Send data when a SYNC message is received once.) • 0x03: Synchronous (Send data when SYNC messages are received twice.) ⋮ • 0xF0: Synchronous (Send data when SYNC messages are received 239 times.) • 0xF1 to 0xFD: System-reserved • 0xFE: Event-driven • 0xFF: Event-driven ■For RPDO Set the RPDO transmission type. • 0x00 to 0xF0: Synchronous*2 • 0xF1 to 0xFD: System-reserved • 0xFE: Event-driven • 0xFF: Event-driven	0x00 to 0xFF (Default: 0xFE)
Inhibit time	Set the minimum time interval between PDO transmissions. (Unit: 100μs) To disable this item, set 0.	0 to 65535 (Default: 0)
Event timer	Set the event timer. (Unit: ms) When an event-driven transfer is not executed by the time the event timer times out, a message containing the current value for the object dictionary will be sent. To disable this item, set 0.	0 to 65535 (Default: 0)

*1 A PDO is sent after a SYNC message is generated. However, it is acyclic and sent only when an event occurs before SYNC message generation.

*2 The received PDO data is processed after the next SYNC message is received, regardless of the transmission rate specified by the transmission type.

Point

- A new PDO can be added in the PDO list by setting a PDO number in an empty row.
- To delete a PDO from the PDO list, select the left end of the target row, and press the Delete button.
- A maximum of 256 PDOs can be added.
- To delete all PDOs from the PDO list, select the upper left of the list, and press the Delete button.

■PDO area selection

To read and write from the PDO list, a choice can be made from "All", "Starting from", and "Range".

When "Starting from" is selected, specify any number for "Start PDO number". (1 to 256)

When "Range" is selected, specify any number for "Start PDO number" and "End PDO number". (1 to 256)

Item	Description
[Read PDO List] button	Reads PDOs from the connected CANopen node.
[Write PDO List] button	Writes the PDOs to the flash ROM of the connected CANopen node.

Point

Stop writing data to all target nodes except this operation to write the data using [Write PDO List] button. (Writing data to a target node can be performed by using the configuration tool, via SDO communications, or by the function (such as a configuration manager) using SDO communications.)

If the writing is not stopped, a communication error may occur or the written parameters may be broken.

The following example shows how to stop writing data using by the RJ71CN91 or CANopen Configuration Tool.

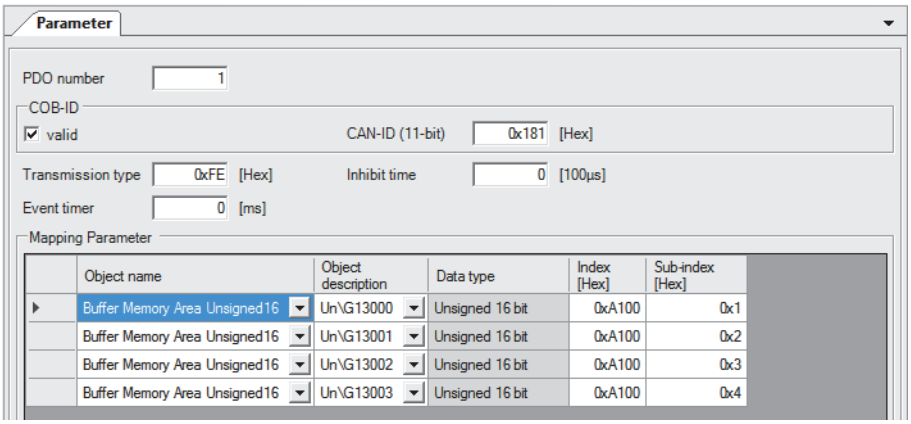
- Set the CPU module which controls the RJ71CN91 to the STOP state. (Stop the SDO communications in the program or system.)
- Stop the operation performed by CANopen Configuration Tool.

For how to stop writing data using by nodes or the configuration tool other than above, refer to the manual for nodes or configuration tool used.

TPDO details window

This window is used for setting detailed parameters for the TPDO.

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [Transmit PDO] ⇨ [PDO 1] (When the PDO number is 1)



Item	Description	Setting range
PDO number	Set the number for identifying the PDO.	1 to 256 (Default: 1)
Transmission type	Set the TPDO transmission type. • 0x00: Synchronous (acyclic)*1 • 0x01: Synchronous (Send data every time a SYNC message is received.) • 0x02: Synchronous (Send data when a SYNC message is received once.) • 0x03: Synchronous (Send data when SYNC messages are received twice.) ⋮ • 0xF0: Synchronous (Send data when SYNC messages are received 239 times.) • 0xF1 to 0xFD: System-reserved • 0xFE: Event-driven • 0xFF: Event-driven	0x00 to 0xFF (Default: 0xFE)
Inhibit time	Set the minimum time interval between PDO transmissions. (Unit: 100µs) To disable this item, set 0.	0 to 65535 (Default: 0)
Event timer	Set the event timer. (Unit: ms) When a data exchange is not executed by the time the event timer times out, a message containing the current value for the object dictionary will be sent. To disable this item, set 0.	0 to 65535 (Default: 0)

*1 A PDO is sent after a SYNC message is generated. However, it is acyclic and sent only when an event occurs before SYNC message generation.

■COB-ID

Set the details of the COB-ID set in the PDO list window.

Item	Description	Setting range
valid	• Selected: The PDO is valid. • Not selected: The PDO is not valid.	• Not selected • Selected (Default: Not selected)
CAN-ID(11-bit)	Set the CAN-ID for the COB-ID. • TPDO1: 0x181 to 0x1FF • TPDO2: 0x281 to 0x2FF • TPDO3: 0x381 to 0x3FF • TPDO4: 0x481 to 0x4FF	0x000 to 0x57F (Default: 0x181 (When the node ID is 1))

■Mapping Parameter

Set the objects to be mapped to the PDO.

Item	Description	Setting range
Object name	Set the object name to be used in the buffer memory area.	• Buffer Memory Area Integer8 • Buffer Memory Area Unsigned8 • Buffer Memory Area Integer16 • Buffer Memory Area Unsigned16 • Buffer Memory Area Integer32 • Buffer Memory Area Unsigned32 • Buffer Memory Area Real32 (Default: Blank)
Object description	Set the buffer memory address to be used in the object. ■When "Buffer Memory Area Integer8" or "Buffer Memory Area Unsigned8" is specified for "Object name" • Un\G13000 LByte • Un\G13000 HByte ⋮ • Un\G14023 LByte • Un\G14023 HByte ■When "Buffer Memory Area Integer16" or "Buffer Memory Area Unsigned16" is specified for "Object name" • Un\G13000 • Un\G13001 ⋮ • Un\G14023 ■When "Buffer Memory Area Integer32", "Buffer Memory Area Unsigned32", or "Buffer Memory Area Real32" is specified for "Object name" • Un\G13000-Un\G13001 • Un\G13002-Un\G13003 ⋮ • Un\G14022-Un\G14023	Refer to the left.
Data type	Displays the data type of send data. The type depends on what is specified for "Object name". • Signed 8 bit • Signed 16 bit • Signed 32 bit • Unsigned 8 bit • Unsigned 16 bit • Unsigned 32 bit • Float 32 bit	—
Index	Specify the index in the object dictionary. For indexes, refer to the following. ☞ Page 191 Object Dictionary	0x0000 to 0xFFFF (Default: Blank)
Sub-index	Specify the subindex in the object dictionary. For subindexes, refer to the following. ☞ Page 191 Object Dictionary	0x00 to 0xFF (Default: Blank)

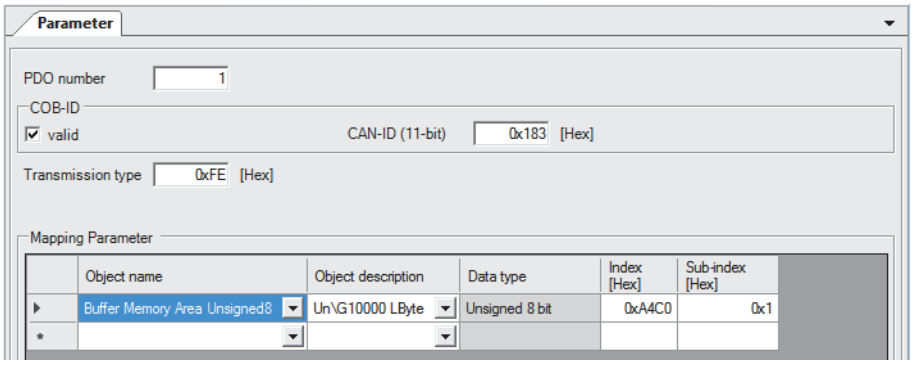
Point

- By setting values for "Object name" and "Object description", the corresponding "Index" and "Sub-index" are automatically displayed.
- By entering values for "Index" and "Sub-index", the corresponding "Object name" and "Object description" are automatically generated.
- Up to 64-bit data can be set for mapping parameters.

RPDO details window

This window is used for setting detailed parameters for the RPDO.

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [Receive PDO] ⇨ [PDO 1] (When the PDO number is 1)



Item	Description	Setting range
PDO number	Set the number for identifying the PDO.	1 to 256 (Default: 1)
Transmission type	Set the RPDO transmission type. • 0x00 to 0xF0: Synchronous*1 • 0xF1 to 0xFD: System-reserved • 0xFE: Event-driven • 0xFF: Event-driven	0x00 to 0xFF (Default: 0xFE)

*1 The received PDO data is processed after the next SYNC message is received, regardless of the transmission rate specified by the transmission type.

■COB-ID

Set the details of the COB-ID set in the PDO list window.

Item	Description	Setting range
valid	• Selected: The PDO is valid. • Not selected: The PDO is not valid.	• Not selected • Selected (Default: Not selected)
CAN-ID(11-bit)	Set the CAN-ID for the COB-ID. • RPDO1: 0x201 to 0x27F • RPDO2: 0x301 to 0x37F • RPDO3: 0x401 to 0x47F • RPDO4: 0x501 to 0x57F	0x000 to 0x57F (Default: 0x201 (When the node ID is 1))

■Mapping Parameter

Set the objects to be mapped to the PDO.

Item	Description	Setting range
Object name	Set the object name to be used in the buffer memory area. If mapping is not required, set dummies.	• Buffer Memory Area Integer8 • Buffer Memory Area Unsigned8 • Buffer Memory Area Integer16 • Buffer Memory Area Unsigned16 • Buffer Memory Area Integer32 • Buffer Memory Area Unsigned32 • Buffer Memory Area Real32 • Dummy Signed 8 bit • Dummy Signed 16 bit • Dummy Signed 32 bit • Dummy Unsigned 8 bit • Dummy Unsigned 16 bit • Dummy Unsigned 32 bit (Default: Blank)
Object description	Set the buffer memory address to be used in the object. ■When "Buffer Memory Area Integer8", "Buffer Memory Area Unsigned8", "Dummy Signed 8 bit", or "Dummy Unsigned 8 bit" is specified for "Object name" • Un\G10000 LByte • Un\G10000 HByte ⋮ • Un\G11023 LByte • Un\G11023 HByte ■When "Buffer Memory Area Integer16", "Buffer Memory Area Unsigned16", "Dummy Signed 16 bit", or "Dummy Unsigned 16 bit" is specified for "Object name" • Un\G10000 • Un\G10001 ⋮ • Un\G11023 ■When "Buffer Memory Area Integer32", "Buffer Memory Area Unsigned32", "Buffer Memory Area Real32", "Dummy Signed 32 bit", or "Dummy Unsigned 32 bit" is specified for "Object name" • Un\G10000-Un\G10001 • Un\G10002-Un\G10003 ⋮ • Un\G11022-Un\G11023	Refer to the left.
Data type	Displays the data type of send data. The type depends on what is specified for "Object name". • Signed 8 bit • Signed 16 bit • Signed 32 bit • Unsigned 8 bit • Unsigned 16 bit • Unsigned 32 bit • Float 32 bit	—
Index	Specify the index in the object dictionary. For indexes, refer to the following. ☞ Page 191 Object Dictionary	0x0000 to 0xFFFF (Default: Blank)
Sub-index	Specify the subindex in the object dictionary. For subindexes, refer to the following. ☞ Page 191 Object Dictionary	0x00 to 0xFF (Default: Blank)

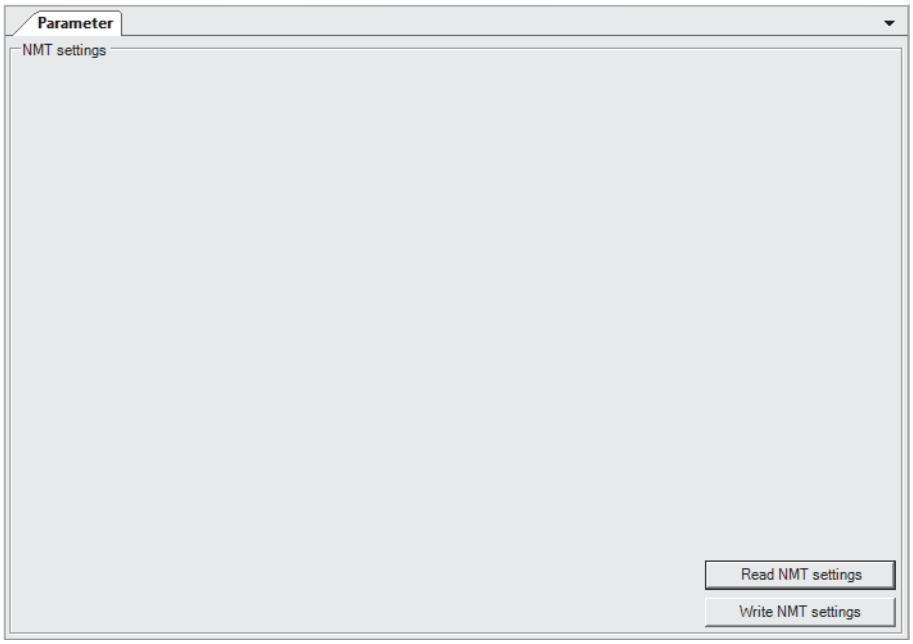
Point

- By setting values for "Object name" and "Object description", the corresponding "Index" and "Sub-index" are automatically displayed.
- By entering values for "Index" and "Sub-index", the corresponding "Object name" and "Object description" are automatically generated.
- Up to 64-bit data can be set for mapping parameters.

"NMT settings" window

This window is used for reading or writing NMT settings.

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [NMT settings]



Item	Description
[Read NMT settings] button	Reads the NMT settings from the connected CANopen node.
[Write NMT settings] button	Writes the NMT settings to the flash ROM of the connected CANopen node.

Point

Stop writing data to all target nodes except this operation to write the data using [Write NMT settings] button. (Writing data to a target node can be performed by using the configuration tool, via SDO communications, or by the function (such as a configuration manager) using SDO communications.)
If the writing is not stopped, a communication error may occur or the written parameters may be broken.
The following example shows how to stop writing data using by the RJ71CN91 or CANopen Configuration Tool.

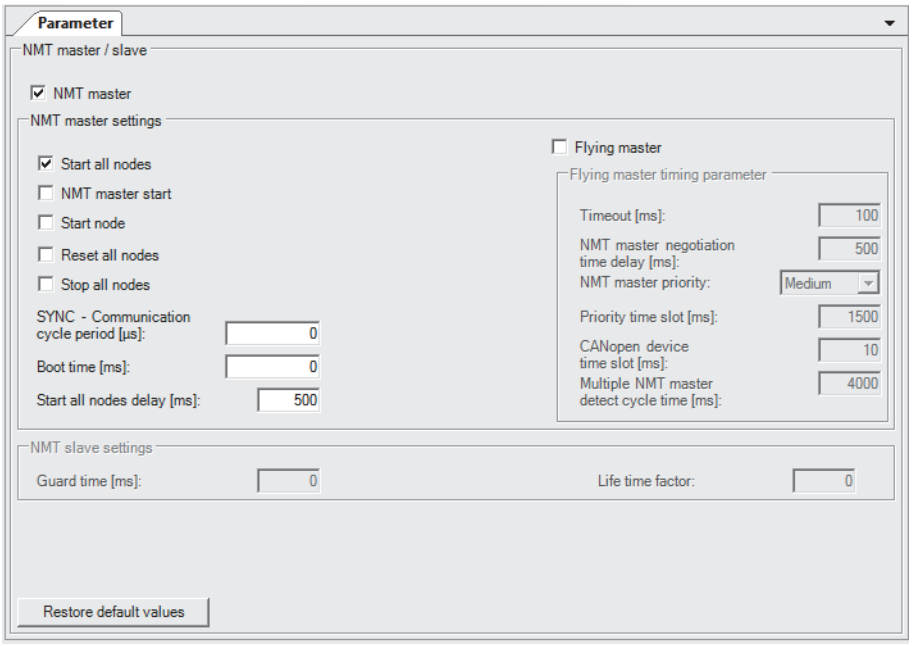
- Set the CPU module which controls the RJ71CN91 to the STOP state. (Stop the SDO communications in the program or system.)
- Stop the operation performed by CANopen Configuration Tool.

For how to stop writing data using by nodes or the configuration tool other than above, refer to the manual for nodes or configuration tool used.

"NMT master / slave" window

This window is used for setting the NMT master and NMT slave parameters to the connected CANopen node.

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [NMT settings] ⇨ [NMT master / slave]



Item	Description	Setting range
NMT master	Set the own node type. • Not selected: NMT slave • Selected: NMT master	• Not selected • Selected (Default: Not selected)

■NMT master settings

Set detailed parameters for the NMT master.

Item	Description	Setting range
Start all nodes	Set the method to start the NMT slaves by sending NMT service. • Not selected: Send Remote node start to each NMT slave. • Selected: Send Remote node start excluding NMT master.	• Not selected • Selected (Default: Not selected)
NMT master start	Set whether to start the node automatically as the NMT master. • Not selected: Shift automatically. • Selected: Do not shift automatically.	• Not selected • Selected (Default: Not selected)
Start node	Set the startup method for NMT slaves. • Not selected: The NMT master shall start the NMT slaves. • Selected: The NMT master shall not start the NMT slaves, and the NMT slaves shall be started by a program.	• Not selected • Selected (Default: Not selected)
Reset all nodes	Set whether to execute a node reset if a mandatory slave fails to respond to node guarding or heartbeat. • Not selected: Execute communication reset in only CANopen nodes where an error occurred. • Selected: Execute communication reset for all nodes. When "Stop all nodes" is selected, this setting is disabled.	• Not selected • Selected (Default: Not selected)
Stop all nodes	Set whether to execute a remote node stop if a mandatory slave fails to respond to node guarding or heartbeat. • Not selected: Do not stop all nodes. • Selected: Stop all nodes.	• Not selected • Selected (Default: Not selected)
SYNC - Communication cycle period	Set the transmission cycle of the SYNC message. (Unit: µs)	0 to 4294967295 (Default: 0)
Boot time	Set the boot time. (Unit: ms)	0 to 4294967295 (Default: 0)
Start all nodes delay	Set the NMT Start all Nodes delay time. (Unit: ms)	0 to 65535 (Default: 500)
Flying master	Set whether to use flying master in the own node. • Not selected: Do not use flying master. • Selected: Use flying master.	• Not selected • Selected (Default: Not selected)

■Flying master timing parameter

Set detailed parameters for flying master.

Item	Description	Setting range
Timeout	Set the NMT master response waiting time. (Unit: ms)	0 to 65535 (Default: 100)
NMT master negotiation time delay	Set the waiting time before starting NMT master negotiation. (Unit: ms) This waiting time is set to secure time to allow other devices to be initialized before deciding the active NMT master.	0 to 65535 (Default: 500)
NMT master priority	Set the NMT master priority level.	• Low • Mid • High (Default: Medium)
Priority time slot	Set a coefficient used to calculate the response time of NMT master negotiation with the priority level used. (Unit: ms) Note that the setting value set for "Priority time slot" must be greater than the setting value of "CANopen device time slot" × 127.	0 to 65535 (Default: 1500)
CANopen device time slot	Set a coefficient used to calculate the response time of NMT master negotiation with the node ID used. (Unit: ms)	0 to 65535 (Default: 10)
Multiple NMT master detect cycle time	Set the interval for sending protocol messages for NMT master negotiation. (Unit: ms)	0 to 65535 (Default: 4000)

■NMT slave settings

Set detailed parameters for NMT slaves.

Item	Description	Setting range
Guard time	Set the guard time of node guarding for CANopen nodes. (Unit: ms) The value 0 disables node guarding.	0 to 65535 (Default: 0)
Life time factor	Set this object to calculate the node life time for node guarding. The value 0 disables node guarding.	0 to 255 (Default: 0)

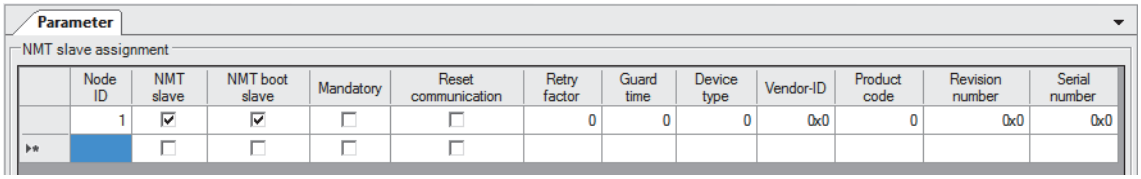
■Operation button

Item	Description
[Restore default values] button	Restores the default NMT master/slave settings.

"NMT slave assignment" window

This window is used for setting the NMT slaves to be assigned to the NMT master on a list basis.

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [NMT settings] ⇨ [NMT master / slave] ⇨ [NMT slave assignment]



Item	Description	Setting range
Node ID	Set the node ID to be assigned to the NMT slave.	1 to 127 (Default: Blank)
NMT slave	• Not selected: Remote node is NMT master or NMT slave that is not assigned. • Selected: Remote node is NMT slave and assigned to this NMT master.	• Not selected • Selected (Default: Not selected)
NMT boot slave	Specify whether the NMT master executes configuration manager or remote node start when starting up the NMT slaves. • Not selected: Not executable • Selected: Executable	• Not selected • Selected (Default: Not selected)
Mandatory	Set this item to indicate the existence of the CANopen node before starting up the network. • Not selected: Not mandatory • Selected: Mandatory	• Not selected • Selected (Default: Not selected)
Reset communication	Set the execution condition for communication reset for the CANopen node. • Not selected: No communication reset condition. Always executable • Selected: Not executable only when the CANopen node is in Operational	• Not selected • Selected (Default: Not selected)
Retry factor	Set the number of resends by the NMT master in case a node guarding event occurs. The value 0 disables node guarding.	0 to 255 (Default: 0)
Guard time	Set the guard time of node guarding for CANopen nodes. (Unit: ms) The value 0 disables node guarding.	0 to 65535 (Default: 0)
Device type*1	Set the device type that has been checked by identification information of the target node. (☞ Page 99 Checking identification information) This setting prohibits the NMT master from starting unregistered nodes. (☞ Page 99 Checking NMT slave verification discrepancy) The device type identification can be disabled by setting 0.	0 to 4294967295 (Default: 0)
Vendor-ID*1	Set the vendor ID that has been checked by identification information of the target node. (☞ Page 99 Checking identification information) This setting prohibits the NMT master from starting unregistered nodes. (☞ Page 99 Checking NMT slave verification discrepancy) The vendor ID identification can be disabled by setting 0x0.	0x0000 to 0xFFFFFFFF (Default: 0x0)
Product code*1	Set the product code that has been checked by identification information of the target node. (☞ Page 99 Checking identification information) This setting prohibits the NMT master from starting unregistered nodes. (☞ Page 99 Checking NMT slave verification discrepancy) The product code identification can be disabled by setting 0.	0 to 4294967295 (Default: 0)
Revision number*1*2	Set the revision number that has been checked by identification information of the target node. (☞ Page 99 Checking identification information) This setting prohibits the NMT master from starting unregistered nodes. (☞ Page 99 Checking NMT slave verification discrepancy) The revision number identification can be disabled by setting 0x0.	0x0000 to 0xFFFFFFFF (Default: 0x0)
Serial number*1*3	Set the serial number that has been checked by identification information of the target node. (☞ Page 99 Checking identification information) This setting prohibits the NMT master from starting unregistered nodes. (☞ Page 99 Checking NMT slave verification discrepancy) The serial number identification can be disabled by setting 0x0.	0x0000 to 0xFFFFFFFF (Default: 0x0)

*1 The settings are enabled by setting "Selected" in "NMT boot slave".

*2 If the revision number verification is enabled, a verification error may occur when the target node is replaced or updated.

*3 If the serial number verification is enabled, a verification error may occur when the target node is replaced.

Point

- An NMT slave can be added in the list by setting a node ID in an empty row. To set a node ID in an empty row, ensure that the NMT master in the "NMT master / slave" window is selected.
- To delete an NMT slave from the list, select the left end of the target row, and press the Delete button.
- A maximum of 126 NMT slave assignments can be set.
- To delete all PDOs from the PDO list, select the upper left of the list, and press the Delete button.

■Checking identification information

The identification information of the target node can be checked as follows.

- Manual for the target device
- Object dictionary
- CANopen Configuration Tool

When checking data from the object dictionary, read the following information.

Item	Index	Sub-index
Device type	1000H	00H
Vendor-ID	1018H	01H
Product code (This code is not defined, which depends on devices.)	1018H	02H
Revision number (This code is not defined, which depends on devices.)	1018H	03H
Serial number (This code is not defined, which depends on devices.)	1018H	04H

When checking the data from CANopen Configuration Tool, operate the following procedure.

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [NMT settings] ⇨ [NMT master / slave] ⇨ [NMT slave assignment] ⇨ [Slave (Node ID: 1)] (When the node ID is 1) ⇨ [Read from slave] button of "NMT slave identification"

■Checking NMT slave verification discrepancy

The target node is identified by the NMT slave verification when the NMT startup is executed. If the verification has detected even one discrepancy between the data, the target node is not started up.

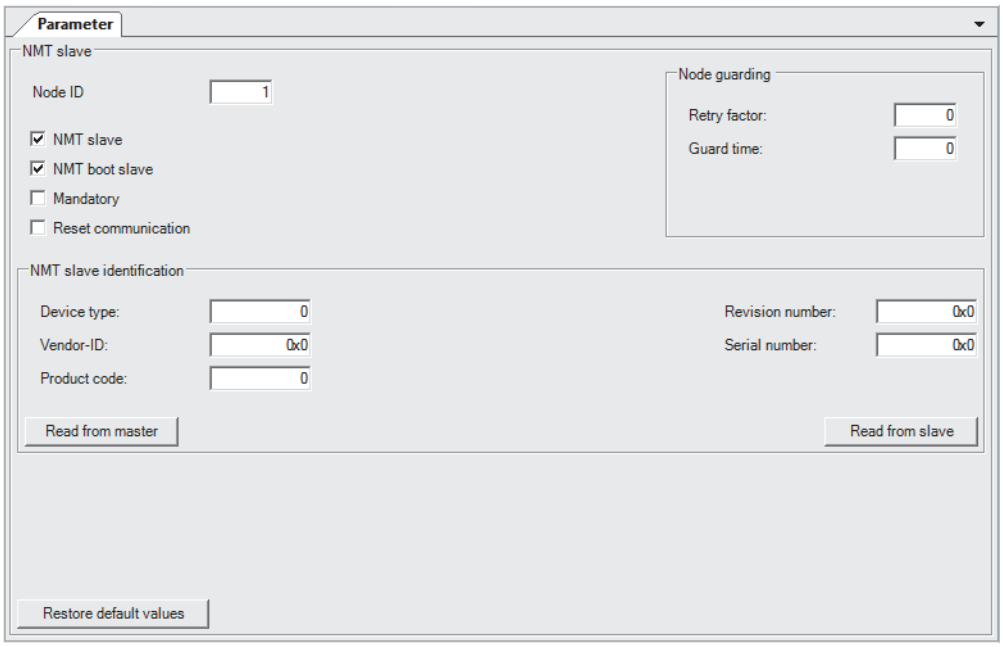
The causes of discrepancy are as follows. (Excluded when the settings are invalid.)

Item	Description
Device type	The registered value is different from the value of actual node.
Vendor-ID	The registered value is different from the value of actual node.
Product code	The registered value is different from the value of actual node.
Revision number	The major version (upper four digits in the registered value) is different from the value of actual node. The minor version (upper four digits in the registered value) is greater than the value of actual node.
Serial number	The registered value is different from the value of actual node.

"NMT slave" window

This window is used for setting the NMT slaves to be assigned to the NMT master on a CANopen node basis.

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [NMT settings] ⇨ [NMT master / slave] ⇨ [NMT slave assignment] ⇨ [Slave (Node ID: 1)] (When the node ID is 1)



Item	Description	Setting range
Node ID	Set the node ID to be assigned to the NMT slave.	1 to 127 (Default: Value set in the "NMT slave assignment" window)
NMT slave	• Not selected: Remote node is NMT master or NMT slave that is not assigned. • Selected: Remote node is NMT slave and assigned to this NMT master.	• Not selected • Selected (Default: Not selected)
NMT boot slave	Specify whether the NMT master executes configuration manager or remote node start when starting up the NMT slaves. • Not selected: Not executable • Selected: Executable	• Not selected • Selected (Default: Not selected)
Mandatory	Set this item to indicate the existence of the CANopen node before starting up the network. • Not selected: Not mandatory • Selected: Mandatory	• Not selected • Selected (Default: Not selected)
Reset communication	Set the execution condition for communication reset for the CANopen node. • Not selected: No communication reset condition. Always executable • Selected: Not executable only when the CANopen node is in Operational	• Not selected • Selected (Default: Not selected)

■Node guarding

Item	Description	Setting range
Retry factor	Set the number of resends by the NMT master in case a node guarding event occurs. The value 0 disables node guarding.	0 to 255 (Default: 0)
Guard time	Set the guard time of node guarding for CANopen nodes. (Unit: ms) The value 0 disables node guarding.	0 to 65535 (Default: 0)

■NMT slave identification

Item	Description	Setting range
Device type*1	Set the device type that has been checked by identification information of the target node. (☞ Page 99 Checking identification information) This setting prohibits the NMT master from starting unregistered nodes. (☞ Page 99 Checking NMT slave verification discrepancy) The device type identification can be disabled by setting 0.	0 to 4294967295 (Default: 0)
Vendor-ID*1	Set the vendor ID that has been checked by identification information of the target node. (☞ Page 99 Checking identification information) This setting prohibits the NMT master from starting unregistered nodes. (☞ Page 99 Checking NMT slave verification discrepancy) The vendor ID identification can be disabled by setting 0x0.	0x0000 to 0xFFFFFFFF (Default: 0x0)
Product code*1	Set the product code that has been checked by identification information of the target node. (☞ Page 99 Checking identification information) This setting prohibits the NMT master from starting unregistered nodes. (☞ Page 99 Checking NMT slave verification discrepancy) The product code identification can be disabled by setting 0.	0 to 4294967295 (Default: 0)
Revision number*1	Set the revision number that has been checked by identification information of the target node. (☞ Page 99 Checking identification information) This setting prohibits the NMT master from starting unregistered nodes. (☞ Page 99 Checking NMT slave verification discrepancy) The revision number identification can be disabled by setting 0x0.	0x0000 to 0xFFFFFFFF (Default: 0x0)
Serial number*1	Set the serial number that has been checked by identification information of the target node. (☞ Page 99 Checking identification information) This setting prohibits the NMT master from starting unregistered nodes. (☞ Page 99 Checking NMT slave verification discrepancy) The serial number identification can be disabled by setting 0x0.	0x0000 to 0xFFFFFFFF (Default: 0x0)
[Read from master] button	Read the NMT slave identification of the target node from the parameter set in the "NMT slave assignment" window of the NMT master. A node specified in "Node ID" of the "CANopen node settings" window will be the target node. (This button reads the identification regardless of operating as an NMT master.)	—
[Read from slave] button	Read the NMT slave identification from the target node. A node specified in "Node ID" of the "NMT slave" window will be the target node. (This button reads the identification regardless of operating as an NMT slave.)	—

*1 The settings are enabled by setting "Selected" in "NMT boot slave".

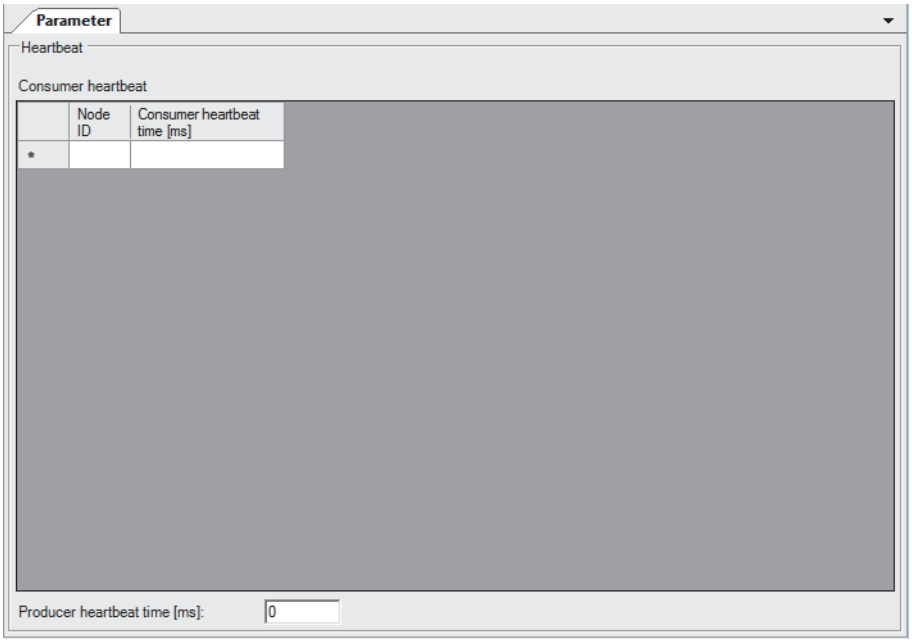
■Operation button

Item	Description
[Restore default values] button	Restores the default NMT slave settings.

"Heartbeat" window

This window is used for setting heartbeat parameters.

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [NMT settings] ⇨ [Heartbeat]



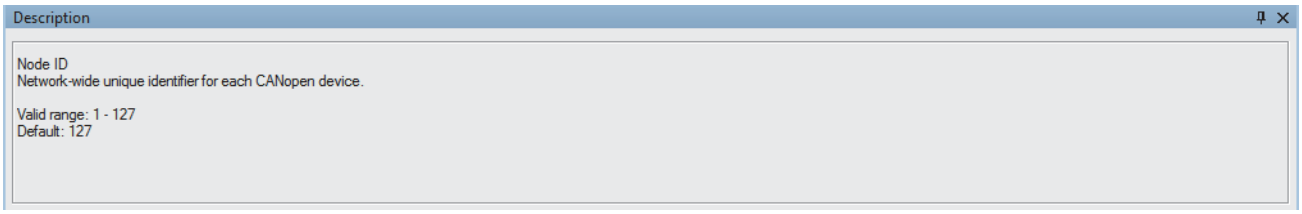
Item	Description	Setting range
Node ID	Set the node ID to use heartbeat.	1 to 127 (Default: Blank)
Consumer heartbeat time	Set the CANopen node to be monitored and the heartbeat time for monitoring that CANopen node. (Unit: ms)	0 to 65535 (Default: 0)
Producer heartbeat time	Set the transmission cycle of heartbeat messages sent from the own node. (Unit: ms)	0 to 65535 (Default: 0)

Point

- A consumer heartbeat parameter can be added in the list by setting a node ID in an empty row.
- To delete a consumer heartbeat parameter from the list, select the left end of the target row, and press the Delete button.
- A maximum of 126 consumer heartbeat times can be set.
- To delete a consumer heartbeat from the list, select the upper left of the list, and press the Delete button.

Description window

The Description window displays information about the items selected in the Parameter window.
Displayed information includes a setting range and default value.



3.2 Setting Procedure

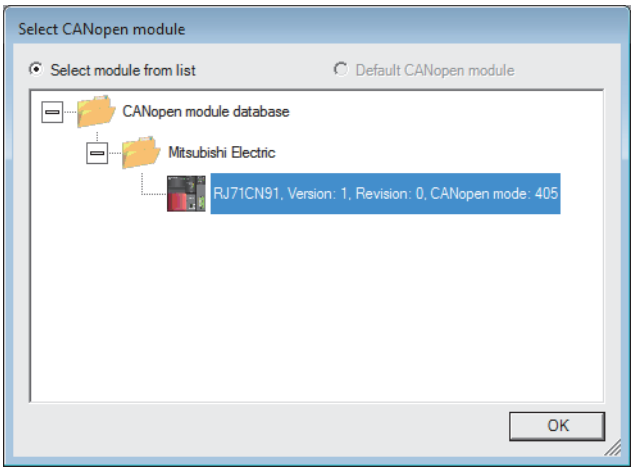
This section describes the procedure for setting CANopen parameters with CANopen Configuration Tool.

1. Start CANopen Configuration Tool and create a project. (☞ Page 104 Creating a new project)
2. To communicate with the CPU module, set the connection destination. (☞ Page 105 Transfer setup)
3. Set parameters for CANopen. (☞ Page 113 Parameter settings)
4. Write the set parameters into the CPU module. (☞ Page 117 Writing the settings)

Creating a new project

Start CANopen Configuration Tool and create a project.

[Project] ⇨ [New]

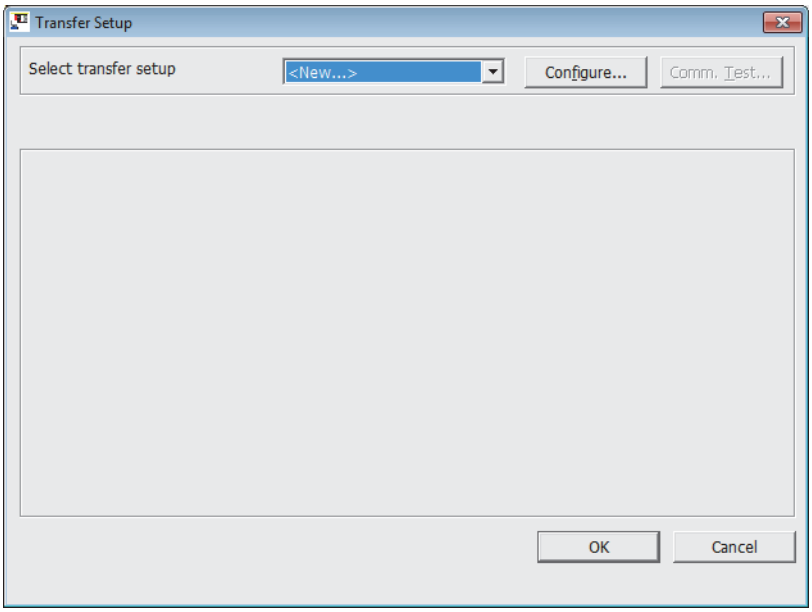


In the above window, select the RJ71CN91 module and click the [OK] button.

Transfer setup

To communicate with the CPU module, set the connection destination.

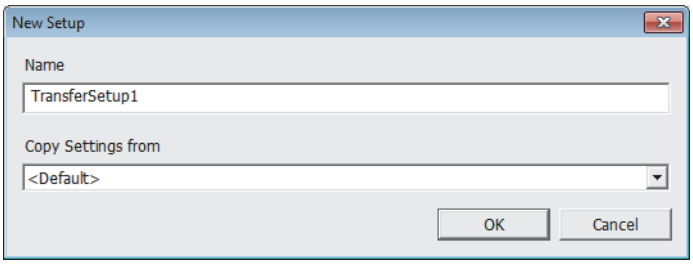
[Online] ⇨ [Transfer Setup]



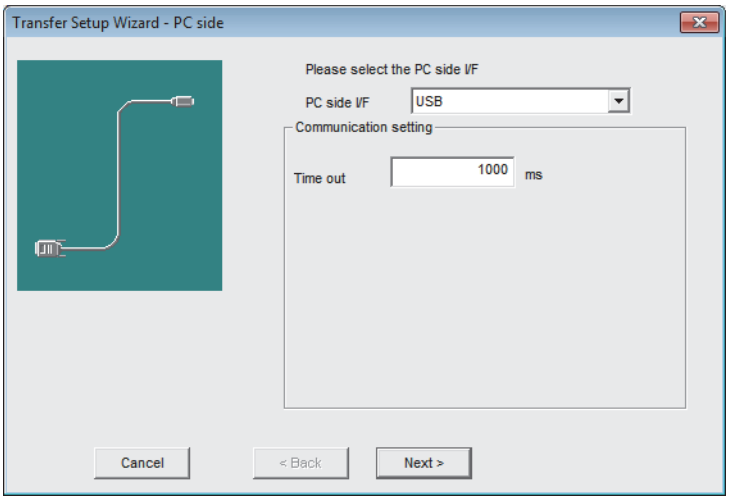
Operating procedure

■When connecting via USB

1. Select "<New>" from "Select transfer setup" in the "Transfer Setup" window, click the [Configure] button and create a setting list. Click the [OK] button to open the setup wizard.

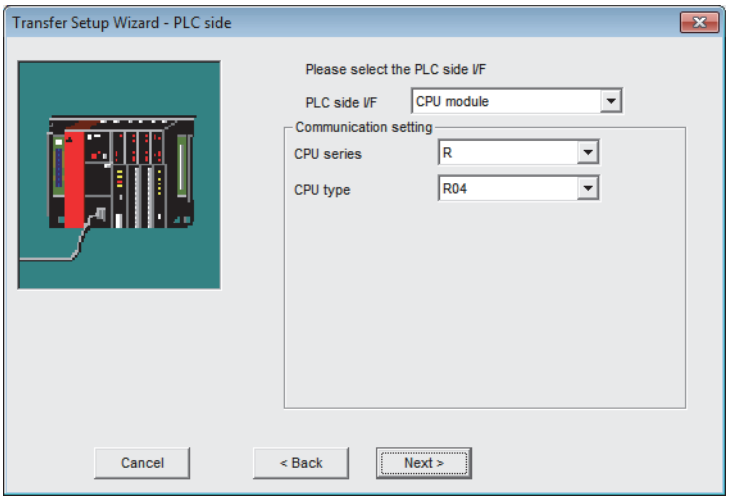


2. Set the interface on the computer side and click the [Next] button.



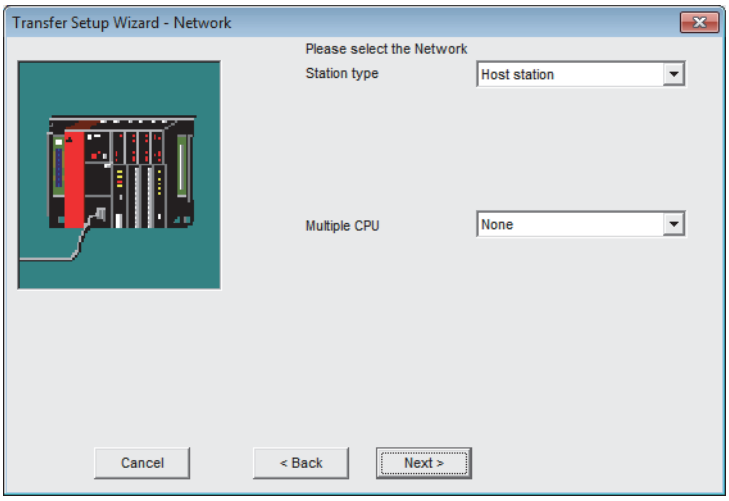
Item	Description	Setting range
PC side I/F	Select an interface on the computer side.	• USB • Ethernet board (Default: USB)
Time out	Set the communication timeout time. (Unit: ms)	1 to 2147483647 (Default: 1000)

3. Set the interface on the PLC side and click the [Next] button.



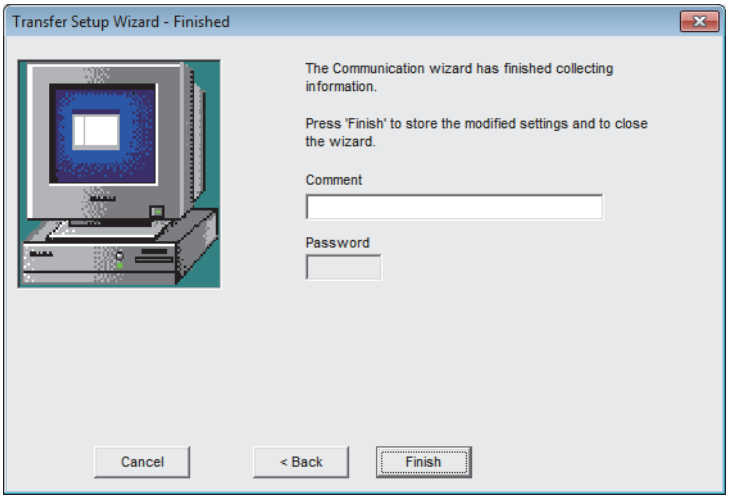
Item	Description	Setting range
PLC side I/F	Select an interface on the PLC side. (Fixed as "CPU module")	—
CPU series	Select a CPU module series.	• R • R Safety
CPU type	Select a CPU module type.	■When "R" is selected from "CPU series" • R04 • R04EN • R08 • R08EN • R08P • R16 • R16EN • R16P • R32 • R32EN • R32P • R120 • R120EN • R120P • R12CCPU-V (Default: R04) ■When "R Safety" is selected from "CPU series" • R08SF • R16SF • R32SF • R120SF (Default: R08SF)

4. Set the communication path and click the [Next] button.



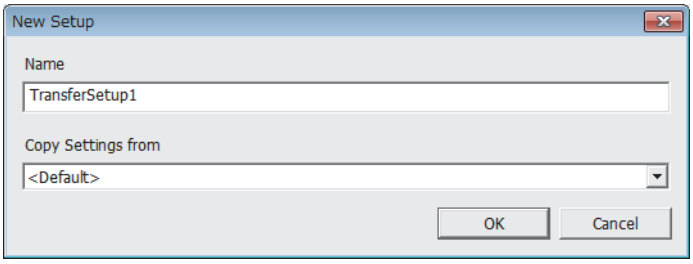
Item	Description	Setting range
Station type	Select the communication path. (Fixed as "Host station")	—
Multiple CPU	Select an access destination for the multiple CPU system.	• None • No.1 • No.2 • No.3 • No.4 (Default: None)

5. Click the [Finish] button to finish the setup wizard.

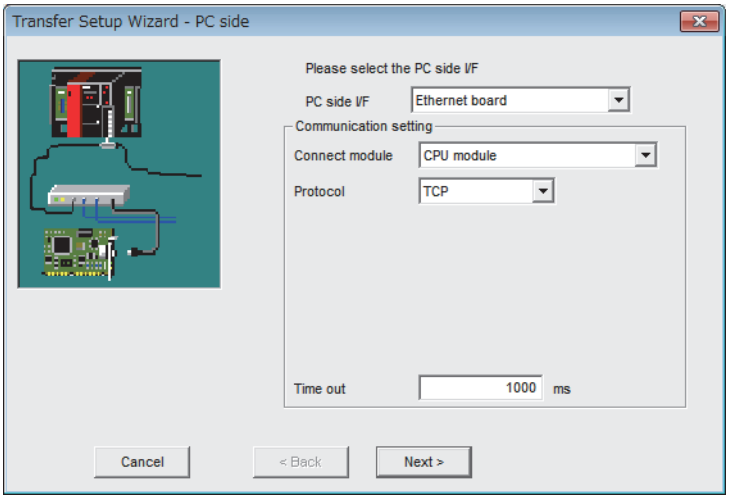


■When connecting via Ethernet

1. Select "<New>" from "Select transfer setup" in the "Transfer Setup" window, click the [Configure] button and create a setting list. Click the [OK] button to open the setup wizard.

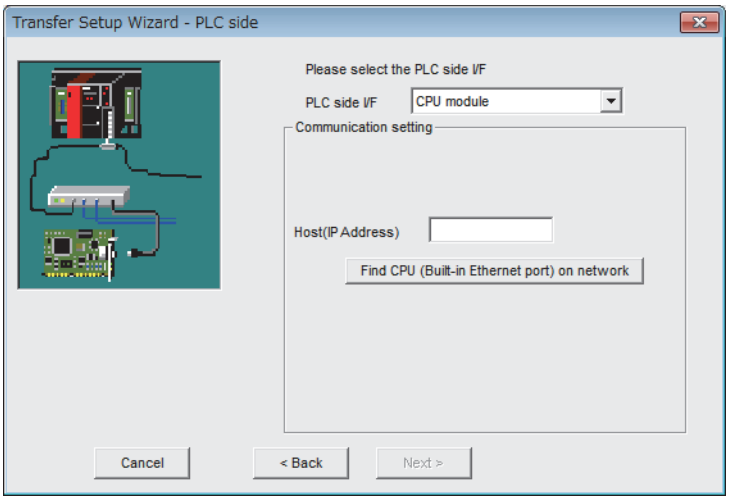


2. Set the interface on the computer side and click the [Next] button.



Item	Description	Setting range
PC side I/F	Select an interface on the computer side.	• USB • Ethernet board (Default: USB)
Connect module	Set the module. (Fixed as "CPU module")	—
Protocol	Set the communication protocol.	• TCP • UDP (Default: TCP)
Time out	Set the communication timeout time. (Unit: ms)	1 to 2147483647 (Default: 1000)

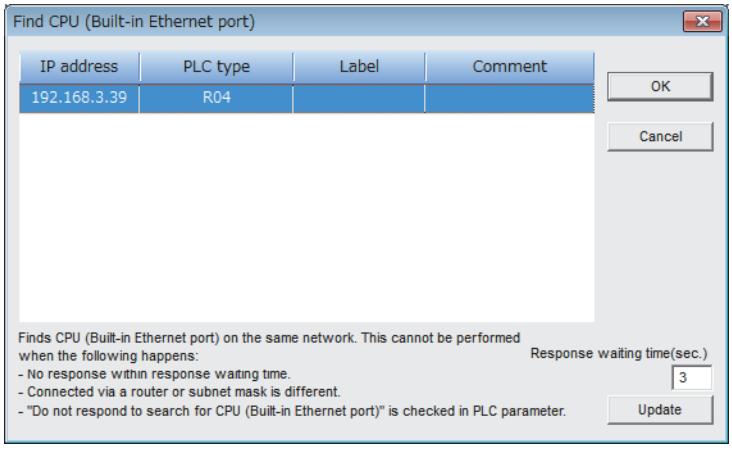
3. Set the interface on the PLC side and click the [Next] button.



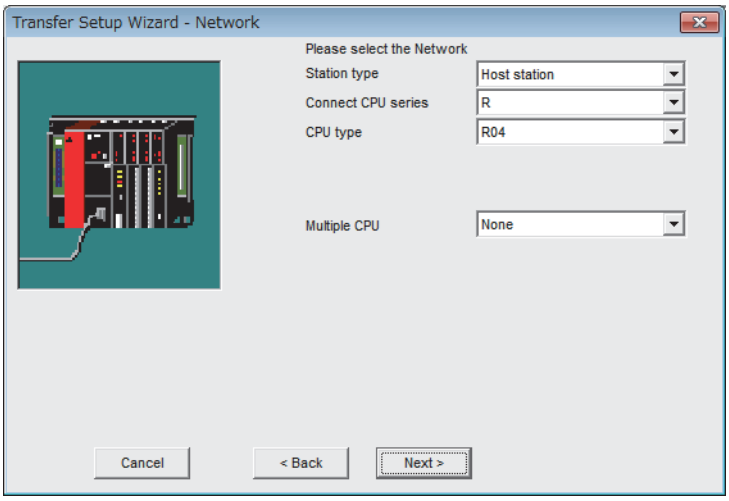
Item	Description	Setting range
PLC side I/F	Select an interface on the PLC side. (Fixed as "CPU module")	—
Host(IP Address)	Set the PLC side IP address.	• Blank • 0.0.0.1 to 223.255.255.254 (Default: Blank)
[Find CPU (Built-in Ethernet port) on network] button	Searches for a CPU with built-in Ethernet port on the same network through connection via a hub.	—

Point

When the [Find CPU (Built-in Ethernet port) on network] button is clicked, a search is made for a CPU with built-in Ethernet port on the same network, and its IP address, CPU type, label, and comment are displayed. Click the [OK] button to input the selected IP address to "Host (IP Address)".

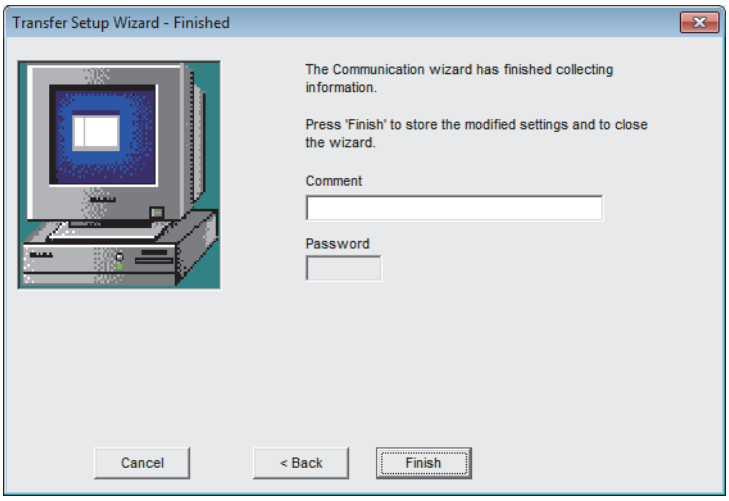


4. Set the communication path and click the [Next] button.

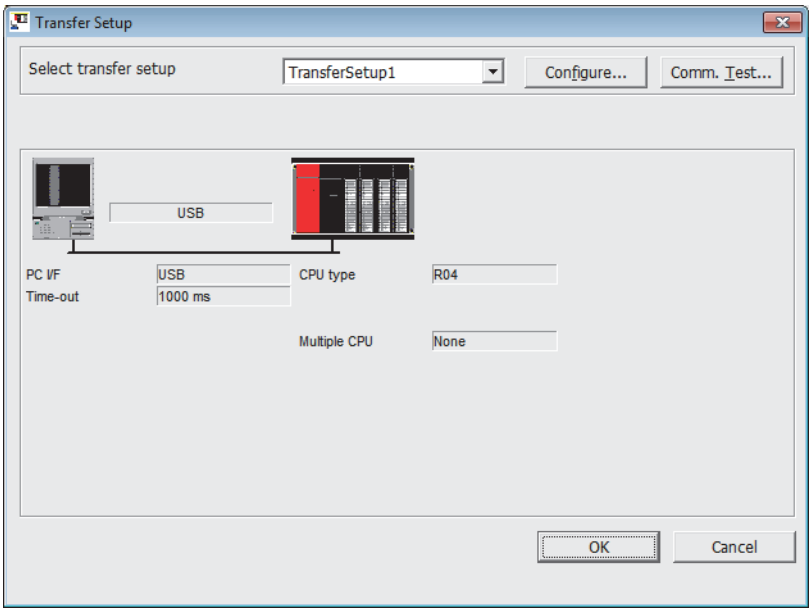


Item	Description	Setting range
Station type	Select the communication path. (Fixed as "Host station")	—
Connect CPU series	Select a CPU module series.	• R • R Safety
CPU type	Select a CPU module type.	■When "R" is selected from "Connect CPU series" • R04 • R04EN • R08 • R08EN • R08P • R16 • R16EN • R16P • R32 • R32EN • R32P • R120 • R120EN • R120P • R12CCPU-V (Default: R04) ■When "R Safety" is selected from "Connect CPU series" • R08SF • R16SF • R32SF • R120SF (Default: R08SF)
Multiple CPU	Select an access destination for the multiple CPU system.	• None • No.1 • No.2 • No.3 • No.4 (Default: None)

5. Click the [Finish] button to finish the setup wizard.

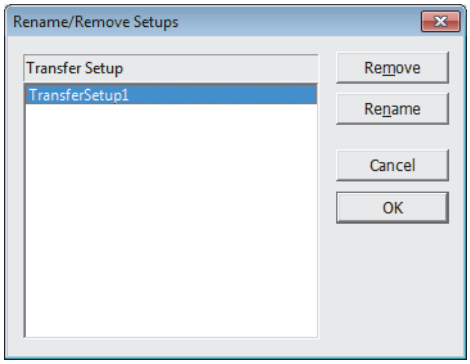


In "Select transfer setup", select one name from the setup list and click the [OK] button.



Point

- Selecting "<Rename/Remove>" in "Select transfer setup" opens the "Rename/Remove Setups" window for changing and deleting the names in the setup list.



- Click the [Comm. Test] button to conduct a communication test.

Parameter settings

Set parameters for CANopen.

Operating procedure

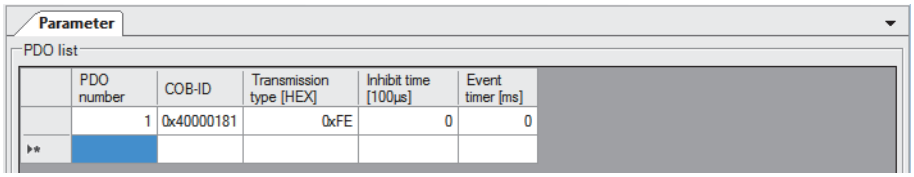
1. In the "CANopen Configuration" window, set items for "Module basic settings". (☞ Page 84 Module basic settings)

[Navigator] ⇨ [CANopen Configuration]

2. In the "PDO list" window, add an entry to the TPDO and RPDO lists. An entry can be added in each list by setting a value for "PDO number". (☞ Page 88 PDO list window)

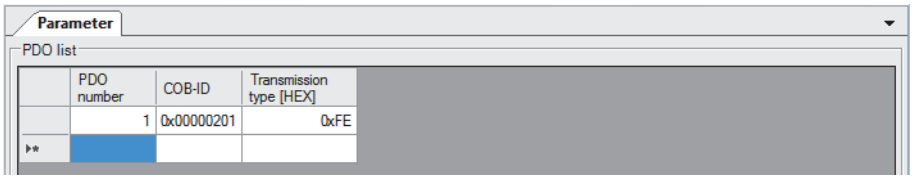
• TPDO

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [Transmit PDO]



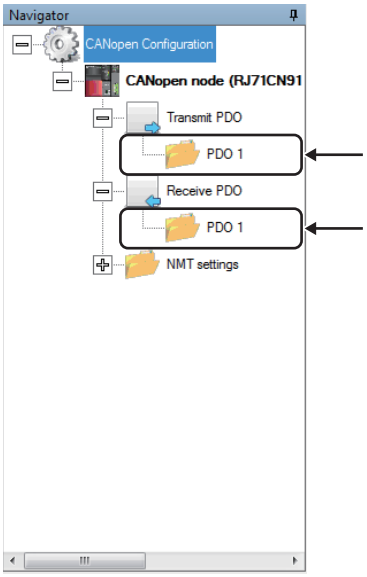
• RPDO

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [Receive PDO]



Point

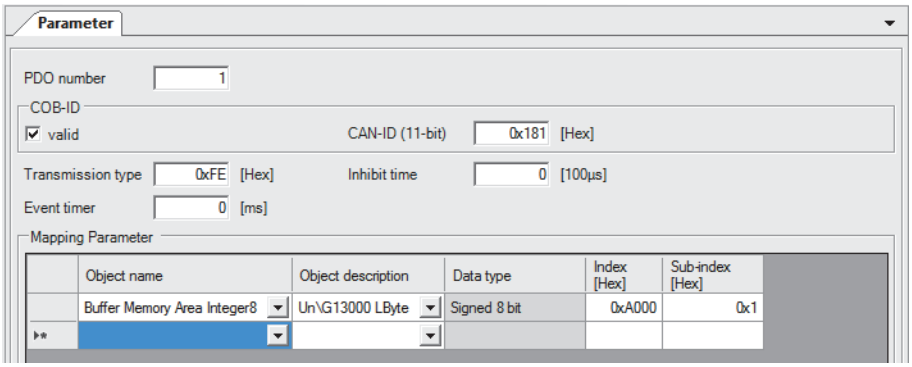
When an entry is added to the TPDO and RPDO lists, its corresponding item is added in the Navigator window.



3. Set parameters for TPDO and RPDO.

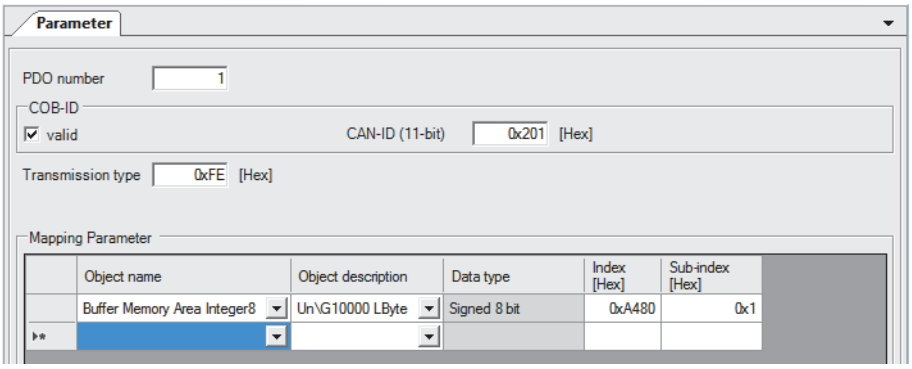
• TPDO (☞ Page 91 TPDO details window)

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [Transmit PDO] ⇨ [PDO 1] (When the PDO number is 1)



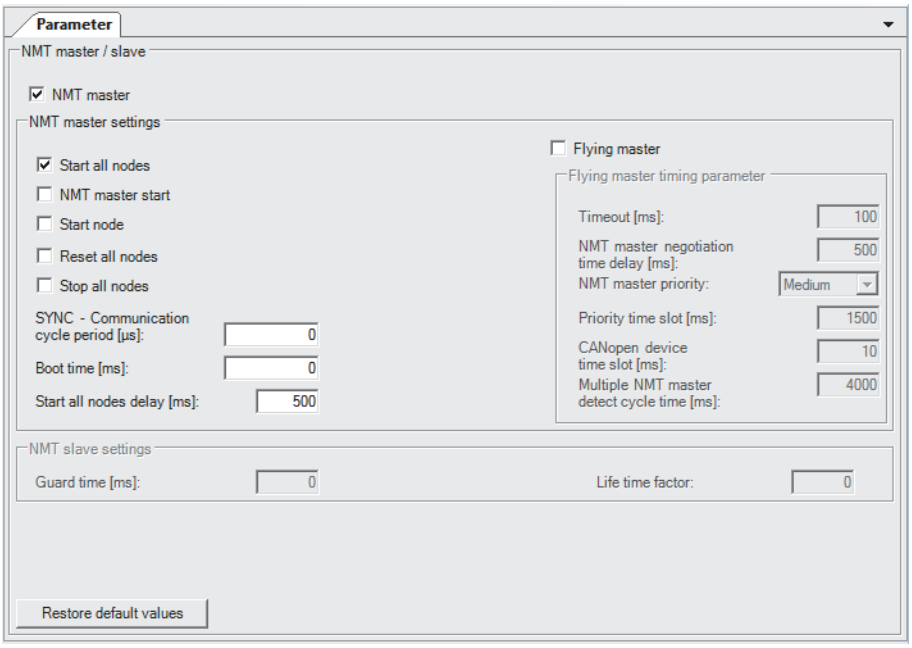
• RPDO (☞ Page 93 RPDO details window)

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [Receive PDO] ⇨ [PDO 1] (When the PDO number is 1)



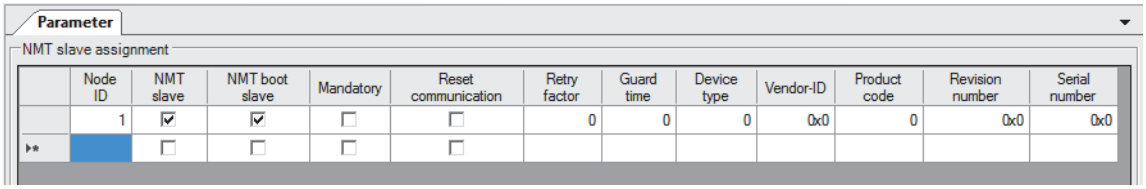
4. In the "NMT master / slave" window, set parameters for the NMT master and NMT slaves. (☞ Page 96 "NMT master / slave" window)

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [NMT settings] ⇨ [NMT master / slave]



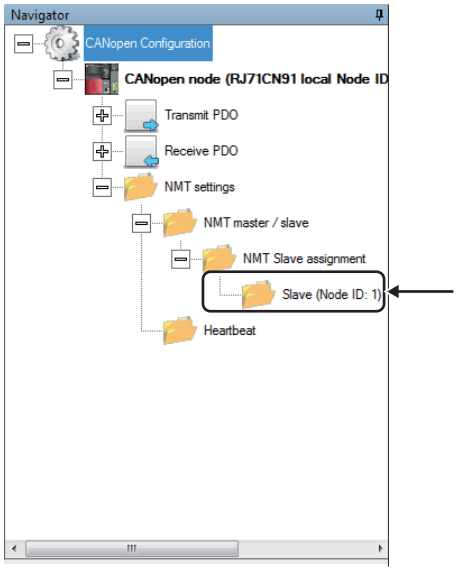
5. In the "NMT slave assignment" window, add an NMT slave to be assigned to the NMT master. An entry can be added in the list by setting a value for "Node ID". (☞ Page 98 "NMT slave assignment" window)

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [NMT settings] ⇨ [NMT master / slave] ⇨ [NMT slave assignment]



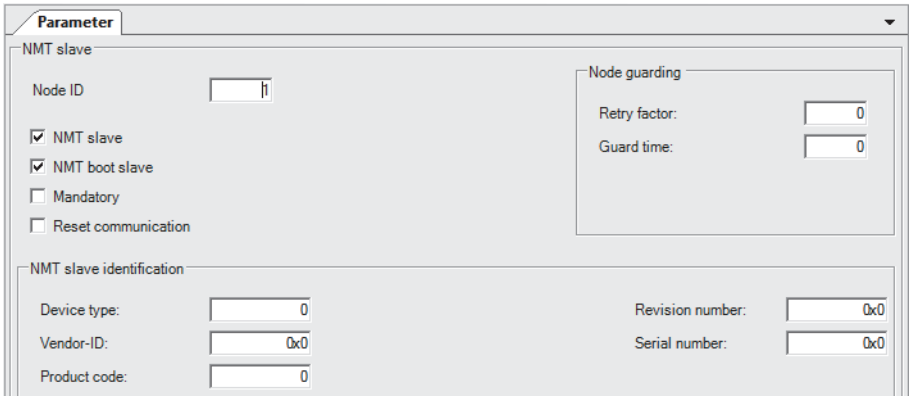
Point

When an entry is added to the NMT slave list, its corresponding item is added in the Navigator window.



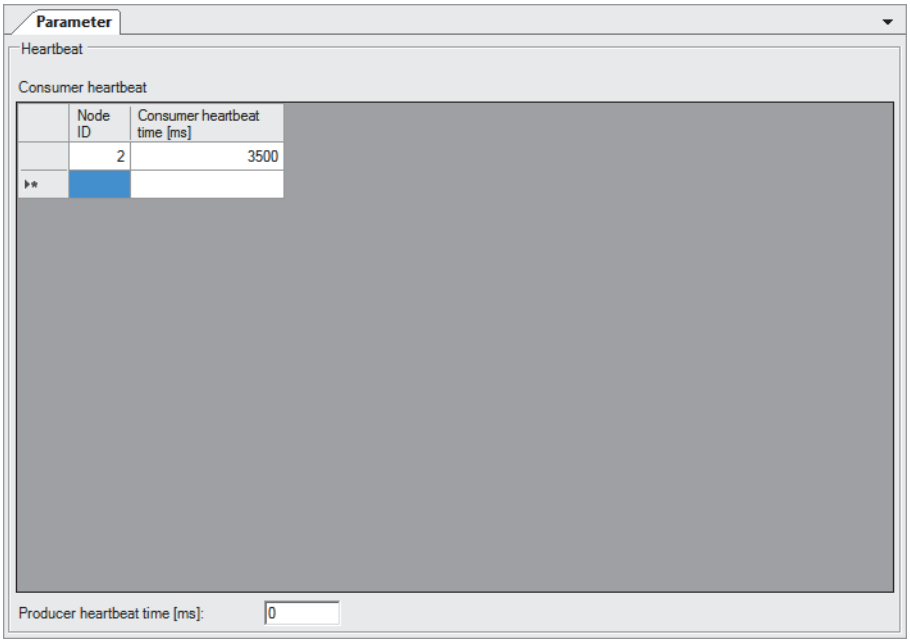
6. In the "NMT slave" window, set parameters for the NMT slave. (☞ Page 100 "NMT slave" window)

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [NMT settings] ⇨ [NMT master / slave] ⇨ [NMT slave assignment] ⇨ [Slave (Node ID: 1)] (When the node ID is 1)



7. In the "Heartbeat" window, set parameters for the heartbeat. An entry can be added in the list by setting a value for "Node ID". (☞ Page 102 "Heartbeat" window)

[Navigator] ⇨ [CANopen Configuration] ⇨ Node name ⇨ [NMT settings] ⇨ [Heartbeat]



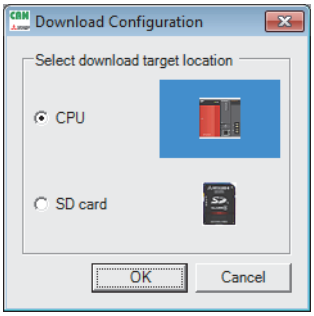
Writing the settings

Write the set parameters into the CPU module.

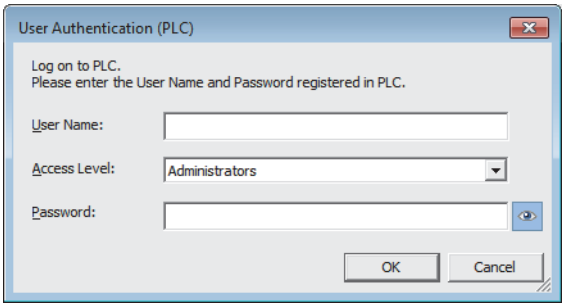
Operating procedure

1. In the "Download Configuration" window, select the download target destination.

[Online] ⇨ [Download Configuration]



2. When the Safety CPU is connected to the CPU module, the "User Authentication (PLC)" window appears. Input user information that is registered in the Safety CPU. (Re-entering information is not required from the second time onward, however, the information needs to be entered again if a certain time has passed or the Safety CPU is reset.)



Point

"Administrators" or "Developers" can be selected from "Access Level". For details on the access level and its authentication, refer to the following.

GX Works3 Operating Manual

3. Click the [OK] button to complete the writing process.

Point

- When the CPU module is in the STOP state, parameters can be written.
- When the writing parameters has failed, an error window appears and the error code is displayed. For details on the error codes, refer to the following.

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

3.3 Functions

This section describes the functions of CANopen Configuration Tool.

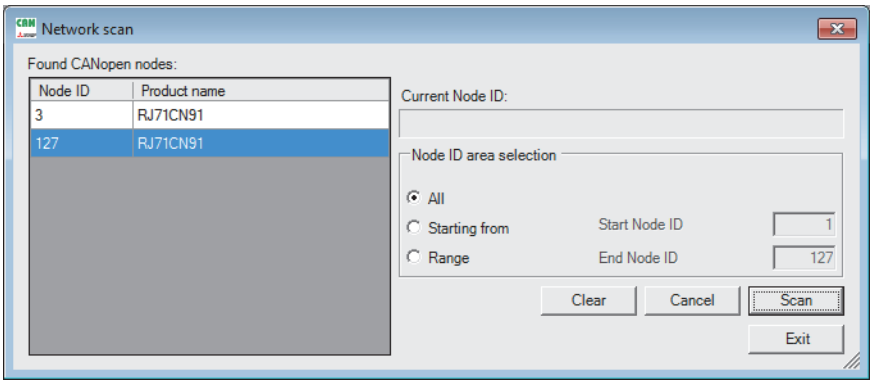
Network scan

CANopen Configuration Tool can scan for all CANopen nodes connected to CANopen.
When a CANopen node is found, information for "Node ID" and "Product name" is displayed in the list on the left side of the "Network scan" window.
The scan time can be shortened by limiting the node ID scan range in "Node ID area selection" and shortening the timeout time of "SDO timeout" in "Basic Setting".

Operation method

The "Network scan" window can be displayed by the following operation.

[Diagnostics] ⇨ [Network scan]



Item	Description
Found CANopen nodes	Click the [Scan] button to display the scan results. Information for "Node ID" and "Product name" is displayed in the list.
Current Node ID	Displays the status of scan progress.
Node ID area selection	For a node ID scan, a choice can be made from "All", "Starting from", and "Range". When "Starting from" is selected, specify any number for "Start Node ID". (1 to 127) When "Range" is selected, specify any number for "Start Node ID" and "End Node ID". (1 to 127)

SDO send/receive

In SDO send/receive, SDO read and SDO write can be executed.

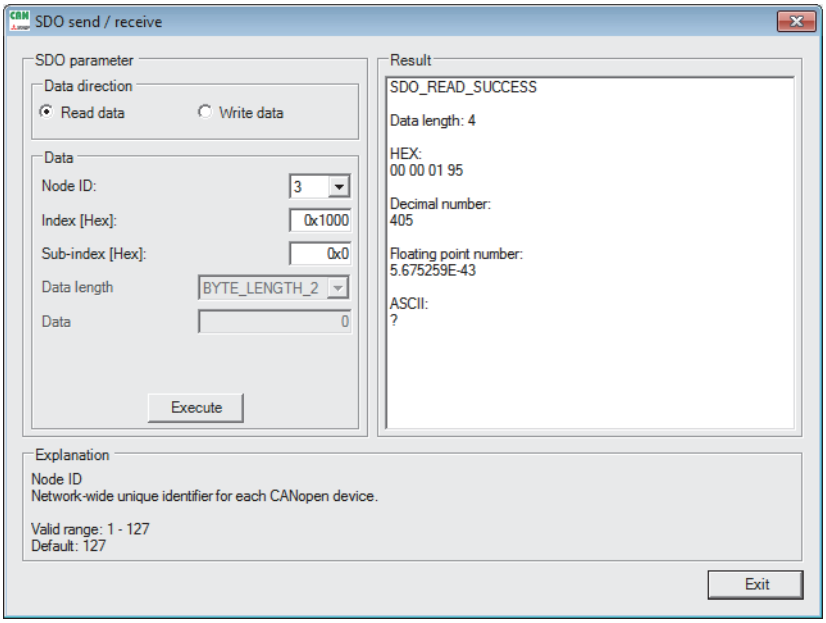
SDO is a function to directly access an object entry in the object dictionary of any CANopen node.

Operation method

The "SDO send / receive" window can be displayed by the following operation.

[Online] ⇨ [SDO send / receive]

Select "Read data" or "Write data", set necessary data, and click the [Execute] button to display the execution result.



Item		Description	Setting range
SDO parameter	Data direction	"Read data" or "Write data" can be selected.	• Read data • Write data (Default: Read data)
Data	Node ID	Set the node ID of the node from which data is read or to which data is written.	1 to 127 (Default: 127)
	Index	Set the index in the object dictionary. For indexes, refer to the following. ☞ Page 191 Object Dictionary	0x0000 to 0xFFFF (Default: 0x0000)
	Sub-index	Set the subindex in the object dictionary. For subindexes, refer to the following. ☞ Page 191 Object Dictionary	0x00 to 0xFF (Default: 0x00)
	Data length	Set the data length of data to transmit.	• STRING • BYTE_LENGTH_1 • BYTE_LENGTH_2 • BYTE_LENGTH_4 • BYTE_LENGTH_6 • BYTE_LENGTH_8 (Default: BYTE_LENGTH_2)
	Data	Set send data. Prepends with "0x" when specifying send data using a hexadecimal number.	— (Default: 0)
Result		Click the [Execute] button to display the SDO send/receive results. When "Read data" is successful, "SDO_READ_SUCCESS" is displayed and the result is displayed in each of the following data types. • Hexadecimal number • Decimal number • Floating point number • ASCII When "Write data" is successful, "SDO_WRITE_SUCCESS" is displayed. If "Read data" or "Write data" fails, the error details are displayed.	—
Explanation		Displays the description of the selected item.	—

Export/import

A project set with CANopen Configuration Tool can be exported as a backup file and imported to another project. This operation allows the user to use the setting details in another project.*1

The extension of the backup file is ".xml".

*1 A project file exported with Version 1.01B cannot be imported with Version 1.00A.

Operating procedure

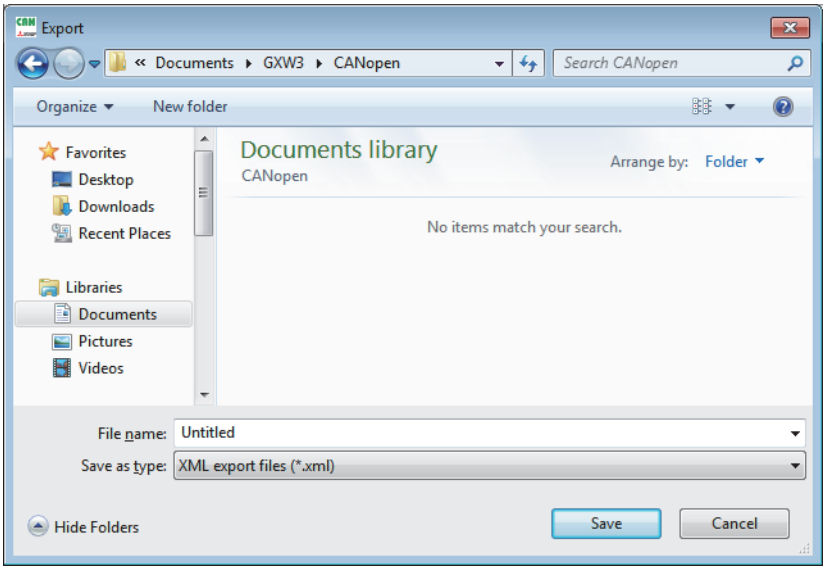
The export and import procedures are as follows.

■Export

1. The "Export" window is displayed by the following operation.

[Project] ⇨ [Export]

2. Set a file name, select the save destination, and click the [Save] button.

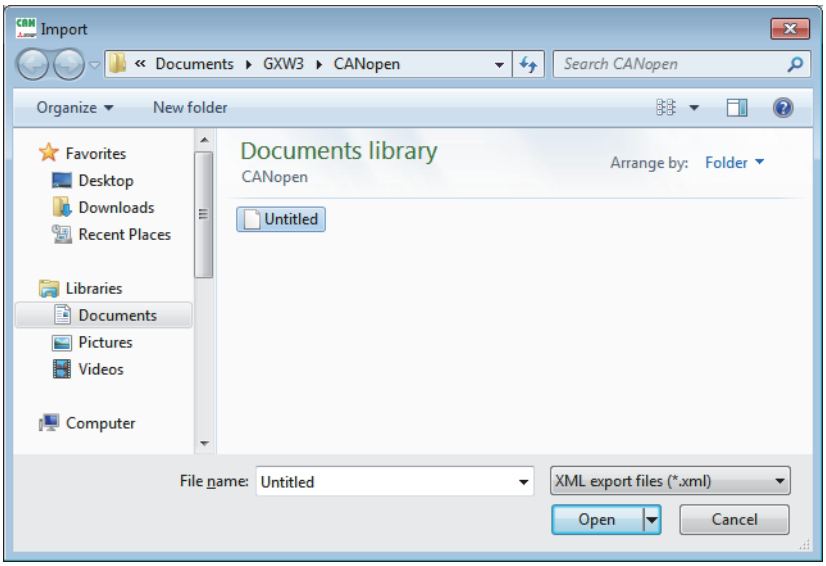


■Import

1. The "Import" window is displayed by the following operation.

[Project] ⇨ [Import]

2. Select a file to be imported and click the [Open] button.



Module status

The status of the connected CANopen node can be checked.

If an error is displayed, resolve the error status, and then click the [Clear error] button to clear the error display.

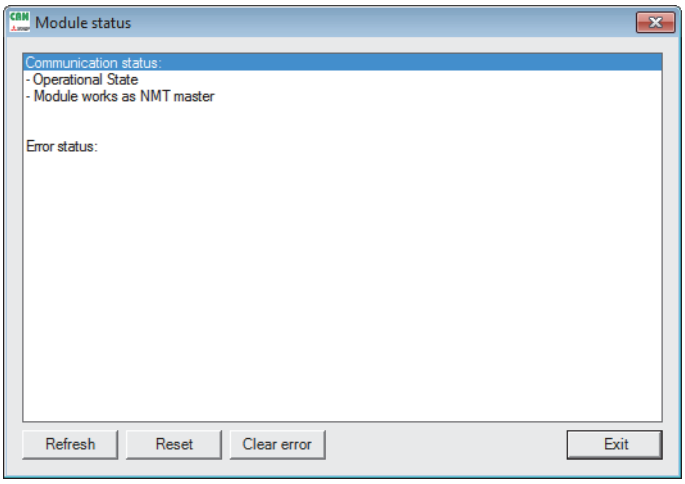
Also, click the [Refresh] button to display the latest status.

Click the [Reset] button to reset the NMT master.

Operation method

The "Module status" window can be displayed by the following operation.

[Diagnostics] ⇨ [Module status]



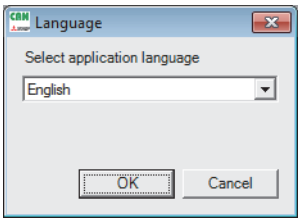
Select language

The display language for CANopen Configuration Tool can be selected. (Default: English)
After selecting the language, restart CANopen Configuration Tool to have the change take effect.

Operation method

The "Language" window can be displayed by the following operation.

[Tools] ⇨ [Language]



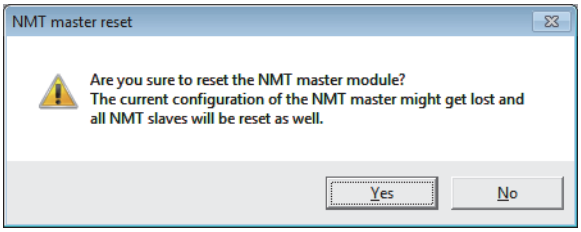
NMT master reset

Resets and restarts the connected NMT master.

Operating procedure

1. Execute NMT master reset using the following operation.

[Online] ⇨ [NMT master reset]

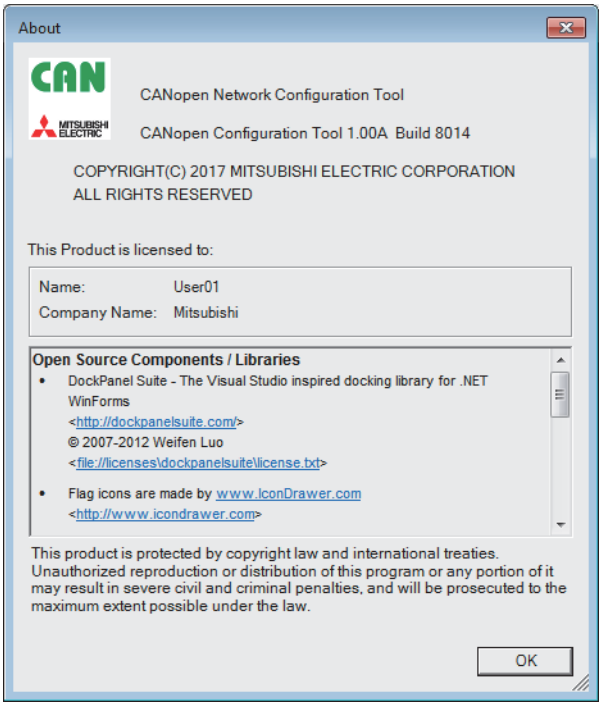


2. Click the [Yes] button to reset the NMT master.

3.4 Checking the Software Version

Check the software version of CANopen Configuration Tool in the following window.

[Help] ⇨ [About]



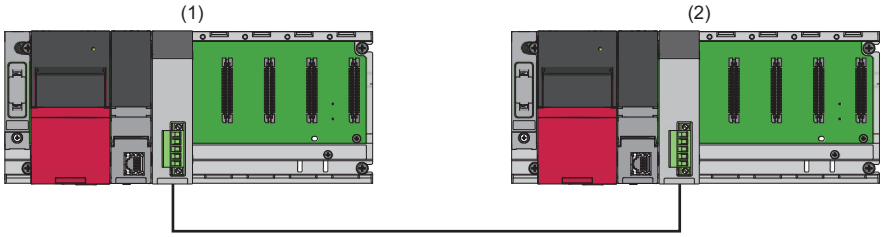
4 PROGRAMMING

This chapter describes programming in the Layer 2 message mode.
For data communication in the CANopen 405 mode, refer to the following.
MELSEC iQ-R CANopen Module User's Manual (Startup)

4.1 Communication Example of Layer 2 Message Mode

This section describes a communication example of message transmission and reception using layer 2 messages.

System configuration



(1) Programmable controller system (Master side)
- Power supply module: R61P
- CPU module: R04CPU
- CAN node: RJ71CN91 (start I/O number: 0000H to 001FH)

(2) Programmable controller system (Slave side)
- Power supply module: R61P
- CPU module: R04CPU
- CAN node: RJ71CN91 (start I/O number: 0000H to 001FH)

Setting details

The following is CAN node setting details.

■Parameter settings

Target node	Item	Description
CAN node (1)	Baud rate	1000kbps
	Data frame type	Standard format (11-bit CAN-ID)
CAN node (2)	Baud rate	1000kbps
	Data frame type	Standard format (11-bit CAN-ID)

■Message settings

CAN-ID	Frame type	Data length	RTR response	Transmit node	Receive node
0501H	Data frame	8 bytes	Disabled	CAN node (1)	CAN node (2)
0581H	Data frame	8 bytes	Disabled	CAN node (2)	CAN node (1)
0601H	Remote frame	—	—	CAN node (2)	CAN node (1)
0601H	Data frame	8 bytes	Enabled	CAN node (1)	CAN node (2)
0681H	Remote frame	—	—	CAN node (1)	CAN node (2)
0681H	Data frame	8 bytes	Enabled	CAN node (2)	CAN node (1)

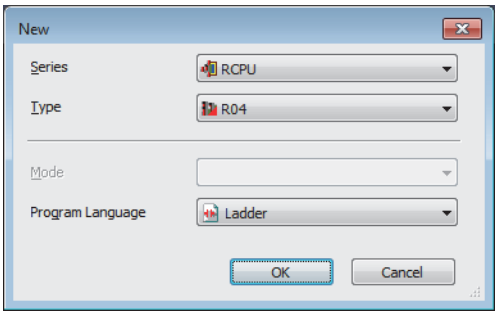
Parameter settings

Connect the engineering tool to the CPU module and set parameters.

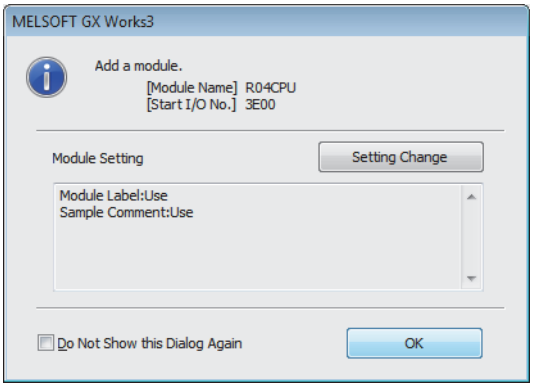
Module parameter setting

1. Set the CPU module as follows.

[Project] ⇨ [New]

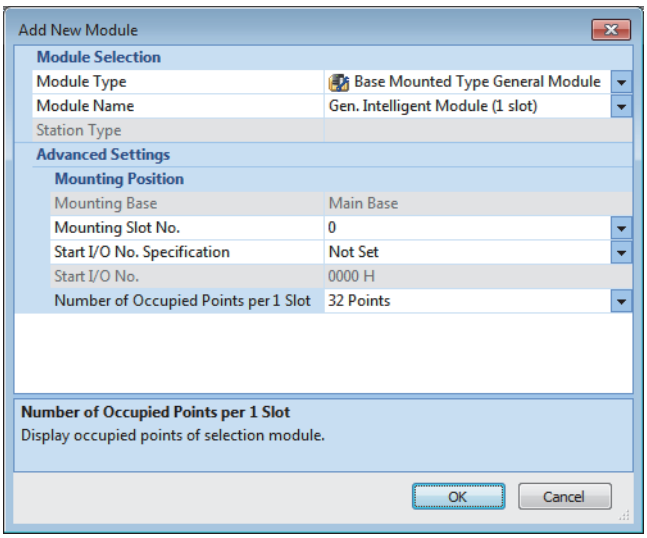


2. Click the [Setting Change] button and set "Use Module Label" to "Yes".



3. Add a general-purpose intelligent module (for single slot). (Set "Number of Occupied Points per 1 Slot" to "32 Points".)

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ Right-click ⇨ [Add New Module]



4. Write the set parameters to the CPU module, and reset the CPU module or power off and on the system.

[Online] ⇨ [Write to PLC]

Point

In the program example, default settings are used for parameters other than the above. For details on parameters, refer to the following.

☞ Page 74 PARAMETER SETTINGS

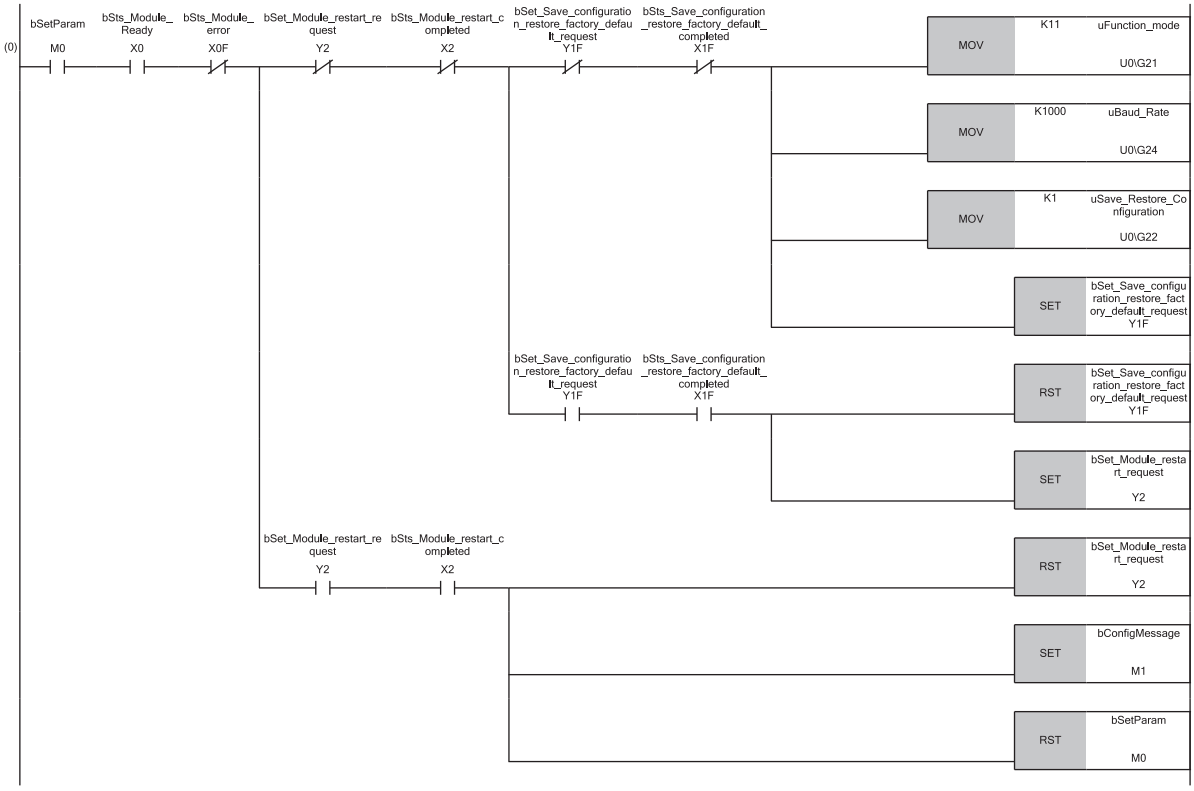
Program example

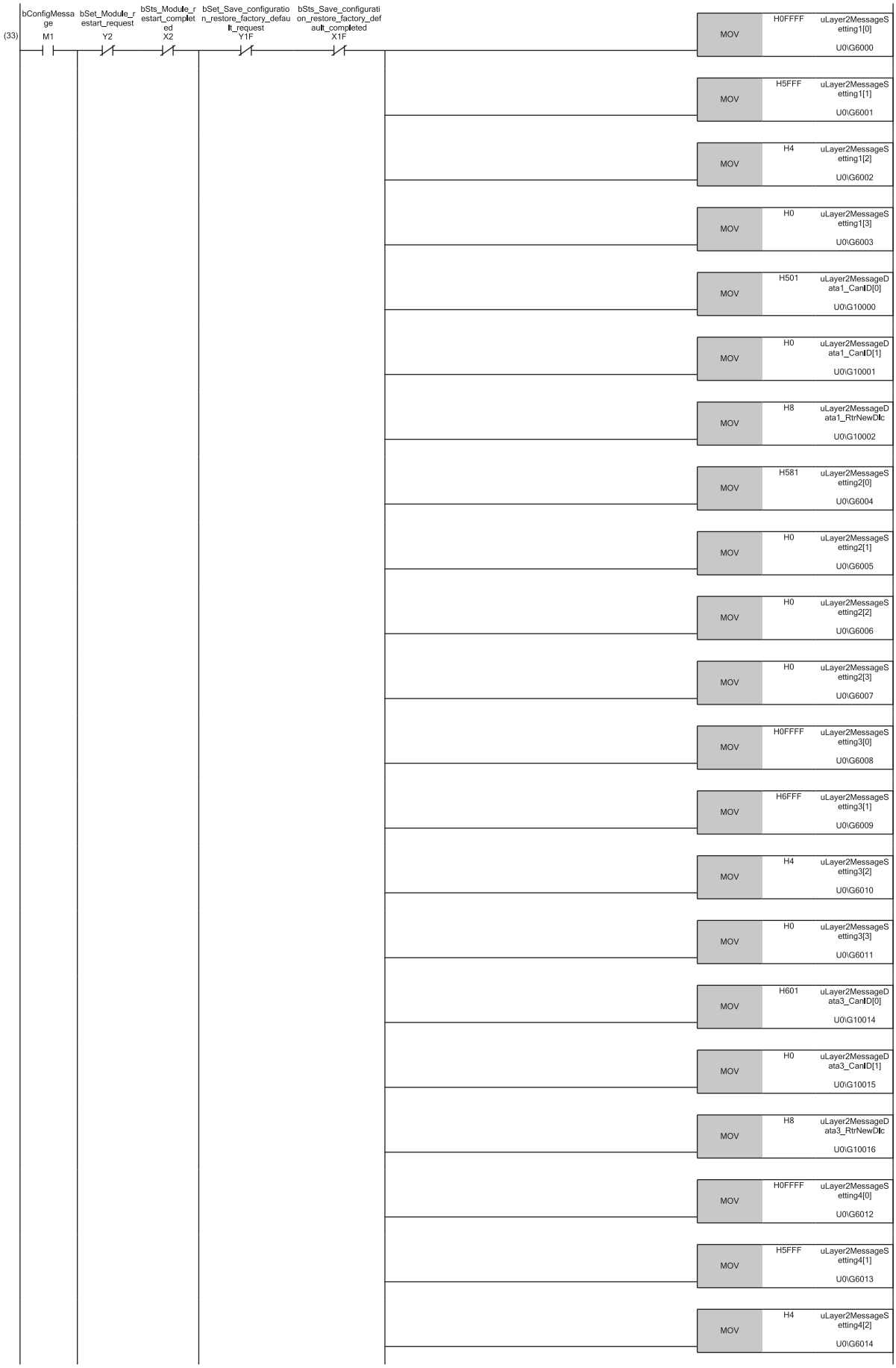
Parameter setting program

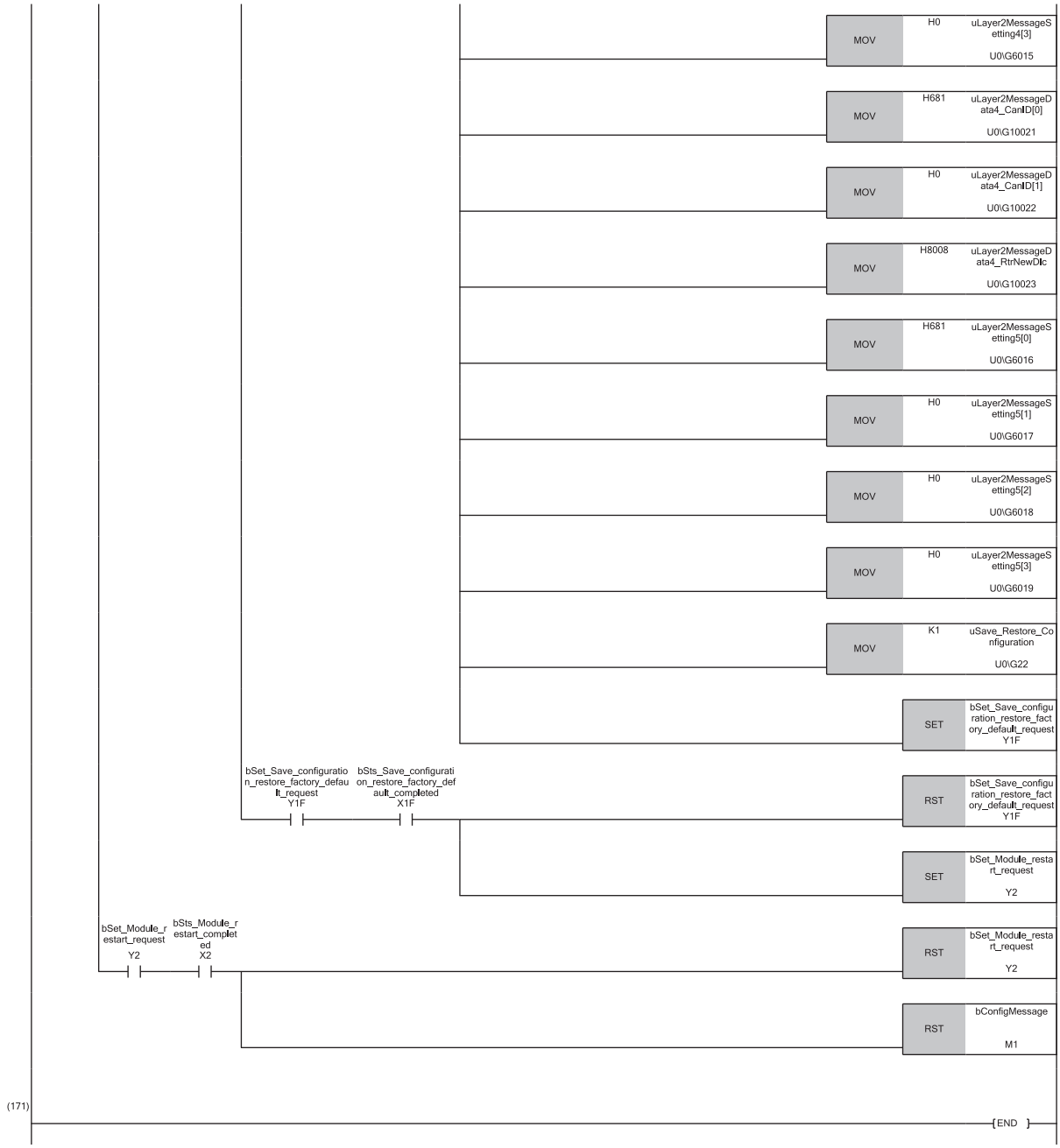
When the RJ71CN91 is used in the Layer 2 message mode, set parameters via the program.

Classification	Label name
Label to be defined	Define the global labels as shown below.

Label Name	Data Type		Class	Assign (Device/Label)
bConfigMessage	Bit	...	VAR_GLOBAL	M1
bSet_Module_restart_request	Bit	...	VAR_GLOBAL	Y2
bSet_Save_configuration_restore_factory_default_request	Bit	...	VAR_GLOBAL	Y1F
bSetParam	Bit	...	VAR_GLOBAL	M0
bSts_Module_error	Bit	...	VAR_GLOBAL	X0F
bSts_Module_Ready	Bit	...	VAR_GLOBAL	X0
bSts_Module_restart_completed	Bit	...	VAR_GLOBAL	X2
bSts_Save_configuration_restore_factory_default_completed	Bit	...	VAR_GLOBAL	X1F
uBaud_Rate	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	U0\G24
uFunction_mode	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	U0\G21
uLayer2MessageData1_CanID	Word [Unsigned]/Bit String [16-bit](0..1)	...	VAR_GLOBAL	U0\G10000
uLayer2MessageData1_RtrNewDlc	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	U0\G10002
uLayer2MessageData2_CanID	Word [Unsigned]/Bit String [16-bit](0..1)	...	VAR_GLOBAL	U0\G10007
uLayer2MessageData2_RtrNewDlc	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	U0\G10009
uLayer2MessageData3_CanID	Word [Unsigned]/Bit String [16-bit](0..1)	...	VAR_GLOBAL	U0\G10014
uLayer2MessageData3_RtrNewDlc	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	U0\G10016
uLayer2MessageData4_CanID	Word [Unsigned]/Bit String [16-bit](0..1)	...	VAR_GLOBAL	U0\G10021
uLayer2MessageData4_RtrNewDlc	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	U0\G10023
uLayer2MessageData5_CanID	Word [Unsigned]/Bit String [16-bit](0..1)	...	VAR_GLOBAL	U0\G10028
uLayer2MessageData5_RtrNewDlc	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	U0\G10030
uLayer2MessageSetting1	Word [Unsigned]/Bit String [16-bit](0..3)	...	VAR_GLOBAL	U0\G6000
uLayer2MessageSetting2	Word [Unsigned]/Bit String [16-bit](0..3)	...	VAR_GLOBAL	U0\G6004
uLayer2MessageSetting3	Word [Unsigned]/Bit String [16-bit](0..3)	...	VAR_GLOBAL	U0\G6008
uLayer2MessageSetting4	Word [Unsigned]/Bit String [16-bit](0..3)	...	VAR_GLOBAL	U0\G6012
uLayer2MessageSetting5	Word [Unsigned]/Bit String [16-bit](0..3)	...	VAR_GLOBAL	U0\G6016
uSave_Restore_Configuration	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	U0\G22




(33)
bConfigMessage
M1
bSet_Module_restart_request
Y2
bSts_Module_restart_completed
X2
bSet_Save_configuration_restore_factory_default_request
Y1F
bSts_Save_configuration_restore_factory_default_completed
X1F
MOV H0FFFF uLayer2MessageSetting1[0] U0\G6000
MOV H5FFF uLayer2MessageSetting1[1] U0\G6001
MOV H4 uLayer2MessageSetting1[2] U0\G6002
MOV H0 uLayer2MessageSetting1[3] U0\G6003
MOV H501 uLayer2MessageData1_CanID[0] U0\G10000
MOV H0 uLayer2MessageData1_CanID[1] U0\G10001
MOV H8 uLayer2MessageData1_RtrNewDlc U0\G10002
MOV H581 uLayer2MessageSetting2[0] U0\G6004
MOV H0 uLayer2MessageSetting2[1] U0\G6005
MOV H0 uLayer2MessageSetting2[2] U0\G6006
MOV H0 uLayer2MessageSetting2[3] U0\G6007
MOV H0FFFF uLayer2MessageSetting3[0] U0\G6008
MOV H6FFF uLayer2MessageSetting3[1] U0\G6009
MOV H4 uLayer2MessageSetting3[2] U0\G6010
MOV H0 uLayer2MessageSetting3[3] U0\G6011
MOV H601 uLayer2MessageData3_CanID[0] U0\G10014
MOV H0 uLayer2MessageData3_CanID[1] U0\G10015
MOV H8 uLayer2MessageData3_RtrNewDlc U0\G10016
MOV H0FFFF uLayer2MessageSetting4[0] U0\G6012
MOV H5FFF uLayer2MessageSetting4[1] U0\G6013
MOV H4 uLayer2MessageSetting4[2] U0\G6014



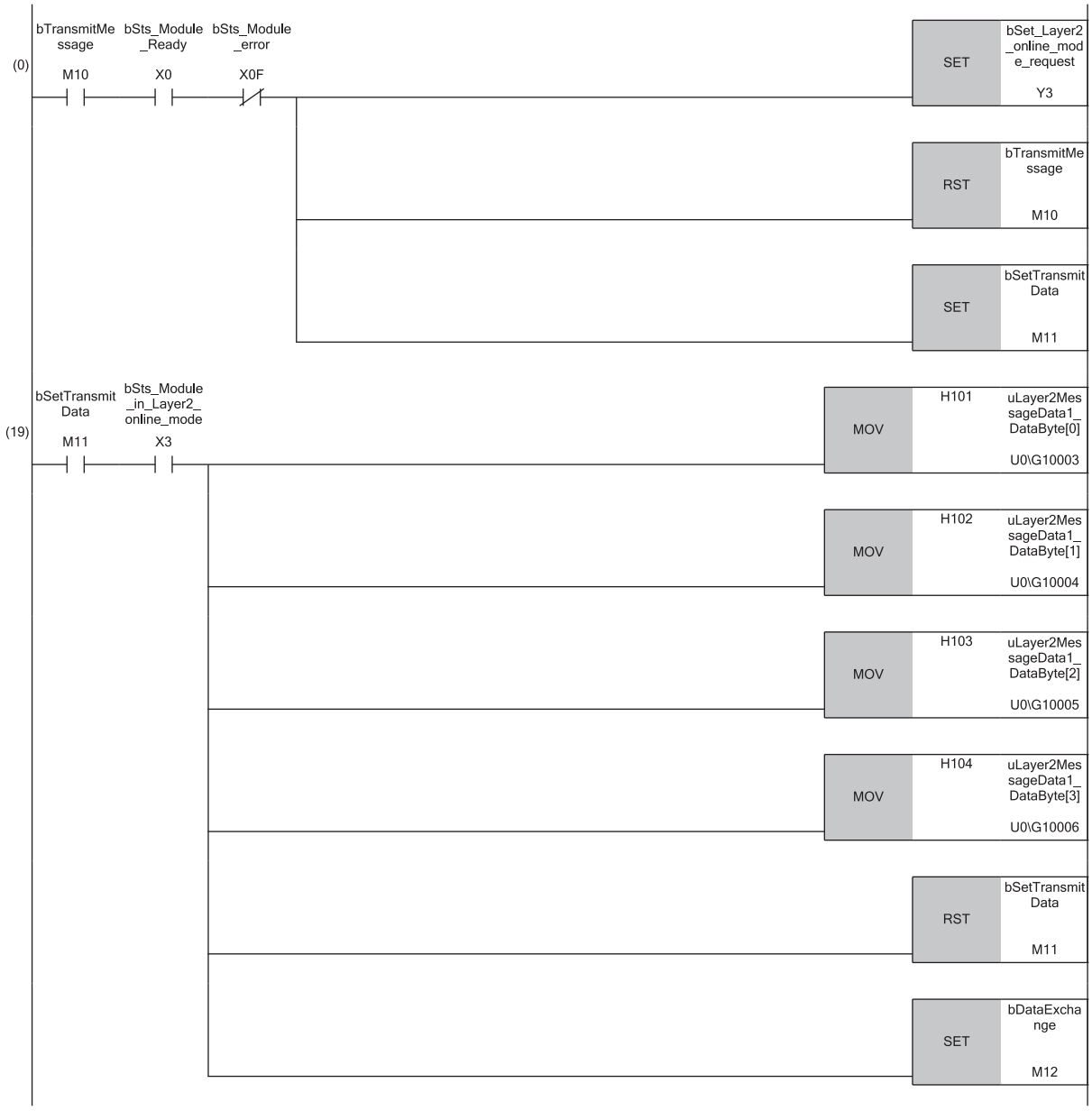
(0) Execute CAN node parameter setting.
(33) Execute CAN node message setting.

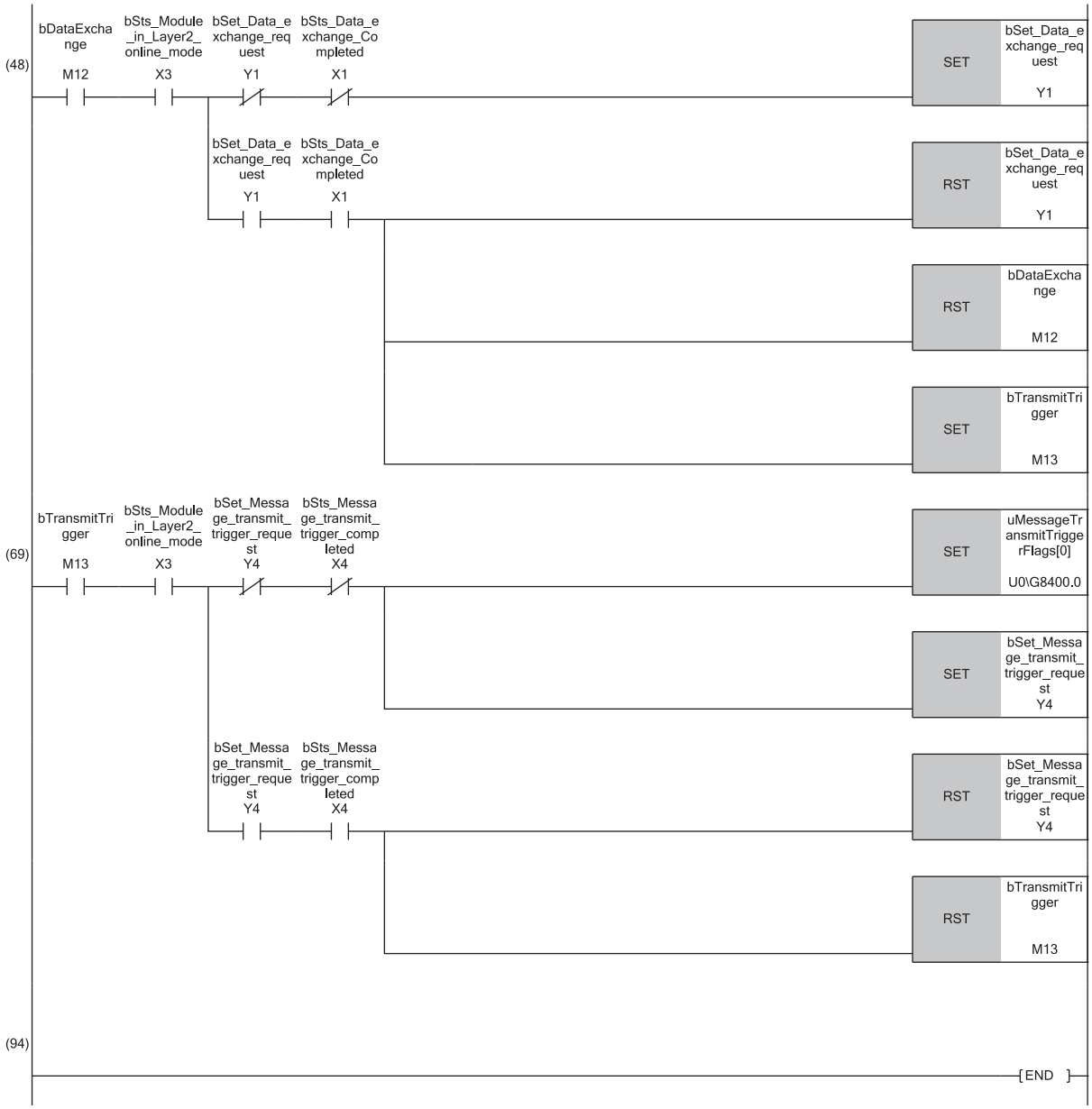
Message transmission program

Send the message of message slot 1.

Classification	Label name
Label to be defined	Define the global labels as shown below.

Label Name	Data Type		Class	Assign (Device/Label)
bTransmitMessage	Bit	...	VAR_GLOBAL	M10
bSetTransmitData	Bit	...	VAR_GLOBAL	M11
bDataExchange	Bit	...	VAR_GLOBAL	M12
bTransmitTrigger	Bit	...	VAR_GLOBAL	M13
bSts_Module_Ready	Bit	...	VAR_GLOBAL	X0
bSts_Data_exchange_Completed	Bit	...	VAR_GLOBAL	X1
bSts_Module_in_Layer2_online_mode	Bit	...	VAR_GLOBAL	X3
bSts_Message_transmit_trigger_completed	Bit	...	VAR_GLOBAL	X4
bSts_Module_error	Bit	...	VAR_GLOBAL	X0F
bSet_Data_exchange_request	Bit	...	VAR_GLOBAL	Y1
bSet_Layer2_online_mode_request	Bit	...	VAR_GLOBAL	Y3
bSet_Message_transmit_trigger_request	Bit	...	VAR_GLOBAL	Y4
uMessageTransmitTriggerFlags	Bit(0..41)	...	VAR_GLOBAL	U0\G8400.0
uLayer2MessageData1_DataByte	Word [Unsigned]/Bit String [16-bit](0..3)	...	VAR_GLOBAL	U0\G10003





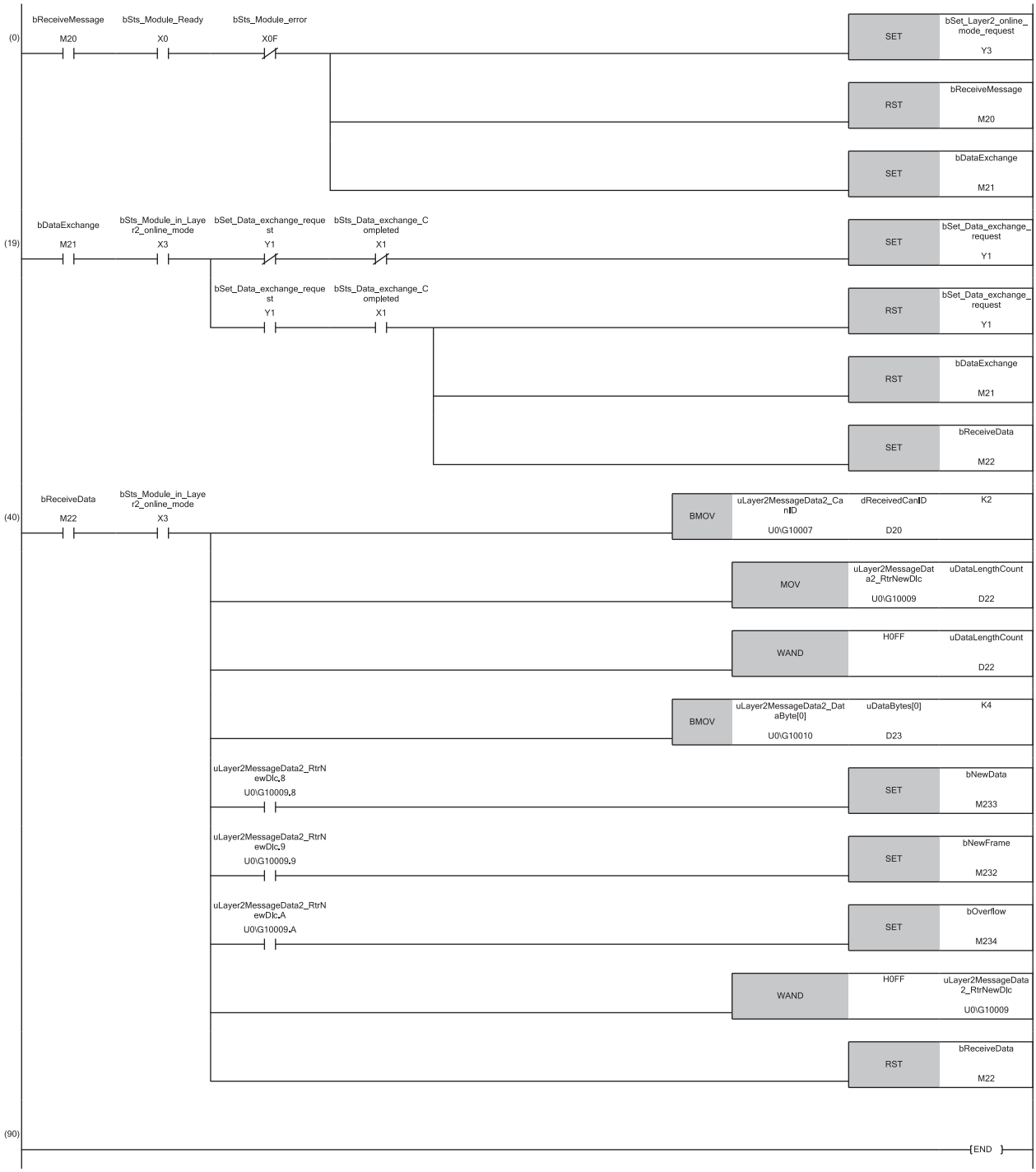
(0) Transition to the online mode.
(19) Register the transmission data.
(48) Execute data exchange.
(69) Execute the transmit trigger request.

Message receive program

Receive the message of message slot 2.

Classification	Label name
Label to be defined	Define the global labels as shown below.

Label Name	Data Type		Class	Assign (Device/Label)
bReceiveMessage	Bit	...	VAR_GLOBAL	M20
bDataExchange	Bit	...	VAR_GLOBAL	M21
bReceiveData	Bit	...	VAR_GLOBAL	M22
bSts_Module_Ready	Bit	...	VAR_GLOBAL	X0
bSts_Data_exchange_Completed	Bit	...	VAR_GLOBAL	X1
bSts_Module_in_Layer2_online_mode	Bit	...	VAR_GLOBAL	X3
bSts_Module_error	Bit	...	VAR_GLOBAL	X0F
bSet_Data_exchange_request	Bit	...	VAR_GLOBAL	Y1
bSet_Layer2_online_mode_request	Bit	...	VAR_GLOBAL	Y3
uLayer2MessageData2_CanID	Word [Unsigned]/Bit String [16-bit](0..1)	...	VAR_GLOBAL	U0\G10007
uLayer2MessageData2_RtrNewDlc	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	U0\G10009
uLayer2MessageData2_DataByte	Word [Unsigned]/Bit String [16-bit](0..3)	...	VAR_GLOBAL	U0\G10010
dReceivedCanID	Double Word [Unsigned]/Bit String [32-bit]	...	VAR_GLOBAL	D20
uDataLengthCount	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	D22
uDataBytes	Word [Unsigned]/Bit String [16-bit](0..3)	...	VAR_GLOBAL	D23
bNewFrame	Bit	...	VAR_GLOBAL	M232
bNewData	Bit	...	VAR_GLOBAL	M233
bOverflow	Bit	...	VAR_GLOBAL	M234



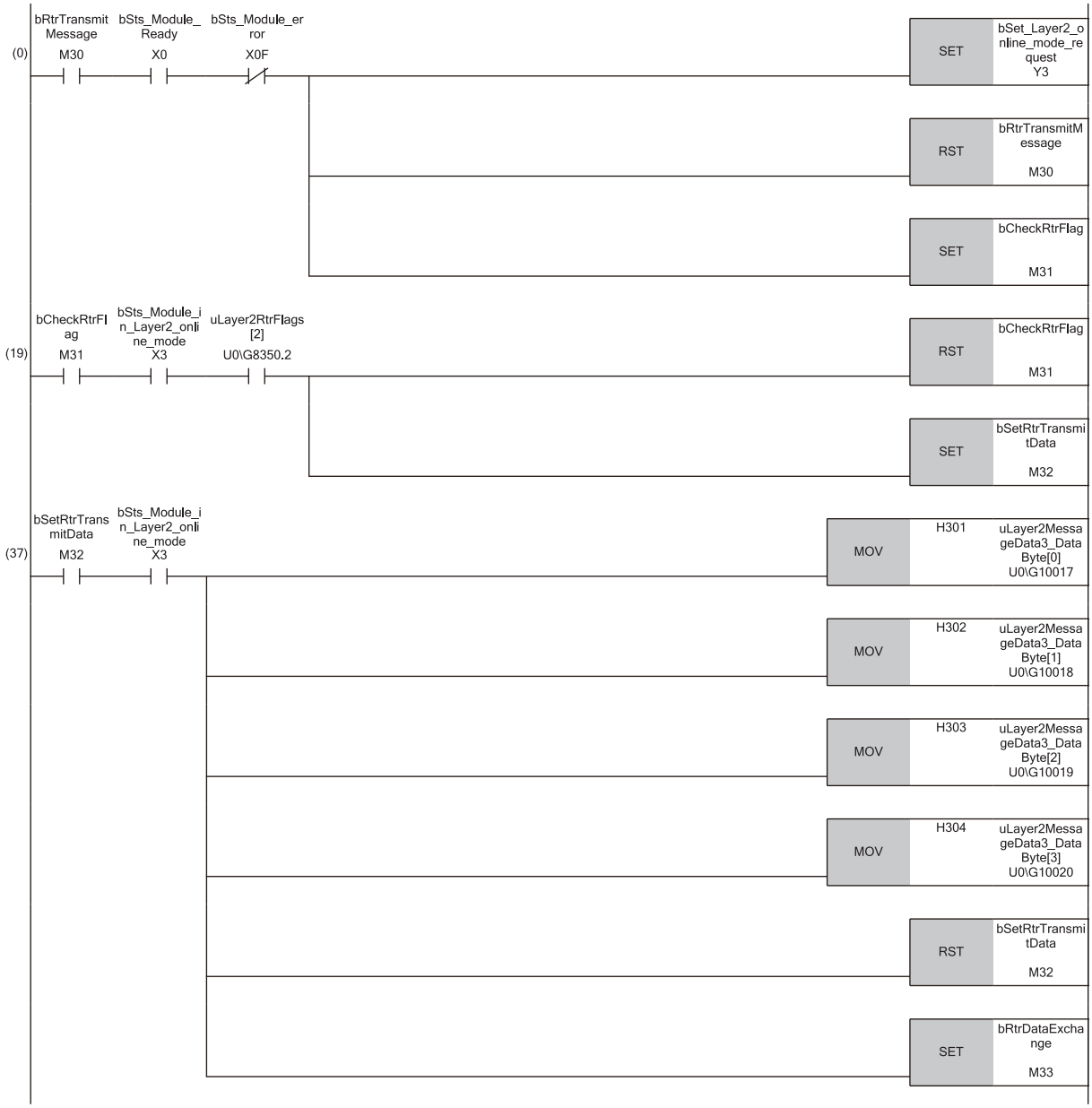
(0) Transition to the online mode.
(19) Execute data exchange.
(40) Retrieve the receive data.

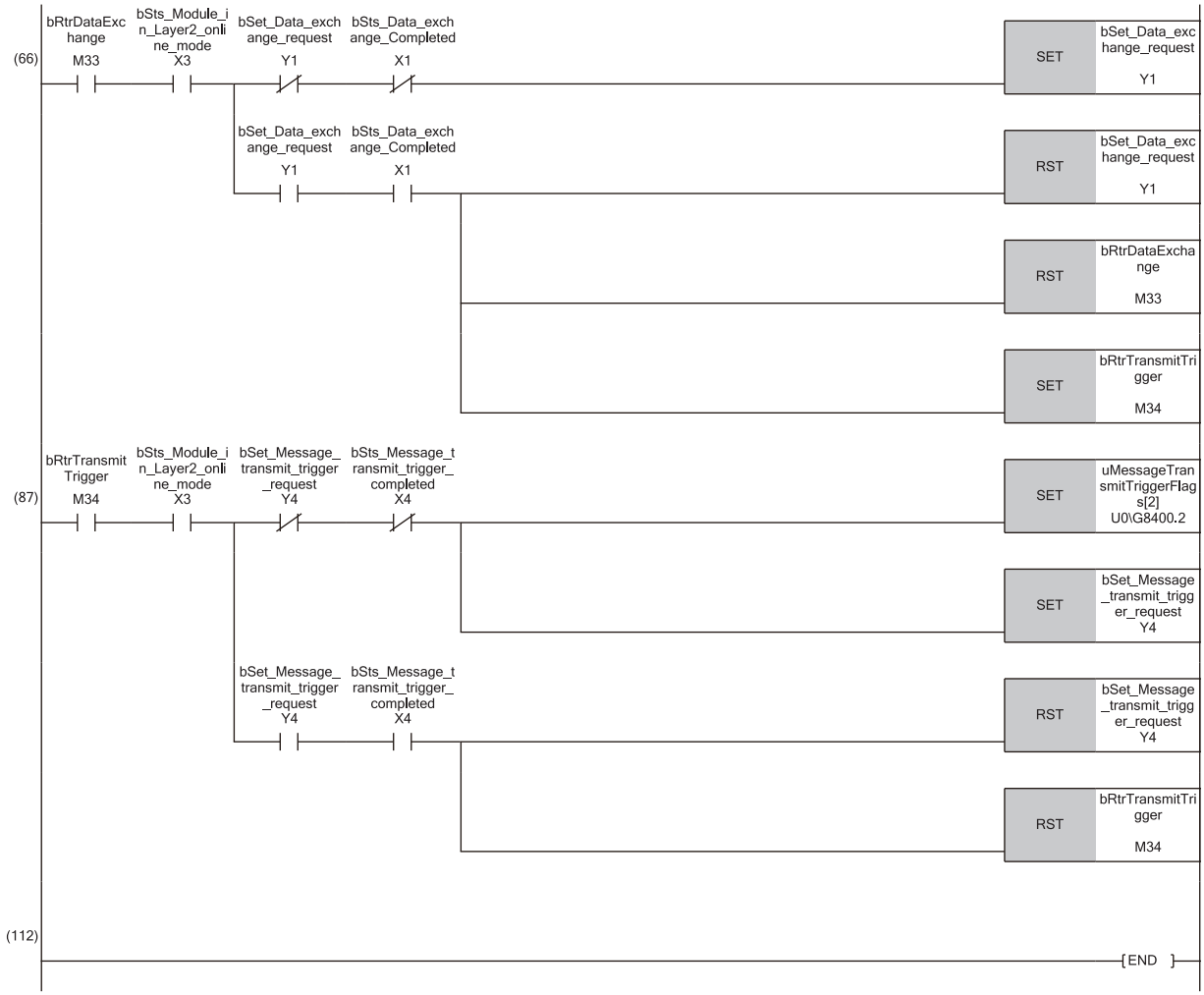
RTR response message transmission program

Check the RTR flag of message slot 3, and send the RTR response message.

Classification	Label name
Label to be defined	Define the global labels as shown below.

Label Name	Data Type		Class	Assign (Device/Label)
bRtrTransmitMessage	Bit	...	VAR_GLOBAL	M30
bCheckRtrFlag	Bit	...	VAR_GLOBAL	M31
bSetRtrTransmitData	Bit	...	VAR_GLOBAL	M32
bRtrDataExchange	Bit	...	VAR_GLOBAL	M33
bRtrTransmitTrigger	Bit	...	VAR_GLOBAL	M34
bSts_Module_Ready	Bit	...	VAR_GLOBAL	X0
bSts_Data_exchange_Completed	Bit	...	VAR_GLOBAL	X1
bSts_Module_in_Layer2_online_mode	Bit	...	VAR_GLOBAL	X3
bSts_Message_transmit_trigger_completed	Bit	...	VAR_GLOBAL	X4
bSts_Module_error	Bit	...	VAR_GLOBAL	X0F
bSet_Data_exchange_request	Bit	...	VAR_GLOBAL	Y1
bSet_Layer2_online_mode_request	Bit	...	VAR_GLOBAL	Y3
bSet_Message_transmit_trigger_request	Bit	...	VAR_GLOBAL	Y4
uLayer2RtrFlags	Bit(0..41)	...	VAR_GLOBAL	U0\G8350.0
uMessageTransmitTriggerFlags	Bit(0..41)	...	VAR_GLOBAL	U0\G8400.0
uLayer2MessageData3_DataByte	Word [Unsigned]/Bit String [16-bit](0..3)	...	VAR_GLOBAL	U0\G10017





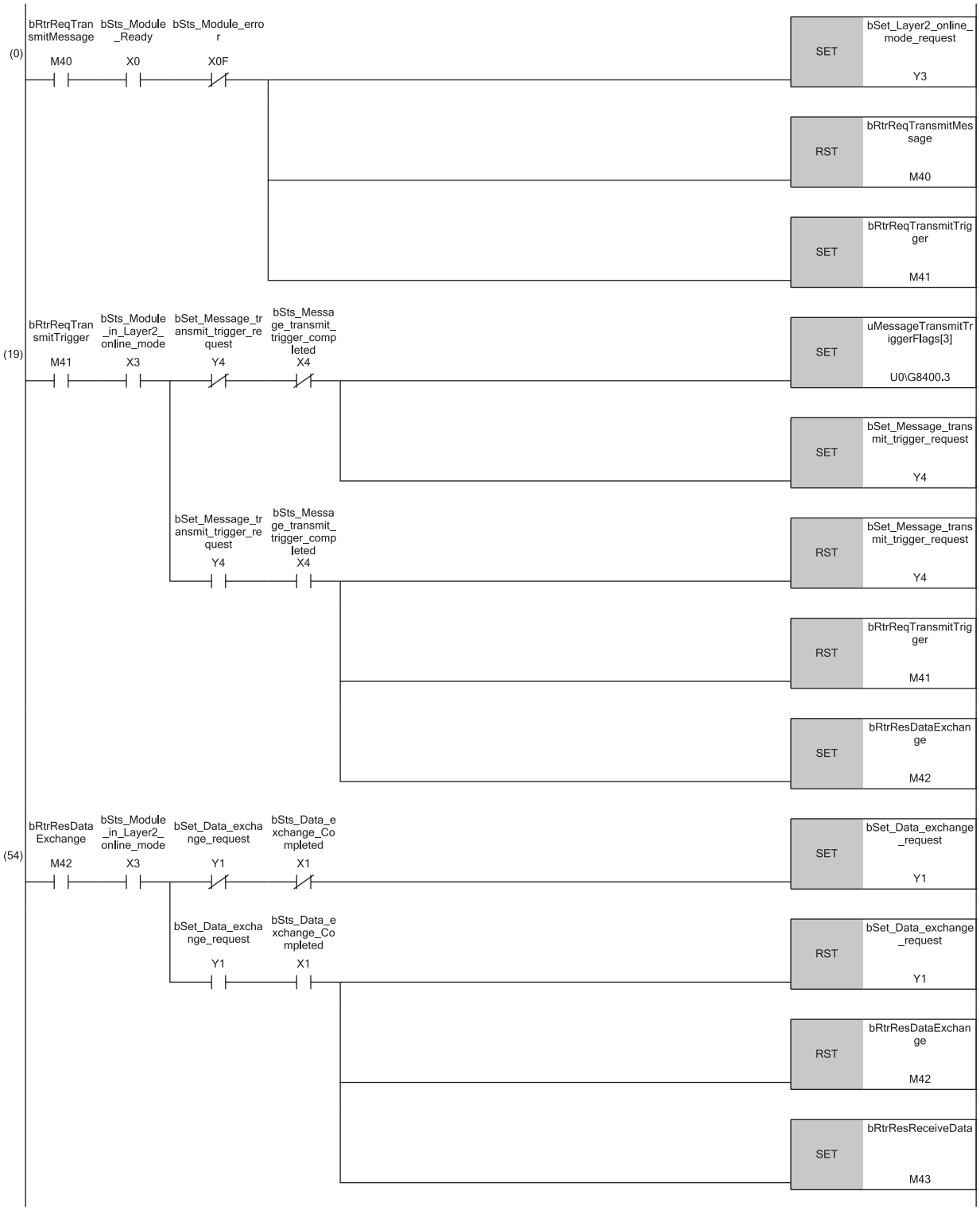
(0) Transition to the online mode.
(20) Check the RTR flag.
(39) Register the transmission data.
(69) Execute data exchange.
(91) Execute the transmit trigger request.

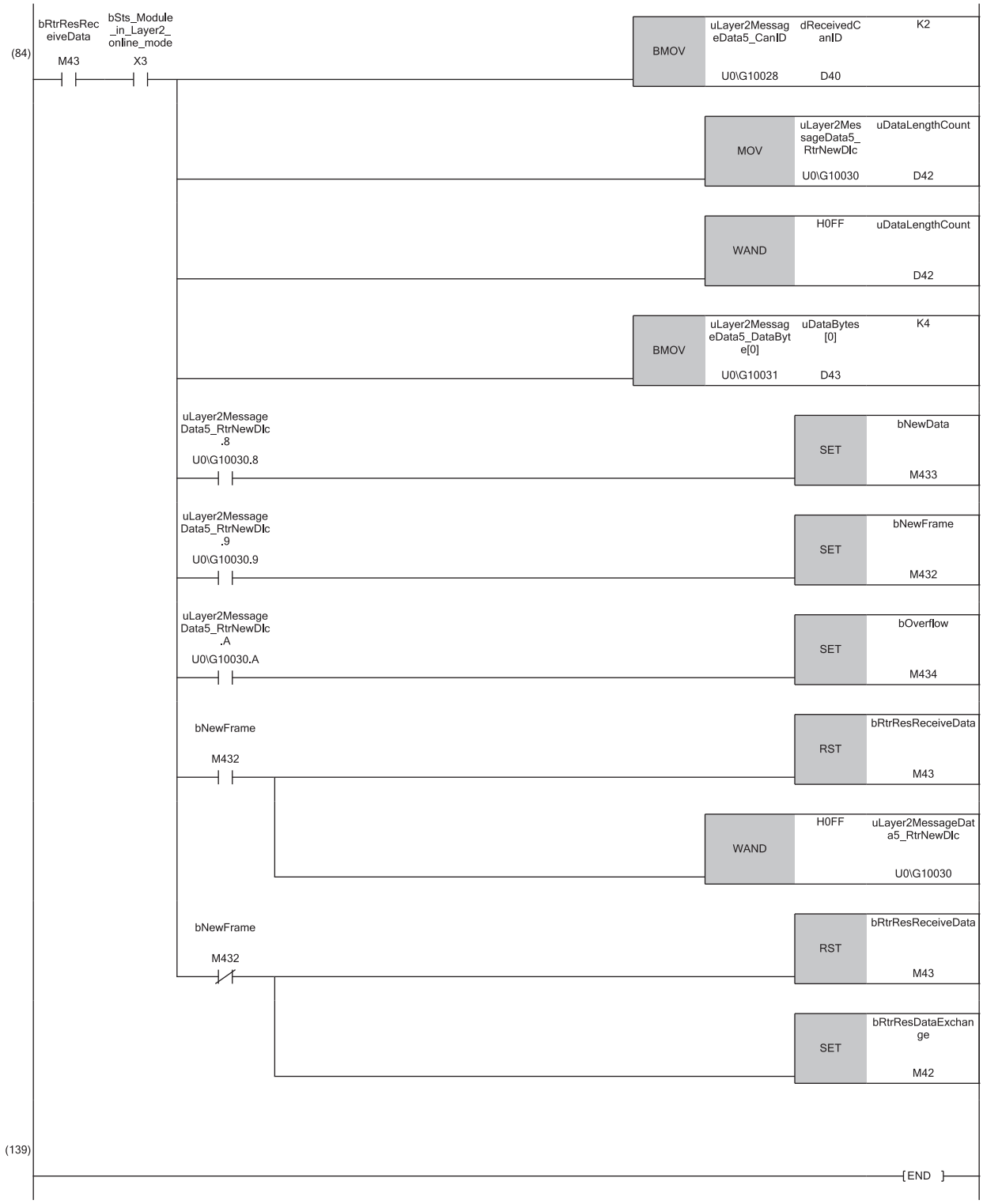
RTR request message transmission program

Send the RTR request message of message slot 4, and retrieve the RTR response message of message slot 5.

Classification	Label name
Label to be defined	Define the global labels as shown below.

Label Name	Data Type		Class	Assign (Device/Label)
bRtrReqTransmitMessage	Bit	...	VAR_GLOBAL	M40
bRtrReqTransmitTrigger	Bit	...	VAR_GLOBAL	M41
bRtrResDataExchange	Bit	...	VAR_GLOBAL	M42
bRtrResReceiveData	Bit	...	VAR_GLOBAL	M43
bSts_Module_Ready	Bit	...	VAR_GLOBAL	X0
bSts_Data_exchange_Completed	Bit	...	VAR_GLOBAL	X1
bSts_Module_in_Layer2_online_mode	Bit	...	VAR_GLOBAL	X3
bSts_Message_transmit_trigger_completed	Bit	...	VAR_GLOBAL	X4
bSts_Module_error	Bit	...	VAR_GLOBAL	X0F
bSet_Data_exchange_request	Bit	...	VAR_GLOBAL	Y1
bSet_Layer2_online_mode_request	Bit	...	VAR_GLOBAL	Y3
bSet_Message_transmit_trigger_request	Bit	...	VAR_GLOBAL	Y4
uMessageTransmitTriggerFlags	Bit(0..41)	...	VAR_GLOBAL	U0\G8400.0
uLayer2MessageData4_DataByte	Word [Unsigned]/Bit String [16-bit](0..3)	...	VAR_GLOBAL	U0\G10024
uLayer2MessageData5_CanID	Word [Unsigned]/Bit String [16-bit](0..1)	...	VAR_GLOBAL	U0\G10028
uLayer2MessageData5_RtrNewDlc	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	U0\G10030
uLayer2MessageData5_DataByte	Word [Unsigned]/Bit String [16-bit](0..3)	...	VAR_GLOBAL	U0\G10031
dReceivedCanID	Double Word [Unsigned]/Bit String [32-bit]	...	VAR_GLOBAL	D40
uDataLengthCount	Word [Unsigned]/Bit String [16-bit]	...	VAR_GLOBAL	D42
uDataBytes	Word [Unsigned]/Bit String [16-bit](0..3)	...	VAR_GLOBAL	D43
bNewFrame	Bit	...	VAR_GLOBAL	M432
bNewData	Bit	...	VAR_GLOBAL	M433
bOverflow	Bit	...	VAR_GLOBAL	M434


(0)
bRtrReqTransmitMessage
M40
bSts_Module_Ready
X0
bSts_Module_error
X0F
SET
bSet_Layer2_online_mode_request
Y3
RST
bRtrReqTransmitMessage
M40
SET
bRtrReqTransmitTrigger
M41
(19)
bRtrReqTransmitTrigger
M41
bSts_Module_in_Layer2_online_mode
X3
bSet_Message_transmit_trigger_request
Y4
bSts_Message_transmit_trigger_completed
X4
SET
uMessageTransmitTriggerFlags[3]
U0\G8400.3
SET
bSet_Message_transmit_trigger_request
Y4
bSet_Message_transmit_trigger_request
Y4
bSts_Message_transmit_trigger_completed
X4
RST
bSet_Message_transmit_trigger_request
Y4
RST
bRtrReqTransmitTrigger
M41
SET
bRtrResDataExchange
M42
(54)
bRtrResDataExchange
M42
bSts_Module_in_Layer2_online_mode
X3
bSet_Data_exchange_request
Y1
bSts_Data_exchange_Completed
X1
SET
bSet_Data_exchange_request
Y1
bSet_Data_exchange_request
Y1
bSts_Data_exchange_Completed
X1
RST
bSet_Data_exchange_request
Y1
RST
bRtrResDataExchange
M42
SET
bRtrResReceiveData
M43



(0) Transition to the online mode.
(19) Execute the transmit trigger request. (Send the RTR request.)
(54) Execute data exchange. (Receive the RTR response.)
(84) Retrieve the receive data.

4.2 Communication Example of CANopen 405 Mode

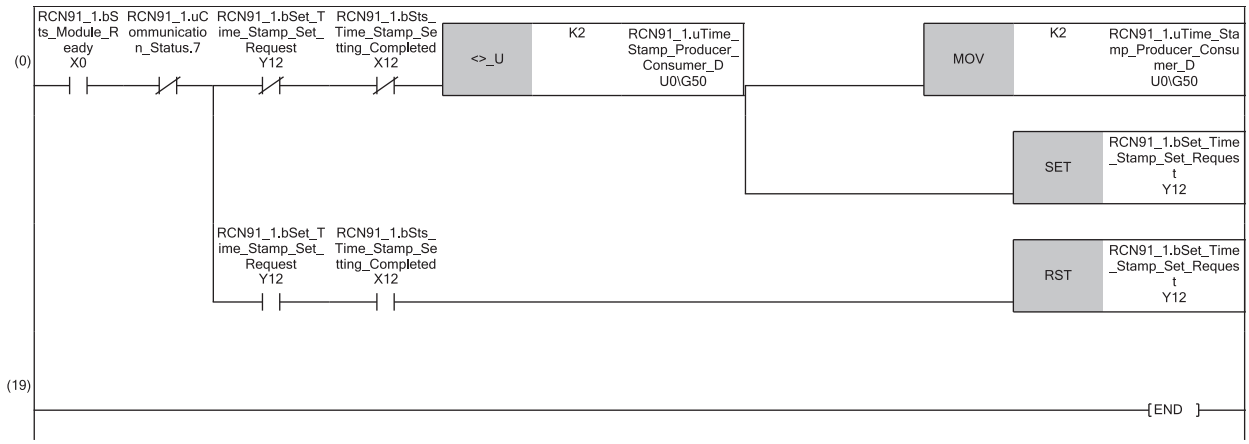
This section shows a program for resetting the values for the object dictionary when an NMT reset occurs in 'Time stamp' (Un\G50 to Un\G59) in the buffer memory (APPENDIX 2).

Program example

Set 2 (producer) to 'Time stamp' (Un\G50.0). (When the start I/O number of the RJ71CN91 is 0.)

Classification	Label name	Description	Device
Module label	RCN91_1.bSts_Module_Ready	Module READY	X0
	RCN91_1.uCommunication_Status.7*1	Initialization status of the RJ71CN91	U0\G25.7
	RCN91_1.bSet_Time_Stamp_Set_Request	Time stamp setting request	Y12
	RCN91_1.bSts_Time_Stamp_Setting_Completed	Time stamp setting completed	X12
	RCN91_1.uTime_Stamp_Producer_Consumer_D	Producer/Consumer	U0\G50

*1 RCN91_1.uCommunication_Status indicates the communication status.



5 TROUBLESHOOTING

This chapter describes troubleshooting for the RJ71CN91.

5.1 Checking with LED

This section describes troubleshooting using LED.

Error status can be determined by status of the RUN LED and the ERR LED.

RUN LED	ERR LED	Error status*1	Description
Off	—	Major error	An error such as hardware failure or memory failure. The module stops operating.
On	Flashing	Moderate error	An error, such as parameter error, which affects module operation. The module stops operating.
On	On	Minor error	An error such as communication failure. The module continues operating. If an error with error code 1804H (bus off state occurrence) occurs, it is necessary to restart the module to restore communication.

*1 When multiple errors occur, the error status is displayed in the order of major, moderate, and minor.

When the RUN LED turns off

When the RUN LED turns off after powering on the RJ71CN91, check the following.

Check item	Action
Is the RJ71CN91 mounted correctly?	Securely mount the RJ71CN91 on the base unit.

When the ERR LED turns on or is flashing

When the ERR LED turns on or is flashing, check the following.

Check item	Action
Does any error occur in the module diagnostics?	Take the actions displayed on the window.

When the CAN ERR LED turns on or is flashing

When the CAN ERR LED turns on or is flashing, check the following.

Check item	Action
Is any error displayed in 'Error status' (Un\G29)?	Refer to the following. ☞ Page 144 Troubleshooting Using the Buffer Memory

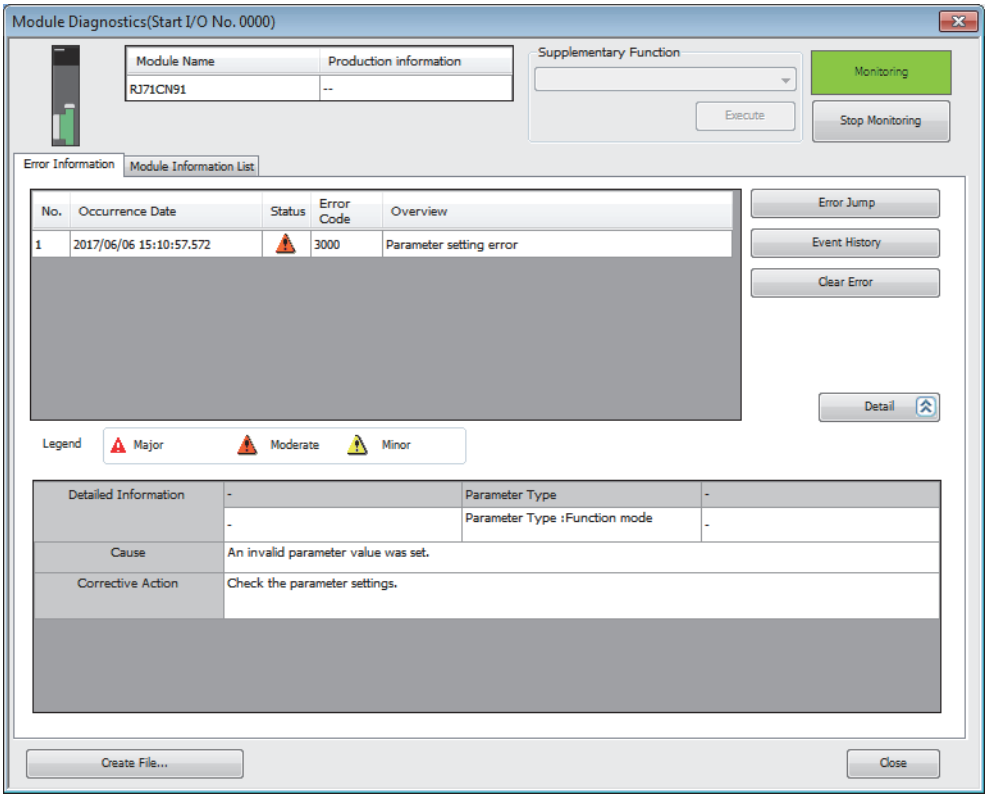
5.2 Checking the Module Status

The following items can be checked in the "Module Diagnostics" window for the RJ71CN91.

Item	Description
Error Information	Displays the details of the errors currently occurring. Click the [Event History] button to check the history of errors that have occurred on the network, errors detected for each module, and operations that have been executed.
Module Information List	Displays various status information of the RJ71CN91.
Clear Error	Clears continuation errors for the module.

Error Information

Check the details of the error currently occurring and action to remove the error.



Item	Description
Status	Major: An error such as hardware failure or memory failure. The module stops operating.
	Moderate: An error, such as parameter error, which affects module operation. The module stops operating.
	Minor: An error such as communication failure. The module continues operating.
Detailed Information	Displays detailed information about each error (maximum of 3 pieces).
Cause	Displays the detailed error causes.
Corrective Action	Displays the actions to eliminate the error causes.

Clear Error

Click the [Clear Error] button in the Error Information tab page in the Module Diagnostics window to execute this function.

This clears all the continuation errors of the RJ71CN91.

The following table lists errors and whether they can be cleared.

○: Can be cleared, ×: Cannot be cleared

Error name	Level	Whether it can be cleared
Continuation error	Minor error	○
Stop error	Moderate error	×
	Major error	×

The following table shows which error notifications/indications can be cleared by the "Clear Error" operation.

○: Can be cleared, ×: Cannot be cleared

Clear target		Whether it can be cleared	Status after error clear
Module	ERR LED	○	Off
	Each error information in buffer memory	×	—
GX Works3	Error status on the system monitor window	○	No error
	Error codes on the system monitor window	○	No error code
	Error information on the Module Diagnostics window	○	No error Information
	Event history	×	—

■Procedure for clearing error

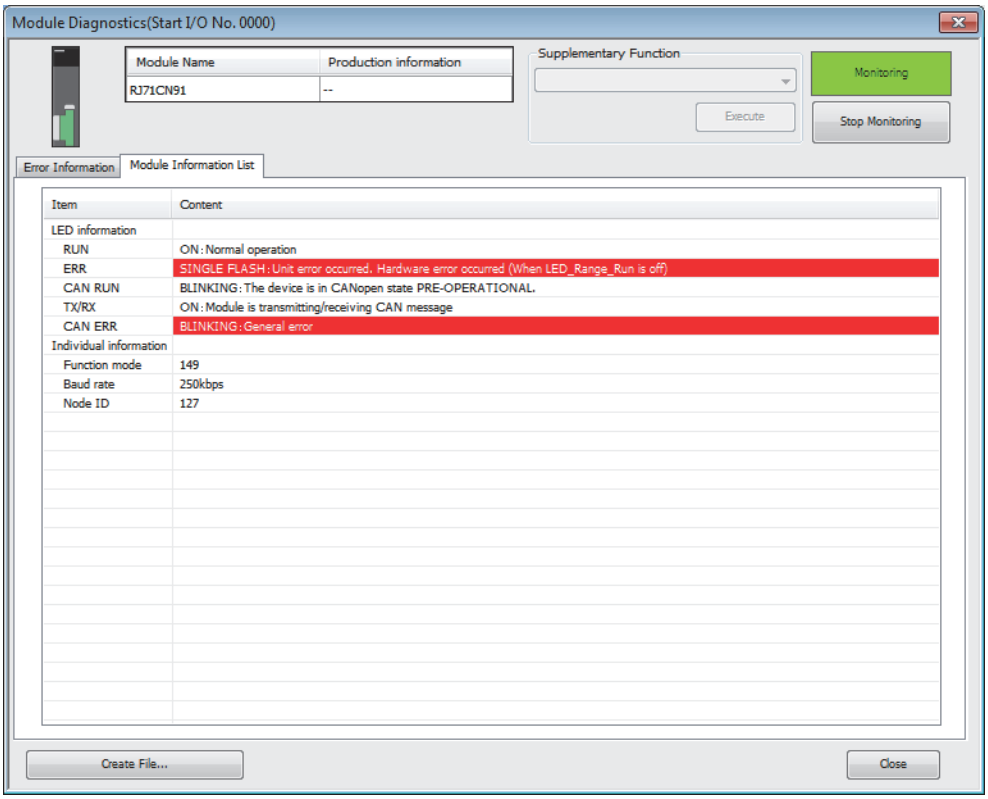
Clear errors using the module diagnostics function of GX Works3. (📖 GX Works3 Operating Manual)

Precautions

- Errors cannot be cleared while a stop error exists.
- Errors may occur and be detected again if the error causes are not removed or another error occurs after clearing.

Module Information List

Switch to the [Module Information List] tab to check various status information of the RJ71CN91.



Item		Description
LED information		Displays the LED status of the RJ71CN91.
Individual information	Function mode	Displays the function mode set for the selected module.
	Baud rate	Displays the baud rate set for the selected module. However, when the Layer 2 offline mode is active, "—" is displayed.
	Node ID	Displays the node ID set for the selected module.

5.3 Troubleshooting Using the Buffer Memory

Check 'Error status' (Un\G29) to confirm the error status of the RJ71CN91.

Bit	Error description	Action
b0	System area	—
b1	Hardware error	If this bit is not cleared when 'Module restart request' (Y2) is turned on or the RJ71CN91 is powered off and on or reset, the RJ71CN91 hardware may be in failure. Please consult your local Mitsubishi representative.
b2	System area	—
b3	Bus off state, or the value of 'CAN transmission error counter' (Un\G35) becomes 256 or more.	Check the following, and then power off and on or reset the RJ71CN91. • Are terminating resistors connected to the both ends of the network? • Is the baud rate the same at all nodes? • Are all node IDs not duplicated? • Are the CAN_H, CAN_L, and CAN_GND wires not disconnected? • Is CAN_SHLD of the CAN bus cable grounded? • Is CAN_SHLD of the CAN bus cable connected to all nodes? • Is the CAN bus cable not short-circuited? • Check that other nodes are running normally.
b4	Flash ROM error	There is invalid data in the flash ROM. If this bit is not cleared when 'Module restart request' (Y2) is turned on or the RJ71CN91 is powered off and on or reset, the RJ71CN91 hardware may be in failure. Please consult your local Mitsubishi representative.
b5	Invalid save/restore configuration request in Layer 2 message mode	Configuration save/factory default configuration restore operation was executed when the online mode of Layer 2 message mode is active. Check the program and execute the configuration save/factory default configuration restore operation when the RJ71CN91 was in the offline mode. For details on the offline mode, refer to the following. (☞ Page 155 Layer 2 online mode status (X3), Layer 2 online mode request (Y3))
b6	Buffer memory setting error	This bit turns on when a value out of the setting range is written to the buffer memory. The buffer memory address where this error occurred is stored in 'Buffer memory setting error display' (Un\G39). (☞ Page 166 Buffer memory setting error display (Un\G39)) Check 'Buffer memory setting error display' (Un\G39), and set the value in the setting range.
b7	System area	—
b8	Send/receive buffer overflow	Extreme bus load caused the send buffer or receive buffer to overflow. Decrease the bus load. Send buffer overflow may occur if data exchange is swiftly performed when the baud rate is low. When a large number of slaves are registered in "NMT slave assignment" for the master and then the master and slaves are powered on, a receive buffer overflow may occur in a slave.
b9	System area	—
b10	NMT error control failure	Check 'NMT error control status' (Un\G401 to Un\G527) and check the status of the node in which an NMT error control error occurred. (☞ Page 173 NMT error control status (Un\G401 to Un\G527))
b11 to b13	System area	—
b14	CAN error active state/passive state	This bit turns on if the RJ71CN91 switches to the error passive state. Check the following details. • Are terminating resistors connected to the both ends of the network? • Do all nodes have the same baud rate setting? • Is any node ID duplicated? • Is any CAN-ID duplicated? • Is any communication cable disconnected? • Are communication cables grounded properly? • Is any communication cable short-circuited? This error will be cleared when the value of 'CAN transmission error counter' (Un\G35) or 'CAN reception error counter' (Un\G36) becomes 128 or less. (☞ Page 165 CAN transmission error counter (Un\G35), Page 165 CAN reception error counter (Un\G36))
b15	Layer 2 message slot specific error	Refer to 'Message slot specific error code list' (Un\G5001 to Un\G5042) to check the error code of each message slot. (☞ Page 181 Message slot specific error code list (Un\G5001 to Un\G5042))

5.4 Troubleshooting by Symptom

This section describes troubleshooting method by symptom. If an error occurs in the RJ71CN91, identify the cause of the error using the engineering tool. (☞ Page 141 Checking the Module Status)

CANopen 405 mode

This section describes troubleshooting by symptom in the CANopen 405 mode.

SDO communication is not available

The following table lists the actions to be taken if SDO communication is not available.

Check item	Action
Is the CAN bus cable connected firmly to the transmission cable terminal block of the RJ71CN91?	Connect the CAN bus cable firmly to the transmission cable terminal block.
Are terminating resistors connected to the both ends of the network?	Connect terminating resistors to both ends of the network.
Is the baud rate the same at all CANopen nodes in the network?	Ensure that the baud rate is the same at all CANopen nodes in the network.
Is the target node capable of normal communication?	Check the target node to see if there is any problem.
Is the RJ71CN91 function mode set to CANopen 405 mode?	Set the RJ71CN91 function mode to CANopen 405 mode.
Is any node ID duplicated?	Change duplicated node IDs of CANopen nodes.
Are parameters set correctly for the RJ71CN91?	Set parameters correctly for the RJ71CN91.
Is the RJ71CN91 in the bus off state?	Check the above check items, take actions, and restart the RJ71CN91.

The NMT state of a CANopen node does not change to "Operational"

If the NMT state of a CANopen node does not change to "Operational", check the following.

Check item	Action
Are parameter settings of all the flying masters correct?	Set parameters correctly for all the flying masters.
Does the parameter setting of the NMT master match the system configuration of the actual NMT slave?	Set parameters correctly for the NMT master. Modify the system configuration correctly.
Is the CPU module with which the RJ71CN91 is installed in RUN state?	Set the CPU module with which the RJ71CN91 is installed in RUN state.
Is the NMT state of the NMT master Operational?	• Enable the NMT master start. • Check whether the startup process failed in the NMT master event history. • Use the node control function to run remote node start for the own node.
Is the NMT state of the NMT slave Operational?	• Check that the NMT slave operates properly. • Check that the NMT state of the NMT master is Operational. • Assign the NMT slave to the NMT master. • Enable the NMT master parameter "NMT boot slave" for the target slave. • From the NMT master, use the node control function to run remote node start.
Do the current node settings match the parameters? (Read the settings using the [Read NMT settings] button in CANopen Configuration Tool, and compare them with the parameters.)	• Delete the unnecessary CDCFs from Concise DCF (index 1F22H, subindex 01H to 7FH) in the object dictionary of the NMT master.*1 • If the object dictionary with unmatched settings was changed through SDO communication, modify or delete the relevant process.

*1 All CDCFs can be deleted by using the [Restore Object Dictionary to default on all nodes] button in the “CANopen Configuration" window.
☞ Page 84 "CANopen Configuration" window

PDO communication is not available

The following table lists the actions to be taken if PDO communication is not available.

Check item	Action
Are the NMT state of the CANopen node for PDO transmission and the NMT state of the CANopen node for PDO reception Operational?	Set both the NMT state of the CANopen node for PDO transmission and the NMT state of the CANopen node for PDO reception as Operational. (☞ Page 145 The NMT state of a CANopen node does not change to "Operational")
Were the TPDO and RPDO for CANopen Configuration Tool set correctly?	By paying attention to the following, set parameters correctly for CANopen Configuration Tool. • Is PDO communication enabled? • Do a TPDO and RPDO that form a pair have the same CAN-ID?
Do the current PDO settings match the parameters? (Read the settings using the [Read PDO List] button in CANopen Configuration Tool, and compare them with the parameters.)	• Delete the unnecessary CDCFs from Concise DCF (index 1F22H, subindex 01H to 7FH) in the object dictionary of the NMT master.*1 • If the object dictionary with unmatched settings was changed through SDO communication, modify or delete the relevant process.
Is 'Data exchange request' (Y1) turned on?	Turn on 'Data exchange request' (Y1) to update send/receive data.
Is the send data changed from the previous transmission when "Transmission type" is Event-driven and "Event timer" is not set?	Change the send data from the previous transmission, and turn on 'Data exchange request' (Y1).
Is the SYNC message transmission of the NMT master enabled when Transmission type is Synchronous?	Enable the SYNC message transmission of the NMT master.

*1 All CDCFs can be deleted by using the [Restore Object Dictionary to default on all nodes] button in the “CANopen Configuration" window.
☞ Page 84 "CANopen Configuration" window

Layer 2 message mode

The following is troubleshooting by symptom in the 11-bit CAN-ID Layer 2 message mode and 29-bit CAN-ID Layer 2 message mode.

Cannot send/receive Layer 2 messages

The following table lists the actions to be taken if Layer 2 message communication is not possible.

Check item	Action
Is the CAN bus cable connected firmly to the transmission cable terminal block of the RJ71CN91?	Connect the CAN bus cable firmly to the transmission cable terminal block.
Are terminating resistors connected to the both ends of the network?	Connect terminating resistors to both ends of the network.
Is the baud rate the same at all CANopen nodes in the network?	Ensure that the baud rate is the same at all CANopen nodes in the network.
When communication was executed, were one or more other stations capable of communication connected?	• Execute communication with one or more other stations connected. • Execute communication after making one or more other stations capable of communication. (If the device type of other stations is RJ71CN91, activate the Layer 2 online mode.)*1
Is the RJ71CN91 function mode set to 11-bit CAN-ID Layer 2 message mode or 29-bit CAN-ID Layer 2 message mode? Is the 11-bit CAN-ID Layer 2 message mode or 29-bit CAN-ID Layer 2 message mode selected according to the format used?	Set the RJ71CN91 function mode to 11-bit CAN-ID Layer 2 message mode or 29-bit CAN-ID Layer 2 message mode. Also, ensure that the 11-bit CAN-ID Layer 2 message mode or 29-bit CAN-ID Layer 2 message mode is selected according to the format used.
Are parameters set correctly for the RJ71CN91?	Set parameters correctly for the RJ71CN91.
Is 'Layer 2 online mode status' (X3) turned on?	Turn on 'Layer 2 online mode request' (Y3).
Is the RJ71CN91 in the bus off state?	Turn on 'Layer 2 online mode request' (Y3) to bring the RJ71CN91 into the Layer 2 online mode again.
Is 'Data exchange request' (Y1) turned on?	Turn on 'Data exchange request' (Y1) to update send/receive data.
Is the send data changed from the previous transmission when "Transmission type" is Event (COS) or Cycle (COS)?	Change the send data from the previous transmission, and turn on 'Data exchange request' (Y1).
Is 'Message transmit trigger request' (Y4) turned on when the transmission type is Transmit trigger?	In 'Message transmit trigger flags (Un\8400 to Un\G8402), turn on the bit corresponding to the sending message slot and turn on 'Message transmit trigger request' (Y4) to send data.

*1 When the device type of other stations is RJ71CN91, communication is disregarded during Layer 2 offline mode.

5.5 List of Error Codes

This section lists the error codes, error definitions and causes, and actions for the errors that occur in processing for data communication between the RJ71CN91 and other nodes or caused by processing requests from the CPU module on the own node.

Error codes are classified into major error, moderate error, and minor error, and can be checked in the [Error Information] tab of the "Module Diagnostics" window of the RJ71CN91. (☞ Page 141 Error Information)

Error code	Error definition and cause	Action	Detailed information
1080H	The number of writes to the flash ROM has exceeded 100000.	Replace the RJ71CN91.	—
1800H	A high load on the bus/communication failure caused the send buffer or receive buffer to overflow.	Check if communication with other nodes is possible. Check the following items if communication is possible. • Is the communication frequency high for SDO or PDO communication? • Is the communication frequency high for node guarding or heartbeat communication? Check the following items if communication is not possible. • Are terminating resistors connected to the both ends of the network? • Do all nodes have the same baud rate setting? • Is any communication cable disconnected? • Are communication cables grounded properly? • Is any communication cable short-circuited? • Are other nodes running normally?	—
1801H	The data length of the received PDO is invalid.	Check the mapping parameter on the PDO sending/receiving side.	PDO number PDO error cause
1802H	An invalid CANopen object was received.	Check the setting of nodes connected to the network.	COB-ID CANopen error cause
1803H	A Layer 2 message slot specific error occurred.	Check the message slot specific error code.	Message slot number
1804H	The module is in bus off state due to a massive amount of transmission errors.	Restart the RJ71CN91. If the same error is displayed again, check the following. • Are terminating resistors connected to the both ends of the network? • Do all nodes have the same baud rate setting? • Is any node ID duplicated? • Is any CAN-ID duplicated? • Is any communication cable disconnected? • Are communication cables grounded properly? • Is any communication cable short-circuited? • Are other nodes running normally?	—
1805H	A configuration save/factory default configuration restore request was executed when the RJ71CN91 was in the Layer 2 online mode state.	Bring the RJ71CN91 into the offline mode state, and then save/restore the settings.	—
1806H	An error occurred during NMT slave start-up.	Check the following. • Is a correct node ID assigned to the NMT slave? • Is the identification information of the NMT slave set in the NMT master correct? • Do values to be set satisfy the NMT slave specifications? • Is any communication cable disconnected? • Are other nodes running normally? • Do the settings for node guarding or heartbeat correspond to the settings on the slave side? • When using heartbeat, is the value for the consumer heart beat time sufficiently larger than the value for the producer heartbeat time? • When using node guarding, is the value for the node life time sufficiently larger than the value for the guard time?	Node ID NMT slave start-up error cause
1807H	An error occurred during NMT start-up.	Check the following. • NMT slave state • Is the boot time setting value sufficient?	NMT start-up error cause

Error code	Error definition and cause	Action	Detailed information
1808H	An error occurred in the NMT slave setting during NMT start-up.	Check the following. • Is a correct node ID assigned to the NMT slave? • Are values to be set correct? • Is the NMT slave running? • NMT slave state • Is the boot time setting value sufficient?	Node ID NMT slave setting error cause
2250H	An error was detected in the extension parameter.	• Write this module extension parameters into the CPU module. • Check and correct the settings for extension parameters and rewrite them to the CPU module. • Exchange the module if an error occurs after rewriting them.	Parameter type
3000H	The set parameter value is invalid.	Check the value set for the parameter. The value may become different from the set value because it can be overwritten by the RJ71CN91 when a parameter error occurs.	Buffer memory address
3001H	Invalid data in the flash ROM might be caused by power-off during a write operation to the flash ROM.	Save parameters in the flash ROM again, and restart the RJ71CN91. If the same error is displayed again, please consult your local Mitsubishi representative.	—
3C00H, 3C02H	A hardware failure has been detected.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the error occurs again even after taking the above, the possible cause is a hardware failure of the RJ71CN91, base unit, or extension cable. Please consult your local Mitsubishi representative.	Cause type code

5.6 Event List

This section lists the events which occur in the RJ71CN91.

System and operation are included in the event types.

System

Event code	Overview	Cause
00400	Start in CANopen 405 mode	The module started in CANopen 405 mode.
00401	Start in 11-bit CAN-ID Layer 2 message mode	The module started in 11-bit CAN-ID Layer 2 message mode.
00402	Start in 29-bit CAN-ID Layer 2 message mode	The module started in 29-bit CAN-ID Layer 2 message mode.
00404	Module restart complete	Module restart is completed.
00405	Layer 2 online mode status	The module switched to Layer 2 online mode status.
00406	Layer 2 offline mode status	The module switched to Layer 2 offline mode status.
00408	Error clear complete	Error clear is completed.
00409	NMT error control failure clear complete	NMT error control failure clear is completed.
0040A	EMCY message area clear complete	EMCY message area clear is completed.
0040C	Configuration save complete	The settings were saved by one of the following operations. · Buffer memory settings save operation was executed. · The object dictionary store parameter was executed. · The Concise DCF was written to the object dictionary. · A new Concise DCF was obtained from the module extension parameters.
0040D	Restore factory default complete	The factory default settings were restored by one of the following operations. · Buffer memory factory default configuration restore operation was executed. · The object dictionary restore default parameter was executed. (The object dictionary and CDCFs are the targets to be restored.)
0040E	CAN error counter warning level	The CAN error counter reached the warning level.
0040F	CAN error counter passive state	The CAN error counter entered passive state.
00410	EMCY message received	An EMCY message was received.
00411	EMCY message transmission command completed	An EMCY message was sent by the command interface (CIF).
00412	NMT error control event	An NMT error control event occurred.
00413	Node guarding event	A node guarding event occurred in the own node which is the NMT slave.
00414	NMT reset executed	CANopen was reset by an NMT reset request.
00415	NMT state changed	The NMT state was changed.
00416	Flying master event	A flying master event occurred.

APPENDICES

Appendix 1 I/O Signals

This section describes the I/O signals for the CPU module. The I/O signal assignment of when the start I/O number of the RJ71CN91 is "0" is listed below.

List of I/O signals

The following tables list I/O signals. The device X is an input signal from the RJ71CN91 to the CPU module. The device Y is an output signal from the CPU module to the RJ71CN91.

CANopen 405 mode

This section lists I/O signals for the CANopen 405 mode.

■Input signals

Device number	Signal name
X0	Module READY
X1	Data exchange completed
X2	Module restart completed
X3 to XE	Use prohibited
XF	RJ71CN91 error
X10	NMT error control failure available
X11	EMCY message available
X12	Time stamp setting completed
X13	Time stamp information available in buffer memory
X14 to X16	Use prohibited
X17	Command execution completed
X18 to X1E	Use prohibited
X1F	Configuration save/factory default configuration restore completed

■Output signals

Device number	Signal name
Y0	Use prohibited
Y1	Data exchange request
Y2	Module restart request
Y3 to YE	Use prohibited
YF	RJ71CN91 error clear request
Y10	NMT error control failure clear request
Y11	EMCY message area clear request
Y12	Time stamp setting request
Y13	Time stamp read request
Y14 to Y16	Use prohibited
Y17	Command execution request
Y18 to Y1E	Use prohibited
Y1F	Configuration save/factory default configuration restore request

A

Layer 2 message mode

This section lists I/O signals in the 11-bit CAN-ID Layer 2 message mode and 29-bit CAN-ID Layer 2 message mode.

■Input signals

Device number	Signal name
X0	Module READY
X1	Data exchange completed
X2	Module restart completed
X3	Layer 2 online mode status
X4	Message transmit trigger completed
X5 to XE	Use prohibited
XF	RJ71CN91 error
X10 to X16	Use prohibited
X17	Command execution completed
X18 to X1E	Use prohibited
X1F	Configuration save/factory default configuration restore completed

■Output signals

Device number	Signal name
Y0	Use prohibited
Y1	Data exchange request
Y2	Module restart request
Y3	Layer 2 online mode request
Y4	Message transmit trigger request
Y5 to YE	Use prohibited
YF	RJ71CN91 error clear request
Y10 to Y16	Use prohibited
Y17	Command execution request
Y18 to Y1E	Use prohibited
Y1F	Configuration save/factory default configuration restore request

Point

Do not use (turn on) any "use prohibited" signals as an input or output signal to the CPU module.
Doing so may cause malfunction of the programmable controller system.

Details of I/O signals

This section describes the ON/OFF timing and condition for each I/O signal.

Module READY (X0)

This signal turns on when the CPU module is powered off and on or reset or when the RJ71CN91 is ready.
It turns off when a watchdog timer error occurs.

- On: Module ready
- Off: The module is in preparation or a watchdog timer error exists.

Data exchange completed (X1), Data exchange request (Y1)

These signals are used for data exchange between the buffer memory, object dictionary, and communication buffer of the RJ71CN91.
When 'Data exchange request' (Y1) is turned on, data exchange starts as follows.

- The latest receive data that has been received by the RJ71CN91 is stored in the buffer memory.
- The send data written in the buffer memory is loaded into the RJ71CN91.

When data exchange is completed, 'Data exchange completed' (X1) is turned on.
After 'Data exchange completed' (X1) is turned on, turn off 'Data exchange request' (Y1). Then 'Data exchange completed' (X1) will be turned off.
The following table lists the data exchange target buffer memory areas.
○: Data exchange target, ×: Not data exchange target

Buffer memory	Data exchange direction	Function mode	
		CANopen 405 mode	11-bit CAN-ID Layer 2 message mode 29-bit CAN-ID Layer 2 message mode
EMCY message buffer	RJ71CN91 → Buffer memory	○	×
RPDO	RJ71CN91 → Buffer memory	○*1	×
TPDO	Buffer memory → RJ71CN91	○*1	×
Receive/Transmit process data	Depending on settings for each slot. Send data: Buffer memory → RJ71CN91 Receive data: RJ71CN91 → Buffer memory	×	○

*1 Data exchange is not performed when the own node is not Operational.
When 'Data exchange control' (Un\G20) is used, the data exchange target buffer memory areas can be changed for each data exchange operation.

Point

- Do not read or write buffer memory send/receive data during data exchange operation.
- Request the next data exchange operation after 'Data exchange completed' (X1) is turned off.
- By following the above two procedures, it becomes possible to prevent the data inconsistency of buffer memory send/receive data and the overwriting of receive data in an unexpected timing.
- For details on the timings in which data after data exchange is actually sent or received on the network, refer to the following.

☞ Page 41 PDO, Page 52 EMCY,Page 62 Layer 2 Message Transmission and Receive

Module restart completed (X2), Module restart request (Y2)

When the following parameters are set in the buffer memory, the RJ71CN91 module must be restarted to enable these set parameters.

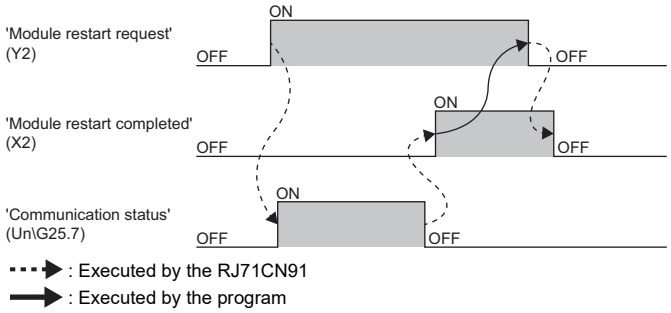
Recovery from the bus off state can be made and the following settings can be changed.

To change any of the following settings, save the settings after change and restart the module. All settings not saved will be lost.

- 'Function mode' (Un\G21) (☞ Page 161 Function mode (Un\G21))
- 'Baud rate' (Un\G24) (For CANopen mode only) (☞ Page 163 Baud rate (Un\G24))
- 'Node ID' (Un\G27) (☞ Page 170 Node ID (Un\G27))

To restart the RJ71CN91, turn on 'Module restart request' (Y2). The RJ71CN91 transitions to NMT state Initialization, and the RJ71CN91 initialization state is stored in 'Communication status' (Un\G25.7). (☞ Page 164 Communication status (Un\G25))

When 'Module restart completed' (X2) is turned on after the RJ71CN91 restarts, turn off 'Module restart request' (Y2). The restart procedure will take approx. 6 seconds to complete.



Point

When the RJ71CN91 is restarted, the error is also cleared.

Layer 2 online mode status (X3), Layer 2 online mode request (Y3)

In 11-bit CAN-ID Layer 2 message mode or 29-bit CAN-ID Layer 2 message mode, bring the own node into the online or offline mode. Activate the online mode to communicate data from/to other nodes in the network. Activate the offline mode to change the settings of the own node.

When 'Layer 2 online mode request' (Y3) is turned on, the settings for the own node are checked. When the settings are normal, the RJ71CN91 transitions to the online mode and 'Layer 2 online mode status' (X3) is turned on.

However, if the settings are abnormal, a setting error occurs and 'Error status' (Un\G29.6) is turned on. In that case, 'Layer 2 online mode status' (X3) will not be turned on. When 'Layer 2 online mode request' (Y3) is turned off, the RJ71CN91 transitions to the offline mode and 'Layer 2 online mode status' (X3) is turned off.

In addition, by turning on 'Layer 2 online mode request' (Y3), it becomes possible to make recovery from the bus off state.

Point

- Data cannot be sent or received during offline mode. Data communication from other nodes are all ignored. Even after the state changes to the online mode, data received during offline mode cannot be retrieved.
- The settings cannot be changed during online mode. Even if any changes are made, they will be ignored.
- Configuration save/factory default configuration restore operation cannot be executed during online mode. If it is executed, an error occurs. For details on configuration save/factory default configuration restore operation, refer to the following.

☞ Page 162 Save/restore configuration (Un\G22)

- For details on the settings for 11-bit CAN-ID Layer 2 message mode or 29-bit CAN-ID Layer 2 message mode, refer to the following.

☞ Page 74 Setting Parameters

However, the function mode cannot be set or changed by using 'Layer 2 online mode request' (Y3). For details, refer to the following.

☞ Page 161 Function mode (Un\G21)

Message transmit trigger completed (X4), Message transmit trigger request (Y4)

'Message transmit trigger request' (Y4) needs to be turned on to send data using 'Message transmit trigger flags' (Un\G8400 to Un\G8402) in the 11-bit CAN-ID Layer 2 message mode or 29-bit CAN-ID Layer 2 message mode.

When all the transmission data is stored in the send buffer, 'Message transmit trigger completed' (X4) turns on. (☞ Page 187 Message transmit trigger flags (Un\G8400 to Un\G8402))

After 'Message transmit trigger completed' (X4) is turned on, turn off 'Message transmit trigger request' (Y4). 'Message transmit trigger completed' (X4) will also be turned off.

Point

When the own node is in any of the following states, messages cannot be sent. If a send request is made continuously in this state, space would run out in the send buffer. In that case, 'Message transmit trigger completed' (X4) is not turned on until transmission becomes possible and available space is allowed in the send buffer.

- Offline state
- Bus off state
- Not connected to other devices capable of communication

RJ71CN91 error (XF), RJ71CN91 error clear request (YF)

'RJ71CN91 error' (XF) turns on if either bit 1, 3, 4, 5, 6, 8, or 15 of 'Error status' (Un\G29) turns on. (☞ Page 165 Error status (Un\G29))

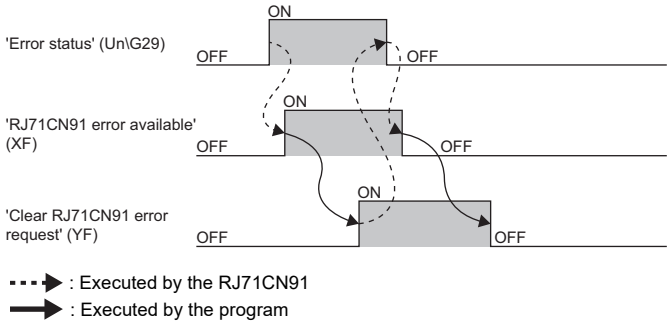
When 'RJ71CN91 error' (XF) turns on, remove the error cause as necessary and then turn on 'RJ71CN91 error clear request' (YF). Turning on 'RJ71CN91 error clear request' (YF) clears each bit in 'Error status' (Un\G29) and turns off 'RJ71CN91 error' (XF).

While 'RJ71CN91 error clear request' (YF) is turned on, an error other than a major error cannot be detected.

And, as long as a major error does not occur, any bit of 'Error status' (Un\G29) and 'RJ71CN91 error' (XF) do not turn on.

When 'RJ71CN91 error clear request' (YF) is turned off, error detection resumes.

All the bits of 'Error status' (Un\G29) automatically turn off when the error cause is removed. They are also turned off while 'RJ71CN91 error' (XF) is turned on, but if the error cause is not removed, they will be turned on again when 'RJ71CN91 error' (XF) is turned off.



Point

- When 'RJ71CN91 error clear request' (YF) is turned on, the ERR LED turns off, and the error information in the Module Diagnostics window is deleted. (Excluding when a major error occurs)
- While 'RJ71CN91 error clear request' (YF) is turned on, the ERR LED neither turns on nor flashes. And, error information is not displayed in the Module Diagnostics window or the event history. (Excluding when a major error occurs)
- The targets to be cleared by 'RJ71CN91 error clear request' (YF) also include an NMT error control failure.

NMT error control failure available (X10), NMT error control failure clear request (Y10)

When NMT error control failed (NMT error control failed in an NMT slave of the RJ71CN91), 'NMT error control failure available' (X10) turns on.

To clear errors of all CANopen nodes, write 0000H (initial value) in 'NMT error clear node' (Un\G400), and turn on 'NMT error control failure clear request' (Y10). NMT error control failure is cleared in all CANopen nodes, and 'NMT error control failure available' (X10) turns off.

To clear errors of a specific CANopen node, write the node ID of the target node in 'NMT error clear node (Un\G400), and turn on 'NMT error control failure clear request' (Y10). NMT error control failure is cleared in the target node, and if there is no other errors, 'NMT error control failure available' (X10) turns off.

While 'NMT error control failure clear request' (Y10) is turned on, an NMT error control failure in the target CANopen node is not detected.

Also, when no NMT error control failure occurs in other CANopen nodes, 'NMT error control failure available' (X10) does not turn on.

When 'NMT error control failure clear request' (Y10) is turned off, the detection of an NMT error control failure in the target CANopen node resumes.

For details on NMT error control failure, refer to the following.

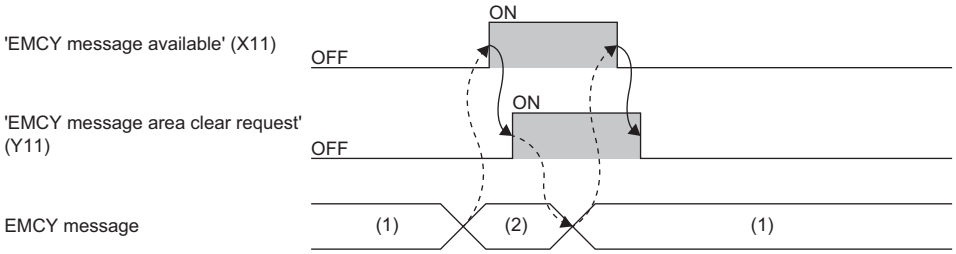
☞ Page 173 NMT error control status (Un\G401 to Un\G527)

Point

While 'NMT error control failure clear request' (Y10) is turned on, even if an NMT error control failure occurs in the target CANopen node, it is not displayed in the event history.

EMCY message available (X11), EMCY message area clear request (Y11)

When the RJ71CN91 receives an EMCY message from another node, 'EMCY message available' (X11) turns on.
The received EMCY message can be checked in 'EMCY message buffer' (Un\G750 to Un\G859). (☞ Page 175 EMCY message buffer (Un\G750 to Un\G859))
To clear all received EMCY messages, turn on 'EMCY message area clear request' (Y11). All EMCY messages will be then cleared and 'EMCY message available' (X11) will turn off.
While 'EMCY message area clear request' (Y11) is turned on, EMCY messages are not received and 'EMCY message available' (X11) will not turn on.
When 'EMCY message area clear request' (Y11) is turned off, the reception of EMCY messages resumes.
For details on EMCY messages, refer to the following.
☞ Page 52 EMCY
☞ Page 175 EMCY message buffer (Un\G750 to Un\G859)



(1) EMCY data not available
(2) EMCY data available
: Executed by the RJ71CN91
: Executed by the program

Point

While 'EMCY message area clear request' (Y11) is turned on, even if an EMCY message is received, it is not displayed in the event history.

Time stamp setting completed (X12), Time stamp setting request (Y12)

The Time stamp of the own node is set.
After setting Time stamp information in 'Time stamp' (Un\G50 to Un\G59), turn on 'Time stamp setting request' (Y12).
The Time stamp is set and 'Time stamp setting completed' (X12) is turned on. After 'Time stamp setting completed' (X12) is turned on, turn off 'Time stamp setting request' (Y12). Then, 'Time stamp setting completed' (X12) is turned off.

Time stamp info. available in buffer memory (X13), Time stamp read request (Y13)

The Time stamp of the own node is read.
To read the Time stamp, turn on 'Time stamp read request' (Y13).
The Time stamp will be stored in 'Time stamp' (Un\G50 to G59), and 'Time stamp information available in buffer memory' (X13) will be turned on.
After 'Time stamp information available in buffer memory' (X13) is turned on, turn off 'Time stamp read request' (Y13). Then, 'Time stamp information available in buffer memory' (X13) will be turned off.

Point

The Time stamp when 'Time stamp information available in buffer memory' (X13) is turned on will be stored in the buffer memory.

Command execution completed (X17), Command execution request (Y17)

These signals are used for execution of CIF.

To execute a command, set 'Command interface (CIF)' (Un\G1000 to Un\G1066), and then turn on 'Command execution request' (Y17). When the command execution is finished, 'Command execution completed' (X17) turns on. (☞ Page 166 Command interface (CIF) (Un\G1000 to Un\G1066))

After 'Command execution completed' (X17) is turned on, turn off 'Command execution request' (Y17). Then, 'Command execution completed' (X17) will be turned off.

Configuration save/factory default configuration restore completed/request (X1F)/(Y1F)

These signals are used to save the buffer memory settings to the flash ROM or restore the factory default settings.

To execute the request, set 'Save/restore configuration' (Un\G22), and then turn on 'Configuration save/factory default configuration restore request' (Y1F). When the request execution is finished, 'Configuration save/factory default configuration restore completed' (X1F) turns on. (☞ Page 162 Save/restore configuration (Un\G22))

After 'Configuration save/factory default configuration restore completed' (X1F) is turned on, turn off 'Configuration save/factory default configuration restore request' (Y1F). Then, 'Configuration save/factory default configuration restore completed' (X1F) will be turned off.

Point

'Configuration save/factory default configuration restore request' (Y1F) cannot be executed during the online mode of Layer 2 message mode. If it is executed, 'Error status' (Un\G29.5) of the buffer memory will be turned on. (☞ Page 165 Error status (Un\G29))

Note that the object dictionary is not saved/restored with 'Configuration save/factory default configuration restore completed' (X1F). To save/restore the object dictionary, use the following method.

- CANopen Configuration Tool (☞ Page 84 "CANopen Configuration" window, Page 86 "CANopen node settings" window)
- Object dictionary (☞ Page 212 Store parameters, Page 213 Restore default parameters)

Appendix 2 Buffer Memory

The buffer memory is used to exchange data between the RJ71CN91 and the CPU module, or between the RJ71CN91 and other nodes. Buffer memory values are defaulted when the CPU module is reset or the system is powered off and on.

List of buffer memory addresses

For the RJ71CN91, buffer memory areas used vary depending on the function mode.
This section lists the buffer memory addresses for each function mode.
○: Available, ×: Not available

Address (decimal)	Address (hexadecimal)	Name		Initial value		Read/ Write	Used/Not used	
				CANopen 405 mode	11-bit CAN-ID Layer 2 message mode, 29-bit CAN-ID Layer 2 message mode		CANopen 405 mode	11-bit CAN-ID Layer 2 message mode, 29-bit CAN-ID Layer 2 message mode
0 to 19	0H to 13H	System area						
20	14H	Data exchange control		0H	0H	Read/Write	○	○
21	15H	Function mode		405	11 or 29[*5]	Read/Write	○	○
22	16H	Save/restore configuration		0H	0H	Read/Write	○	○
23	17H	System area						
24	18H	Baud rate		250[*5]	250[*5]	Read/Write	○	○
25	19H	Communication status		80H	80H	Read	○	○
26	1AH	System area						
27	1BH	Node ID		127[*5]	0	Read/Write	○	×
28	1CH	System area						
29	1DH	Error status		0H	0H	Read	○	○
30 to 34	1EH to 22H	System area						
35	23H	CAN transmission error counter		0H	0H	Read	○	○
36	24H	CAN reception error counter		0H	0H	Read	○	○
37	25H	Baud rate display		2500	2500	Read	○	○
38	26H	Sampling point display		875	875	Read	○	○
39	27H	Buffer memory setting error display		0H	0H	Read	○	○
40 to 49	28H to 31H	System area						
50	32H	Time stamp	Producer/ Consumer	1	0	Read/Write	○	×
51	33H		Year	2014	0	Read/Write	○	×
52	34H		Month	8	0	Read/Write	○	×
53	35H		Day	1	0	Read/Write	○	×
54	36H		Hour	0	0	Read/Write	○	×
55	37H		Minute	0	0	Read/Write	○	×
56	38H		Second	0	0	Read/Write	○	×
57	39H		Day of the week	5	0	Read	○	×
58	3AH		Transmission interval	0	0	Read/Write	○	×
59	3BH		Daily correction	0[*5]	0	Read/Write	○	×
60 to 69	3CH to 45H	System area						
70	46H	NMT all nodes start delay time		500[*5]	0	Read/Write	○	×
71	47H	SDO timeout		500[*5]	0	Read/Write	○	×
72 to 399	48H to 18FH	System area						
400	190H	NMT error clear node		0H	0H	Read/Write	○	○
401 to 527	191H to 20FH	NMT error control status		0H	0H	Read	○	×
528 to 600	210H to 258H	System area						

Address (decimal)	Address (hexadecimal)	Name	Initial value		Read/ Write	Used/Not used	
			CANopen 405 mode	11-bit CAN-ID Layer 2 message mode, 29-bit CAN-ID Layer 2 message mode		CANopen 405 mode	11-bit CAN-ID Layer 2 message mode, 29-bit CAN-ID Layer 2 message mode
601 to 727	259H to 2D7H	NMT state	0H	0H	Read	○	×
728 to 749	2D8H to 2EDH	System area					
750 to 859	2EEH to 35BH	EMCY message buffer	0H	0H	Read/ Write*3	○	×
860 to 999	35CH to 3E7H	System area					
1000 to 1066	3E8H to 42AH	Command interface (CIF)	0H	0H	Read/Write	○	○
1067 to 5000	42BH to 1388H	System area					
5001 to 5042	1389H to 13B2H	Message slot specific error code list	0H	0H	Read/Write	×	○
5043 to 5999	13B3H to 176FH	System area					
6000 to 6167	1770H to 1817H	Pre-defined Layer 2 message configuration	0H	*1*5	Read/ Write*2	×	○
6168 to 8349	1818H to 209DH	System area					
8350 to 8352	209EH to 20A0H	Layer 2 RTR flags	0H	0H	Read	×	○
8353 to 8399	20A1H to 20CFH	System area					
8400 to 8402	20D0H to 20D2H	Message transmit trigger flags	0H	0H	Read/Write	×	○
8403 to 8449	20D3H to 2101H	System area					
8450 to 8477	2102H to 211DH	CPU module STOP transition message	0H	*4*5	Read/Write	×	○
8478 to 9999	211EH to 270FH	System area					

*1 For the initial value, check 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167). (☞ Page 181 Pre-defined Layer 2 message configuration (Un\G6000 to Un\G6167))

*2 Reading and writing is possible via the program. However, write access to these buffer memory areas is possible only in Layer 2 offline mode. (☞ Page 155 Layer 2 online mode status (X3), Layer 2 online mode request (Y3))

*3 Writing is possible only in Un\G750.

*4 For the initial value, check 'CPU module STOP transition message' (Un\G8450 to Un\G8477). (☞ Page 188 CPU module STOP transition message (Un\G8450 to Un\G8477))

*5 When the setting is saved in the flash ROM, the saved value is stored as the initial value.

The following describes the buffer memory assignment that varies per function mode.

• In CANopen 405 mode

Address (decimal)	Address (hexadecimal)	Name	Initial value	Read/Write
10000 to 11023	2710H to 2B0FH	RPDO	0H	Read
11024 to 12999	2B10H to 32C7H	System area		
13000 to 14023	32C8H to 36C7H	TPDO	0H	Read/Write
14024 to 32767	36C8H to 7FFFH	System area		

• In 11-bit CAN-ID Layer 2 message mode and 29-bit CAN-ID Layer 2 message mode

Address (decimal)	Address (hexadecimal)	Name	Initial value	Read/Write
10000 to 10293	2710H to 2835H	Receive/Transmit process data	*7*8	Read/Write*6
10294 to 32767	2836H to 7FFFH	System area		

*6 Reading and writing is possible via the program. However, for writing to the relevant buffer memory area, the following conditions exist. Writing is not possible under any condition other than the conditions below.
· When Layer 2 offline mode is active
· When Layer 2 online mode is active and the program data is written to the b8, b9, and b10 of "RTR/new/DLC" of the receive message and the "data byte" area of the send message
For details on the Layer 2 online mode and Layer 2 offline mode, refer to the following.
☞ Page 155 Layer 2 online mode status (X3), Layer 2 online mode request (Y3)

*7 For the initial value, check 'Receive/Transmit process data' (Un\G10000 to Un\G10293). (☞ Page 189 Receive/Transmit process data (Un\G10000 to Un\G10293))

*8 When the setting is saved in the flash ROM, the saved value is stored as the initial value.

Details of buffer memory areas

Common to CANopen 405 mode and Layer 2 message mode

The following describes the buffer memory areas common to the CANopen 405 mode, 11-bit CAN-ID Layer 2 message mode, and 29-bit CAN-ID Layer 2 message mode.

■Data exchange control (Un\G20)

Data for data exchange can be selected with 'Data exchange request' (Y1). This function is used if it is desired to update only the EMCY message buffer without changing any RPDO or TPDO.

To select only one data type as the data exchange target, set 1 for the bit for the data exchange target.

○: Available, ×: Not available

Bit	Description	Function mode	
		CANopen 405 mode	11-bit CAN-ID Layer 2 message mode, 29-bit CAN-ID Layer 2 message mode
b0 to b7	System area	—	—
b8	Specify whether to set 'RPDO' (Un\G10000 to Un\G11023) and 'TPDO' (Un\G13000 to Un\G14023) as data exchange targets. • 0: Not target*1 • 1: Target	○	×
b9 to b11	System area	—	—
b12	Specify whether to set 'EMCY message buffer' (Un\G750 to Un\G859) as a data exchange target. • 0: Not target*1 • 1: Target	○	×
b13 to b15	System area	—	—

*1 When all the bits of this area are 0, this area becomes a data exchange target.

■Function mode (Un\G21)

Set the RJ71CN91 function mode.

- 405: CANopen 405 mode
- 11: 11-bit CAN-ID Layer 2 message mode
- 29: 29-bit CAN-ID Layer 2 message mode

If a value other than above is set, 'Error status' (Un\G29.6) turns on. (☞ Page 144 Troubleshooting Using the Buffer Memory)

Point

To change this setting, save the buffer memory settings into the flash ROM, and then restart the module.

On how to save into the flash ROM, refer to the following.

☞ Page 158 Configuration save/factory default configuration restore completed/request (X1F)/(Y1F),Page 162 Save/restore configuration (Un\G22)

For how to restart the module, refer to the following.

☞ Page 154 Module restart completed (X2), Module restart request (Y2)

■Save/restore configuration (Un\G22)

This area is used to save the buffer memory settings to the flash ROM or restore the factory default settings.
To enable this setting, turn on 'Configuration save/factory default configuration restore request' (Y1F). (☞ Page 158 Configuration save/factory default configuration restore completed/request (X1F)/(Y1F))
When the processing is finished, 'Save/restore configuration' (Un\G22.0) and 'Save/restore configuration' (Un\G22.1) will automatically turn off, and 'Configuration save/factory default configuration restore completed' (X1F) will turn on.

Bit	Description
b0	1: Save configuration to flash ROM
b1	1: Restore factory default configuration (not saved to flash ROM)
b2 to b15	System area

The following table shows the buffer memory areas whose settings are to be saved and restored.
The save target of Save/restore configuration are buffer memory areas supported by the function mode used only. The restoration target of Save/restore configuration are all areas.

Address (Decimal)	Address (Hexadecimal)	Item	Remarks
21	15H	Function mode	—
24	18H	Baud rate	—
27	1BH	Node ID*2	—
59	3BH	Time stamp (daily correction)*2	—
70	46H	NMT all nodes start delay time*2	—
71	47H	SDO timeout*2	—
6000 to 6167	1770H to 1817H	Pre-defined Layer 2 message configuration*1	—
8450 to 8477	2102H to 211DH	CPU module STOP transition message*1	—
10000 to 10293	2710H to 2835H	Receive/Transmit process data*1	When the Layer 2 message mode is active, this area is partially saved. For details, refer to the following. ☞ Page 189 Receive/Transmit process data (Un\G10000 to Un\G10293)

*1 These areas are not save target in the CANopen 405 mode.
*2 These areas are not save target in the11-bit CAN-ID Layer 2 message mode or 29-bit CAN-ID Layer 2 message mode.

Point

If both bits 'Save/restore configuration' (Un\G22.0) and 'Save/restore configuration' (Un\G22.1) are turned on simultaneously, the buffer memory and flash ROM will be reset to factory default settings.
If only 'Save/restore configuration' (Un\G22.1) is turned on, the buffer memory settings will not be saved in the flash ROM when reset to the factory default settings. Therefore, save the buffer memory settings in the flash ROM in advance.
The settings saved with CANopen Configuration Tool have priority over the settings saved with this function. Note that the object dictionary is not saved/restored with 'Save/restore configuration' (Un\G22). To save/restore the object dictionary, use the following method.

- CANopen Configuration Tool (☞ Page 84 "CANopen Configuration" window, Page 86 "CANopen node settings" window)
- Object dictionary (☞ Page 212 Store parameters, Page 213 Restore default parameters)

■Baud rate (Un\G24)

Set the baud rate for communications. The current value for the baud rate can be checked with 'Baud rate display' (Un\G37). (☞ Page 165 Baud rate display (Un\G37))

- 10: 10kbps
- 20: 20kbps
- 50: 50kbps
- 100: 100kbps
- 125: 125kbps
- 250: 250kbps
- 500: 500kbps
- 800: 800kbps
- 1000: 1000kbps

If a value other than above is set, 'Error status' (Un\G29.6) turns on. (☞ Page 144 Troubleshooting Using the Buffer Memory)

Point

- Send buffer overflow may occur if data exchange is performed quickly when the baud rate is low or the bus load is high. (☞ Page 144 Troubleshooting Using the Buffer Memory)
- To change this setting when the 11-bit CAN-ID Layer 2 message mode or 29-bit CAN-ID Layer 2 message mode is active, bring the RJ71CN91 into the offline mode, change the setting, and then bring the RJ71CN91 into the online mode. When the RJ71CN91 transitions to the online mode, the setting is enabled.

To change this setting when the CANopen 405 mode is active, save the data in the buffer memory into the flash ROM, and then restart the RJ71CN91.*1

*1 For details on how to save into the flash ROM, refer to the following.
☞ Page 158 Configuration save/factory default configuration restore completed/request (X1F)/(Y1F)
☞ Page 162 Save/restore configuration (Un\G22)
For details on how to restart the module, refer to the following.
☞ Page 154 Module restart completed (X2), Module restart request (Y2)

A

■Communication status (Un\G25)

This area indicates the communication status of the RJ71CN91.

Bit	Description
b0	Stores the NMT state of the RJ71CN91 when the function mode is CANopen 405 mode. 0: The NMT state is other than Operational 1: The NMT state is Operational
b1	Error counter warning level 0: Except warning level (Below warning level, error passive, or bus off state) 1: Warning level The warning level is judged by whichever value that is greater from the value of 'CAN transmission error counter' (Un\G35) and 'CAN reception error counter' (Un\G36). The value is automatically set to 0 if it can no longer reach the warning level.
b2	System area
b3	
b4	
b5	
b6	Stores the node reset status of the RJ71CN91. 0: NMT service node reset or communication reset was not received. 1: The application was reset by NMT service node reset or communication reset. All settings not saved in the object dictionary will be cleared, and the factory default settings or the previously saved settings are restored. Write "0" to reset this bit. This bit is set to 0 in the beginning of the reset process.
b7	Stores the initialization status of the RJ71CN91. 0: Initialization completed 1: Initialization in progress The RJ71CN91 is initializing its internal data and the buffer memory. While this bit is on, only perform read from this area or 'Error status' (Un\G29), and do not write to any of the areas in the buffer memory. This bit turns on when restart of the RJ71CN91 is requested by 'Module restart request' (Y2). Always monitor this bit by the program since 'Error status' (Un\G29) does not indicate the error state.
b8	Stores the NMT state of the RJ71CN91. b8 = 0, b9 = 0: Stop b8 = 1, b9 = 0: Pre-operational b8 = 0, b9 = 1: Operational
b9	
b10	Stores the LSS master routine status of the RJ71CN91. 0: LSS master routine inactive 1: LSS master routine active This bit is turned on when the LSS master is searching and configuring LSS slaves.
b11	Stores the error state at start-up of the RJ71CN91. 0: No error 1: A mandatory slave failed to start, the NMT master stopped its start-up, or the NMT master restarted to resume NMT start-up. If this bit is on when all mandatory slaves are running, the NMT master may have been configured incorrectly. Check the settings of the NMT master.*1
b12	Stores the TIME message reception status of the RJ71CN91. 0: A TIME message was not received or this bit was reset. 1: A TIME message was received. (When the RJ71CN91 is set as a TIME consumer) Write "0" to reset this bit.
b13	Stores the start status of NMT slaves that are not mandatory slaves. 0: No error 1: Optional NMT slave failed to start. If this bit is 1 and b14 is 0, the NMT master needs to be reset to resume its start-up process that was stopped. If this bit is on when all optional slaves are running, the NMT master may have been configured incorrectly. Check the settings of the NMT master.
b14	Stores the start status of NMT slaves when the RJ71CN91 is in NMT start-up state. 0: No NMT slave start-up in progress 1: NMT slave start-up in progress This bit is turned on during NMT start-up, or when NMT slave is restarting due to an error occurred during NMT start-up.
b15	Stores the status of whether the RJ71CN91 is running as the NMT master or a NMT slave. 0: Module works as NMT slave 1: Module works as NMT master

*1 If the mandatory slave Boot time (index 1F89H) is set, check that the boot time is sufficient for the last mandatory slave to start. (☞ Page 224 Boot time)

■Error status (Un\G29)

This area indicates the error status of the RJ71CN91. For details, refer to the following.

☞ Page 144 Troubleshooting Using the Buffer Memory

■CAN transmission error counter (Un\G35)

This area stores the current value of the CAN transmission error counter.

The CAN transmission error counter counts up in increments of 8 if a transmission error is detected. The counter value decreases by 1 every time transmission succeeded.

Monitoring is performed to see if messages to be sent from the own node can be communicated in the network correctly.

When an error is detected in a sent message, or if ACK is not returned by other nodes, it is judged as a send error.

When an error is detected in a sent message, regardless of whether the own node is in the error active or error passive state, an error flag is sent to disable the message.

Stored value	Description
0 to 127	Error active state
96 to 127	Warning level
128 to 255	Error passive state
256	Bus off state

Point

'CAN transmission error counter' (Un\G35) can be checked in 'Communication status' (Un\G25) and 'Error status' (Un\G29). (☞ Page 164 Communication status (Un\G25), Page 144 Troubleshooting Using the Buffer Memory)

■CAN reception error counter (Un\G36)

This area stores the current value of the CAN reception error counter.

The CAN reception error counter counts up in increments of 8 if a reception error is detected. The counter value decreases by 1 every time reception succeeded.

By receiving all messages in the network except those sent from the own node, monitoring is performed to see if they are being communicated normally. When an error is detected in a received message, it is judged as a reception error.

When an error is detected in a received message, the action to take varies depending on whether the own node is in the error active or error passive state.

- Error active state: When an error is detected in a received message, an error flag is returned and the message is disabled.
- Error passive state: Even when an error is detected in a received message, an error flag is not returned.

Stored value	Description
0 to 127	Error active state
96 to 127	Warning level
128	Error passive state
256	Bus off state

Point

'CAN reception error counter' (Un\G36) can be checked in 'Communication status' (Un\G25) and 'Error status' (Un\G29). (☞ Page 164 Communication status (Un\G25), Page 144 Troubleshooting Using the Buffer Memory)

■Baud rate display (Un\G37)

This area indicates the current baud rate of the CAN Controller. (In units of 0.1kbps)

When the Layer 2 offline mode is active, the area indicates 0 because communication is not performed.

■Sampling point display (Un\G38)

This area indicates the current sampling point of the CAN Controller. (In units of 0.1%)

■Buffer memory setting error display (Un\G39)

This area stores the buffer memory address where an error occurs when a value out of the setting range is written to the buffer memory and 'Error status' (Un\G29.6) turns on. If multiple errors occur, the buffer memory address where the first error occurs is stored.

To reset this area, turn on 'RJ71CN91 error clear request' (YF).

■Command interface (CIF) (Un\G1000 to Un\G1066)

This area is used to access the object dictionary of a local node or network node. Access using commands are for SDO read/write and EMCY message communication. For how to run a command, refer to the following.

☞ Page 158 Command execution completed (X17), Command execution request (Y17)

Address	Description	
	Transmit message	Receive message
Un\G1000	Command code	Command execution result code
Un\G1001 to Un\G1066	Command parameter	Command parameter read back or detailed error information

Point

- The area will not be cleared after command execution. Execute a command of Display current parameter to check the previously written command parameters.
- Before write access to this area, check that FFFFH (CIF busy) is not stored in Un\G1000. If write access is attempted in the CIF busy state, an error message will be stored in the receive message.

• Transmit message

For transmit messages, the following commands are available.

○: Available, ×: Not available

Command	Command code	CANopen 405 mode	11-bit CAN-ID Layer 2 message mode, 29-bit CAN-ID Layer 2 message mode	Reference
SDO read	0004H	○	×	☞ Page 31 SDO read
SDO multi read	0008H	○	×	☞ Page 33 SDO multi read
SDO write	0002H	○	×	☞ Page 36 SDO write
SDO multi write	0006H	○	×	☞ Page 38 SDO multi write
EMCY message transmission	000AH	○	×	☞ Page 54 When sending a message manually
Display current parameter	0000H	○	○	Refer to this section.
"CIF busy" error clear/reset	FFFFH	○	○	Refer to this section.
Layer 2 message transmission	000CH	×	○	☞ Page 71 Data/RTR transmission via CIF

• Receive message (except error message)

A receive message in response to the executed command is stored. For details, refer to the section regarding each function.

• Receive message (error message)

If command execution is completed with an error, the error message is stored in the receive message.

The following table shows the details of the error message.

Address	Description
	Receive message (error message)
Un\G1000	Command execution result code • 000FH: Error message
Un\G1001	Error type For details, refer to the following table.
Un\G1002 to Un\G1066	Additional information For details, refer to the following table.

Information stored as additional information varies depending on the error type.

The following table lists the error types and error details, as well as whether additional information is available.

Error type	Name	Description	Additional information
0003H	SDO error	An SDO error occurred.	Available
000FH	Message slot error	An error occurred in CIF while the command is being executed.	Available
0064H	Unknown command	An unknown command was executed.	Not available
0F0FH	Invalid device status	Cannot execute the command.	Not available
B0FFH	Bus OFF	Cannot receive a CAN message in the bus off state.	Not available
FFFFH	Unauthorized access in the CIF busy state	'Command interface (CIF)' (Un\G1000 to Un\G1066) was altered by unauthorized write access in the CIF busy state.	Not available

The following table details the SDO error additional information.

Address	Description
Un\G1002	Node ID
Un\G1003 to Un\G1004	Error code (☞ Page 30 SDO abort codes) • 1003 (3EBH): Low word of the error code • 1004 (3ECH): High word of the error code
Un\G1005 to Un\G1066	System area

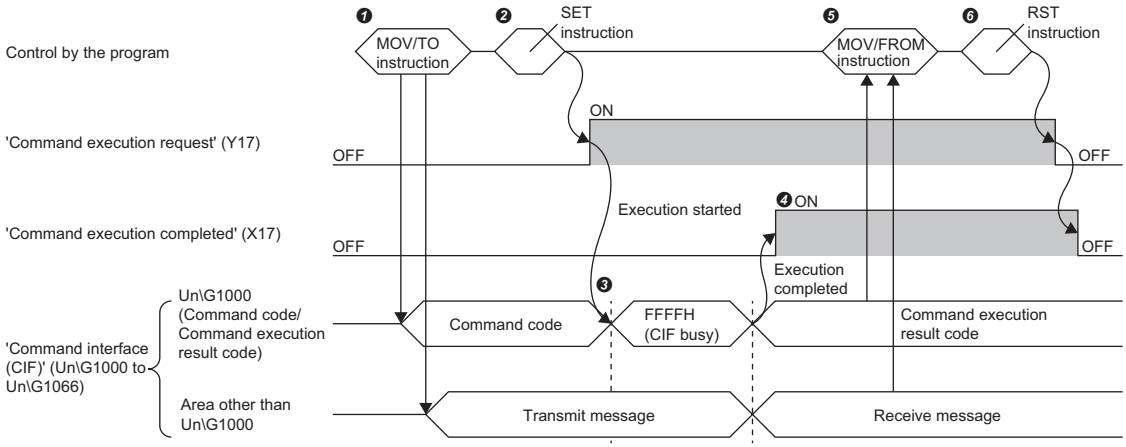
The following table lists additional information for message slot errors.

Address	Description	Cause	Action
Un\G1002	Error code (2002H (Timeout))	Timeout without execution of the message transmission. This error is caused by the following: • Communications fail due to the network disconnection or others. • CAN nodes which can communicate do not exist on the network. • Sending message cannot be performed due to the network overload.	• Check that the cable connection status or wiring method. • Check that CAN nodes which can communicate exist other than the own node. Check that the RJ71CN91 is in online mode when the communication destination is set to the RJ71CN91. • Reduce the network load. Or, use a high-priority CAN-ID.
Un\G1003 to Un\G1066	System area	—	—

• Display current parameter

This process displays the command parameters of the previously executed command in 'Command interface (CIF)' (Un\G1000 to Un\G1066). The command code is not displayed.

The following is the procedure for executing a command of Display current parameter.



❶ Set the transmit message in 'Command interface (CIF)' (Un\G1000 to Un\G1066) via the program.
❷ Turn on 'Command execution request' (Y17) via the program.
❸ FFFFH (CIF busy) is stored in Un\G1000 while the command is being processed.
❹ When the command execution is finished, 'Command execution completed' (X17) turns on.
❺ Then, 0000H (Completed) is stored in Un\G1000.
The receive message is stored in an area other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066).
Retrieve the receive message via the program to check the command parameters of the previously executed command.
❻ After the command (Display current parameter) is completed, turn off 'Command execution request' (Y17) via the program. When 'Command execution request' (Y17) is turned off, 'Command execution completed' (X17) also turns off.

For the current parameter display, the buffer memory areas are assigned as follows.

Address	Description	
	Transmit message	**Receive message**
Un\G1000	Command code • 0000H: Display current parameter	Command execution result code • 0000H: Completed
Un\G1001 to Un\G1066	System area	Command parameters of the previously executed command

• Clear error of unauthorized access in the CIF busy state

If unauthorized access occurs in the CIF busy state, no command can be executed. Before executing any command, execute "CIF busy" error clear/reset to clear the errors.

The procedure for executing "CIF busy" error clear/reset is the same as the procedure for executing a command of Display current parameter.

For "CIF busy" error clear/reset, receive messages are not stored in areas other than Un\G1000 of 'Command interface (CIF)' (Un\G1000 to Un\G1066).

For "CIF busy" error clear/reset, the buffer memory areas are assigned as follows.

Address	Description	
	Transmit message	**Receive message**
Un\G1000	Command code • FFFFH: "CIF busy" error clear/reset	Command execution result code • 0000H: Completed
Un\G1001 to Un\G1066	System area	System area

CANopen 405 mode

This section describes the details of the buffer memory areas used only in CANopen 405 mode.

■Node ID (Un\G27)

Set the node ID. The setting range is 1 to 127.

When the CANopen 405 mode is active, if a value outside the specified range is written, 'Error status' (Un\G29.6) turns on.

For details on the error status, refer to the following.

☞ Page 165 Error status (Un\G29)

To change this setting, save the buffer memory settings into the flash ROM, and then restart the module.*1

*1 For details on how to save into the flash ROM, refer to the following.

☞ Page 158 Configuration save/factory default configuration restore completed/request (X1F)/(Y1F)

☞ Page 162 Save/restore configuration (Un\G22)

For details on how to restart the module, refer to the following.

☞ Page 154 Module restart completed (X2), Module restart request (Y2)

■Time stamp (Un\G50 to Un\G59)

A TIME consumer CANopen node receives a TIME message to synchronize the time data with the TIME producer.

When the CPU module is powered off and on or reset, the clock data of the RJ71CN91 is reset to the initial value and the time counter stops.

Set whether to run the RJ71CN91 as a TIME consumer or TIME producer in 'Producer/Consumer' (Un\G50).

When the RJ71CN91 is setup as a TIME consumer, the time counter will start after receiving the first TIME message.

When the RJ71CN91 is setup as a TIME producer, the time counter will start after setup of this area and turning on 'Time stamp setting request' (Y12). (☞ Page 157 Time stamp setting completed (X12), Time stamp setting request (Y12))

A TIME message will be produced only if the RJ71CN91 is the active NMT master and in NMT state Operational or Pre-operational.

The RJ71CN91 counts the time in seconds. 'Communication status' (Un\G25.12) turns on when a TIME message is received. (☞ Page 164 Communication status (Un\G25))

The clock time of the RJ71CN91 will be read from the clock data when 'Time stamp read request' (Y13) is turned on, and the clock time will be written to the clock data when 'Time stamp setting request' (Y12) is turned on. (☞ Page 157 Time stamp setting completed (X12), Time stamp setting request (Y12))

When 'Time stamp setting request' (Y12) turns on, the clock time data in the RJ71CN91 is checked. If the clock data is out of the setting range, 'Error status' (Un\G29.6) turns on. (☞ Page 144 Troubleshooting Using the Buffer Memory)

When 'Time stamp setting request' (Y12) turns on, the clock time data in the RJ71CN91 is checked. If the value for the clock time data is not valid as a date and time, 'Error status' (Un\G29.6) turns on. (☞ Page 144 Troubleshooting Using the Buffer Memory)

If the date and time is outside the valid range for this area, the default value is set. (☞ Page 51 TIME)

The clock data will be enabled after 'Time stamp setting request' (Y12) turns on.

To synchronize the clock time with the CPU module, use the DATERD and DATEWR instructions to read clock data from the CPU module and write it to the RJ71CN91.

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

Point

- The RJ71CN91 does not support daylight saving time.
- The actual clock time data will be delayed by the time required for internal processing of the RJ71CN91 and data communication processing in the network.
- Leap years are automatically adjusted.

Address	Name		Setting range
Un\G50	Time stamp	Producer/Consumer	Directly access the consumer/producer bit of the Time COB-ID in the object dictionary.*3 0: Time stamp disabled 1: Consumer 2: Producer*1 3: Producer*1/Consumer
Un\G51		Year	2000 to 2079*2
Un\G52		Month	1 to 12*2
Un\G53		Day	1 to 31*2 Do not set dates that do not exist, such as "February 30".
Un\G54		Hour	0 to 23 (24-hour notation)*2
Un\G55		Minute	0 to 59*2
Un\G56		Second	0 to 59*2
Un\G57		Day of the week	0: Sunday, 1: Monday, 2: Tuesday, 3: Wednesday, 4: Thursday, 5: Friday, 6: Saturday*2 The Day of the week will be set automatically during the setup of the clock data.
Un\G58		Transmission interval	Set the interval (minutes) for sending TIME messages from the RJ71CN91.*2 0: Do not send TIME messages 1 to 1440: 1 to 1440 minutes (24 hours)
Un\G59		Daily correction	Set the time (s/day) to correct the clock time of the RJ71CN91 periodically. -60 to +60

*1 A TIME message will be produced only if the RJ71CN91 is the active NMT master.

*2 If the RJ71CN91 is a TIME consumer, this value is disabled.

*3 If an NMT reset occurs, the set values of the object dictionary that have not been saved yet are written over by their initial values. For this reason, operation different from the operation before an NMT reset occurred may be performed. If the set values of the object dictionary are written over by different values, create a program for writing the set values again and re-configuring the object directory with 'Time stamp setting request' (Y12).
Program example (When 2 (Producer) is set in 'Time stamp' (Un\G50.0) with the start I/O number of the RJ71CN91 at 0)
☞ Page 139 Program example

■NMT all nodes start delay time (Un\G70)

When the NMT master is used, Communication reset or Remote node start runs in accordance with the parameter setting when the NMT master starts.

Set the minimum transmission interval (ms) for node control messages in milliseconds in this area so that NMT slaves that take some time to start can recognize the remote node start.

The setting range is 0 to 65535.

Point

For details on NMT, refer to the following.

☞ Page 19 NMT

■SDO timeout (Un\G71)

Set the timeout time (ms) for SDO communication.

The setting range is 50 to 32767.

Point

For details on SDO, refer to the following.

☞ Page 29 SDO

■NMT error clear node (Un\G400)

Specify the node ID of a CANopen node, where NMT error control failed, to clear errors in that node.

For details on how to clear errors in a CANopen node where NMT error control failed, refer to "NMT error control failure clear request (Y10)". (☞ Page 156 NMT error control failure available (X10), NMT error control failure clear request (Y10))

For details on CANopen node errors, refer to "NMT error control status (Un\G401 to Un\G527)". (☞ Page 173 NMT error control status (Un\G401 to Un\G527))

■NMT error control status (Un\G401 to Un\G527)

This area indicates the Node guarding and Heartbeat status.

Address	Description
Un\G401	CANopen node 1 status
Un\G402	CANopen node 2 status
⋮	⋮
Un\G526	CANopen node 126 status
Un\G527	CANopen node 127 status

The following table lists the bits in this area, and describes what each bit means when it is turned on.

○: Enabled, ×: Disabled

Bit	Item	Description	Bit enabled/disabled by node		
			Node guarding		Heartbeat
			NMT master	NMT slave	Consumer
b0	Node guarding enabled	Node guarding is in progress. (This bit turns on each time node guarding starts. Then, it turns off when node guarding is stopped or a node guarding timeout is detected.)	○	○	×
b1	Heartbeat enabled	Heartbeat is in progress. (This bit turns on each time heartbeat starts. Then, it turns off when heartbeat is stopped or a heartbeat timeout is detected.)	×	×	○
b2	Node guarding message error detected	There was no polling/response within the guard time. Or, an invalid response is received.	○	○	×
b3	Node guarding timeout detected (NMT master)	There was no response from an NMT slave within the node life time.	○	×	×
b4	NMT startup failure	Node guarding or heartbeat did not start when an NMT slave booted.	○	×	○ (Enabled for the active NMT master only)
b5	NMT state error detected	The monitoring target node was not in the expected NMT state.	○	×	○
b6	Node guarding timeout detected (NMT slave)	There was no polling from the NMT master within the life time.	×	○	×
b7	Heartbeat timeout detected	A heartbeat message was not received within the heartbeat time.	×	×	○
b8 to 15	System area		×	×	×

If any of bits b2 to b7 in this area turns on, 'NMT error control failure available' (X10) turns on.

While heartbeat is being used, if b1 repeatedly turns on and off, or b7 turns on, heartbeat may not be operating properly. Check the settings again as follows.

- Set the value for the consumer heartbeat time so that it is sufficiently large.
- Set the value for the consumer heartbeat time to be sufficiently larger than the value for the producer heartbeat time.

With node guarding in use, if b0 turns on and off repeatedly, or if one of b2, b3, and b6 turns on, node guarding may not be running normally. Check the settings again as follows.

- Set the value for the guard time so that it is sufficiently large.
- Set the value for the life time to be sufficiently larger than the value for the guard time.

For details on how to clear errors in a CANopen node where NMT error control failed, refer to "NMT error control failure clear request (Y10)". (☞ Page 156 NMT error control failure available (X10), NMT error control failure clear request (Y10))

■NMT state (Un\G601 to Un\G727)

This area indicates the NMT state of each CANopen node. (☞ Page 19 NMT state)

The NMT state of the own node is always stored.

For the NMT states of other nodes, the states obtained as follows are stored. If the state is not obtained, 00H (NMT state unknown) is stored.

Status	Action
When the own node is the active NMT master	The latest NMT state among the following NMT states is stored. • The NMT state in which the own node instructed the target nodes through the NMT startup function or node control • The NMT state obtained from the target node through node guarding or heartbeat consumer
When the own node is not the active NMT master	The NMT state obtained from the target node through heartbeat consumer is stored.

For details on heartbeat and node guarding, refer to the following.

☞ Page 59 Heartbeat

☞ Page 57 Node Guarding

For details on the value displayed in this area, refer to the following.

☞ Page 222 Request NMT

Point

- To obtain the NMT state of the target node from this area when own node is the active NMT master, check that node guarding or heartbeat consumer is in progress for the target node in 'NMT error control status' (Un\G401 to Un\G527). (For the NMT error control status, check that node guarding is enabled or that heartbeat is enabled.)
- Depending on the timing of communication, the NMT state of the target node in this area may become temporarily inconsistent.

Address	Description
Un\G601	CANopen node 1
Un\G602	CANopen node 2
⋮	⋮
Un\G726	CANopen node 126
Un\G727	CANopen node 127

■EMCY message buffer (Un\G750 to Un\G859)

A maximum of 22 received EMCY messages are stored.

The buffer memory area is divided into the 11 stack buffer (Un\G750 to Un\G804) and 11 ring buffer (Un\G805 to Un\G859).

The stack buffer will store the first 11 EMCY messages received after power-on or after the EMCY message buffer was cleared the last time.

The ring buffer will store the next eleven EMCY messages.

EMCY messages delivered thereafter will be added to the end of the ring buffer, and the oldest message at the top will be deleted.

The stack buffer will not be overwritten.

To ensure that the EMCY data are handled in a consistent way, it is necessary to refresh the data before reading the EMCY data.

(☞ Page 161 Data exchange control (Un\G20), Page 153 Data exchange completed (X1), Data exchange request (Y1))

To clear the entire buffer area, turn on 'EMCY message area clear request' (Y11).

The following table lists the EMCY message buffer addresses.

Address	Name	Description		
		Buffer memory area	Upper byte	Lower byte
Un\G750	Node ID*1	Stack buffer	The node-ID number of the node which sent the EMCY message to the network is displayed.	
Un\G751	EMCY data*1*3		Emergency error code	
Un\G752			Byte 0 of manufacturer-specific error code	Error register
Un\G753			Byte 2 of manufacturer-specific error code	Byte 1 of manufacturer-specific error code
Un\G754			Byte 4 of manufacturer-specific error code	Byte 3 of manufacturer-specific error code
:	:		:	:
Un\G800	Node ID*2		The node-ID number of the node which sent the EMCY message to the network is displayed.	
Un\G801	EMCY data*2*3		Emergency error code	
Un\G802			Byte 0 of manufacturer-specific error code	Error register
Un\G803			Byte 2 of manufacturer-specific error code	Byte 1 of manufacturer-specific error code
Un\G804			Byte 4 of manufacturer-specific error code	Byte 3 of manufacturer-specific error code
Un\G805	Node ID*1	Ring buffer	The node-ID number of the node which sent the EMCY message to the network is displayed.	
Un\G806	EMCY data*1*3		Emergency error code	
Un\G807			Byte 0 of manufacturer-specific error code	Error register
Un\G808			Byte 2 of manufacturer-specific error code	Byte 1 of manufacturer-specific error code
Un\G809			Byte 4 of manufacturer-specific error code	Byte 3 of manufacturer-specific error code
:	:		:	:
Un\G855	Node ID*2		The node-ID number of the node which sent the EMCY message to the network is displayed.	
Un\G856	EMCY data*2*3		Emergency error code	
Un\G857			Byte 0 of manufacturer-specific error code	Error register
Un\G858			Byte 2 of manufacturer-specific error code	Byte 1 of manufacturer-specific error code
Un\G859			Byte 4 of manufacturer-specific error code	Byte 3 of manufacturer-specific error code

*1 The oldest message is stored.

*2 The latest message is stored.

*3 For details on emergency error codes, manufacturer-specific error codes, and error register, refer to the following.

☞ Page 52 Emergency error codes

☞ Page 53 Manufacturer-specific error code

☞ Page 211 Error register

■RPDO (Un\G10000 to Un\G11023), TPDO (Un\G13000 to Un\G14023)

Configure these area to run RPDOs and TPDOs for data transfer.

To ensure that the RJ71CN91 can handle send/receive data in a consistent way, it is necessary to turn on 'Data exchange request' (Y1) to exchange data before reading and after writing the PDO transmit/receive data. The data exchange control signal ensures, by internal buffer exchange, which data in the CPU module will be transmitted simultaneously by the same PDO. (☞ Data exchange control (Un\G20))

Point

The data will only be exchanged when the RJ71CN91 is in NMT state Operational.

• Direct receive buffer memory access to the CiA-405 Object

Use the FROM or MOV instruction to read data from 'RPDO' (Un\G10000 to Un\G11023).

The initial setting of RPDO mapping is assigned to unsigned 16-bit objects. To change this setting, execute SDO or use CANopen configuration software.

Point

If the data is written to object entry (index-subindex) assigned to 'RPDO' (Un\G10000 to Un\G11023) using SDO, the data is reflected to the buffer memory area as well. If the data is written to the buffer memory area, however, the data is not reflected to the object entry.

The following table shows direct receive buffer memory access for unsigned and signed 8-bit objects.

Index		Subindex	Buffer memory address	
Unsigned 8-bit object	Signed 8-bit object			
A4C0H	A480H	01H	Un\G10000	Lower 8 bits
		02H		Upper 8 bits
		03H	Un\G10001	Lower 8 bits
		04H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G10126	Lower 8 bits
		FEH		Upper 8 bits
A4C1H	A481H	01H	Un\G10127	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G10253	Lower 8 bits
		FEH		Upper 8 bits
A4C2H	A482H	01H	Un\G10254	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G10380	Lower 8 bits
		FEH		Upper 8 bits
A4C3H	A483H	01H	Un\G10381	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G10507	Lower 8 bits
		FEH		Upper 8 bits
A4C4H	A484H	01H	Un\G10508	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G10634	Lower 8 bits
		FEH		Upper 8 bits
A4C5H	A485H	01H	Un\G10635	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G10761	Lower 8 bits
		FEH		Upper 8 bits

Index		Subindex	Buffer memory address	
Unsigned 8-bit object	Signed 8-bit object			
A4C6H	A486H	01H	Un\G10762	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G10888	Lower 8 bits
		FEH		Upper 8 bits
A4C7H	A487H	01H	Un\G10889	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G11015	Lower 8 bits
		FEH		Upper 8 bits
A4C8H	A488H	01H	Un\G11016	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		0FH	Un\G11023	Lower 8 bits
		10H		Upper 8 bits

The following table shows direct receive buffer memory access for unsigned and signed 16-bit objects.

Index		Subindex	Buffer memory address
Unsigned 16-bit object	Signed 16-bit object		
A580H	A540H	01H	Un\G10000
		02H	Un\G10001
		⋮	⋮
		FEH	Un\G10253
A581H	A541H	01H	Un\G10254
		⋮	⋮
		FEH	Un\G10507
A582H	A542H	01H	Un\G10508
		⋮	⋮
		FEH	Un\G10761
A583H	A543H	01H	Un\G10762
		⋮	⋮
		FEH	Un\G11015
A584H	A544H	01H	Un\G11016
		⋮	⋮
		08H	Un\G11023

A

The following table shows direct receive buffer memory access for float 32-bit objects, and unsigned and signed 32-bit objects.

Index			Subindex	Buffer memory address
Float 32-bit object	Unsigned 32-bit object	Signed 32-bit object		
A6C0H	A680H	A640H	01H	Un\G10000
				Un\G10001
			02H	Un\G10002
				Un\G10003
			⋮	⋮
			FEH	Un\G10506
				Un\G10507
A6C1H	A681H	A641H	01H	Un\G10508
				Un\G10509
			⋮	⋮
			FEH	Un\G11014
				Un\G11015
A6C2H	A682H	A642H	01H	Un\G11016
				Un\G11017
			⋮	⋮
			04H	Un\G11022
				Un\G11023

• Direct transmit buffer memory access to the CiA-405 Object

Use the TO or MOV instruction to write data to 'TPDO' (Un\G13000 to Un\G14023).

The initial setting of TPDO mapping is assigned to unsigned 16-bit objects. To change this setting, execute SDO or use CANopen configuration software.

Point

If 'TPDO' (Un\G13000 to Un\G14023) is mapped to TPDO of the object dictionary, data written to the buffer memory area is reflected to the mapped object entry. The data is not reflected to the object entry that is not mapped. If data is written to object entry using SDO, the data is not reflected to the buffer memory area.

The following table shows direct transmit buffer memory access for unsigned and signed 8-bit objects.

Index		Subindex	Buffer memory address	
Unsigned 8-bit object	Signed 8-bit object			
A040H	A000H	01H	Un\G13000	Lower 8 bits
		02H		Upper 8 bits
		03H	Un\G13001	Lower 8 bits
		04H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G13126	Lower 8 bits
		FEH		Upper 8 bits
A041H	A001H	01H	Un\G13127	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G13253	Lower 8 bits
		FEH		Upper 8 bits
A042H	A002H	01H	Un\G13254	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G13380	Lower 8 bits
		FEH		Upper 8 bits
A043H	A003H	01H	Un\G13381	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G13507	Lower 8 bits
		FEH		Upper 8 bits
A044H	A004H	01H	Un\G13508	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G13634	Lower 8 bits
		FEH		Upper 8 bits
A045H	A005H	01H	Un\G13635	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G13761	Lower 8 bits
		FEH		Upper 8 bits
A046H	A006H	01H	Un\G13762	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G13888	Lower 8 bits
		FEH		Upper 8 bits
A047H	A007H	01H	Un\G13889	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		FDH	Un\G14015	Lower 8 bits
		FEH		Upper 8 bits
A048H	A008H	01H	Un\G14016	Lower 8 bits
		02H		Upper 8 bits
		⋮	⋮	
		0FH	Un\G14023	Lower 8 bits
		10H		Upper 8 bits

The following table shows direct transmit buffer memory access for unsigned and signed 16-bit objects.

Index		Subindex	Buffer memory address
Unsigned 16-bit object	Signed 16-bit object		
A100H	A0C0H	01H	Un\G13000
		02H	Un\G13001
		⋮	⋮
		FEH	Un\G13253
A101H	A0C1H	01H	Un\G13254
		⋮	⋮
		FEH	Un\G13507
A102H	A0C2H	01H	Un\G13508
		⋮	⋮
		FEH	Un\G13761
A103H	A0C3H	01H	Un\G13762
		⋮	⋮
		FEH	Un\G14015
A104H	A0C4H	01H	Un\G14016
		⋮	⋮
		08H	Un\G14023

The following table shows direct transmit buffer memory access for float 32-bit objects, and unsigned and signed 32-bit objects.

Index			Subindex	Buffer memory address
Float 32-bit object	Unsigned 32-bit object	Signed 32-bit object		
A240H	A200H	A1C0H	01H	Un\G13000
				Un\G13001
			02H	Un\G13002
				Un\G13003
			⋮	⋮
			FEH	Un\G13506
				Un\G13507
A241H	A201H	A1C1H	01H	Un\G13508
				Un\G13509
			⋮	⋮
			FEH	Un\G14014
				Un\G14015
A242H	A202H	A1C2H	01H	Un\G14016
				Un\G14017
			⋮	⋮
			04H	Un\G14022
				Un\G14023

Layer 2 message mode

The following describes the details of the buffer memory areas used for the 11-bit CAN-ID Layer 2 message mode and 29-bit CAN-ID Layer 2 message mode.

■Message slot specific error code list (Un\G5001 to Un\G5042)

This area stores error codes of each message slot.

Address	Description
Un\G5001	Message 1 error code
Un\G5002	Message 2 error code
⋮	⋮
Un\G5041	Message 41 error code
Un\G5042	Message 42 error code

The following table lists and describes the error codes stored in each message slot.

Error code	Description	Cause	Action
0000H	No error code	—	—
2000H	Receive buffer overflow	A message was not received since the receive processing of the RJ71CN91 failed.	Take the following actions to extend the receiving interval of the message. • Increase the data size of send data. • Extend the transmission interval. • Set the baud rate lower in the network.

■Pre-defined Layer 2 message configuration (Un\G6000 to Un\G6167)

This area is used to determine if the corresponding Layer 2 message in 'Receive/Transmit process data' (Un\G10000 to Un\G10293) is a transmit or receive message.

Address	Description		Initial value*1
Un\G6000	Layer 2 message 1 parameter A	Layer 2 message 1 parameter	FFFFH
Un\G6001	Layer 2 message 1 parameter B		FFFFH
Un\G6002	Layer 2 message 1 parameter C		0000H
Un\G6003	Layer 2 message 1 parameter D		0000H
Un\G6004	Layer 2 message 2 parameter A	Layer 2 message 2 parameter	FFFFH
Un\G6005	Layer 2 message 2 parameter B		FFFFH
Un\G6006	Layer 2 message 2 parameter C		0000H
Un\G6007	Layer 2 message 2 parameter D		0000H
⋮	⋮	⋮	⋮
Un\G6160	Layer 2 message 41 parameter A	Layer 2 message 41 parameter	FFFFH
Un\G6161	Layer 2 message 41 parameter B		FFFFH
Un\G6162	Layer 2 message 41 parameter C		0000H
Un\G6163	Layer 2 message 41 parameter D		0000H
Un\G6164	Layer 2 message 42 parameter A	Layer 2 message 42 parameter	FFFFH
Un\G6165	Layer 2 message 42 parameter B		FFFFH
Un\G6166	Layer 2 message 42 parameter C		0000H
Un\G6167	Layer 2 message 42 parameter D		0000H

*1 Initial value when the CPU module is powered off and on or reset

Point

- Layer 2 message n parameters A and B define if the corresponding Layer 2 message in 'Receive/Transmit process data' (Un\G10000 to Un\G10293) is used as a transmit or receive message. If Layer 2 message n is not used, set Layer 2 message n parameters A and B to FFFFH to disable the message slot.
- If an invalid value is set in this area, 'Error status' (Un\G29.6) turns on, and the address where this error occurred is stored in 'Buffer memory setting error display' (Un\G39).
- Change these settings during offline mode. When the RJ71CN91 transitions to the online mode, the settings are enabled.

• Pre-defined Layer 2 transmit messages

The meaning of the parameters A to D for a transmit message is as follows.

Set Layer 2 message n parameter A to FFFFH, and set Layer 2 message n parameter B to 7FFFH, 6FFFH or 5FFFH to assign the buffer of this area to the transmit message.

Parameter	Description	Initial value*1
Layer 2 message n parameter A*2	Fixed to FFFFH	FFFFH
Layer 2 message n parameter B*2	7FFFH: Auto RTR response 6FFFH: Manual RTR response 5FFFH: RTR handling disabled FFFFH: Message disabled	FFFFH
Layer 2 message n parameter C	Transmission type	0000H
Layer 2 message n parameter D	Cycle time in (10ms)	0000H

*1 Initial value when the CPU module is powered off and on or reset

*2 To disable the message slot, set Layer 2 message n parameters A and B to FFFFH.

Setting data

■Layer 2 message n parameter A, B

- When this is set to Auto RTR response, the RJ71CN91 will automatically respond to an RTR if the 11-bit CAN-ID and 29-bit CAN-ID set in 'Receive/Transmit process data' (Un\G10000 to Un\G10293) match the CAN-ID of the RTR message. The result of RTR message reception is not stored in 'Layer 2 RTR flags' (Un\G8350 to Un\G8352).
- When this is set to Manual RTR response, the RJ71CN91 does not respond to RTR automatically. The result of RTR message reception is stored in 'Layer 2 RTR flags' (Un\G8350 to Un\G8352).
- If this is set to RTR handling disabled, the RJ71CN91 does not receive all RTR messages that match the CAN-ID of a message slot where RTR handling is disabled.

Point

Up to 28 message slots can be set to Auto RTR response or Manual RTR response (parameter B = 7FFFH or 6FFFH).

The 29th message slot and message slots thereafter are forcibly set as RTR handling disabled (parameter B = 5FFFH). In addition, 'Error status' (Un\G29.6) is turned on.

■Layer 2 message n parameter C

The transmission type defines under which conditions a transmit message is sent.

Setting value	Description*1*2
0000H	Event • Transmit each time 'Data exchange request' (Y1) is made.
0001H	Event (COS) • Transmit each time 'Data exchange request' (Y1) is made. • Do not transmit if there is no data change.
0002H	Cycle • Transmit at specified time intervals. • The time interval is set with Layer 2 message n parameter D.
0003H	Cycle (COS) • Transmit at specified time intervals. • Do not transmit if there is no data change. • The time interval is set with Layer 2 message n parameter D.
0004H	Transmission trigger only • Transmit only by a message transmit trigger request or Auto RTR response.

*1 Regardless of the send message transmission type setting, 'Data exchange request' (Y1) is required for the refresh of send data.

*2 Regardless of the send message transmission type setting, data is sent in any of the following cases.

· Transition is made to the Layer 2 online mode. (☞ Page 155 Layer 2 online mode status (X3), Layer 2 online mode request (Y3))

· A send request is made with 'Message transmit trigger request' (Y4). (☞ Page 155 Message transmit trigger completed (X4), Message transmit trigger request (Y4))

· A remote transmission request is received when parameter B is set to "Auto RTR response".

■Layer 2 message n parameter D

This is only enabled if Layer 2 message n parameter C is set to 0002H or 0003H.

Specify the interval for message transmission in units of 10ms. Setting this parameter to 0000H sets the interval to 0001H (10ms) automatically.

- When the Layer 2 message n parameter C is set to 0002H, a message is transmitted every time regardless of whether data is changed or not.
- When the Layer 2 message n parameter C is set to 0003H, a message is transmitted only if data is changed from the previous transmission.
- A parameter error occurs if the cycle time is too short against the baud rate, whereby operation is unable to be performed with the set cycle time.

• Pre-defined Layer 2 receive messages

The meaning of the parameters A to D for a receive message is as follows.

Parameter	Description		Initial value*1
Layer 2 message n parameter A*2	Reception CAN-ID	Low word	FFFFH
Layer 2 message n parameter B*2		High word	FFFFH
Layer 2 message n parameter C	Reception ID filter bit mask	Low word	0000H
Layer 2 message n parameter D		High word	0000H

*1 Initial value when the CPU module is powered off and on or reset

*2 To disable the message slot, set Layer 2 message n parameters A and B to FFFFH.

Setting data

■Layer 2 message n parameter A, B

Set the 11-bit/29-bit CAN-ID of messages to be received in the message slot of the corresponding Layer 2 message.

■Layer 2 message n parameter C, D

Set the filter for the CAN-ID that was set using Layer 2 message n parameters A and B.

If this is set to 00000000H, whether the CAN-ID of a received message matches the CAN-ID set in Layer 2 message n parameters A and B is verified.

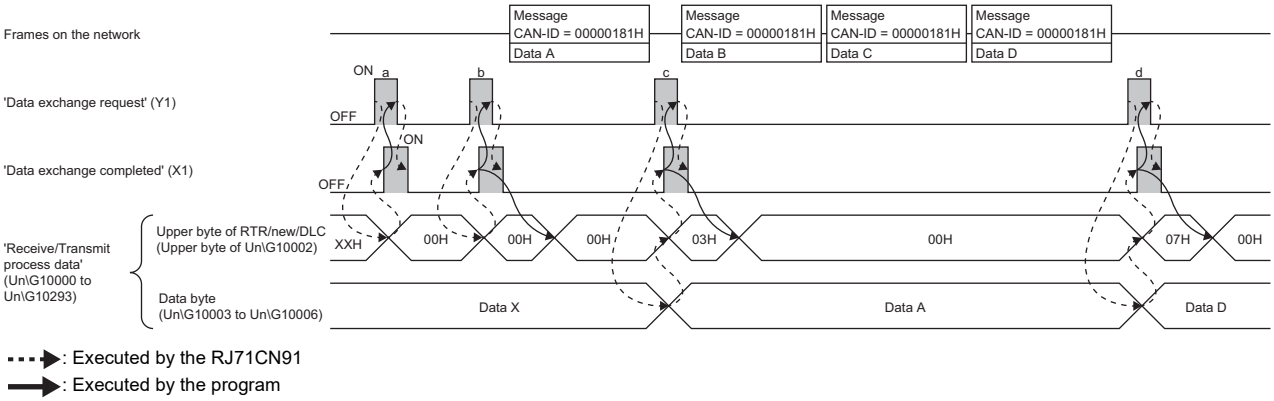
If either bit is turned on, the bit turned on is ignored when verifying whether the CAN-ID of a received message matches the CAN-ID set in Layer 2 message n parameters A and B.

Ex.

When Layer 2 message 1 parameters A and B are 00000181H, and Layer 2 message 1 parameters C and D are 00000000H

Only receive messages whose CAN-ID is 00000181H are stored in the message slot of Layer 2 message 1 in 'Receive/Transmit process data' (Un\G10000 to Un\G10293).

Each time data exchange (a, b, c, d) is performed, the upper byte of RTR/new/DLC is cleared by the program.



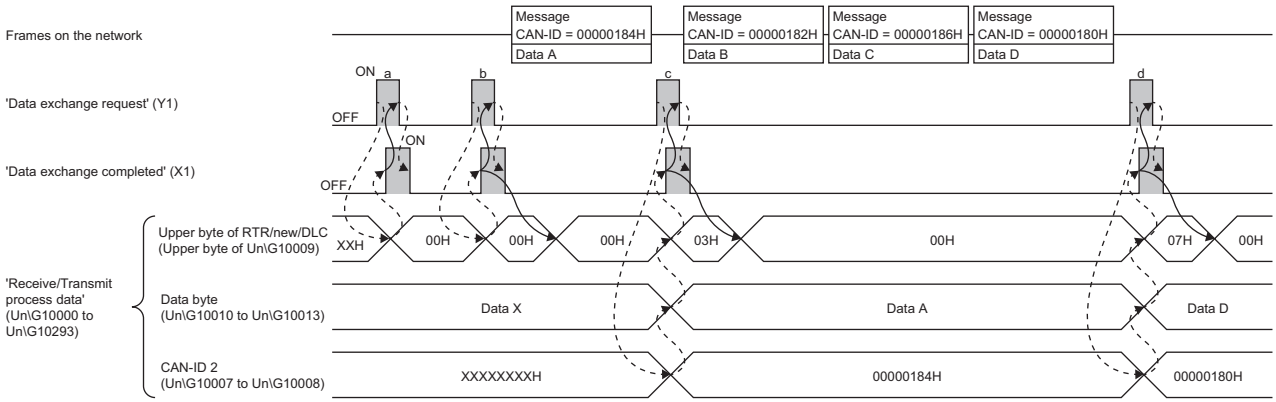
Data exchange b: No message is received between completion of data exchange a (the upper byte of RTR/new/DLC cleared) and data exchange request b; 00H is therefore stored in the upper byte of RTR/new/DLC.

Data exchange c: A data A message is received (a new frame + new data received) between completion of data exchange b (the upper byte of RTR/new/DLC cleared) and data exchange request c; 03H is therefore stored in the upper byte of RTR/new/DLC. Data A is stored in the data byte.

Data exchange d: A data B message is received (a new frame + new data received) between completion of data exchange c (the upper byte of RTR/new/DLC cleared) and data exchange request d, and a data C message and a data D message are received (new data received) and then the buffer memory area is overwritten (an overflow occurs for data C and data D each); 07H is therefore stored in the upper byte of RTR/new/DLC. Data D, which has been received last, is stored in the data byte.

Ex.

When Layer 2 message 2 parameters A and B are 00000180H, and Layer 2 message 2 parameters C and D are 00000006H Receive messages whose CAN-ID is 00000180H, 00000182H, 00000184H, or 00000186H are stored in the message slot of Layer 2 message 2 in 'Receive/Transmit process data' (Un\G10000 to Un\G10293) since bit 1 and bit 2 of each CAN-ID are ignored in verification.



: Executed by the RJ71CN91

: Executed by the program

The behavior up to (d) is the same as that described in the above example.

In this example, it is required to check the 11-bit/29-bit CAN-ID of the corresponding message slot of 'Receive/Transmit process data' (Un\G10000 to Un\G10293) to determine which CAN-ID message was received. Since the CAN-ID of the last message is 00000180H, only the data of this message is stored in the data bytes of 'Receive/Transmit process data' (Un\G10000 to Un\G10293). Data of messages whose CAN-ID is 00000182H or 00000186H will be lost.

Point

- In this example, four messages are stored in the corresponding message slot in 'Receive/Transmit process data' (Un\G10000 to Un\G10293). If two or more messages whose CAN-ID is 00000180H, 00000182H, 00000184H, or 00000186H are received while turning on 'Data exchange request' (Y1), only the last message will be stored in 'Receive/Transmit process data' (Un\G10000 to Un\G10293).
- In this example, at (a), (b), (c), and (d), it is necessary to read data stored in the data bytes of 'Receive/Transmit process data' (Un\G10000 to Un\G10293) and then clear RTR/new//DLC in 'Receive/Transmit process data' (Un\G10000 to Un\G10293) by the program.

A

■Layer 2 RTR flags (Un\G8350 to Un\G8352)

When the following conditions are met, message slots that received RTR messages can be checked.

- The CAN-ID of an RTR message and the CAN-ID of the corresponding message slot match.
- The corresponding message slot is configured as a transmit message slot.
- The corresponding message slot is set to 6FFFH (Manual RTR).

"Layer 2 RTR flag" is updated without regard to output signal Y1.

Each bit in this area will turn on when a valid RTR message is received. The timing of RTR response varies depending on the setting. (☞ Page 68 RTR response timing)

When an RTR response message is transmitted, the corresponding bit in this area will be cleared automatically.

Address	Bit	Description
Un\G8350	b0	RTR message for Layer 2 message slot 1 received
	b1	RTR message for Layer 2 message slot 2 received
	⋮	⋮
	b14	RTR message for Layer 2 message slot 15 received
	b15	RTR message for Layer 2 message slot 16 received
Un\G8351	b0	RTR message for Layer 2 message slot 17 received
	b1	RTR message for Layer 2 message slot 18 received
	⋮	⋮
	b14	RTR message for Layer 2 message slot 31 received
	b15	RTR message for Layer 2 message slot 32 received
Un\G8352	b0	RTR message for Layer 2 message slot 33 received
	b1	RTR message for Layer 2 message slot 34 received
	⋮	⋮
	b9	RTR message for Layer 2 message slot 42 received
	b10	System area
	⋮	⋮
	b15	System area

■Message transmit trigger flags (Un\G8400 to Un\G8402)

Set the send request message slot.

For the message slot set for the send message, turn on the corresponding message slot in this area.

For the message slot set for the receive message, if the corresponding message slot in this area is turned on, the setting is discarded.

Set the send request message slot in this area, and turn on 'Message transmit trigger request' (Y4). Messages are transmitted when the send buffer becomes available.

When a message is written to the send buffer, the corresponding message slot in this area will be cleared automatically.

'Message transmit trigger completed' (X4) will turn on when all the messages are written into the send buffer.

Address	Bit	Description
Un\G8400	b0	Message slot 1
	b1	Message slot 2
	⋮	⋮
	b14	Message slot 15
	b15	Message slot 16
Un\G8401	b0	Message slot 17
	b1	Message slot 18
	⋮	⋮
	b14	Message slot 31
	b15	Message slot 32
Un\G8402	b0	Message slot 33
	b1	Message slot 34
	⋮	⋮
	b9	Message slot 42
	b10	System area
	⋮	⋮
	b15	System area

A

■CPU module STOP transition message (Un\G8450 to Un\G8477)

The RJ71CN91 can define up to four transmit messages to send them when the CPU module state changes from RUN to STOP.

CPU module STOP transition messages 1 to 4 are sent when the CPU module state changes from RUN to STOP.

Point

Best practice is to use only one CPU module STOP transition message. This increases the chances of sending the message when the CPU module state changes from RUN to STOP. If multiple messages are defined, they are sent in the order of "CPU module STOP transition message" 1 to 4.

Address	Description				Initial value*1
Un\G8450	CAN-ID 1 LW	11-bit/29-bit CAN-ID (low word)		CPU module STOP transition message 1	FFFFH
Un\G8451	CAN-ID 1 HW	11-bit/29-bit CAN-ID (high word)			FFFFH
Un\G8452	DLC*2	Data length count			0000H
Un\G8453	Data bytes	Lower byte	Data byte 1		0000H
		Upper byte	Data byte 2		
Un\G8454		Lower byte	Data byte 3		0000H
		Upper byte	Data byte 4		
Un\G8455		Lower byte	Data byte 5		0000H
		Upper byte	Data byte 6		
Un\G8456		Lower byte	Data byte 7		0000H
		Upper byte	Data byte 8		
⋮	⋮	⋮		⋮	⋮
Un\G8471	CAN-ID 4 LW	11-bit/29-bit CAN-ID (low word)		CPU module STOP transition message 4	FFFFH
Un\G8472	CAN-ID 4 HW	11-bit/29-bit CAN-ID (high word)			FFFFH
Un\G8473	DLC*2	Data length count			0000H
Un\G8474	Data bytes	Lower byte	Data byte 1		0000H
		Upper byte	Data byte 2		
Un\G8475		Lower byte	Data byte 3		0000H
		Upper byte	Data byte 4		
Un\G8476		Lower byte	Data byte 5		0000H
		Upper byte	Data byte 6		
Un\G8477		Lower byte	Data byte 7		0000H
		Upper byte	Data byte 8		

*1 Initial value when the CPU module is powered off and on or reset

*2 DLC is an abbreviation of Data Length Count.

The following describes settings of each buffer memory area.

Buffer memory area	Description
CAN ID n (CAN-ID n LW, CAN-ID n HW)	Set the CAN-ID of the message. To disable the message, set CAN-ID n LW and CAN-ID n HW to FFFFH.
DLC	Lower byte: Set the number of data bytes to transmit (0 to 8).
	Upper byte: Set to 00H (Send data frame).*3
Data bytes	Set data to transmit.

*3 01H (Send RTR frame) is not available.

Point

Change these settings during offline mode. When the RJ71CN91 transitions to the online mode, the settings are enabled.

■Receive/Transmit process data (Un\G10000 to Un\G10293)

The RJ71CN91 can send/receive up to 42 messages pre-defined by the user.

CIF can be used to send messages. (☞ Page 71 Data/RTR transmission via CIF)

Point

- This area is updated every time 'Data exchange request' (Y1) is turned on. (☞ Page 153 Data exchange completed (X1), Data exchange request (Y1))
- The receive messages stored in this area are initialized when the RJ71CN91 transitions to the online mode.
- Change these settings during offline mode. When the RJ71CN91 transitions to the online mode, the settings are enabled.
- The data byte sections are not used for parameters, and thus not included in the setting saving targets.

Address	Description				Initial value*1	Configuration save target*4
Un\G10000	CAN-ID 1 LW	11-bit/29-bit CAN-ID (low word)		Layer 2 message 1	FFFFH	○
Un\G10001	CAN-ID 1 HW	11-bit/29-bit CAN-ID (high word)			FFFFH	○
Un\G10002	RTR/new/DLC*2	Lower byte	Data length count		0000H	○
		Upper byte	RTR			
Un\G10003	Data bytes*3	Lower byte	Data byte 1		0000H	×
		Upper byte	Data byte 2			
Un\G10004		Lower byte	Data byte 3		0000H	×
		Upper byte	Data byte 4			
Un\G10005		Lower byte	Data byte 5		0000H	×
		Upper byte	Data byte 6			
Un\G10006		Lower byte	Data byte 7		0000H	×
		Upper byte	Data byte 8			
⋮	⋮	⋮		⋮	⋮	⋮
Un\G10287	CAN-ID 42 LW	11-bit/29-bit CAN-ID (low word)		Layer 2 message 42	FFFFH	○
Un\G10288	CAN-ID 42 HW	11-bit/29-bit CAN-ID (high word)			FFFFH	○
Un\G10289	RTR/new/DLC*2	Lower byte	Data length count		0000H	○
		Upper byte	RTR			
Un\G10290	Data bytes*3	Lower byte	Data byte 1		0000H	×
		Upper byte	Data byte 2			
Un\G10291		Lower byte	Data byte 3		0000H	×
		Upper byte	Data byte 4			
Un\G10292		Lower byte	Data byte 5		0000H	×
		Upper byte	Data byte 6			
Un\G10293		Lower byte	Data byte 7		0000H	×
		Upper byte	Data byte 8			

*1 Initial value when the CPU module is powered off and on or reset

*2 DLC is an abbreviation of Data Length Count.

*3 Receive messages are read-only. Transmit messages can be read and written.

*4 An area to be saved into the flash ROM by buffer memory settings save operation (○: Can be saved, ×: Cannot be saved)
For how to store the settings, refer to Save/restore configuration (Un\G22). (☞ Page 162 Save/restore configuration (Un\G22))
The settings can be saved only when Send message is set as the setting in 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167) (when parameter A = FFFFH, and parameter B = 7FFFH, 6FFFH, or 5FFFH).

The following describes settings of each buffer memory area.

Buffer memory area	Description	
	Transmit message	Receive message[*1]
CAN ID n (CAN-ID n LW, CAN-ID n HW)[*7]	Set the CAN-ID of the transmit message.	Reception CAN-ID
RTR/new/DLC	■Upper byte[*2] b12 = 1: RTR DLC precise check[*3] b15 = 0: Send data frame[*4] b15 = 1: Send RTR frame[*4]	■Upper byte[*5] 00H = New data not received b8 = 1: New data received b9 = 1: New frame received b10 = 1: Overflow
	■Lower byte Set the number of bytes for send data (0 to 8)[*4].	■Lower byte Data length count (DLC) of the received CAN frame
Data bytes	Data bytes 1 to 8 The number of attached data bytes that is defined by DLC	Received data bytes[*6]

*1 When a receive message passes the filter set in 'Pre-defined Layer 2 message configuration' (Un\G6000 to Un\G6167), the reception CAN-ID is updated. The CAN-ID, RTR/new/DLC, and data displayed in the receive message is always latest.

*2 b12 and b15 cannot be set to 1 at the same time.

*3 When b12 is set to 0, only the CAN-ID of an receive RTR frame is checked for a match with the ID of the user message.
When b12 is set to 1, the message will not be sent if the CAN-ID and DLC of the RTR frame do not match those of the user message.
Also, the corresponding message slot in 'Layer 2 RTR flags' (Un\G8350 to Un\G8352) will not turn on.
b12 can be set to 1 when parameter B is 6FFFH or 7FFFH.

*4 b15 sets whether to send a message as a send data frame or send RTR frame.
b15 can be set to 1 when parameter B is 5FFFH
b15 can be set to 1 when parameter C is 0, 2, or 4.

*5 When b8 is 1, a new message including new data has been received and stored.
If b9 is 1 but b8 is 0, the same message (same CAN-ID, DLC and data) has been received.
If b10 is 1, at least one message has been stored in the receive buffer while b8 was 1, which caused an overflow condition.

*6 In case the received DLC is lower than 8, unused data bytes are set to 00H.

*7 For the CAN-ID of a send message, the same CAN-ID as the CAN-ID of another send message cannot be used. If the same CAN-ID is set, a parameter error occurs.

The following describes status of b8 to b10 of RTR/new/DLC for receive messages.

- New frame, no new data: b8 = 0, b9 = 1, b10 = 0
- New frame, new data: b8 = 1, b9 = 1, b10 = 0
- New frame, no new data, overflow: b8 = 0, b9 = 1, b10 = 1
- New frame, new data, overflow: b8 = 1, b9 = 1, b10 = 1
- No receive data: b8 = 0 or 1, b9 = 0, b10 = 0 or 1

Appendix 3 Object Dictionary

Object dictionary list

The object dictionary is structured in indexes and subindexes. Each index addresses a single parameter, a set of parameters, network input/output data or other data. A subindex addresses a subset of the parameter or data of the index.

Index	Object	Reference
0000H	System area	—
0001H to 009FH	Data type definitions	Page 191 Data type definitions
00A0H to 0FFFH	System area	—
1000H to 1FFFH	Communication profile (CiA-301/CiA-302)	Page 192 Communication profile
2000H to 5FFFH	Manufacturer specific	—
6000H to 9FFFH	System area	—
A000H to AFFFH	Standard interface profile (CiA-405)	Page 208 Standard interface profile
B000H to FFFFH	System area	—

Data type definitions

This area provides the data type definition object. Use this object as a mapping destination for data not used in RPDO receive data (not mapped to any object). SDO access to this object will cause an error.

For details on mapping to RPDO, refer to the following.

☞ Page 41 PDO

Index	Subindex	Object	Description		Data type
0001H	00H	Data type definition	System area		—
0002H	00H		Signed	8 bits	I8
0003H	00H			16 bits	I16
0004H	00H			32 bits	I32
0005H	00H		Unsigned	8 bits	U8
0006H	00H			16 bits	U16
0007H	00H			32 bits	U32
0008H	00H		Float	32 bits	Real32
0009H to 009FH	00H		System area		—

A

Communication profile

This section provides a brief description of the CANopen object dictionary of the RJ71CN91 module and related information.

○: Writable, ×: Not writable

Index	Subindex	Object	Description	Data type	Initial value*1	Read/ Write*2	Write to flash ROM*3
1000H	00H	Device Type	Describes the device profile or the application profile. Can be changed by setting 'Function mode' (Un\G21).	U32	405	Read	×
1001H	00H	Error register	☞ Page 211 Error register	U8	00H	Read	×
1002H	00H	System area	—	—	—	—	—
1003H	00H	Pre-defined error field	☞ Page 211 Pre-defined error field	U8	00H	Read/Write	×
	01H to 0FH			U32	00H	Read	
1004H	00H	System area	—	—	—	—	—
1005H	00H	COB-ID of SYNC message	☞ Page 211 COB-ID of SYNC message	U32	80H	Read/Write	○
1006H	00H	Communication cycle period	☞ Page 212 Communication cycle period	U32	00H	Read/Write	○
1007H	00H	System area	—	—	—	—	—
1008H	00H	Device Name	8-byte ASCII code string	Visible string	RJ71CN91	Read	×
1009H	00H	Hardware Version	2-byte ASCII code string		—*4	Read	×
100AH	00H	Software Version	2-byte ASCII code string		—*5	Read	×
100BH	00H	System area	—	—	—	—	—
100CH	00H	Guard time	☞ Page 212 Guard time	U16	00H	Read/Write	○
100DH	00H	Life time factor	☞ Page 212 Life time factor	U8	00H	Read/Write	○
100EH to 100FH	00H	System area	—	—	—	—	—
1010H	00H	Store parameters	Maximum subindex	U8	01H	Read	×
	01H		Save all parameters ☞ Page 212 Store parameters	U32	01H	Read/Write	
1011H	00H	Restore default parameters	Maximum subindex	U8	01H	Read	×
	01H		Restore all parameters ☞ Page 213 Restore default parameters	U32	01H	Read/Write	
1012H	00H	COB-ID Time stamp	☞ Page 213 COB-ID Time stamp	U32	80000100H	Read/Write	○
1013H	00H	System area	—	—	—	—	—
1014H	00H	COB-ID EMCY	☞ Page 213 COB-ID EMCY	U32	80H + node ID	Read	×
1015H	00H	Inhibit time EMCY	☞ Page 213 Inhibit time EMCY	U16	00H	Read/Write	○
1016H	00H	Consumer heartbeat time	Maximum subindex	U8	7FH	Read	×
	01H to 7FH		☞ Page 214 Consumer heartbeat time	U32	00H	Read/Write	○
1017H	00H	Producer heartbeat time	☞ Page 214 Producer heartbeat time	U16	0	Read/Write	○
1018H	00H	Identity Object	Maximum subindex	U8	03H	Read	×
	01H		Vendor ID	U32	01D0H		
	02H		Product code	U32	0100H		
	03H		Revision number	U32	—*6		
1019H to 101FH	00H	System area	—	—	—	—	—

Index	Subindex	Object	Description		Data type	Initial value*1	Read/ Write*2	Write to flash ROM*3
1020H	00H	Verify Configuration	Maximum subindex		U8	02H	Read	×
	01H		☞ Page 214 Verify Configuration		U32	00H	Read/Write	○
	02H				U32	00H	Read/Write	○
1021H to 1027H	00H	System area	—		—	—	—	—
1028H	00H	Emergency consumer object	Maximum subindex		U8	7FH	Read	×
	01H to 7FH		☞ Page 214 Emergency consumer object		U32	80H + node ID	Read/Write	○
1029H	00H	Error behavior	Maximum subindex		U8	01H	Read	×
	01H		☞ Page 215 Error behavior		U8	00H	Read/Write	○
102AH	00H	NMT inhibit time	☞ Page 215 NMT inhibit time		U16	00H	Read	×
102BH to 13FFH	00H	System area	—		—	—	—	—
1400H to 14FFH	00H	RPDO communication parameter	Maximum subindex		U8	☞ Page 196 RPDO communication parameter initial values		×
	01H		☞ Page 215 RPDO communication parameter	COB-ID	U32			○
	02H			Transmission type	U8			○
1500H to 15FFH	00H	System area	—		—	—	—	—
1600H to 16FFH	00H	RPDO mapping parameter	☞ Page 216 RPDO mapping parameter	Number of valid object entries	U8	☞ Page 196 RPDO mapping parameter initial values		○
	01H			First mapped object	U32			
	02H			Second mapped object	U32			
	03H			Third mapped object	U32			
	04H			Fourth mapped object	U32			
	05H			Fifth mapped object	U32			
	06H			Sixth mapped object	U32			
	07H			Seventh mapped object	U32			
	08H			Eighth mapped object	U32			
1700H to 17FFH	00H	System area	—		—	—	—	—
1800H to 18FFH	00H	TPDO communication parameter	Maximum subindex		U8	☞ Page 201 TPDO communication parameter initial values		×
	01H		☞ Page 217 TPDO communication parameter	COB-ID	U32			○
	02H			Transmission type	U8			○
	03H			Inhibit time	U16			○
	04H			Compatibility entry	U8			×
	05H			Event time	U16			○
1900H to 19FFH	00H	System area	—		—	—	—	—

A

Index	Subindex	Object	Description		Data type	Initial value*1	Read/ Write*2	Write to flash ROM*3
1A00H to 1AFFH	00H	TPDO mapping parameter	☞ Page 218 TPDO mapping parameter	Number of valid object entries	U8	☞ Page 202 TPDO mapping parameter initial values		○
	01H			First mapped object	U32			
	02H			Second mapped object	U32			
	03H			Third mapped object	U32			
	04H			Fourth mapped object	U32			
	05H			Fifth mapped object	U32			
	06H			Sixth mapped object	U32			
	07H			Seventh mapped object	U32			
	08H			Eighth mapped object	U32			
1B00H to 1F21H	00H	System area	—		—	—	—	—
1F22H	00H	Concise DCF	Maximum subindex		U8	7FH	Read	×
	01H to 7FH		☞ Page 218 Concise DCF	Node ID	DOMAIN	—	Read/Write	
1F23H to 1F24H	00H	System area	—		—	—	—	—
1F25H	00H	Configuration request	Maximum subindex		U8	80H	Read	×
	01H to 7FH		☞ Page 219 Configuration request	Node ID	U32	—	Write	
	80H			All nodes				
1F26H	00H	Expected configuration date	Maximum subindex		U8	7FH	Read	×
	01H to 7FH		☞ Page 219 Expected configuration date	Node ID	U32	00H	Read/Write	○
1F27H	00H	Expected configuration time	Maximum subindex		U8	7FH	Read	×
	01H to 7FH		☞ Page 219 Expected configuration time	Node ID	U32	00H	Read/Write	○
1F28H to 1F7FH	00H	System area	—		—	—	—	—
1F80H	00H	NMT start-up	☞ Page 220 NMT start-up		U32	00H	Read/Write	○
1F81H	00H	NMT slave assignment	Maximum subindex		U8	7FH	Read	×
	01H to 7FH		☞ Page 221 NMT slave assignment	Node ID	U32	00H	Read/Write	○
1F82H	00H	Request NMT	Maximum subindex		U8	80H	Read	×
	01H to 7FH		☞ Page 222 Request NMT	Node ID	U8	00H	Read/Write	
	80H			All nodes		—	Write	
1F83H	00H	Request node guarding	Maximum subindex		U8	80H	Read	×
	01H to 7FH		☞ Page 223 Request node guarding	Node ID	U8	00H	Read/Write	
	80H			All nodes			Write	
1F84H	00H	Device Type	Maximum subindex		U8	7FH	Read	×
	01H to 7FH		☞ Page 223 Device Type		U32	00H	Read/Write	○

Index	Subindex	Object	Description		Data type	Initial value*1	Read/ Write*2	Write to flash ROM*3
1F85H	00H	Vendor identification	Maximum subindex		U8	7FH	Read	×
	01H to 7FH		☞ Page 223 Vendor identification		U32	00H	Read/Write	○
1F86H	00H	Product code	Maximum subindex		U8	7FH	Read	×
	01H to 7FH		☞ Page 223 Product code		U32	00H	Read/Write	○
1F87H	00H	Revision number	Maximum subindex		U8	7FH	Read	×
	01H to 7FH		☞ Page 224 Revision number		U32	00H	Read/Write	○
1F88H	00H	Serial number	Maximum subindex		U8	7FH	Read	×
	01H to 7FH		☞ Page 224 Serial number		U32	00H	Read/Write	○
1F89H	00H	Boot time	☞ Page 224 Boot time		U32	00H	Read/Write	○
1F8AH to 1F8FH	00H	System area	—		—	—	—	—
1F90H	00H	NMT flying master timing parameters	Maximum subindex		U8	06H	Read	×
	01H		☞ Page 225 NMT flying master timing parameters	Active NMT master detection timeout	U16	100	Read/Write	○
	02H			NMT master negotiation time delay	U16	500	Read/Write	
	03H			Master priority level	U16	1	Read/Write	
	04H			Priority time slot	U16	1500	Read/Write	
	05H			Node time slot	U16	10	Read/Write	
	06H			Multiplex master detection cycle time	U16	4000 + 10 × Node ID	Read/Write	
1F91H to 1FFFH	00H	System area	—		—	—	—	—

*1 Initial value when the CPU module is powered off and on or reset

*2 This indicates whether reading from and writing to the network are enabled.

*3 Data will be saved in the flash ROM by using the Store parameters (index 1010H, subindex 01H). When writing data, note that the maximum number of write accesses to a flash ROM is 100000.

*4 The hardware version depends on the one of the RJ71CN91. (Example: 01)
Check the actual value by using the CANopen Configuration Tool. (☞ Page 85 CANopen node list)

*5 The software version matches the firmware version of the RJ71CN91. (Example: 01)

*6 The revision number depends on the firmware version of the RJ71CN91. For the supported firmware version, refer to the following. However, the actual value must be checked by using the CANopen Configuration Tool. (☞ Page 85 CANopen node list)

Firmware version	Major revision number	Minor revision number
01 or later	0001H	0000H or later (The firmware version is updated, the value increases.)

RPDO communication parameter initial values

The following table lists the RPDO communication parameter initial values.

Index	Subindex initial values		
	00H (Readable)	01H (Readable/Writable)	02H (Readable/Writable)
1400H	02H	200 + node ID	FEH
1401H	02H	300 + node ID	FEH
1402H	02H	400 + node ID	FEH
1403H	02H	500 + node ID	FEH
1404H to 14FFH	02H	80000000	FEH

RPDO mapping parameter initial values

The following table lists the RPDO mapping parameter initial values.

Index	Subindex initial values								
	00H (Readable /Writable)	01H (Readable/ Writable)	02H (Readable/ Writable)	03H (Readable/ Writable)	04H (Readable/ Writable)	05H (Readable /Writable)	06H (Readable /Writable)	07H (Readable /Writable)	08H (Readable /Writable)
1600H	04H	A5800110H	A5800210H	A5800310H	A5800410H	00H	00H	00H	00H
1601H	04H	A5800510H	A5800610H	A5800710H	A5800810H	00H	00H	00H	00H
1602H	04H	A5800910H	A5800A10H	A5800B10H	A5800C10H	00H	00H	00H	00H
1603H	04H	A5800D10H	A5800E10H	A5800F10H	A5801010H	00H	00H	00H	00H
1604H	04H	A5801110H	A5801210H	A5801310H	A5801410H	00H	00H	00H	00H
1605H	04H	A5801510H	A5801610H	A5801710H	A5801810H	00H	00H	00H	00H
1606H	04H	A5801910H	A5801A10H	A5801B10H	A5801C10H	00H	00H	00H	00H
1607H	04H	A5801D10H	A5801E10H	A5801F10H	A5802010H	00H	00H	00H	00H
1608H	04H	A5802110H	A5802210H	A5802310H	A5802410H	00H	00H	00H	00H
1609H	04H	A5802510H	A5802610H	A5802710H	A5802810H	00H	00H	00H	00H
160AH	04H	A5802910H	A5802A10H	A5802B10H	A5802C10H	00H	00H	00H	00H
160BH	04H	A5802D10H	A5802E10H	A5802F10H	A5803010H	00H	00H	00H	00H
160CH	04H	A5803110H	A5803210H	A5803310H	A5803410H	00H	00H	00H	00H
160DH	04H	A5803510H	A5803610H	A5803710H	A5803810H	00H	00H	00H	00H
160EH	04H	A5803910H	A5803A10H	A5803B10H	A5803C10H	00H	00H	00H	00H
160FH	04H	A5803D10H	A5803E10H	A5803F10H	A5804010H	00H	00H	00H	00H
1610H	04H	A5804110H	A5804210H	A5804310H	A5804410H	00H	00H	00H	00H
1611H	04H	A5804510H	A5804610H	A5804710H	A5804810H	00H	00H	00H	00H
1612H	04H	A5804910H	A5804A10H	A5804B10H	A5804C10H	00H	00H	00H	00H
1613H	04H	A5804D10H	A5804E10H	A5804F10H	A5805010H	00H	00H	00H	00H
1614H	04H	A5805110H	A5805210H	A5805310H	A5805410H	00H	00H	00H	00H
1615H	04H	A5805510H	A5805610H	A5805710H	A5805810H	00H	00H	00H	00H
1616H	04H	A5805910H	A5805A10H	A5805B10H	A5805C10H	00H	00H	00H	00H
1617H	04H	A5805D10H	A5805E10H	A5805F10H	A5806010H	00H	00H	00H	00H
1618H	04H	A5806110H	A5806210H	A5806310H	A5806410H	00H	00H	00H	00H
1619H	04H	A5806510H	A5806610H	A5806710H	A5806810H	00H	00H	00H	00H
161AH	04H	A5806910H	A5806A10H	A5806B10H	A5806C10H	00H	00H	00H	00H
161BH	04H	A5806D10H	A5806E10H	A5806F10H	A5807010H	00H	00H	00H	00H
161CH	04H	A5807110H	A5807210H	A5807310H	A5807410H	00H	00H	00H	00H
161DH	04H	A5807510H	A5807610H	A5807710H	A5807810H	00H	00H	00H	00H
161EH	04H	A5807910H	A5807A10H	A5807B10H	A5807C10H	00H	00H	00H	00H
161FH	04H	A5807D10H	A5807E10H	A5807F10H	A5808010H	00H	00H	00H	00H
1620H	04H	A5808110H	A5808210H	A5808310H	A5808410H	00H	00H	00H	00H
1621H	04H	A5808510H	A5808610H	A5808710H	A5808810H	00H	00H	00H	00H
1622H	04H	A5808910H	A5808A10H	A5808B10H	A5808C10H	00H	00H	00H	00H
1623H	04H	A5808D10H	A5808E10H	A5808F10H	A5809010H	00H	00H	00H	00H

Index	Subindex initial values								
	00H (Readable /Writable)	01H (Readable/ Writable)	02H (Readable/ Writable)	03H (Readable/ Writable)	04H (Readable/ Writable)	05H (Readable /Writable)	06H (Readable /Writable)	07H (Readable /Writable)	08H (Readable /Writable)
1624H	04H	A5809110H	A5809210H	A5809310H	A5809410H	00H	00H	00H	00H
1625H	04H	A5809510H	A5809610H	A5809710H	A5809810H	00H	00H	00H	00H
1626H	04H	A5809910H	A5809A10H	A5809B10H	A5809C10H	00H	00H	00H	00H
1627H	04H	A5809D10H	A5809E10H	A5809F10H	A580A010H	00H	00H	00H	00H
1628H	04H	A580A110H	A580A210H	A580A310H	A580A410H	00H	00H	00H	00H
1629H	04H	A580A510H	A580A610H	A580A710H	A580A810H	00H	00H	00H	00H
162AH	04H	A580A910H	A580AA10H	A580AB10H	A580AC10H	00H	00H	00H	00H
162BH	04H	A580AD10H	A580AE10H	A580AF10H	A580B010H	00H	00H	00H	00H
162CH	04H	A580B110H	A580B210H	A580B310H	A580B410H	00H	00H	00H	00H
162DH	04H	A580B510H	A580B610H	A580B710H	A580B810H	00H	00H	00H	00H
162EH	04H	A580B910H	A580BA10H	A580BB10H	A580BC10H	00H	00H	00H	00H
162FH	04H	A580BD10H	A580BE10H	A580BF10H	A580C010H	00H	00H	00H	00H
1630H	04H	A580C110H	A580C210H	A580C310H	A580C410H	00H	00H	00H	00H
1631H	04H	A580C510H	A580C610H	A580C710H	A580C810H	00H	00H	00H	00H
1632H	04H	A580C910H	A580CA10H	A580CB10H	A580CC10H	00H	00H	00H	00H
1633H	04H	A580CD10H	A580CE10H	A580CF10H	A580D010H	00H	00H	00H	00H
1634H	04H	A580D110H	A580D210H	A580D310H	A580D410H	00H	00H	00H	00H
1635H	04H	A580D510H	A580D610H	A580D710H	A580D810H	00H	00H	00H	00H
1636H	04H	A580D910H	A580DA10H	A580DB10H	A580DC10H	00H	00H	00H	00H
1637H	04H	A580DD10H	A580DE10H	A580DF10H	A580E010H	00H	00H	00H	00H
1638H	04H	A580E110H	A580E210H	A580E310H	A580E410H	00H	00H	00H	00H
1639H	04H	A580E510H	A580E610H	A580E710H	A580E810H	00H	00H	00H	00H
163AH	04H	A580E910H	A580EA10H	A580EB10H	A580EC10H	00H	00H	00H	00H
163BH	04H	A580ED10H	A580EE10H	A580EF10H	A580F010H	00H	00H	00H	00H
163CH	04H	A580F110H	A580F210H	A580F310H	A580F410H	00H	00H	00H	00H
163DH	04H	A580F510H	A580F610H	A580F710H	A580F810H	00H	00H	00H	00H
163EH	04H	A580F910H	A580FA10H	A580FB10H	A580FC10H	00H	00H	00H	00H
163FH	04H	A580FD10H	A580FE10H	A5810110H	A5810210H	00H	00H	00H	00H
1640H	04H	A5810310H	A5810410H	A5810510H	A5810610H	00H	00H	00H	00H
1641H	04H	A5810710H	A5810810H	A5810910H	A5810A10H	00H	00H	00H	00H
1642H	04H	A5810B10H	A5810C10H	A5810D10H	A5810E10H	00H	00H	00H	00H
1643H	04H	A5810F10H	A5811010H	A5811110H	A5811210H	00H	00H	00H	00H
1644H	04H	A5811310H	A5811410H	A5811510H	A5811610H	00H	00H	00H	00H
1645H	04H	A5811710H	A5811810H	A5811910H	A5811A10H	00H	00H	00H	00H
1646H	04H	A5811B10H	A5811C10H	A5811D10H	A5811E10H	00H	00H	00H	00H
1647H	04H	A5811F10H	A5812010H	A5812110H	A5812210H	00H	00H	00H	00H
1648H	04H	A5812310H	A5812410H	A5812510H	A5812610H	00H	00H	00H	00H
1649H	04H	A5812710H	A5812810H	A5812910H	A5812A10H	00H	00H	00H	00H
164AH	04H	A5812B10H	A5812C10H	A5812D10H	A5812E10H	00H	00H	00H	00H
164BH	04H	A5812F10H	A5813010H	A5813110H	A5813210H	00H	00H	00H	00H
164CH	04H	A5813310H	A5813410H	A5813510H	A5813610H	00H	00H	00H	00H
164DH	04H	A5813710H	A5813810H	A5813910H	A5813A10H	00H	00H	00H	00H
164EH	04H	A5813B10H	A5813C10H	A5813D10H	A5813E10H	00H	00H	00H	00H
164FH	04H	A5813F10H	A5814010H	A5814110H	A5814210H	00H	00H	00H	00H
1650H	04H	A5814310H	A5814410H	A5814510H	A5814610H	00H	00H	00H	00H
1651H	04H	A5814710H	A5814810H	A5814910H	A5814A10H	00H	00H	00H	00H
1652H	04H	A5814B10H	A5814C10H	A5814D10H	A5814E10H	00H	00H	00H	00H
1653H	04H	A5814F10H	A5815010H	A5815110H	A5815210H	00H	00H	00H	00H
1654H	04H	A5815310H	A5815410H	A5815510H	A5815610H	00H	00H	00H	00H
1655H	04H	A5815710H	A5815810H	A5815910H	A5815A10H	00H	00H	00H	00H

Index	Subindex initial values								
	00H (Readable /Writable)	01H (Readable/ Writable)	02H (Readable/ Writable)	03H (Readable/ Writable)	04H (Readable/ Writable)	05H (Readable /Writable)	06H (Readable /Writable)	07H (Readable /Writable)	08H (Readable /Writable)
1656H	04H	A5815B10H	A5815C10H	A5815D10H	A5815E10H	00H	00H	00H	00H
1657H	04H	A5815F10H	A5816010H	A5816110H	A5816210H	00H	00H	00H	00H
1658H	04H	A5816310H	A5816410H	A5816510H	A5816610H	00H	00H	00H	00H
1659H	04H	A5816710H	A5816810H	A5816910H	A5816A10H	00H	00H	00H	00H
165AH	04H	A5816B10H	A5816C10H	A5816D10H	A5816E10H	00H	00H	00H	00H
165BH	04H	A5816F10H	A5817010H	A5817110H	A5817210H	00H	00H	00H	00H
165CH	04H	A5817310H	A5817410H	A5817510H	A5817610H	00H	00H	00H	00H
165DH	04H	A5817710H	A5817810H	A5817910H	A5817A10H	00H	00H	00H	00H
165EH	04H	A5817B10H	A5817C10H	A5817D10H	A5817E10H	00H	00H	00H	00H
165FH	04H	A5817F10H	A5818010H	A5818110H	A5818210H	00H	00H	00H	00H
1660H	04H	A5818310H	A5818410H	A5818510H	A5818610H	00H	00H	00H	00H
1661H	04H	A5818710H	A5818810H	A5818910H	A5818A10H	00H	00H	00H	00H
1662H	04H	A5818B10H	A5818C10H	A5818D10H	A5818E10H	00H	00H	00H	00H
1663H	04H	A5818F10H	A5819010H	A5819110H	A5819210H	00H	00H	00H	00H
1664H	04H	A5819310H	A5819410H	A5819510H	A5819610H	00H	00H	00H	00H
1665H	04H	A5819710H	A5819810H	A5819910H	A5819A10H	00H	00H	00H	00H
1666H	04H	A5819B10H	A5819C10H	A5819D10H	A5819E10H	00H	00H	00H	00H
1667H	04H	A5819F10H	A581A010H	A581A110H	A581A210H	00H	00H	00H	00H
1668H	04H	A581A310H	A581A410H	A581A510H	A581A610H	00H	00H	00H	00H
1669H	04H	A581A710H	A581A810H	A581A910H	A581AA10H	00H	00H	00H	00H
166AH	04H	A581AB10H	A581AC10H	A581AD10H	A581AE10H	00H	00H	00H	00H
166BH	04H	A581AF10H	A581B010H	A581B110H	A581B210H	00H	00H	00H	00H
166CH	04H	A581B310H	A581B410H	A581B510H	A581B610H	00H	00H	00H	00H
166DH	04H	A581B710H	A581B810H	A581B910H	A581BA10H	00H	00H	00H	00H
166EH	04H	A581BB10H	A581BC10H	A581BD10H	A581BE10H	00H	00H	00H	00H
166FH	04H	A581BF10H	A581C010H	A581C110H	A581C210H	00H	00H	00H	00H
1670H	04H	A581C310H	A581C410H	A581C510H	A581C610H	00H	00H	00H	00H
1671H	04H	A581C710H	A581C810H	A581C910H	A581CA10H	00H	00H	00H	00H
1672H	04H	A581CB10H	A581CC10H	A581CD10H	A581CE10H	00H	00H	00H	00H
1673H	04H	A581CF10H	A581D010H	A581D110H	A581D210H	00H	00H	00H	00H
1674H	04H	A581D310H	A581D410H	A581D510H	A581D610H	00H	00H	00H	00H
1675H	04H	A581D710H	A581D810H	A581D910H	A581DA10H	00H	00H	00H	00H
1676H	04H	A581DB10H	A581DC10H	A581DD10H	A581DE10H	00H	00H	00H	00H
1677H	04H	A581DF10H	A581E010H	A581E110H	A581E210H	00H	00H	00H	00H
1678H	04H	A581E310H	A581E410H	A581E510H	A581E610H	00H	00H	00H	00H
1679H	04H	A581E710H	A581E810H	A581E910H	A581EA10H	00H	00H	00H	00H
167AH	04H	A581EB10H	A581EC10H	A581ED10H	A581EE10H	00H	00H	00H	00H
167BH	04H	A581EF10H	A581F010H	A581F110H	A581F210H	00H	00H	00H	00H
167CH	04H	A581F310H	A581F410H	A581F510H	A581F610H	00H	00H	00H	00H
167DH	04H	A581F710H	A581F810H	A581F910H	A581FA10H	00H	00H	00H	00H
167EH	04H	A581FB10H	A581FC10H	A581FD10H	A581FE10H	00H	00H	00H	00H
167FH	04H	A5820110H	A5820210H	A5820310H	A5820410H	00H	00H	00H	00H
1680H	04H	A5820510H	A5820610H	A5820710H	A5820810H	00H	00H	00H	00H
1681H	04H	A5820910H	A5820A10H	A5820B10H	A5820C10H	00H	00H	00H	00H
1682H	04H	A5820D10H	A5820E10H	A5820F10H	A5821010H	00H	00H	00H	00H
1683H	04H	A5821110H	A5821210H	A5821310H	A5821410H	00H	00H	00H	00H
1684H	04H	A5821510H	A5821610H	A5821710H	A5821810H	00H	00H	00H	00H
1685H	04H	A5821910H	A5821A10H	A5821B10H	A5821C10H	00H	00H	00H	00H
1686H	04H	A5821D10H	A5821E10H	A5821F10H	A5822010H	00H	00H	00H	00H
1687H	04H	A5822110H	A5822210H	A5822310H	A5822410H	00H	00H	00H	00H

Index	Subindex initial values								
	00H (Readable /Writable)	01H (Readable/ Writable)	02H (Readable/ Writable)	03H (Readable/ Writable)	04H (Readable/ Writable)	05H (Readable /Writable)	06H (Readable /Writable)	07H (Readable /Writable)	08H (Readable /Writable)
1688H	04H	A5822510H	A5822610H	A5822710H	A5822810H	00H	00H	00H	00H
1689H	04H	A5822910H	A5822A10H	A5822B10H	A5822C10H	00H	00H	00H	00H
168AH	04H	A5822D10H	A5822E10H	A5822F10H	A5823010H	00H	00H	00H	00H
168BH	04H	A5823110H	A5823210H	A5823310H	A5823410H	00H	00H	00H	00H
168CH	04H	A5823510H	A5823610H	A5823710H	A5823810H	00H	00H	00H	00H
168DH	04H	A5823910H	A5823A10H	A5823B10H	A5823C10H	00H	00H	00H	00H
168EH	04H	A5823D10H	A5823E10H	A5823F10H	A5824010H	00H	00H	00H	00H
168FH	04H	A5824110H	A5824210H	A5824310H	A5824410H	00H	00H	00H	00H
1690H	04H	A5824510H	A5824610H	A5824710H	A5824810H	00H	00H	00H	00H
1691H	04H	A5824910H	A5824A10H	A5824B10H	A5824C10H	00H	00H	00H	00H
1692H	04H	A5824D10H	A5824E10H	A5824F10H	A5825010H	00H	00H	00H	00H
1693H	04H	A5825110H	A5825210H	A5825310H	A5825410H	00H	00H	00H	00H
1694H	04H	A5825510H	A5825610H	A5825710H	A5825810H	00H	00H	00H	00H
1695H	04H	A5825910H	A5825A10H	A5825B10H	A5825C10H	00H	00H	00H	00H
1696H	04H	A5825D10H	A5825E10H	A5825F10H	A5826010H	00H	00H	00H	00H
1697H	04H	A5826110H	A5826210H	A5826310H	A5826410H	00H	00H	00H	00H
1698H	04H	A5826510H	A5826610H	A5826710H	A5826810H	00H	00H	00H	00H
1699H	04H	A5826910H	A5826A10H	A5826B10H	A5826C10H	00H	00H	00H	00H
169AH	04H	A5826D10H	A5826E10H	A5826F10H	A5827010H	00H	00H	00H	00H
169BH	04H	A5827110H	A5827210H	A5827310H	A5827410H	00H	00H	00H	00H
169CH	04H	A5827510H	A5827610H	A5827710H	A5827810H	00H	00H	00H	00H
169DH	04H	A5827910H	A5827A10H	A5827B10H	A5827C10H	00H	00H	00H	00H
169EH	04H	A5827D10H	A5827E10H	A5827F10H	A5828010H	00H	00H	00H	00H
169FH	04H	A5828110H	A5828210H	A5828310H	A5828410H	00H	00H	00H	00H
16A0H	04H	A5828510H	A5828610H	A5828710H	A5828810H	00H	00H	00H	00H
16A1H	04H	A5828910H	A5828A10H	A5828B10H	A5828C10H	00H	00H	00H	00H
16A2H	04H	A5828D10H	A5828E10H	A5828F10H	A5829010H	00H	00H	00H	00H
16A3H	04H	A5829110H	A5829210H	A5829310H	A5829410H	00H	00H	00H	00H
16A4H	04H	A5829510H	A5829610H	A5829710H	A5829810H	00H	00H	00H	00H
16A5H	04H	A5829910H	A5829A10H	A5829B10H	A5829C10H	00H	00H	00H	00H
16A6H	04H	A5829D10H	A5829E10H	A5829F10H	A582A010H	00H	00H	00H	00H
16A7H	04H	A582A110H	A582A210H	A582A310H	A582A410H	00H	00H	00H	00H
16A8H	04H	A582A510H	A582A610H	A582A710H	A582A810H	00H	00H	00H	00H
16A9H	04H	A582A910H	A582AA10H	A582AB10H	A582AC10H	00H	00H	00H	00H
16AAH	04H	A582AD10H	A582AE10H	A582AF10H	A582B010H	00H	00H	00H	00H
16ABH	04H	A582B110H	A582B210H	A582B310H	A582B410H	00H	00H	00H	00H
16ACH	04H	A582B510H	A582B610H	A582B710H	A582B810H	00H	00H	00H	00H
16ADH	04H	A582B910H	A582BA10H	A582BB10H	A582BC10H	00H	00H	00H	00H
16AEH	04H	A582BD10H	A582BE10H	A582BF10H	A582C010H	00H	00H	00H	00H
16AFH	04H	A582C110H	A582C210H	A582C310H	A582C410H	00H	00H	00H	00H
16B0H	04H	A582C510H	A582C610H	A582C710H	A582C810H	00H	00H	00H	00H
16B1H	04H	A582C910H	A582CA10H	A582CB10H	A582CC10H	00H	00H	00H	00H
16B2H	04H	A582CD10H	A582CE10H	A582CF10H	A582D010H	00H	00H	00H	00H
16B3H	04H	A582D110H	A582D210H	A582D310H	A582D410H	00H	00H	00H	00H
16B4H	04H	A582D510H	A582D610H	A582D710H	A582D810H	00H	00H	00H	00H
16B5H	04H	A582D910H	A582DA10H	A582DB10H	A582DC10H	00H	00H	00H	00H
16B6H	04H	A582DD10H	A582DE10H	A582DF10H	A582E010H	00H	00H	00H	00H
16B7H	04H	A582E110H	A582E210H	A582E310H	A582E410H	00H	00H	00H	00H
16B8H	04H	A582E510H	A582E610H	A582E710H	A582E810H	00H	00H	00H	00H
16B9H	04H	A582E910H	A582EA10H	A582EB10H	A582EC10H	00H	00H	00H	00H

Index	Subindex initial values								
	00H (Readable /Writable)	01H (Readable/ Writable)	02H (Readable/ Writable)	03H (Readable/ Writable)	04H (Readable/ Writable)	05H (Readable /Writable)	06H (Readable /Writable)	07H (Readable /Writable)	08H (Readable /Writable)
16BAH	04H	A582ED10H	A582EE10H	A582EF10H	A582F010H	00H	00H	00H	00H
16BBH	04H	A582F110H	A582F210H	A582F310H	A582F410H	00H	00H	00H	00H
16BCH	04H	A582F510H	A582F610H	A582F710H	A582F810H	00H	00H	00H	00H
16BDH	04H	A582F910H	A582FA10H	A582FB10H	A582FC10H	00H	00H	00H	00H
16BEH	04H	A582FD10H	A582FE10H	A5830110H	A5830210H	00H	00H	00H	00H
16BFH	04H	A5830310H	A5830410H	A5830510H	A5830610H	00H	00H	00H	00H
16C0H	04H	A5830710H	A5830810H	A5830910H	A5830A10H	00H	00H	00H	00H
16C1H	04H	A5830B10H	A5830C10H	A5830D10H	A5830E10H	00H	00H	00H	00H
16C2H	04H	A5830F10H	A5831010H	A5831110H	A5831210H	00H	00H	00H	00H
16C3H	04H	A5831310H	A5831410H	A5831510H	A5831610H	00H	00H	00H	00H
16C4H	04H	A5831710H	A5831810H	A5831910H	A5831A10H	00H	00H	00H	00H
16C5H	04H	A5831B10H	A5831C10H	A5831D10H	A5831E10H	00H	00H	00H	00H
16C6H	04H	A5831F10H	A5832010H	A5832110H	A5832210H	00H	00H	00H	00H
16C7H	04H	A5832310H	A5832410H	A5832510H	A5832610H	00H	00H	00H	00H
16C8H	04H	A5832710H	A5832810H	A5832910H	A5832A10H	00H	00H	00H	00H
16C9H	04H	A5832B10H	A5832C10H	A5832D10H	A5832E10H	00H	00H	00H	00H
16CAH	04H	A5832F10H	A5833010H	A5833110H	A5833210H	00H	00H	00H	00H
16CBH	04H	A5833310H	A5833410H	A5833510H	A5833610H	00H	00H	00H	00H
16CCH	04H	A5833710H	A5833810H	A5833910H	A5833A10H	00H	00H	00H	00H
16CDH	04H	A5833B10H	A5833C10H	A5833D10H	A5833E10H	00H	00H	00H	00H
16CEH	04H	A5833F10H	A5834010H	A5834110H	A5834210H	00H	00H	00H	00H
16CFH	04H	A5834310H	A5834410H	A5834510H	A5834610H	00H	00H	00H	00H
16D0H	04H	A5834710H	A5834810H	A5834910H	A5834A10H	00H	00H	00H	00H
16D1H	04H	A5834B10H	A5834C10H	A5834D10H	A5834E10H	00H	00H	00H	00H
16D2H	04H	A5834F10H	A5835010H	A5835110H	A5835210H	00H	00H	00H	00H
16D3H	04H	A5835310H	A5835410H	A5835510H	A5835610H	00H	00H	00H	00H
16D4H	04H	A5835710H	A5835810H	A5835910H	A5835A10H	00H	00H	00H	00H
16D5H	04H	A5835B10H	A5835C10H	A5835D10H	A5835E10H	00H	00H	00H	00H
16D6H	04H	A5835F10H	A5836010H	A5836110H	A5836210H	00H	00H	00H	00H
16D7H	04H	A5836310H	A5836410H	A5836510H	A5836610H	00H	00H	00H	00H
16D8H	04H	A5836710H	A5836810H	A5836910H	A5836A10H	00H	00H	00H	00H
16D9H	04H	A5836B10H	A5836C10H	A5836D10H	A5836E10H	00H	00H	00H	00H
16DAH	04H	A5836F10H	A5837010H	A5837110H	A5837210H	00H	00H	00H	00H
16DBH	04H	A5837310H	A5837410H	A5837510H	A5837610H	00H	00H	00H	00H
16DCH	04H	A5837710H	A5837810H	A5837910H	A5837A10H	00H	00H	00H	00H
16DDH	04H	A5837B10H	A5837C10H	A5837D10H	A5837E10H	00H	00H	00H	00H
16DEH	04H	A5837F10H	A5838010H	A5838110H	A5838210H	00H	00H	00H	00H
16DFH	04H	A5838310H	A5838410H	A5838510H	A5838610H	00H	00H	00H	00H
16E0H	04H	A5838710H	A5838810H	A5838910H	A5838A10H	00H	00H	00H	00H
16E1H	04H	A5838B10H	A5838C10H	A5838D10H	A5838E10H	00H	00H	00H	00H
16E2H	04H	A5838F10H	A5839010H	A5839110H	A5839210H	00H	00H	00H	00H
16E3H	04H	A5839310H	A5839410H	A5839510H	A5839610H	00H	00H	00H	00H
16E4H	04H	A5839710H	A5839810H	A5839910H	A5839A10H	00H	00H	00H	00H
16E5H	04H	A5839B10H	A5839C10H	A5839D10H	A5839E10H	00H	00H	00H	00H
16E6H	04H	A5839F10H	A583A010H	A583A110H	A583A210H	00H	00H	00H	00H
16E7H	04H	A583A310H	A583A410H	A583A510H	A583A610H	00H	00H	00H	00H
16E8H	04H	A583A710H	A583A810H	A583A910H	A583AA10H	00H	00H	00H	00H
16E9H	04H	A583AB10H	A583AC10H	A583AD10H	A583AE10H	00H	00H	00H	00H
16EAH	04H	A583AF10H	A583B010H	A583B110H	A583B210H	00H	00H	00H	00H
16EBH	04H	A583B310H	A583B410H	A583B510H	A583B610H	00H	00H	00H	00H

Index	Subindex initial values								
	00H (Readable /Writable)	01H (Readable/ Writable)	02H (Readable/ Writable)	03H (Readable/ Writable)	04H (Readable/ Writable)	05H (Readable /Writable)	06H (Readable /Writable)	07H (Readable /Writable)	08H (Readable /Writable)
16ECH	04H	A583B710H	A583B810H	A583B910H	A583BA10H	00H	00H	00H	00H
16EDH	04H	A583BB10H	A583BC10H	A583BD10H	A583BE10H	00H	00H	00H	00H
16EEH	04H	A583BF10H	A583C010H	A583C110H	A583C210H	00H	00H	00H	00H
16EFH	04H	A583C310H	A583C410H	A583C510H	A583C610H	00H	00H	00H	00H
16F0H	04H	A583C710H	A583C810H	A583C910H	A583CA10H	00H	00H	00H	00H
16F1H	04H	A583CB10H	A583CC10H	A583CD10H	A583CE10H	00H	00H	00H	00H
16F2H	04H	A583CF10H	A583D010H	A583D110H	A583D210H	00H	00H	00H	00H
16F3H	04H	A583D310H	A583D410H	A583D510H	A583D610H	00H	00H	00H	00H
16F4H	04H	A583D710H	A583D810H	A583D910H	A583DA10H	00H	00H	00H	00H
16F5H	04H	A583DB10H	A583DC10H	A583DD10H	A583DE10H	00H	00H	00H	00H
16F6H	04H	A583DF10H	A583E010H	A583E110H	A583E210H	00H	00H	00H	00H
16F7H	04H	A583E310H	A583E410H	A583E510H	A583E610H	00H	00H	00H	00H
16F8H	04H	A583E710H	A583E810H	A583E910H	A583EA10H	00H	00H	00H	00H
16F9H	04H	A583EB10H	A583EC10H	A583ED10H	A583EE10H	00H	00H	00H	00H
16FAH	04H	A583EF10H	A583F010H	A583F110H	A583F210H	00H	00H	00H	00H
16FBH	04H	A583F310H	A583F410H	A583F510H	A583F610H	00H	00H	00H	00H
16FCH	04H	A583F710H	A583F810H	A583F910H	A583FA10H	00H	00H	00H	00H
16FDH	04H	A583FB10H	A583FC10H	A583FD10H	A583FE10H	00H	00H	00H	00H
16FEH	04H	A5840110H	A5840210H	A5840310H	A5840410H	00H	00H	00H	00H
16FFH	04H	A5840510H	A5840610H	A5840710H	A5840810H	00H	00H	00H	00H

TPDO communication parameter initial values

The following table lists the TPDO communication parameter initial values.

Index	Subindex initial values					
	00H (Readable)	01H (Readable/Writable)	02H (Readable/Writable)	03H (Readable/Writable)	04H	05H (Readable/Writable)
1800H	05H	40000180H + node ID	FEH	00H	System area	00H
1801H	05H	40000280H + node ID	FEH	00H	System area	00H
1802H	05H	40000380H + node ID	FEH	00H	System area	00H
1803H	05H	40000480H + node ID	FEH	00H	System area	00H
1804H to 18FFH	05H	C0000000H	FEH	00H	System area	00H

TPDO mapping parameter initial values

The following table lists the TPDO mapping parameter initial values.

Index	Subindex initial values								
	00H (Readable/ Writable)	01H (Readable/ Writable)	02H (Readable/ Writable)	03H (Readable/ Writable)	04H (Readable/ Writable)	05H (Readable/ Writable)	06H (Readable/ Writable)	07H (Readable/ Writable)	08H (Readable/ Writable)
1A00H	04H	A1000110H	A1000210H	A1000310H	A1000410H	00H	00H	00H	00H
1A01H	04H	A1000510H	A1000610H	A1000710H	A1000810H	00H	00H	00H	00H
1A02H	04H	A1000910H	A1000A10H	A1000B10H	A1000C10H	00H	00H	00H	00H
1A03H	04H	A1000D10H	A1000E10H	A1000F10H	A1001010H	00H	00H	00H	00H
1A04H	04H	A1001110H	A1001210H	A1001310H	A1001410H	00H	00H	00H	00H
1A05H	04H	A1001510H	A1001610H	A1001710H	A1001810H	00H	00H	00H	00H
1A06H	04H	A1001910H	A1001A10H	A1001B10H	A1001C10H	00H	00H	00H	00H
1A07H	04H	A1001D10H	A1001E10H	A1001F10H	A1002010H	00H	00H	00H	00H
1A08H	04H	A1002110H	A1002210H	A1002310H	A1002410H	00H	00H	00H	00H
1A09H	04H	A1002510H	A1002610H	A1002710H	A1002810H	00H	00H	00H	00H
1A0AH	04H	A1002910H	A1002A10H	A1002B10H	A1002C10H	00H	00H	00H	00H
1A0BH	04H	A1002D10H	A1002E10H	A1002F10H	A1003010H	00H	00H	00H	00H
1A0CH	04H	A1003110H	A1003210H	A1003310H	A1003410H	00H	00H	00H	00H
1A0DH	04H	A1003510H	A1003610H	A1003710H	A1003810H	00H	00H	00H	00H
1A0EH	04H	A1003910H	A1003A10H	A1003B10H	A1003C10H	00H	00H	00H	00H
1A0FH	04H	A1003D10H	A1003E10H	A1003F10H	A1004010H	00H	00H	00H	00H
1A10H	04H	A1004110H	A1004210H	A1004310H	A1004410H	00H	00H	00H	00H
1A11H	04H	A1004510H	A1004610H	A1004710H	A1004810H	00H	00H	00H	00H
1A12H	04H	A1004910H	A1004A10H	A1004B10H	A1004C10H	00H	00H	00H	00H
1A13H	04H	A1004D10H	A1004E10H	A1004F10H	A1005010H	00H	00H	00H	00H
1A14H	04H	A1005110H	A1005210H	A1005310H	A1005410H	00H	00H	00H	00H
1A15H	04H	A1005510H	A1005610H	A1005710H	A1005810H	00H	00H	00H	00H
1A16H	04H	A1005910H	A1005A10H	A1005B10H	A1005C10H	00H	00H	00H	00H
1A17H	04H	A1005D10H	A1005E10H	A1005F10H	A1006010H	00H	00H	00H	00H
1A18H	04H	A1006110H	A1006210H	A1006310H	A1006410H	00H	00H	00H	00H
1A19H	04H	A1006510H	A1006610H	A1006710H	A1006810H	00H	00H	00H	00H
1A1AH	04H	A1006910H	A1006A10H	A1006B10H	A1006C10H	00H	00H	00H	00H
1A1BH	04H	A1006D10H	A1006E10H	A1006F10H	A1007010H	00H	00H	00H	00H
1A1CH	04H	A1007110H	A1007210H	A1007310H	A1007410H	00H	00H	00H	00H
1A1DH	04H	A1007510H	A1007610H	A1007710H	A1007810H	00H	00H	00H	00H
1A1EH	04H	A1007910H	A1007A10H	A1007B10H	A1007C10H	00H	00H	00H	00H
1A1FH	04H	A1007D10H	A1007E10H	A1007F10H	A1008010H	00H	00H	00H	00H
1A20H	04H	A1008110H	A1008210H	A1008310H	A1008410H	00H	00H	00H	00H
1A21H	04H	A1008510H	A1008610H	A1008710H	A1008810H	00H	00H	00H	00H
1A22H	04H	A1008910H	A1008A10H	A1008B10H	A1008C10H	00H	00H	00H	00H
1A23H	04H	A1008D10H	A1008E10H	A1008F10H	A1009010H	00H	00H	00H	00H
1A24H	04H	A1009110H	A1009210H	A1009310H	A1009410H	00H	00H	00H	00H
1A25H	04H	A1009510H	A1009610H	A1009710H	A1009810H	00H	00H	00H	00H
1A26H	04H	A1009910H	A1009A10H	A1009B10H	A1009C10H	00H	00H	00H	00H
1A27H	04H	A1009D10H	A1009E10H	A1009F10H	A100A010H	00H	00H	00H	00H
1A28H	04H	A100A110H	A100A210H	A100A310H	A100A410H	00H	00H	00H	00H
1A29H	04H	A100A510H	A100A610H	A100A710H	A100A810H	00H	00H	00H	00H
1A2AH	04H	A100A910H	A100AA10H	A100AB10H	A100AC10H	00H	00H	00H	00H
1A2BH	04H	A100AD10H	A100AE10H	A100AF10H	A100B010H	00H	00H	00H	00H
1A2CH	04H	A100B110H	A100B210H	A100B310H	A100B410H	00H	00H	00H	00H
1A2DH	04H	A100B510H	A100B610H	A100B710H	A100B810H	00H	00H	00H	00H
1A2EH	04H	A100B910H	A100BA10H	A100BB10H	A100BC10H	00H	00H	00H	00H

Index	Subindex initial values								
	00H (Readable/ Writable)	01H (Readable/ Writable)	02H (Readable/ Writable)	03H (Readable/ Writable)	04H (Readable/ Writable)	05H (Readable/ Writable)	06H (Readable/ Writable)	07H (Readable/ Writable)	08H (Readable/ Writable)
1A2FH	04H	A100BD10H	A100BE10H	A100BF10H	A100C010H	00H	00H	00H	00H
1A30H	04H	A100C110H	A100C210H	A100C310H	A100C410H	00H	00H	00H	00H
1A31H	04H	A100C510H	A100C610H	A100C710H	A100C810H	00H	00H	00H	00H
1A32H	04H	A100C910H	A100CA10H	A100CB10H	A100CC10H	00H	00H	00H	00H
1A33H	04H	A100CD10H	A100CE10H	A100CF10H	A100D010H	00H	00H	00H	00H
1A34H	04H	A100D110H	A100D210H	A100D310H	A100D410H	00H	00H	00H	00H
1A35H	04H	A100D510H	A100D610H	A100D710H	A100D810H	00H	00H	00H	00H
1A36H	04H	A100D910H	A100DA10H	A100DB10H	A100DC10H	00H	00H	00H	00H
1A37H	04H	A100DD10H	A100DE10H	A100DF10H	A100E010H	00H	00H	00H	00H
1A38H	04H	A100E110H	A100E210H	A100E310H	A100E410H	00H	00H	00H	00H
1A39H	04H	A100E510H	A100E610H	A100E710H	A100E810H	00H	00H	00H	00H
1A3AH	04H	A100E910H	A100EA10H	A100EB10H	A100EC10H	00H	00H	00H	00H
1A3BH	04H	A100ED10H	A100EE10H	A100EF10H	A100F010H	00H	00H	00H	00H
1A3CH	04H	A100F110H	A100F210H	A100F310H	A100F410H	00H	00H	00H	00H
1A3DH	04H	A100F510H	A100F610H	A100F710H	A100F810H	00H	00H	00H	00H
1A3EH	04H	A100F910H	A100FA10H	A100FB10H	A100FC10H	00H	00H	00H	00H
1A3FH	04H	A100FD10H	A100FE10H	A1010110H	A1010210H	00H	00H	00H	00H
1A40H	04H	A1010310H	A1010410H	A1010510H	A1010610H	00H	00H	00H	00H
1A41H	04H	A1010710H	A1010810H	A1010910H	A1010A10H	00H	00H	00H	00H
1A42H	04H	A1010B10H	A1010C10H	A1010D10H	A1010E10H	00H	00H	00H	00H
1A43H	04H	A1010F10H	A1011010H	A1011110H	A1011210H	00H	00H	00H	00H
1A44H	04H	A1011310H	A1011410H	A1011510H	A1011610H	00H	00H	00H	00H
1A45H	04H	A1011710H	A1011810H	A1011910H	A1011A10H	00H	00H	00H	00H
1A46H	04H	A1011B10H	A1011C10H	A1011D10H	A1011E10H	00H	00H	00H	00H
1A47H	04H	A1011F10H	A1012010H	A1012110H	A1012210H	00H	00H	00H	00H
1A48H	04H	A1012310H	A1012410H	A1012510H	A1012610H	00H	00H	00H	00H
1A49H	04H	A1012710H	A1012810H	A1012910H	A1012A10H	00H	00H	00H	00H
1A4AH	04H	A1012B10H	A1012C10H	A1012D10H	A1012E10H	00H	00H	00H	00H
1A4BH	04H	A1012F10H	A1013010H	A1013110H	A1013210H	00H	00H	00H	00H
1A4CH	04H	A1013310H	A1013410H	A1013510H	A1013610H	00H	00H	00H	00H
1A4DH	04H	A1013710H	A1013810H	A1013910H	A1013A10H	00H	00H	00H	00H
1A4EH	04H	A1013B10H	A1013C10H	A1013D10H	A1013E10H	00H	00H	00H	00H
1A4FH	04H	A1013F10H	A1014010H	A1014110H	A1014210H	00H	00H	00H	00H
1A50H	04H	A1014310H	A1014410H	A1014510H	A1014610H	00H	00H	00H	00H
1A51H	04H	A1014710H	A1014810H	A1014910H	A1014A10H	00H	00H	00H	00H
1A52H	04H	A1014B10H	A1014C10H	A1014D10H	A1014E10H	00H	00H	00H	00H
1A53H	04H	A1014F10H	A1015010H	A1015110H	A1015210H	00H	00H	00H	00H
1A54H	04H	A1015310H	A1015410H	A1015510H	A1015610H	00H	00H	00H	00H
1A55H	04H	A1015710H	A1015810H	A1015910H	A1015A10H	00H	00H	00H	00H
1A56H	04H	A1015B10H	A1015C10H	A1015D10H	A1015E10H	00H	00H	00H	00H
1A57H	04H	A1015F10H	A1016010H	A1016110H	A1016210H	00H	00H	00H	00H
1A58H	04H	A1016310H	A1016410H	A1016510H	A1016610H	00H	00H	00H	00H
1A59H	04H	A1016710H	A1016810H	A1016910H	A1016A10H	00H	00H	00H	00H
1A5AH	04H	A1016B10H	A1016C10H	A1016D10H	A1016E10H	00H	00H	00H	00H
1A5BH	04H	A1016F10H	A1017010H	A1017110H	A1017210H	00H	00H	00H	00H
1A5CH	04H	A1017310H	A1017410H	A1017510H	A1017610H	00H	00H	00H	00H
1A5DH	04H	A1017710H	A1017810H	A1017910H	A1017A10H	00H	00H	00H	00H
1A5EH	04H	A1017B10H	A1017C10H	A1017D10H	A1017E10H	00H	00H	00H	00H
1A5FH	04H	A1017F10H	A1018010H	A1018110H	A1018210H	00H	00H	00H	00H
1A60H	04H	A1018310H	A1018410H	A1018510H	A1018610H	00H	00H	00H	00H

Index	Subindex initial values								
	00H (Readable/ Writable)	01H (Readable/ Writable)	02H (Readable/ Writable)	03H (Readable/ Writable)	04H (Readable/ Writable)	05H (Readable/ Writable)	06H (Readable/ Writable)	07H (Readable/ Writable)	08H (Readable/ Writable)
1A61H	04H	A1018710H	A1018810H	A1018910H	A1018A10H	00H	00H	00H	00H
1A62H	04H	A1018B10H	A1018C10H	A1018D10H	A1018E10H	00H	00H	00H	00H
1A63H	04H	A1018F10H	A1019010H	A1019110H	A1019210H	00H	00H	00H	00H
1A64H	04H	A1019310H	A1019410H	A1019510H	A1019610H	00H	00H	00H	00H
1A65H	04H	A1019710H	A1019810H	A1019910H	A1019A10H	00H	00H	00H	00H
1A66H	04H	A1019B10H	A1019C10H	A1019D10H	A1019E10H	00H	00H	00H	00H
1A67H	04H	A1019F10H	A101A010H	A101A110H	A101A210H	00H	00H	00H	00H
1A68H	04H	A101A310H	A101A410H	A101A510H	A101A610H	00H	00H	00H	00H
1A69H	04H	A101A710H	A101A810H	A101A910H	A101AA10H	00H	00H	00H	00H
1A6AH	04H	A101AB10H	A101AC10H	A101AD10H	A101AE10H	00H	00H	00H	00H
1A6BH	04H	A101AF10H	A101B010H	A101B110H	A101B210H	00H	00H	00H	00H
1A6CH	04H	A101B310H	A101B410H	A101B510H	A101B610H	00H	00H	00H	00H
1A6DH	04H	A101B710H	A101B810H	A101B910H	A101BA10H	00H	00H	00H	00H
1A6EH	04H	A101BB10H	A101BC10H	A101BD10H	A101BE10H	00H	00H	00H	00H
1A6FH	04H	A101BF10H	A101C010H	A101C110H	A101C210H	00H	00H	00H	00H
1A70H	04H	A101C310H	A101C410H	A101C510H	A101C610H	00H	00H	00H	00H
1A71H	04H	A101C710H	A101C810H	A101C910H	A101CA10H	00H	00H	00H	00H
1A72H	04H	A101CB10H	A101CC10H	A101CD10H	A101CE10H	00H	00H	00H	00H
1A73H	04H	A101CF10H	A101D010H	A101D110H	A101D210H	00H	00H	00H	00H
1A74H	04H	A101D310H	A101D410H	A101D510H	A101D610H	00H	00H	00H	00H
1A75H	04H	A101D710H	A101D810H	A101D910H	A101DA10H	00H	00H	00H	00H
1A76H	04H	A101DB10H	A101DC10H	A101DD10H	A101DE10H	00H	00H	00H	00H
1A77H	04H	A101DF10H	A101E010H	A101E110H	A101E210H	00H	00H	00H	00H
1A78H	04H	A101E310H	A101E410H	A101E510H	A101E610H	00H	00H	00H	00H
1A79H	04H	A101E710H	A101E810H	A101E910H	A101EA10H	00H	00H	00H	00H
1A7AH	04H	A101EB10H	A101EC10H	A101ED10H	A101EE10H	00H	00H	00H	00H
1A7BH	04H	A101EF10H	A101F010H	A101F110H	A101F210H	00H	00H	00H	00H
1A7CH	04H	A101F310H	A101F410H	A101F510H	A101F610H	00H	00H	00H	00H
1A7DH	04H	A101F710H	A101F810H	A101F910H	A101FA10H	00H	00H	00H	00H
1A7EH	04H	A101FB10H	A101FC10H	A101FD10H	A101FE10H	00H	00H	00H	00H
1A7FH	04H	A1020110H	A1020210H	A1020310H	A1020410H	00H	00H	00H	00H
1A80H	04H	A1020510H	A1020610H	A1020710H	A1020810H	00H	00H	00H	00H
1A81H	04H	A1020910H	A1020A10H	A1020B10H	A1020C10H	00H	00H	00H	00H
1A82H	04H	A1020D10H	A1020E10H	A1020F10H	A1021010H	00H	00H	00H	00H
1A83H	04H	A1021110H	A1021210H	A1021310H	A1021410H	00H	00H	00H	00H
1A84H	04H	A1021510H	A1021610H	A1021710H	A1021810H	00H	00H	00H	00H
1A85H	04H	A1021910H	A1021A10H	A1021B10H	A1021C10H	00H	00H	00H	00H
1A86H	04H	A1021D10H	A1021E10H	A1021F10H	A1022010H	00H	00H	00H	00H
1A87H	04H	A1022110H	A1022210H	A1022310H	A1022410H	00H	00H	00H	00H
1A88H	04H	A1022510H	A1022610H	A1022710H	A1022810H	00H	00H	00H	00H
1A89H	04H	A1022910H	A1022A10H	A1022B10H	A1022C10H	00H	00H	00H	00H
1A8AH	04H	A1022D10H	A1022E10H	A1022F10H	A1023010H	00H	00H	00H	00H
1A8BH	04H	A1023110H	A1023210H	A1023310H	A1023410H	00H	00H	00H	00H
1A8CH	04H	A1023510H	A1023610H	A1023710H	A1023810H	00H	00H	00H	00H
1A8DH	04H	A1023910H	A1023A10H	A1023B10H	A1023C10H	00H	00H	00H	00H
1A8EH	04H	A1023D10H	A1023E10H	A1023F10H	A1024010H	00H	00H	00H	00H
1A8FH	04H	A1024110H	A1024210H	A1024310H	A1024410H	00H	00H	00H	00H
1A90H	04H	A1024510H	A1024610H	A1024710H	A1024810H	00H	00H	00H	00H
1A91H	04H	A1024910H	A1024A10H	A1024B10H	A1024C10H	00H	00H	00H	00H
1A92H	04H	A1024D10H	A1024E10H	A1024F10H	A1025010H	00H	00H	00H	00H

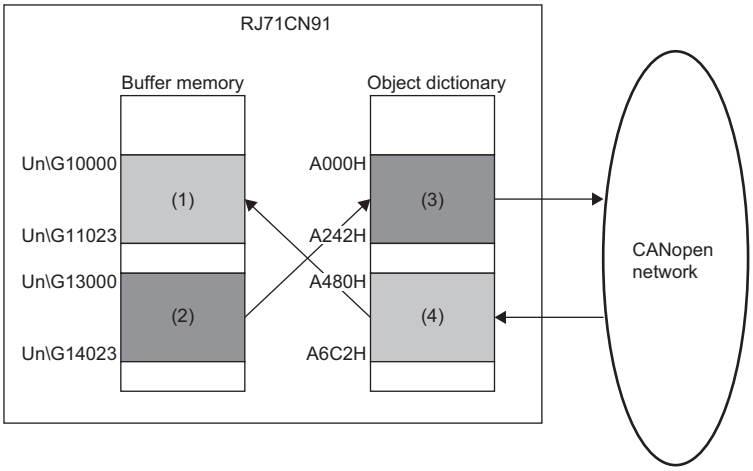
Index	Subindex initial values								
	00H (Readable/ Writable)	01H (Readable/ Writable)	02H (Readable/ Writable)	03H (Readable/ Writable)	04H (Readable/ Writable)	05H (Readable/ Writable)	06H (Readable/ Writable)	07H (Readable/ Writable)	08H (Readable/ Writable)
1A93H	04H	A1025110H	A1025210H	A1025310H	A1025410H	00H	00H	00H	00H
1A94H	04H	A1025510H	A1025610H	A1025710H	A1025810H	00H	00H	00H	00H
1A95H	04H	A1025910H	A1025A10H	A1025B10H	A1025C10H	00H	00H	00H	00H
1A96H	04H	A1025D10H	A1025E10H	A1025F10H	A1026010H	00H	00H	00H	00H
1A97H	04H	A1026110H	A1026210H	A1026310H	A1026410H	00H	00H	00H	00H
1A98H	04H	A1026510H	A1026610H	A1026710H	A1026810H	00H	00H	00H	00H
1A99H	04H	A1026910H	A1026A10H	A1026B10H	A1026C10H	00H	00H	00H	00H
1A9AH	04H	A1026D10H	A1026E10H	A1026F10H	A1027010H	00H	00H	00H	00H
1A9BH	04H	A1027110H	A1027210H	A1027310H	A1027410H	00H	00H	00H	00H
1A9CH	04H	A1027510H	A1027610H	A1027710H	A1027810H	00H	00H	00H	00H
1A9DH	04H	A1027910H	A1027A10H	A1027B10H	A1027C10H	00H	00H	00H	00H
1A9EH	04H	A1027D10H	A1027E10H	A1027F10H	A1028010H	00H	00H	00H	00H
1A9FH	04H	A1028110H	A1028210H	A1028310H	A1028410H	00H	00H	00H	00H
1AA0H	04H	A1028510H	A1028610H	A1028710H	A1028810H	00H	00H	00H	00H
1AA1H	04H	A1028910H	A1028A10H	A1028B10H	A1028C10H	00H	00H	00H	00H
1AA2H	04H	A1028D10H	A1028E10H	A1028F10H	A1029010H	00H	00H	00H	00H
1AA3H	04H	A1029110H	A1029210H	A1029310H	A1029410H	00H	00H	00H	00H
1AA4H	04H	A1029510H	A1029610H	A1029710H	A1029810H	00H	00H	00H	00H
1AA5H	04H	A1029910H	A1029A10H	A1029B10H	A1029C10H	00H	00H	00H	00H
1AA6H	04H	A1029D10H	A1029E10H	A1029F10H	A102A010H	00H	00H	00H	00H
1AA7H	04H	A102A110H	A102A210H	A102A310H	A102A410H	00H	00H	00H	00H
1AA8H	04H	A102A510H	A102A610H	A102A710H	A102A810H	00H	00H	00H	00H
1AA9H	04H	A102A910H	A102AA10H	A102AB10H	A102AC10H	00H	00H	00H	00H
1AAAH	04H	A102AD10H	A102AE10H	A102AF10H	A102B010H	00H	00H	00H	00H
1AABH	04H	A102B110H	A102B210H	A102B310H	A102B410H	00H	00H	00H	00H
1AACH	04H	A102B510H	A102B610H	A102B710H	A102B810H	00H	00H	00H	00H
1AADH	04H	A102B910H	A102BA10H	A102BB10H	A102BC10H	00H	00H	00H	00H
1AAEH	04H	A102BD10H	A102BE10H	A102BF10H	A102C010H	00H	00H	00H	00H
1AAFH	04H	A102C110H	A102C210H	A102C310H	A102C410H	00H	00H	00H	00H
1AB0H	04H	A102C510H	A102C610H	A102C710H	A102C810H	00H	00H	00H	00H
1AB1H	04H	A102C910H	A102CA10H	A102CB10H	A102CC10H	00H	00H	00H	00H
1AB2H	04H	A102CD10H	A102CE10H	A102CF10H	A102D010H	00H	00H	00H	00H
1AB3H	04H	A102D110H	A102D210H	A102D310H	A102D410H	00H	00H	00H	00H
1AB4H	04H	A102D510H	A102D610H	A102D710H	A102D810H	00H	00H	00H	00H
1AB5H	04H	A102D910H	A102DA10H	A102DB10H	A102DC10H	00H	00H	00H	00H
1AB6H	04H	A102DD10H	A102DE10H	A102DF10H	A102E010H	00H	00H	00H	00H
1AB7H	04H	A102E110H	A102E210H	A102E310H	A102E410H	00H	00H	00H	00H
1AB8H	04H	A102E510H	A102E610H	A102E710H	A102E810H	00H	00H	00H	00H
1AB9H	04H	A102E910H	A102EA10H	A102EB10H	A102EC10H	00H	00H	00H	00H
1ABAH	04H	A102ED10H	A102EE10H	A102EF10H	A102F010H	00H	00H	00H	00H
1ABBH	04H	A102F110H	A102F210H	A102F310H	A102F410H	00H	00H	00H	00H
1ABCH	04H	A102F510H	A102F610H	A102F710H	A102F810H	00H	00H	00H	00H
1ABDH	04H	A102F910H	A102FA10H	A102FB10H	A102FC10H	00H	00H	00H	00H
1ABEH	04H	A102FD10H	A102FE10H	A1030110H	A1030210H	00H	00H	00H	00H
1ABFH	04H	A1030310H	A1030410H	A1030510H	A1030610H	00H	00H	00H	00H
1AC0H	04H	A1030710H	A1030810H	A1030910H	A1030A10H	00H	00H	00H	00H
1AC1H	04H	A1030B10H	A1030C10H	A1030D10H	A1030E10H	00H	00H	00H	00H
1AC2H	04H	A1030F10H	A1031010H	A1031110H	A1031210H	00H	00H	00H	00H
1AC3H	04H	A1031310H	A1031410H	A1031510H	A1031610H	00H	00H	00H	00H
1AC4H	04H	A1031710H	A1031810H	A1031910H	A1031A10H	00H	00H	00H	00H

Index	Subindex initial values								
	00H (Readable/ Writable)	01H (Readable/ Writable)	02H (Readable/ Writable)	03H (Readable/ Writable)	04H (Readable/ Writable)	05H (Readable/ Writable)	06H (Readable/ Writable)	07H (Readable/ Writable)	08H (Readable/ Writable)
1AC5H	04H	A1031B10H	A1031C10H	A1031D10H	A1031E10H	00H	00H	00H	00H
1AC6H	04H	A1031F10H	A1032010H	A1032110H	A1032210H	00H	00H	00H	00H
1AC7H	04H	A1032310H	A1032410H	A1032510H	A1032610H	00H	00H	00H	00H
1AC8H	04H	A1032710H	A1032810H	A1032910H	A1032A10H	00H	00H	00H	00H
1AC9H	04H	A1032B10H	A1032C10H	A1032D10H	A1032E10H	00H	00H	00H	00H
1ACAH	04H	A1032F10H	A1033010H	A1033110H	A1033210H	00H	00H	00H	00H
1ACBH	04H	A1033310H	A1033410H	A1033510H	A1033610H	00H	00H	00H	00H
1ACCH	04H	A1033710H	A1033810H	A1033910H	A1033A10H	00H	00H	00H	00H
1ACDH	04H	A1033B10H	A1033C10H	A1033D10H	A1033E10H	00H	00H	00H	00H
1ACEH	04H	A1033F10H	A1034010H	A1034110H	A1034210H	00H	00H	00H	00H
1ACFH	04H	A1034310H	A1034410H	A1034510H	A1034610H	00H	00H	00H	00H
1AD0H	04H	A1034710H	A1034810H	A1034910H	A1034A10H	00H	00H	00H	00H
1AD1H	04H	A1034B10H	A1034C10H	A1034D10H	A1034E10H	00H	00H	00H	00H
1AD2H	04H	A1034F10H	A1035010H	A1035110H	A1035210H	00H	00H	00H	00H
1AD3H	04H	A1035310H	A1035410H	A1035510H	A1035610H	00H	00H	00H	00H
1AD4H	04H	A1035710H	A1035810H	A1035910H	A1035A10H	00H	00H	00H	00H
1AD5H	04H	A1035B10H	A1035C10H	A1035D10H	A1035E10H	00H	00H	00H	00H
1AD6H	04H	A1035F10H	A1036010H	A1036110H	A1036210H	00H	00H	00H	00H
1AD7H	04H	A1036310H	A1036410H	A1036510H	A1036610H	00H	00H	00H	00H
1AD8H	04H	A1036710H	A1036810H	A1036910H	A1036A10H	00H	00H	00H	00H
1AD9H	04H	A1036B10H	A1036C10H	A1036D10H	A1036E10H	00H	00H	00H	00H
1ADAH	04H	A1036F10H	A1037010H	A1037110H	A1037210H	00H	00H	00H	00H
1ADBH	04H	A1037310H	A1037410H	A1037510H	A1037610H	00H	00H	00H	00H
1ADCH	04H	A1037710H	A1037810H	A1037910H	A1037A10H	00H	00H	00H	00H
1ADDH	04H	A1037B10H	A1037C10H	A1037D10H	A1037E10H	00H	00H	00H	00H
1ADEH	04H	A1037F10H	A1038010H	A1038110H	A1038210H	00H	00H	00H	00H
1ADFH	04H	A1038310H	A1038410H	A1038510H	A1038610H	00H	00H	00H	00H
1AE0H	04H	A1038710H	A1038810H	A1038910H	A1038A10H	00H	00H	00H	00H
1AE1H	04H	A1038B10H	A1038C10H	A1038D10H	A1038E10H	00H	00H	00H	00H
1AE2H	04H	A1038F10H	A1039010H	A1039110H	A1039210H	00H	00H	00H	00H
1AE3H	04H	A1039310H	A1039410H	A1039510H	A1039610H	00H	00H	00H	00H
1AE4H	04H	A1039710H	A1039810H	A1039910H	A1039A10H	00H	00H	00H	00H
1AE5H	04H	A1039B10H	A1039C10H	A1039D10H	A1039E10H	00H	00H	00H	00H
1AE6H	04H	A1039F10H	A103A010H	A103A110H	A103A210H	00H	00H	00H	00H
1AE7H	04H	A103A310H	A103A410H	A103A510H	A103A610H	00H	00H	00H	00H
1AE8H	04H	A103A710H	A103A810H	A103A910H	A103AA10H	00H	00H	00H	00H
1AE9H	04H	A103AB10H	A103AC10H	A103AD10H	A103AE10H	00H	00H	00H	00H
1AEAH	04H	A103AF10H	A103B010H	A103B110H	A103B210H	00H	00H	00H	00H
1AEBH	04H	A103B310H	A103B410H	A103B510H	A103B610H	00H	00H	00H	00H
1AECH	04H	A103B710H	A103B810H	A103B910H	A103BA10H	00H	00H	00H	00H
1AEDH	04H	A103BB10H	A103BC10H	A103BD10H	A103BE10H	00H	00H	00H	00H
1AEEH	04H	A103BF10H	A103C010H	A103C110H	A103C210H	00H	00H	00H	00H
1AEFH	04H	A103C310H	A103C410H	A103C510H	A103C610H	00H	00H	00H	00H
1AF0H	04H	A103C710H	A103C810H	A103C910H	A103CA10H	00H	00H	00H	00H
1AF1H	04H	A103CB10H	A103CC10H	A103CD10H	A103CE10H	00H	00H	00H	00H
1AF2H	04H	A103CF10H	A103D010H	A103D110H	A103D210H	00H	00H	00H	00H
1AF3H	04H	A103D310H	A103D410H	A103D510H	A103D610H	00H	00H	00H	00H
1AF4H	04H	A103D710H	A103D810H	A103D910H	A103DA10H	00H	00H	00H	00H
1AF5H	04H	A103DB10H	A103DC10H	A103DD10H	A103DE10H	00H	00H	00H	00H
1AF6H	04H	A103DF10H	A103E010H	A103E110H	A103E210H	00H	00H	00H	00H

Index	Subindex initial values								
	00H (Readable/ Writable)	01H (Readable/ Writable)	02H (Readable/ Writable)	03H (Readable/ Writable)	04H (Readable/ Writable)	05H (Readable/ Writable)	06H (Readable/ Writable)	07H (Readable/ Writable)	08H (Readable/ Writable)
1AF7H	04H	A103E310H	A103E410H	A103E510H	A103E610H	00H	00H	00H	00H
1AF8H	04H	A103E710H	A103E810H	A103E910H	A103EA10H	00H	00H	00H	00H
1AF9H	04H	A103EB10H	A103EC10H	A103ED10H	A103EE10H	00H	00H	00H	00H
1AFAH	04H	A103EF10H	A103F010H	A103F110H	A103F210H	00H	00H	00H	00H
1AFBH	04H	A103F310H	A103F410H	A103F510H	A103F610H	00H	00H	00H	00H
1AFCH	04H	A103F710H	A103F810H	A103F910H	A103FA10H	00H	00H	00H	00H
1AFDH	04H	A103FB10H	A103FC10H	A103FD10H	A103FE10H	00H	00H	00H	00H
1AFEH	04H	A1040110H	A1040210H	A1040310H	A1040410H	00H	00H	00H	00H
1AFFH	04H	A1040510H	A1040610H	A1040710H	A1040810H	00H	00H	00H	00H

Standard interface profile

This section provides a brief description of the standard interface profile of the RJ71CN91 module and related information.
This area is an interface between the CANopen network and the RJ71CN91. This area has two types of objects: Network input variable object for input from the RJ71CN91 to the CANopen network, and network output variable object for output from the CANopen network to the RJ71CN91.
In the RJ71CN91, these objects are mapped to 'TPDO' (Un\G13000 to Un\G14023) and 'RPDO' (Un\G10000 to Un\G11023).
Turn on 'Data exchange request' (Y1) to exchange data between these objects and 'TPDO' (Un\G13000 to Un\G14023), and 'RPDO' (Un\G10000 to Un\G11023).
The RJ71CN91 uses these objects as PDO mapping destinations.



(1) 'RPDO' (Un\G10000 to Un\G11023)
(2) 'TPDO' (Un\G13000 to Un\G14023)
(3) Network input variable object (index A000H to A242H)
(4) Network output variable object (index A480H to A6C2H)

Network input variable

The following describes Network input variable object (index A000H to A242H) and the corresponding buffer memory areas.

■List of network input variable objects and their initial values

The following table lists the RJ71CN91 network input variable objects, data types, and initial values.

Index	Subindex	Description	Data type	Initial value*1	Read/Write*2
A000H to A007H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Signed 8-bit integer	I8	0	Read
A008H	00H	Maximum subindex	U8	10H	Read
	01H to 10H	Signed 8-bit integer	I8	0	Read
A040H to A047H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Unsigned 8-bit integer	U8	0	Read
A048H	00H	Maximum subindex	U8	10H	Read
	01H to 10H	Unsigned 8-bit integer	U8	0	Read
A0C0H to A0C3H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Signed 16-bit integer	I16	0	Read
A0C4H	00H	Maximum subindex	U8	08H	Read
	01H to 08H	Signed 16-bit integer	I16	0	Read
A100H to A103H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Unsigned 16-bit integer	U16	0	Read
A104H	00H	Maximum subindex	U8	08H	Read
	01H to 08H	Unsigned 16-bit integer	U16	0	Read
A1C0H to A1C1H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Signed 32-bit integer	I32	0	Read
A1C2H	00H	Maximum subindex	U8	04H	Read
	01H to 04H	Signed 32-bit integer	I32	0	Read
A200H to A201H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Unsigned 32-bit integer	U32	0	Read
A202H	00H	Maximum subindex	U8	04H	Read
	01H to 04H	Unsigned 32-bit integer	U32	0	Read
A240H to A241H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Float 32 bit	Real32	0	Read
A242H	00H	Maximum subindex	U8	04H	Read
	01H to 04H	Float 32 bit	Real32	0	Read

*1 Initial value when the CPU module is powered off and on or reset

*2 This indicates whether reading from and writing to the network are enabled.

■Mapping to the buffer memory area

The RJ71CN91 network input variable objects (index A000H to A242H) are mapped to the entire 'TPDO' (Un\G13000 to Un\G14023) for each data type. (☞ Page 176 RPDO (Un\G10000 to Un\G11023), TPDO (Un\G13000 to Un\G14023))

Network output variable

The following describes Network output variable object (index A480H to A6C2H) and the corresponding buffer memory areas.

■List of network output variable objects and their initial values

The following table lists the RJ71CN91 network output variable objects, data types, and initial values.

Index	Subindex	Description	Data type	Initial value*1	Read/Write*2
A480H to A487H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Signed 8-bit integer	I8	0	Read/Write
A488H	00H	Maximum subindex	U8	10H	Read
	01H to 10H	Signed 8-bit integer	I8	0	Read/Write
A4C0H to A4C7H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Unsigned 8-bit integer	U8	0	Read/Write
A4C8H	00H	Maximum subindex	U8	10H	Read
	01H to 10H	Unsigned 8-bit integer	U8	0	Read/Write
A540H to A543H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Signed 16-bit integer	I16	0	Read/Write
A544H	00H	Maximum subindex	U8	08H	Read
	01H to 08H	Signed 16-bit integer	I16	0	Read/Write
A580H to A583H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Unsigned 16-bit integer	U16	0	Read/Write
A584H	00H	Maximum subindex	U8	08H	Read
	01H to 08H	Unsigned 16-bit integer	U16	0	Read/Write
A640H to A641H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Signed 32-bit integer	I32	0	Read/Write
A642H	00H	Maximum subindex	U8	04H	Read
	01H to 04H	Signed 32-bit integer	I32	0	Read/Write
A680H to A681H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Unsigned 32-bit integer	U32	0	Read/Write
A682H	00H	Maximum subindex	U8	04H	Read
	01H to 04H	Unsigned 32-bit integer	U32	0	Read/Write
A6C0H to A6C1H	00H	Maximum subindex	U8	FEH	Read
	01H to FEH	Float 32 bit	Real32	0	Read/Write
A6C2H	00H	Maximum subindex	U8	04H	Read
	01H to 04H	Float 32 bit	Real32	0	Read/Write

*1 Initial value when the CPU module is powered off and on or reset

*2 This indicates whether reading from and writing to the network are enabled.

■Mapping to the buffer memory area

The RJ71CN91 network output variable objects are mapped to the entire 'RPDO' (Un\G10000 to Un\G11023) for each data type. (☞ Page 176 RPDO (Un\G10000 to Un\G11023), TPDO (Un\G13000 to Un\G14023))

Object dictionary details

Error register

• Index: 1001H

This object provides error information. The CANopen node maps internal errors into this object. It is part of the EMCY message.

• b0: Generic error
• b1: Current
• b2: Voltage
• b3: Temperature
• b4: Communication error (overrun error state)
• b5: Device profile specific
• b6: Fixed to 0
• b7: Manufacturer specific

The RJ71CN91 uses b0, b4, and b7. However, EMCY message transmission commands are excluded.

Bit b0 of this object will turn on if the EMCY error code value is larger than 00FFH.

This object can be cleared by Pre-defined error field (index 1003H). (☞ Page 211 Pre-defined error field)

Bits of this object can be set by the EMCY message transmission command. (☞ Page 54 When sending a message manually)

Pre-defined error field

• Index: 1003H
• Subindex: 01H to 02H

This object provides the errors that occurred in the RJ71CN91 and were notified via the EMCY message.

■Subindex 00H: Number of errors

This indicates the number of errors occurred. Setting this subindex to 0 clears all the emergency error codes stored in subindexes 01H to 0FH. Any value other than 0 cannot be used.

■Subindexes 01H to 0FH: Standard error field

A maximum of 15 latest emergency error codes sent by the RJ71CN91 are stored. EMCY messages are stored in the order starting from 01H. (☞ Page 52 Emergency error codes)

The data type of this object is U32. For that reason, when a manufacturer-specific error code is received, only the upper 2 byte-portion is stored. (Example: 4D45H)

COB-ID of SYNC message

• Index: 1005H

Set the COB-ID of the SYNC message. To quickly execute synchronization, the SYNC message is given a high priority CAN-ID.

Bit	Item	Description
b0 to b10	11-bit CAN-ID	11-bit CAN-ID of the CAN base frame
b11 to b29	Fixed to 0	
b30	gen.	0: Don't generate SYNC message 1: Generate SYNC message*1
b31	System area	

*1 To generate a SYNC message, the CANopen node must be the active NMT master.
Set Communication cycle period (index 1006H) before setting b30 of this object to 1.

Communication cycle period

• Index: 1006H

Set the transmission cycle of the SYNC message.

Set a 32-bit value in μs units. Note that a value smaller than 1ms will be adjusted to 1ms because the RJ71CN91 counts the time in ms units. Setting the value 0 to this object disables SYNC message generation. To generate a SYNC message, the RJ71CN91 must be the active NMT master.

• Setting range: 0 to 4294967295

Guard time

• Index: 100CH

This object indicates the waiting time for an NMT slave to return a response for node guarding.

Set a 16-bit value in ms units. Setting the value 0 to this object disables life guarding.

Life time factor

• Index: 100DH

Set this object to calculate the node life time for node guarding.

Set the value for the life time to be sufficiently larger than the value for the guard time on the NMT master side.

The node life time within which the NMT master polls an NMT slave can be calculated by multiplying this 8-bit object by the Guard time (index 100CH). Setting the value 0 to this object disables life guarding.

To use node guarding, set both this object and Guard time (index 100CH). The order in which Guard time and Life time factor are set does not matter.

Store parameters

• Index: 1010H

• Subindex: 01H

This object saves all parameters of the object dictionary in the flash ROM.

Write 65766173H (ISO8859 string code evas (save)) in subindex 01H to save the parameters. Saved parameters will be enabled when the CPU module is powered off and on or reset.

Read this object to read the following information.

• b0: Fixed to 1 (CANopen node saved parameters on command.)

• b1: Fixed to 0 (CANopen node did not save parameters automatically.)

• b2 to b31: System area

Point

Concise DCF (index 1F22H, subindex 01H to 7FH) need not be saved by this object because it is saved when Concise DCF (index 1F22H, subindex 01H to 7FH) is updated.

Restore default parameters

- Index: 1011H
- Subindex: 01H

This object resets all parameters of the object dictionary to the factory default settings.

Write 64616F6CH (ISO8859 string code daol (load)) in subindex 01H, power off and on or reset the CPU module to reset all parameters stored in the object dictionary of the RJ71CN91 to the factory default settings.

- b0: Fixed to 1 (CANopen node reset parameters to factory defaults on command.)
- b1 to b31: System area

Point

- Do not execute Store parameters (index 1010H, subindex 01H) before the CPU module is powered off and on or reset. Parameters reset to the factory default settings will be overwritten with the current settings again.
- CDCF saved in Concise DCF (index 1F22H, subindex 01H to 7FH) will also be cleared by this object.

COB-ID Time stamp

- Index: 1012H

Set the COB-ID of the TIME message.

Bit	Item	Description
b0 to b10	11-bit CAN-ID	11-bit CAN-ID of the CAN base frame
b11 to b29	Fixed to 0	
b30	Produce	0: Do not send TIME messages 1: Send TIME messages*1
b31	Consume	0: Do not receive TIME messages 1: Receive TIME messages

*1 To generate a TIME message, the CANopen node must be the active NMT master.

COB-ID EMCY

- Index: 1014H

The COB-ID of the EMCY message is stored.

Bit	Item	Description
b0 to b10	11-bit CAN-ID	11-bit CAN-ID of the CAN base frame
b11 to b30	Fixed to 0	
b31	Enabled	0: EMCY message generation enabled 1: EMCY message generation disabled

Point

For the RJ71CN91, the values of this object are fixed and cannot be changed.

Inhibit time EMCY

- Index: 1015H

Set the minimum transmission cycle for sending EMCY messages.

Set a 16-bit value in 100μ units. Setting the value 0 to this object disables the EMCY inhibit time.

Note that a value smaller than 1ms will be adjusted to 1ms internally because the RJ71CN91 counts the time in ms units.

Consumer heartbeat time

- Index: 1016H
- Subindex: 01H to 7FH

Set the CANopen node to monitor and the heartbeat time for monitoring the CANopen node in heartbeat.
Monitoring of the producer starts after reception of the first heartbeat message.
The heartbeat time set by this object must be larger than the one set by the corresponding Producer heartbeat time (index 1017H). (Approximately a value 1.5 times larger)
Before reception of the first heartbeat message, the status of the producer is unknown.

- b0 to b15: Heartbeat time in ms units
- b16 to b23: Node ID
- b24 to b31: Fixed to 0

If the heartbeat time is 0 or the node ID is 0 or greater than 127, the corresponding object entry is not used.

Producer heartbeat time

- Index: 1017H

Set the transmission cycle of heartbeat messages sent from the own node in heartbeat.
Set a 16-bit value in ms units. Setting the value 0 to this object disables heartbeat message transmission.

Verify Configuration

- Index: 1020H
- Subindex: 01H to 02H

This object is used to indicate the date and time when the NMT slave parameters are set on the configuration manager.
The configuration manager uses this object to verify whether reconfiguration of the NMT slave after it is reset is necessary. If the setting of the object dictionary of the NMT slave has been changed, subindexes 01H and 02H of this object are set to 0.
During NMT slave boot-up, the configuration manager compares RJ71CN91 settings of Expected configuration date (index 1F26H, subindex 01H to 7FH) and Expected configuration time (index 1F27H, subindex 01H to 7FH) with the NMT slave object entry corresponding to this object to determine if reconfiguration is necessary. This reduces the NMT slave boot-up time.

■Subindex 01H: Configuration date

This object stores the elapsed days since January 1, 1984.

■Subindex 02H: Configuration time

This object stores the elapsed time from midnight. (ms units)

Emergency consumer object

- Index: 1028H
- Subindex: 01H to 7FH

Set the COB-ID of EMCY messages to be received by the RJ71CN91. The subindex indicates the related node ID.

Bit	Item	Description
b0 to b10	11-bit CAN-ID	11-bit CAN-ID of the CAN base frame
b11 to b30	Fixed to 0	
b31	Enabled	0: EMCY message reception by remote node is valid. 1: EMCY message reception by remote node is invalid.

Error behavior

- Index: 1029H
- Subindex: 01H

Set operation at error occurrence.

Setting value	Description
00H	Switch NMT state to Pre-operational. (Only if the current NMT state is Operational)
01H	No change of the NMT state*1
02H	Switch NMT state to Stop.

*1 When the CPU module switches from RUN state to STOP state, the NMT state of the RJ71CN91 transitions to Pre-operational. Note that the module can transition NMT state back to Operational even if the CPU module is in the STOP state.

NMT inhibit time

- Index: 102AH

Set the minimum transmission cycle for sending node control messages. Set a 16-bit value in 100μs units. Setting the value 0 to this object disables the Inhibit time.

For the RJ71CN91, the setting range is fixed to 0.

RPDO communication parameter

- Index: 1400H to 14FFH
- Subindex: 01H to 02H

Set the COB-ID and transmission type of RPDO.

■Subindex 01H: COB-ID

Set the COB-ID of RPDO.

Bit	Item	Description
b0 to b10	11-bit CAN-ID	11-bit CAN-ID of the CAN base frame
b11 to b30	Fixed to 0	
b31	Enabled	0: Valid 1: Invalid

■Subindex 02H: Transmission type

Set the transmission type of RPDO.

Setting value	Description
00H to F0H	Synchronous setting Received data will be processed after the subsequent SYNC message is received.
FEH, FFH	Event-driven transmission (CANopen 405 mode)

For SDO communication, execute the following steps to change the RPDO communication parameter.

Step	Description	Object to be used	Remarks
1	Disable the target RPDO.	Subindex 01H of the RPDO communication parameter	—
2	Change the RPDO communication parameter if required.	Subindex 01H to 02H of the RPDO communication parameter	—
3	Enable the target RPDO.	Subindex 01H of the RPDO communication parameter	—

RPDO mapping parameter

- Index: 1600H to 16FFH
- Subindex: 01H to 08H

Set the objects to be mapped to RPDO.

In the initial setting, unsigned integer 16-bit objects are mapped to RPDO. (☞ Page 196 RPDO mapping parameter initial values)

Bit	Item	Description
b0 to b7	Length	Length of the mapped object (bits)
b8 to b15	Subindex	Subindex of the mapped object
b16 to b31	Index	Index of the mapped object

For SDO communication, execute the following steps to change the RPDO mapping parameter.

Step	Description	Object to be used	Remarks
1	Disable the target RPDO.	Subindex 01H of the RPDO communication parameter	—
2	Disable the RPDO mapping parameter.*1	Subindex 00H of the RPDO mapping parameter	—
3	Set the RPDO mapping parameter if required.	Subindex 01H to 08H of the RPDO mapping parameter	—
4	Enable the RPDO mapping parameter.*2	Subindex 00H of the RPDO mapping parameter	The subindexes become valid in sequence, starting from subindex 01H.
5	Enable the target RPDO.	Subindex 01H of the RPDO communication parameter	—

*1 Change the number of valid object entries to 0.

*2 For the number of valid object entries, set the number of subindexes to be made valid.

TPDO communication parameter

- Index: 1800H to 18FFH
- Subindex: 01H to 05H

Set the COB-ID and transmission type of TPDO.

■Subindex 01H: COB-ID

Bit	Item	Description
b0 to b10	11-bit CAN-ID	11-bit CAN-ID of the CAN base frame
b11 to b29	Fixed to 0	
b30	RTR*1	0: RTR permitted 1: RTR prohibited
b31	Enabled	0: Valid 1: Invalid

*1 This bit is constantly set to 1 in the RJ71CN91.

■Subindex 02H: Transmission type

Setting value	Description
00H	Synchronous (acyclic) If data was changed with 'Data exchange request' (Y1) on, data will be sent when the next SYNC message is received.
01H	Synchronous (Send data every time a SYNC message is received.)
02H	Synchronous (Send data when SYNC messages are received twice.)
03H	Synchronous (Send data when SYNC messages are received three times.)
⋮	⋮
F0H	Synchronous (Send data when SYNC messages are received 240 times.)
FEH, FFH	Event-driven transmission (CANopen 405 mode)

■Subindex 03H: Inhibit time

Set the minimum transmission cycle for sending data.

Set a value in 100μs units. The value 0 disables this setting.

Subindex 03H is used only if subindex 02H (transmission type) is set to Event-driven transmission (CANopen 405 mode).

It is not possible to turn on 'Data exchange request' (Y1) to send data until the set time passes.

■Subindex 05H: Event time

Set the maximum transmission cycle for sending data.

Set a value in ms units. The value 0 disables this setting.

Data will be sent regardless of whether the data is changed if there is no transmission by the time set by the Event timer.

For SDO communication, execute the following steps to change the TPDO communication parameter.

Step	Description	Object to be used	Remarks
1	Disable the target TPDO.	Subindex 01H of the TPDO communication parameter	—
2	Change the TPDO communication parameter if required.	Subindex 01H to 02H of the TPDO communication parameter	—
3	Enable the target TPDO.	Subindex 01H of the TPDO communication parameter	—

TPDO mapping parameter

- Index: 1A00H to 1AFFH
- Subindex: 01H to 08H

Set the objects to be mapped to TPDO.

In the initial setting, unsigned integer 16-bit objects are mapped to TPDO. (☞ Page 202 TPDO mapping parameter initial values)

Bit	Item	Description
b0 to b7	Length	Length of the mapped object (bits)
b8 to b15	Subindex	Subindex of the mapped object
b16 to b31	Index	Index of the mapped object

For SDO communication, execute the following steps to change the TPDO mapping parameter.

Step	Description	Object to be used	Remarks
1	Disable the target TPDO.	Subindex 01H of the TPDO communication parameter	—
2	Disable the TPDO mapping parameter.*1	Subindex 00H of the TPDO mapping parameter	—
3	Set the TPDO mapping parameter if required.	Subindex 01H to 08H of the TPDO mapping parameter	—
4	Enable the TPDO mapping parameter.*2	Subindex 00H of the TPDO mapping parameter	The subindexes become valid in sequence, starting from subindex 01H.
5	Enable the target TPDO.	Subindex 01H of the TPDO communication parameter	—

*1 Change the number of valid object entries to 0.

*2 For the number of valid object entries, set the number of subindexes to be made valid.

Concise DCF

- Index: 1F22H
- Subindex: 01H to 7FH

Use the configuration manager to store the configuration file that is used to set parameters of NMT slaves.

The configuration file is saved in the object in the CDCF format. The subindex indicates the corresponding node ID.

A maximum of 60 CDCFs can be registered with the RJ71CN91. The maximum size of a file is 65531 bytes.

Point

- To delete a subindex entry, write 0 with data length 1. During a write operation, a new CDCF cannot be saved. If the flash ROM is in busy state, an SDO error (SDO abort code 06060000H) will occur. (☞ Page 30 SDO abort codes)
- If an SDO error (SDO abort code 06010002H) occurs in the RJ71CN91 during the SDO write to the subindex, this subindex is already being used. Delete the subindex entry using the above procedure before writing. (☞ Page 30 SDO abort codes)
- If an SDO error (SDO abort code 06070010H) occurs in the RJ71CN91 during the SDO write to the subindex, the CDCF size is greater than 65531bytes or this subindex is already being used. Check the file size and delete the subindex entry using the above procedure. (☞ Page 30 SDO abort codes)
- If the CANopen configuration software cannot auto transfer a CDCF normally, a flash ROM busy error will occur. In this case, select and download the file. (If the NMT slave supports this function)
- All subindexes of Concise DCF (index 1F22H, subindex 01H to 7FH) can also be deleted by Restore default parameters (index 1011H, subindex 01H). (☞ Page 213 Restore default parameters)
- The own node ID subindex setting is not supported.
- The CDCF is saved directly in the flash ROM. Store parameters (index 1010H, subindex 01H) need not be operated. (☞ Page 212 Store parameters)

Configuration request

- Index: 1F25H
- Subindex: 01H to 80H

This object requests the configuration manager to configure a CANopen node.
Write 666E6F63H (ISO8859 string code fnoc (conf)) in the subindex corresponding to the node ID of the CANopen node to be configured. If the data is written in subindex 80H, all the CANopen nodes in the network that store CDCFs are requested for configuration. A configuration request for the own node ID is ignored and no error will occur.
If a CDCF is not saved in subindexes 01H to 7FH of the specified node ID, an SDO error (SDO abort code 08000024H) will occur. (☞ Page 30 SDO abort codes)
A configuration request for the own node ID is ignored.

Expected configuration date

- Index: 1F26H
- Subindex: 01H to 7FH

This object is used by CANopen configuration software to verify the setting date of the CANopen nodes in the network. This object stores the elapsed days since January 1, 1984.

Expected configuration time

- Index: 1F27H
- Subindex: 01H to 7FH

This object is used by CANopen configuration software to verify the setting time of the CANopen nodes in the network. This object stores the elapsed time (ms) from midnight.

NMT start-up

• Index: 1F80H

Set the startup operation of the own node. If the own node is not a flying master, the node starts as NMT master and ignores node control from the network targeting all nodes.

After setting the own node as a NMT master, it is necessary to store the parameters, and restart the own node by turning on 'Module restart request' (Y2) or by node reset.

Bit	Item	Description	Remarks
b0	NMT master	Set the own node type. 0: NMT slave 1: NMT master	• When this bit is 0, the settings of this object and the NMT slave assignment object (index 1F81H, subindex 01H to 7FH) will be ignored. • Only one active NMT master is allowed in the network.
b1	Start all nodes	Set the method to start the NMT slaves by sending NMT service. 0: Send Remote node start to each NMT slave. 1: Send Remote node start excluding NMT master.	When this bit is 1, don't start remote nodes which are not assigned to the NMT master via the NMT slave assignment (index 1F81H, subindex 01H to 7FH).
b2	NMT master start	Set whether to start the node automatically as the NMT master. 0: Shift automatically 1: Do not shift automatically	When this bit is 1, the NMT master has to be shifted manually into Operational. Execute SDO write in Request NMT (index 1F82H, subindex 01H to 80H). The start-up process will be suspended as long as the CANopen node is not in Operational.
b3	Start node	Set the startup method for NMT slaves. 0: The NMT master shall start the NMT slaves. 1: The NMT master does not start the NMT slaves. The NMT slaves are started up via the PLC application.	—
b4	Reset all nodes	Set whether to execute node reset if a mandatory slave fails to respond to node guarding or heartbeat. 0: Execute communication reset in only CANopen nodes where an error occurred. 1: Execute communication reset in all nodes.	• This setting is ignored if b6 is 1. • This setting is enabled only if the own node is the NMT master.
b5	Flying master	Set whether to use flying master in the own node. 0: Do not use flying master. 1: Use flying master.	• If a CANopen node does not shift to the NMT master in NMT flying master negotiation, the CANopen node shall operate as NMT slave. • When using flying master, all NMT masters in the network need to be set as flying masters.
b6	Stop all nodes	Set whether to execute remote node stop when a mandatory slave fails to respond to node guarding or heartbeat. 0: Do not stop all nodes. 1: Stop all nodes.	• If this bit is 1, b4 setting is ignored. • To restart the network, turn on 'Module restart request' (Y2), or execute SDO write in Request NMT (index 1F82H, subindex 01H to 80H) to reset the NMT master manually (execute communication reset or node reset excluding NMT master).

NMT slave assignment

- Index: 1F81H
- Subindex: 01H to 7FH

Set NMT slave assignment to the NMT master and node guarding.

Each subindex corresponds to the node ID of each CANopen node in the network. The subindex corresponding to the NMT master node ID will be ignored.

Set b0, b8 to b31 of this object to enable node guarding.

Bit	Item	Description
b0 to b7	Configuration	☞ Page 221 Configuration
b8 to b15	Retry factor	Set the number of resends by the NMT master in case a node guarding event occurs. The value 0 disables node guarding for CANopen nodes. • Setting range: 0 to 255
b16 to b31	Guard time	Set the guard time of node guarding for CANopen nodes (ms). The value 0 disables node guarding for CANopen nodes. If the heartbeat consumer object is set to other than 0, heartbeat will have priority over node guarding. • Setting range: 0 to 65535

■Configuration

Set the NMT slave assigned to the NMT master.

Bit	Item	Description	Remarks
b0	NMT slave	Set whether the remote node is an NMT slave under the RJ71CN91. 0: Remote node is NMT master or NMT slave that is not assigned 1: Remote node is NMT slave and assigned to this NMT master	• If the NMT master shall start up the NMT slave and/or execute node guarding to the NMT slave, it's mandatory to set this bit to 1. • Also when flying master is used and the active NMT master shall start up other NMT masters as NMT slaves, it's mandatory to set this bit to 1.
b1	System area		
b2	NMT boot slave	Specify whether the NMT master is allowed to execute configuration manager or remote node start when starting up the NMT slaves. 0: Not allowed 1: Allowed	—
b3	Mandatory slave	Set whether the CANopen node is a mandatory slave. 0: Not mandatory 1: Mandatory	For LSS slave, select this bit to enable the LSS service for this NMT slave.
b4	Communication reset condition	Set the execution condition for communication reset for the CANopen node. 0: No communication reset condition Always executable 1: Not executable only when the CANopen node is in Operational	—
b5 to b7	System area		

Request NMT

- Index: 1F82H
- Subindex: 01H to 80H

Set this object for node control.

Each subindex corresponds to the node ID of each CANopen node in the network. Set subindex 80H to control all nodes.

Only the NMT master is allowed to execute node control.

Read this object to read the NMT state of the node ID corresponding to the subindex.

Point

- The NMT states of other nodes that will be read are the NMT states obtained by the RJ71CN91 through the heartbeat function (when operating as a consumer) or the node guarding function (when operating as the active NMT master).
- If the above function is not used or the NMT states have not been obtained, "NMT state unknown" will be read. However, when the own node is the active NMT master, the NMT states directed by node control will be read.
- For the RJ71CN91, the NMT states that can be read from this object are also stored in the buffer memory. For details on the NMT state, refer to the following.

☞ Page 174 NMT state (Un\G601 to Un\G727)

Setting value	Description	
	Read	Write
00H	NMT state unknown	—
01H	CANopen node missing	—
04H	Stop	Executes remote node stop.
05H	Operational	Executes remote node start.
06H	—	Executes node reset.
07H	Remote node: Reserved Local node: Communication reset state	Executes communication reset.
7FH	Pre-operational	Executes Pre-operational transition.
84H	—	Executes remote node stop excluding NMT master. NMT slaves will be set to Stop, but the NMT master will stay in its current NMT state.
85H	—	Executes remote node start excluding NMT master. NMT slaves will be set to Operational, but the NMT master will stay in its current NMT state.
86H	—	Executes node reset excluding NMT master. NMT slaves will be set to Reset node, but the NMT master will stay in its current NMT state.
87H	—	Executes communication reset excluding NMT master. NMT slaves will be set to Reset communication, but the NMT master will stay in its current NMT state.
8FH	—	Executes pre-operational transition excluding NMT master NMT slaves will be set to Pre-operational, but the NMT master will stay in its current NMT state.

Request node guarding

- Index: 1F83H
- Subindex: 01H to 80H

Set the node guarding status for CANopen nodes in the network.
Also, node guarding can be stopped/started for each node. However, node guarding cannot be started if it is not set.
Each subindex corresponds to the node ID of each CANopen node in the network. Set subindex 80H to set the node guarding status of all nodes.
Read this object to read the node guarding status of the node ID corresponding to the subindex.

Setting value	Description	
	Read	Write
00H	Node guarding stop status	Node guarding stop
01H	Node guarding in progress	Node guarding start

Device Type

- Index: 1F84H
- Subindex: 01H to 7FH

To identify NMT slaves via the LSS master or NMT start-up, set the device type ID of the NMT slave.
The subindex corresponds to the node ID. Set this object to the same value as the index 1000H and subindex 00H of the object dictionary of the corresponding node ID.
If the value of this object is not 0, it is compared with the device type ID of the CANopen node in the network. If these values are inconsistent, an error event will be generated. If the value of this object is 0, it is not compared with the device type ID of the CANopen node in the network.
The value refers to the index 1000H, subindex 00H of the corresponding node ID.

Vendor identification

- Index: 1F85H
- Subindex: 01H to 7FH

To identify NMT slaves via the LSS master or NMT start-up, set the vendor ID of the NMT slave.
The subindex corresponds to the node ID. Set this object to the same value as the index 1018H and subindex 01H of the object dictionary of the corresponding node ID.
If the value of this object is not 0, it is compared with the vendor ID of the CANopen node in the network. If these values are inconsistent, an error event will be generated. If the value of this object is 0, it is not compared with the vendor ID of the CANopen node in the network.

Product code

- Index: 1F86H
- Subindex: 01H to 7FH

To identify NMT slaves via the LSS master or NMT start-up, set the product code of the NMT slave.
The subindex corresponds to the node ID. Set this object to the same value as the index 1018H and subindex 02H of the object dictionary of the corresponding node ID.
If the value of this object is not 0, it is compared with the product code of the CANopen node in the network. If these values are inconsistent, an error event will be generated. If the value of this object is 0, it is not compared with the product code of the CANopen node in the network.

Revision number

- Index: 1F87H
- Subindex: 01H to 7FH

To identify NMT slaves via the LSS master or NMT start-up, set the revision number of the NMT slave.
The subindex corresponds to the node ID. Set this object to the same value as the index 1018H and subindex 03H of the object dictionary of the corresponding node ID.
If the value of this object is not 0, it is compared with the revision number of the CANopen node in the network. If these values are inconsistent, an error event will be generated. If the value of this object is 0, it is not compared with the revision number of the CANopen node in the network.
Inconsistency is determined under the following cases.

- The major revision numbers are different.
- The minor revision number of the CANopen node is smaller than the minor revision number set in this object.

Serial number

- Index: 1F88H
- Subindex: 01H to 7FH

To identify NMT slaves via the LSS master or NMT start-up, set the serial number of the NMT slave.
The subindex corresponds to the node ID. Set this object to the same value as the index 1018H and subindex 04H of the object dictionary of the corresponding node ID.
If the value of this object is not 0, it is compared with the serial number of the CANopen node in the network. If these values are inconsistent, an error event will be generated. If the value of this object is 0, it is not compared with the serial number of the CANopen node in the network.

Boot time

- Index: 1F89H

This object stores the timeout time (ms) for boot-up of NMT slaves.
The time from the start of NMT slave boot to when all mandatory slaves are booted is measured. If the boot time passes before all mandatory slaves are booted, NMT start-up is stopped and disabled.
Setting the value 0 to this object disables this object.

- Setting range: 0 to 4294967295

NMT flying master timing parameters

- Index: 1F90H
- Subindex: 01H to 06H

Set parameters used for NMT flying master negotiation when the flying master function is used.

■Subindex 01H: Active NMT master detection timeout

Set the active NMT master response waiting time (ms).

■Subindex 02H: NMT master negotiation time delay

Set the waiting time (ms) before starting NMT master negotiation.

This waiting time is set to secure time to allow other devices to be initialized before deciding the active NMT master.

■Subindex 03H: Master priority level

Set the NMT master priority level.

Setting value	Description	
00H	Priority	High
01H		Mid
02H		Low

■Subindex 04H: Priority time slot

Set a coefficient used to calculate the response time of NMT flying master negotiation with the priority level used.

Note that the setting value set for "Priority time slot" must be greater than the setting value of "CANopen device time slot" × 127.

■Subindex 05H: Node time slot

Set a coefficient used to calculate the response time of NMT flying master negotiation with the node ID used.

■Subindex 06H: Multiplex master detection cycle time

Set the interval (ms) for sending protocol messages for NMT flying master negotiation.

Appendix 4 Operation of the RJ71CN91 When the CPU Module Operating Status Is Changed

When the operating status of the CPU module with which the RJ71CN91 is installed is changed, the communication status of the RJ71CN91 is also changed. (Including the case in which the CPU module goes into the STOP state due to a stop error.) The following tables show the details of how the RJ71CN91 communication status is changed.

CANopen 405 mode

■NMT slave

CPU module operating status	RJ71CN91 communication status	Remarks
STOP→RUN (Including the case in which the status is RUN immediately after startup)	Not changed	—
RUN→STOP (Including the case in which the status is STOP immediately after startup)	• The NMT state is changed to the state set by "Operation at Error Occurrence". • An EMCY message (manufacturer-specific error code: 4D45303032H) is sent.	The transmission/reception status for PDO, SDO, SYNC, EMCY, and TIME changes according to the NMT state.*1

*1 For details on the NMT state, refer to the following.
☞ Page 19 NMT state

■NMT master (when the flying master function is not used)

CPU module operating status	RJ71CN91 communication status	Remarks
STOP→RUN (Including the case in which the status is RUN immediately after startup)	• Start heartbeat. • Start operations as the NMT master.*1 • Start NMT startup.	—
RUN→STOP (Including the case in which the status is STOP immediately after startup)	• The NMT state is changed to the state set by "Operation at Error Occurrence". • Stop operations as the NMT master.*1 • Stop heartbeat. • An EMCY message (manufacturer-specific error code: 4D45303032H) is sent.	The transmission/reception status for PDO, SDO, SYNC, EMCY, and TIME changes according to the NMT state.*2

*1 The operations include NMT startup (including LSS and configuration manager), SYNC producer, TIME producer, and node guarding.
*2 For details on the NMT state, refer to the following.
☞ Page 19 NMT state

■NMT master (when the flying master function is used)

CPU module operating status	RJ71CN91 communication status	Remarks
STOP→RUN (Including the case in which the status is RUN immediately after startup)	• Start heartbeat. • Start operations as the NMT master.*1	• When the priority of the own node among the flying masters is the highest, the own node becomes the active NMT master. • The transmission/reception status for PDO, SDO, SYNC, EMCY, and TIME changes according to the NMT state.*2
RUN→STOP (Including the case in which the status is STOP immediately after startup)	• The NMT state is changed to the state set by "Operation at Error Occurrence". • Stop operations as the NMT master.*1 • Stop heartbeat. • An EMCY message (manufacturer-specific error code: 4D45303032H) is sent.	• When the own node was the active NMT master before STOP, a remaining flying master becomes the active NMT master. • The transmission/reception status for PDO, SDO, SYNC, EMCY, and TIME changes according to the NMT state.*2

*1 The operations include NMT startup (including LSS and configuration manager), SYNC producer, TIME producer, and node guarding.
*2 For details on the NMT state, refer to the following.
☞ Page 19 NMT state

Layer 2 message mode

CPU module operating status	RJ71CN91 communication status	Remarks
STOP→RUN (Including the case in which the status is RUN immediately after startup)	Not changed	—
RUN→STOP (Including the case in which the status is STOP immediately after startup)	• A CPU module STOP transition message is sent. • The offline mode becomes active.*1	After a CPU module STOP transition message is sent, the offline mode becomes active.

*1 When the CPU module goes into the STOP state and 'Layer 2 online mode request' (Y3) is cleared, the offline mode becomes active. If 'Layer 2 online mode request' (Y3) is retained, the state before the module goes into the STOP state is retained.

A

Appendix 5 How to Set Parameters When Not Using CANopen Configuration Tool

The following describes how to set parameters with the program in CANopen 405 mode without using CANopen Configuration Tool.

When using a project of the MELSEC-Q series for example, set parameters as follows.

Setting parameters

This section describes how to set parameters with the program or the device/buffer memory batch monitor in the engineering tool.

1. Create module parameters with the engineering tool. (☞ Page 74 When used in the CANopen 405 mode Steps 1 to 4)
2. To change the settings such as the function mode, baud rate, and node ID, set buffer memory areas and save data in the RJ71CN91.*1
3. When the settings such as the function mode, baud rate, and node ID have been changed, restart the RJ71CN91.*2
4. To change parameters in the object dictionary, set an object dictionary in SDO and save it into the RJ71CN91.*3

*1 For buffer memory areas to be set, refer to the following.
☞ Page 228 List of buffer memory areas where parameters are set
For details on how to save data into the RJ71CN91, refer to the following.
☞ Page 158 Configuration save/factory default configuration restore completed/request (X1F)/(Y1F),Page 162 Save/restore configuration (Un\G22)

*2 To restart the RJ71CN91, the following methods are available.
· Reset the CPU module or power off and on the system
· Turn on 'Module restart request' (Y2) (☞ Page 154 Module restart completed (X2), Module restart request (Y2))

*3 For parameters in an object dictionary to be set, refer to the following.
☞ Page 229 List of object dictionary to be set
For details on how to access the object dictionary, refer to the following.
☞ Page 29 SDO
To save data in the RJ71CN91, save in the flash ROM by using the Store parameters (index 1010H, subindex 01H). (☞ Page 212 Store parameters)

List of buffer memory areas where parameters are set

The following table lists the parameters to be set in the buffer memory.

For details on the buffer memory, refer to the following.

☞ Page 159 Buffer Memory

Buffer memory address	Name	Corresponding CANopen Configuration Tool parameter	Timing of changes being applied
Un\G21	Function mode	"Function mode" in the "CANopen Configuration" window	When the RJ71CN91 is restarted
Un\G24	Baud rate*1	"Baud-Rate" in the "CANopen Configuration" window	When the RJ71CN91 is restarted
Un\G27	Node ID*2	"Node ID" in the "CANopen Configuration" window	When the RJ71CN91 is restarted
Un\G59	Time stamp (daily correction)*2	Cannot be set	Immediately*3
Un\G70	NMT all nodes start delay time*2	"Start all nodes delay" in the "NMT master/slave" window	Immediately*3
Un\G71	SDO timeout*2	Cannot be set	Immediately*3

*1 Can be set or changed together with the function mode.
*2 Can be set or changed only when the function mode is CANopen 405 mode.
*3 Applied immediately when the RJ71CN91 detects a parameter change.

List of object dictionary to be set

The following table shows a list of parameters to be set in an object dictionary.

For details on the object dictionary, refer to the following.

☞ Page 191 Object Dictionary

Index	Subindex	Object	Corresponding CANopen Configuration Tool parameter
1005H	00H	COB-ID of SYNC message	Cannot be set
1006H	00H	Communication cycle period	"SYNC - Communication cycle period" in the "NMT master/slave" window
100CH	00H	Guard time	"Guard time" in the "NMT master/slave" window
100DH	00H	Life time factor	"Life time factor" in the "NMT master/slave" window
1012H	00H	COB-ID Time stamp	Cannot be set
1015H	00H	Inhibit time EMCY	Cannot be set
1016H	01H to 7FH	Consumer heartbeat time	"Consumer heartbeat time" in the "Heartbeat" window
1017H	00H	Producer heartbeat time	"Producer heartbeat time" in the "Heartbeat" window
1028H	01H to 7FH	Emergency consumer object	Cannot be set
1029H	01H	Error behavior	Cannot be set
1400H to 14FFH	01H to 02H	RPDO communication parameter*1	The following items in the RPDO details window • "COB-ID" • "Transmission type"
1600H to 16FFH	01H to 08H	RPDO mapping parameter*1	"Mapping Parameter" in the RPDO details window
1800H to 18FFH	01H to 05H	TPDO communication parameter*1	The following items in the TPDO details window • "COB-ID" • "Transmission type" • "Inhibit time" • "Event timer"
1A00H to 1AFFH	01H to 08H	TPDO mapping parameter*1	"Mapping Parameter" in the TPDO details window
1F22H	01H to 7FH	Concise DCF*2	"CANopen node settings" other than the local node setting
1F80H	00H	NMT start-up	The following items in the "NMT master/slave" window • "NMT master" • "Start all nodes" • "NMT master start" • "Start node" • "Reset all nodes" • "Stop all nodes" • "Flying master"
1F81H	01H to 7FH	NMT slave assignment	The following items in the "NMT slave assignment" window • "NMT slave" • "NMT boot slave" • "Mandatory" • "Reset communication" • "Retry factor" • "Guard time"
1F84H	01H to 7FH	Device Type	"Device type" in the "NMT slave assignment" window
1F85H	01H to 7FH	Vendor identification	"Vendor-ID" in the "NMT slave assignment" window
1F86H	01H to 7FH	Product code	"Product code" in the "NMT slave assignment" window
1F87H	01H to 7FH	Revision number	"Revision number" in the "NMT slave assignment" window
1F88H	01H to 7FH	Serial number	"Serial number" in the "NMT slave assignment" window
1F89H	00H	Boot time	"Boot time" in the "NMT master/slave" window
1F90H	01H to 06H	NMT flying master timing parameters	"Flying master timing parameter" in the "NMT master/slave" window

*1 Care should be taken in the procedure for changing a parameter. For details, refer to the following.
RPDO communication parameter: ☞ Page 215 RPDO communication parameter
RPDO mapping parameter: ☞ Page 216 RPDO mapping parameter
TPDO communication parameter: ☞ Page 217 TPDO communication parameter
TPDO mapping parameter: ☞ Page 218 TPDO mapping parameter

*2 Cannot be set by CIF.

Appendix 6 Added and Enhanced Functions

The following table lists the added and enhanced functions in the RJ71CN91.

Added and enhanced function	Firmware version of the RJ71CN91	Software version of CANopen Configuration Tool
Safety CPU supported	"02" or later	Version 1.01B or later

MEMO

INDEX

MEMO

REVISIONS

*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
July 2017	SH(NA)-081736ENG-A	First edition
September 2017	SH(NA)-081736ENG-B	■Added or modified parts Chapter 1, 2, 3, 5, Appendix 1, 2, 5
December 2019	SH(NA)-081736ENG-C	■Added or modified parts SAFETY PRECAUTIONS, TERMS, GENERIC TERMS AND ABBREVIATIONS, Section 1.3, 1.7, 2.1, 2.3, 3.1, 3.2, 3.3, Appendix 2, 6

Japanese manual number: SH-081735-C

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.

TRADEMARKS

CiA® and CANopen® are registered Community Trademarks of CAN in Automation e.V.

The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.

In some cases, trademark symbols such as '™' or '®' are not specified in this manual.

SH(NA)-081736ENG-C(1912)MEE
MODEL: RJ71CN91-U-OU-E
MODEL CODE: 13JX71

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN