MITSUBISHI ELECTRIC

Programmable Controller

MELSEC iQ-R series

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

-R04CPU
-R04ENCPU
-R08CPU
-R08ENCPU
-R08PCPU
-R08SFCPU
-R16CPU
-R16ENCPU
-R16PCPU
-R16SFCPU
-R32CPU
-R32ENCPU
-R32PCPU
-R32SFCPU
-R120CPU
-R120ENCPU
-R120PCPU
-R120SFCPU
-R6RFM
-R6SFM

SAFETY PRECAUTIONS

(Read these precautions before using this product.)

Before using MELSEC iQ-R series programmable controllers, please read the manuals for the product and the relevant manuals introduced in those manuals carefully, and pay full attention to safety to handle the product correctly.

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.

[Precautions for using Safety CPUs]

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

[Design Precautions]

⚠WARNING

- Create an interlock circuit which uses reset buttons so that the system does not restart automatically after executing safety functions and turning off outputs.
- In the case of a communication failure in the network, the status of the error station will be as follows:
  (1) All inputs from remote I/O stations are turned off.
  (2) All outputs from remote I/O stations are turned off.
  Check the communication status information and configure an interlock circuit in the program to ensure that the entire system will operate safely. Failure to do so may result in an accident due to an incorrect output or malfunction.
- Outputs may remain on or off due to a failure of the CC-Link IE Field Network remote I/O module (with safety functions). Configure an external circuit for monitoring output signals that could cause a serious accident.

[Precautions for using redundant function modules]

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

[Design Precautions]

⚠CAUTION

- Do not install the control lines or communication cables together with the main circuit lines or power cables. 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 in Program" for "Open Method Setting" of "Module Parameter". If "OPEN in Program" is selected, an execution of the remote STOP function causes the communication line to close. Consequently, the CPU module cannot reopen the line, and external devices cannot execute the remote RUN function.

[Precautions for using Safety CPUs]

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

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

[Precautions for using Safety CPUs]

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

[Wiring Precautions]

⚠ WARNING

- Shut off the external power supply (all phases) used in the system before installation and wiring. Failure to do so may result in electric shock or cause the module to fail or malfunction.
- After installation and wiring, attach the included terminal cover to the module before turning it on 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 terminal layout before wiring to the module, and connect the cables correctly. Connecting a power supply with a different voltage rating or incorrect wiring may cause fire or failure.
- Connectors for external devices must be crimped or pressed with the tool specified by the manufacturer, or must be correctly soldered.
- Securely connect the connector to the module. Poor contact may cause malfunction.
- Do not install the control lines or communication cables together with the main circuit lines or power cables. 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 cable may swing or inadvertently be pulled, resulting in damage to the module or cables or malfunction due to poor contact. Do not clamp the extension cables with the jacket stripped. Doing so may change the characteristics of the cables, resulting in malfunction.
- Check the interface type and correctly connect the cable. Incorrect wiring (connecting the cable to an incorrect interface) may cause failure of the module and external device.
- Tighten the terminal screws or connector screws within the specified torque range. Undertightening can cause drop of the screw, short circuit, fire, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
- When disconnecting the cable from the module, do not pull the cable by the cable part. For the cable with connector, hold the connector part of the cable. For the cable connected to the terminal block, loosen the terminal screw. Pulling the cable connected to the module may result in malfunction or damage to the module or cable.
- 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.

[Precautions for using redundant function modules]

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

[Startup and Maintenance Precautions]

⚠ WARNING

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

[Startup and Maintenance Precautions]

⚠ CAUTION

- When connecting an external device with a CPU module or intelligent function module to modify data of a running programmable controller, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification, parameter change, forced output, or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Improper operation may damage machines or cause accidents.
- Especially, when a remote programmable controller is controlled by an external device, immediate action cannot be taken if a problem occurs in the programmable controller due to a communication failure. To prevent this, configure an interlock circuit in the program, and determine corrective actions to be taken between the external device and CPU module in case of a communication failure.
- Do not disassemble or modify the modules. Doing so may cause failure, malfunction, injury, or a fire.
- Use any radio communication device such as a cellular phone or PHS (Personal Handy-phone System) 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 mount/remove the module to/from the base unit, and the terminal block to/from the module, and do not insert/remove the extended SRAM cassette to/from the CPU module more than 50 times (IEC 61131-2 compliant) respectively. Exceeding the limit may cause malfunction.
- After the first use of the product, do not insert/remove the SD memory card to/from the CPU module more than 500 times. Exceeding the limit may cause malfunction.
- Do not touch the metal terminals on the back side of the SD memory card. Doing so may cause malfunction or failure of the module.
- Do not touch the integrated circuits on the circuit board of an extended SRAM cassette. Doing so may cause malfunction or failure of the module.
- Do not drop or apply shock to the battery to be installed in the module. Doing so may damage the battery, causing the battery fluid to leak inside the battery. If the battery is dropped or any shock is applied to it, dispose of it without using.

[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 also may 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.

- For Safety CPUs

(1) Although MELCO has obtained the certification for Product's compliance to the international safety standards IEC61508, EN954-1/ISO13849-1 from TUV Rheinland, this fact does not guarantee that Product will be free from any malfunction or failure. The user of this Product shall comply with any and all applicable safety standard, regulation or law and take appropriate safety measures for the system in which the Product is installed or used and shall take the second or third safety measures other than the Product. MELCO is not liable for damages that could have been prevented by compliance with any applicable safety standard, regulation or law.

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

INTRODUCTION

Thank you for purchasing the Mitsubishi Electric MELSEC iQ-R series programmable controllers.
This manual describes the memory, functions, devices, and parameters of the relevant products listed below.
Before using this product, please read this manual and the relevant manuals carefully and develop familiarity with the functions and performance of the MELSEC iQ-R series programmable controller to handle the product correctly.
When applying the program examples provided in this manual to an actual system, ensure the applicability and confirm that it will not cause system control problems.
Please make sure that the end users read this manual.

Relevant products

R04CPU, R04ENCPU, R08CPU, R08ENCPU, R08PCPU, R08SFCPU, R16CPU, R16ENCPU, R16PCPU, R16SFCPU, R32CPU, R32ENCPU, R32PCPU, R32SFCPU, R120CPU, R120ENCPU, R120PCPU, R120SFCPU, R6RFM, R6SFM

CONTENTS

SAFETY PRECAUTIONS 1
CONDITIONS OF USE FOR THE PRODUCT 10
INTRODUCTION 11
RELEVANT MANUALS 26
TERMS 27

PART 1 PROGRAMMING

CHAPTER 1 RUNNING A PROGRAM 32

1.1 Scan Configuration 32
Initial processing (when powered on or switched to RUN) 32
I/O refresh 32
Program operation 33
END processing 33
1.2 Scan Time 33
Initial scan time 33
1.3 Data Communication and I/O Processing 34
Data communication 34
I/O processing and response delay 34
Refresh mode 35
Direct mode 37
1.4 Program Flow 40
1.5 Program Execution Type 41
Initial execution type program 41
Scan execution type program 42
Fixed scan execution type program 42
Event execution type program 47
Standby type program 51
Execution type change 52
Group setting for refresh 53
1.6 Program Type 54
Subroutine program 54
Interrupt program 55

CHAPTER 2 CPU MODULE OPERATING STATUS 71

2.1 Operation Processing According to Operating Status of the CPU Module 71
2.2 Operation Processing When Operating Status of the CPU Module Changed 72

CHAPTER 3 OPERATION PROCESSING AT MOMENTARY POWER FAILURE 73

CHAPTER 4 MEMORY CONFIGURATION OF THE CPU MODULE 75

4.1 Memory Configuration 75
Program memory/program cache memory 76
Device/label memory 77
Data memory 78
Refresh memory 78
CPU buffer memory 78

SD memory card . . . 78
4.2 File Size Unit in Memory . . . 79
4.3 Memory Operation . . . 80
Precautions . . . 80
4.4 Files . . . 81
File types and storage memory . . . 81
File operation available . . . 83
File size . . . 84

PART 2 FUNCTIONS

CHAPTER 5 SCAN MONITORING FUNCTION 89

5.1 Scan Time Monitoring Time Setting . . . 89
5.2 Watchdog Timer Reset . . . 89
5.3 Precautions . . . 90
Measurement error of watchdog timer . . . 90
Resetting the watchdog timer when repeatedly executing a program . . . 90
Scan time when the WDT reset instruction is used . . . 90

CHAPTER 6 CLOCK FUNCTION 91

6.1 Time Setting . . . 91
Clock data . . . 91
Changing the clock data . . . 91
Reading the clock data . . . 92
Precautions on the clock data . . . 92
6.2 Setting Time Zone . . . 93
6.3 Daylight Saving Time Function . . . 94
Daylight saving time setting . . . 94
Timing of daylight saving time adjustment . . . 95
Daylight saving time function operation check . . . 96
Operation of other functions using clock data during daylight saving time . . . 96
Precautions . . . 96
6.4 System Clock . . . 97
Special relay used for system clock . . . 97
Special register used for system clock . . . 97

CHAPTER 7 ONLINE CHANGE 98

7.1 Online Ladder Block Change . . . 98
Editable contents . . . 99
Range changeable in a single session . . . 99
Reserved area for online change . . . 100
Online ladder block change during the boot operation . . . 100
Setting the initial value for registering/changing label definition . . . 100
Precautions . . . 101

CHAPTER 8 INTERRUPT FUNCTION 106

8.1 Multiple Interrupt Function . . . 106
Interrupt priority . . . 106
Multiple interrupt execution sequence . . . 109

Precautions . . . 110

CHAPTER 9 PID CONTROL FUNCTION 111

CHAPTER 10 PROCESS CONTROL FUNCTION 113

10.1 Process Control by Using Process Control Function Blocks . . . 113
10.2 Process Control by Using Process Control Instructions . . . 114

CHAPTER 11 CONSTANT SCAN 115

11.1 Setting Constant Scan . . . 115
11.2 Accuracy of Constant Scan . . . 116
11.3 Precautions . . . 116

CHAPTER 12 DATABASE FUNCTION 117

12.1 Specifications . . . 118
12.2 Database Access Instruction . . . 120
Usage Procedure . . . 120
Creating Unicode Text Files . . . 121
Transactions for Databases . . . 124
Folder Configuration of Databases . . . 124
Timing of Database Update . . . 125
Precautions . . . 125
12.3 CPU Module Database Access (From External Device) Function . . . 126
Usage procedure . . . 126
Built-in database access setting . . . 128
Creating a database . . . 129
Adding a database to the ODBC data source . . . 130
Application example . . . 132
Precautions . . . 143

CHAPTER 13 REMOTE OPERATION 146

13.1 Remote RUN/STOP . . . 146
Executing method . . . 146
Precautions . . . 147
13.2 Remote PAUSE . . . 147
Executing method . . . 147
13.3 Setting RUN-PAUSE Contacts . . . 148
13.4 Remote RESET . . . 148
Enabling remote RESET . . . 148
Executing method . . . 148

CHAPTER 14 DEVICE/LABEL MEMORY AREA SETTING 149

14.1 Default Capacity of Each Area . . . 150
14.2 The Setting Range of the Capacity of Each Area . . . 151
14.3 Device/Label Memory Area Setting . . . 153
14.4 Device Setting . . . 154
Range of use of device points . . . 155

CHAPTER 15 INTERNAL BUFFER CAPACITY SETTING 156

CHAPTER 16 DEVICE/LABEL INITIAL VALUE SETTINGS 158
16.1 Setting Initial Device/Label Values. . . . 159
Setting initial device values . . . 159
Setting initial label values . . . 160
16.2 Applicable Devices/Labels . . . 161
16.3 Precautions . . . 161
CHAPTER 17 OUTPUT MODE AT OPERATING STATUS CHANGE (STOP TO RUN) 162
17.1 Operation When the Operating Status Is Changed from STOP to RUN . . . 162
Outputting the output (Y) status before STOP. . . . 162
Clearing the output (Y) (output after one scan) . . . 162
17.2 Setting the Output Mode. . . . 163
CHAPTER 18 LABEL ACCESS SETTING FROM EXTERNAL DEVICE 164
18.1 Configuration Procedure . . . 165
18.2 Label Communication Data Storage Location and Operation at the Time of Each Setting Operation . . 165
18.3 File Operation . . . 165
CHAPTER 19 ROUTING SETTING 166
19.1 Setting Method. . . . 166
19.2 Setting Example. . . . 167
19.3 Precautions . . . 167
CHAPTER 20 LATCH FUNCTION 168
20.1 Types of Latch . . . 168
20.2 Applicable Devices and Labels . . . 168
Applicable devices. . . . 168
Applicable labels . . . 169
20.3 Latch Setting . . . 169
Setting latch on devices. . . . 169
Setting latch on labels . . . 173
20.4 Clearing Latch Range Data. . . . 173
20.5 Precautions . . . 173
CHAPTER 21 MEMORY CARD FUNCTION 174
21.1 SD Memory Card Forced Disable. . . . 174
How to forcibly disable the SD memory card. . . . 174
Operation of functions accessing the SD memory card . . . 175
Canceling SD memory card forced disable . . . 175
Precautions . . . 175
21.2 Boot Operation . . . 176
Boot operation procedure . . . 176
Specifiable file types . . . 177
Allowed maximum number of boot files. . . . 177
Configuring the boot setting. . . . 178
Operation when security functions are enabled . . . 179
Precautions . . . 179
CHAPTER 22 DEVICE/LABEL ACCESS SERVICE PROCESSING SETTING 180
22.1 Operations Enabled by Setting Details . . . 181

CONTENTS

Execute the Process as Scan Time Proceeds . . . 182
Set Processing Time . . . 183
Set Processing Counts . . . 184
Execute END Processing between Programs . . . 185
22.2 Setting Method . . . 186
22.3 Precautions . . . 186

CHAPTER 23 DATA LOGGING FUNCTION 187

23.1 Data to Be Collected . . . 188
Number of data points . . . 188
Data type . . . 188
Data to be collected . . . 188
23.2 Data Collection Conditions . . . 189
Each scan . . . 189
Time specification . . . 189
Interrupt occurrence . . . 190
Condition specification . . . 191
23.3 Logging Type . . . 194
Operating procedure for continuous logging . . . 194
Operating procedure for trigger logging . . . 195
23.4 Trigger Condition . . . 196
Condition specification . . . 196
When trigger instruction executed . . . 197
23.5 States of the Data Logging Function . . . 198
Data logging states . . . 198
Storage states . . . 198
LED status . . . 199
23.6 Steps Until the Collected Data Is Saved . . . 200
Switching to a storage file . . . 200
23.7 Internal Buffer . . . 203
Internal buffer capacity setting . . . 203
Amount of internal buffer consumed . . . 203
23.8 Data Logging File . . . 203
Storage format of data logging files . . . 203
23.9 Missing Data . . . 204
Conditions under which missing data occurs . . . 204
23.10 Errors Generated During Data Logging . . . 204
23.11 Special Relay and Special Register Used by the Data Logging Function . . . 204
23.12 Other Functions . . . 205
Setting the behavior at the time of a transition to RUN . . . 205
Auto logging . . . 206
SD memory card replacement . . . 207
23.13 SD Memory Card Life When the Data Logging Function Is Used . . . 209
23.14 Precautions to Take When Using the Data Logging Function . . . 210

CHAPTER 24 MEMORY DUMP FUNCTION 216

24.1 Object Data . . . 216
Data to be collected . . . 216
24.2 Trigger Condition . . . 217
Device specification . . . 217

Error code specification . . . 218
Combining trigger conditions . . . 219
24.3 Procedure for Memory Dump . . . 219
24.4 Flow of Data Collection . . . 220
Effect on the scan time . . . 220
24.5 Memory Dump File . . . 221
Save file name . . . 221
24.6 States of the Memory Dump Function . . . 222
Memory dump status . . . 222
LED status . . . 222
24.7 Sizes of Files Used for the Memory Dump Function . . . 222
Capacity of the memory dump setting file . . . 222
Capacity of the memory dump file . . . 222
24.8 Special Relay and Special Register Used in the Memory Dump Function . . . 223
24.9 Precautions for the Memory Dump Function . . . 224

CHAPTER 25 REAL-TIME MONITOR FUNCTION 227

CHAPTER 26 RAS FUNCTIONS 229
26.1 Self-Diagnostics Function . . . 229
Self-diagnostics timing . . . 229
Check method of error . . . 229
CPU module operation upon error detection setting . . . 230
26.2 Error Clear . . . 233
26.3 Event History Function . . . 236
Event history settings . . . 236
Logging of the event history . . . 237
Viewing the event history . . . 239
Clearing the event history . . . 239
Precautions . . . 240
26.4 Program Cache Memory Auto Recovery Function . . . 241

CHAPTER 27 CPU MODULE DATA BACKUP/RESTORATION FUNCTION 242
27.1 Backup Function . . . 247
Backup processing triggered by turning on SM1351 . . . 249
Automatic backup using SD944 . . . 250
Checking backup errors . . . 252
Precautions . . . 253
27.2 Restoration Function . . . 256
Restoration processing triggered by turning on SM1354 . . . 258
Automatic restoration using SD955 . . . 259
Checking restoration errors . . . 259
Precautions . . . 260

CHAPTER 28 EXTERNAL INPUT/OUTPUT FORCED ON/OFF FUNCTION 265

CHAPTER 29 MULTIPLE CPU SYSTEM FUNCTION 272
29.1 Out-of-group I/O Fetch . . . 273
Accessing controlled module . . . 273
Accessing non-controlled module . . . 273

29.2 Operation Settings . . . 276
Stop setting . . . 276
Synchronous startup setting . . . 276
Clock data . . . 279
29.3 Multiple CPU Parameter Checking . . . 280
29.4 Data Communication Between CPU Modules . . . 281
Memory to be used . . . 281
Fixed scan communication setting . . . 286
Error detection setting . . . 286
Module-by-module data guarantee . . . 287
Communication through refresh . . . 289
Communication through direct access . . . 293
Data guarantee by program . . . 296
Communication between CPU modules in error state . . . 299
29.5 Multiple CPU Synchronous Interrupt . . . 300
Execution timing . . . 301
Multiple interrupt . . . 301
Precautions . . . 301

CHAPTER 30 SECURITY FUNCTION 302

CHAPTER 31 SEQUENCE SCAN SYNCHRONIZATION SAMPLING FUNCTION 304

CHAPTER 32 LABEL INITIALIZATION FUNCTION 306

32.1 Initialization of Labels After Rebuilt All (Reassignment) . . . 306
32.2 Label Initial Value Reflection Setting . . . 308

PART 3 DEVICES AND CONSTANTS

CHAPTER 33 DEVICES 312

33.1 Device List . . . 312
33.2 User Device . . . 314
Input (X) . . . 314
Output (Y) . . . 314
Internal relay (M) . . . 315
Latch relay (L) . . . 315
Link relay (B) . . . 315
Annunciator (F) . . . 315
Link special relay (SB) . . . 318
Edge relay (V) . . . 318
Step relay (S) . . . 319
Timer . . . 319
Counter . . . 325
Data register (D) . . . 327
Link register (W) . . . 327
Link special register (SW) . . . 328
33.3 System Device . . . 329
Function device (FX/FY/FD) . . . 329
Special relay (SM) . . . 330
Special register (SD) . . . 330

33.4 Link Direct Device . . . 331
Specification method . . . 331
Specification range . . . 331
Difference from link refresh . . . 332
33.5 Module Access Device . . . 333
Precautions . . . 334
33.6 CPU Buffer Memory Access Device . . . 334
Specification method . . . 334
33.7 Index Register (Z/LZ) . . . 335
Types of index registers . . . 335
Index register setting . . . 336
33.8 File Register (R/ZR) . . . 337
Specification method . . . 337
Setting file registers . . . 338
Clearing file registers . . . 339
33.9 Refresh Data Register (RD) . . . 339
Refresh memory setting . . . 339
33.10 Nesting (N) . . . 340
33.11 Pointer (P) . . . 341
Global pointer . . . 341
Local pointer . . . 342
Pointer setting . . . 343
33.12 Interrupt Pointer (I) . . . 344
Interrupt factors of the interrupt pointer numbers . . . 344
The priority for the interrupt pointer numbers and interrupt factors . . . 345
33.13 Other Devices . . . 346
Network No. specification device (J) . . . 346
I/O No. specification device (U) . . . 346
SFC block device (BL) . . . 346
SFC transition device (TR) . . . 346

CHAPTER 34 CONSTANTS 347

34.1 Decimal Constant (K) . . . 347
34.2 Hexadecimal Constant (H) . . . 347
34.3 Real Constant (E) . . . 348
Setting range for real numbers . . . 348
Activity when operation is in progress . . . 348
Expression on program . . . 348
34.4 Character String Constant . . . 349
Rules for special character . . . 349

CHAPTER 35 HOW TO USE DEVICES CONVENIENTLY 350

35.1 Global Device . . . 350
35.2 Local Device . . . 350
Devices available as local device . . . 350
Local device area . . . 351
When local device is used in subroutine program: . . . 351
When local device is used in interrupt and other programs: . . . 351
Clearing local device . . . 352
Setting method for the local devices . . . 353

Specification method for the local devices. . . . 354
Precautions . . . 354
35.3 Index Modification . . . 355
16-bit index modification . . . 355
32-bit index modification . . . 355
Device for which Index modification can be performed . . . 355
Combination of index modification. . . . 355
Precautions . . . 356
35.4 Indirect Specification . . . 358
Indirect address of device . . . 358
Devices that can allow indirect specification . . . 358

PART 4 WHEN BUILDING A REDUNDANT SYSTEM

CHAPTER 36 BASIC CONCEPT 360

36.1 System . . . 360
36.2 System Switching Between the Control System and Standby System . . . 361
36.3 System Consistency Check . . . 361
36.4 Operation Mode of the Process CPU . . . 361
36.5 Operation Modes of the Redundant System . . . 362
36.6 Scan Configuration . . . 363
36.7 Determination of Control System/Standby System . . . 364
When starting up both systems simultaneously . . . 364
When starting up one system first . . . 365
When one system is started automatically even though a tracking communication error has occurred . . . 367
When starting up the previous control system as the control system . . . 371
36.8 State Transition of the Redundant System . . . 373

CHAPTER 37 PROCEDURE FOR STARTING UP A REDUNDANT SYSTEM 374

37.1 Overview. . . . 374
Starting up both systems at a time . . . 374
Starting up the systems one by one . . . 376
Precautions on starting up the system when the data logging function is used . . . 379
37.2 Wiring . . . 380
Wiring the redundant function modules . . . 380
Wiring to the power supply modules in a redundant system . . . 381
37.3 Creating a Project . . . 382
37.4 Connecting a Personal Computer and a CPU Module . . . 382
37.5 System A/B Settings . . . 383
37.6 Writing Data to the Programmable Controller . . . 385
37.7 Monitoring the Program . . . 386

CHAPTER 38 FUNCTIONS 387

38.1 Operation Mode Change . . . 389
Switching procedure . . . 389
Precautions . . . 390
38.2 System Switching . . . 392
System switching method . . . 392
Operation at system switching. . . . 397
Execution availability of system switching . . . 398

Check method of system switching information . . . 400
Precautions . . . 402
38.3 Tracking Transfer. . . . 405
Tracking data . . . 406
Tracking block and tracking trigger . . . 409
Setting procedure for tracking transfer . . . 410
Tracking transfer setting . . . 411
Tracking mode . . . 415
Precautions . . . 418
38.4 Memory Copy from Control System to Standby System . . . 420
Automatic memory copy . . . 423
Memory copy with the engineering tool . . . 424
Memory copy with the special relay and special register . . . 425
Precautions . . . 427
38.5 System Consistency Check . . . 428
File . . . 430
Operating status . . . 431
Mounting status of main base unit . . . 431
SD memory card . . . 431
38.6 Program Execution in Both Systems . . . 433
Operation of a program executed in both systems . . . 435
Precautions . . . 437
38.7 Redundant System Operation Setting . . . 439
Standby system output setting . . . 440
38.8 Redundant Function Module Communication Test . . . 441
Execution procedure of the module communication test . . . 441
38.9 Clock Function. . . . 442
Time synchronization . . . 442
38.10 Online Change . . . 442
38.11 Constant scan . . . 443
38.12 Remote Operation . . . 443
Remote RUN/Remote STOP/Remote PAUSE . . . 443
Remote RESET . . . 445
Precautions . . . 445
38.13 Boot Operation . . . 446
38.14 SLMP communication . . . 447
38.15 Data Logging Function . . . 448
Data logging states . . . 448
LED status . . . 449
Data collection conditions . . . 449
Trigger logging . . . 449
Switching to a storage file . . . 450
Writing the data logging setting . . . 450
Auto logging function . . . 450
38.16 Clearing an Error on the Standby System CPU Module from the Control System CPU Module . . . 451

CHAPTER 39 PRECAUTIONS ON PROGRAMMING 452

39.1 Instructions Not Available in Redundant System . . . 452
Instructions that cause stop errors . . . 452
Instructions that need to be executed again in a new control system . . . 452

Instructions whose operations vary depending on tracking of the signal flow memory . . . 455
Instructions that affect the status of another instruction when executed . . . 457
Instruction that causes different operation results between the control system and standby system . . . 457
Precautions for using the COM or ZCOM instruction . . . 458
Precautions for using the ADRSET instruction . . . 458
39.2 Interrupt from Modules . . . 459
System switching in backup mode . . . 459
System switching in separate mode . . . 459
When the operation mode is switched to the separate mode . . . 459
When the operation mode is switched to the backup mode . . . 460
39.3 Precautions for Using the Annunciator (F) . . . 461
39.4 Precautions on Timers and Timer Function Blocks . . . 462
39.5 Precautions on Access to Intelligent Function Module or External Devices . . . 463
39.6 Precautions on Writing Data from GOT or External Devices . . . 464
39.7 Precautions on Outputting in the Middle of the Scan . . . 465

CHAPTER 40 MAINTENANCE AND INSPECTION FOR A REDUNDANT SYSTEM 466

40.1 Module Replacement in the Redundant System . . . 466
Replacing a CPU module . . . 467
Replacing a power supply module . . . 469
Replacing a redundant power supply module . . . 469
Replacing a redundant function module . . . 470
Replacing I/O modules . . . 470
Replacing a network module . . . 471
Replacing a main base unit . . . 471

PART 5 WHEN USING THE SAFETY CPU

CHAPTER 41 RUNNING A SAFETY PROGRAM 474

41.1 Overview . . . 474
41.2 Safety Program . . . 475
41.3 Safety Cycle Time . . . 477
Setting method . . . 477
41.4 Safety CPU Operating Status . . . 478
Operation processing based on safety CPU operating status . . . 478
Operation processing when operating status of the Safety CPU changes . . . 478

CHAPTER 42 MEMORY SPECIFICATIONS 479

42.1 Memory Configuration . . . 479
Device/label memory . . . 479
42.2 File Size Unit in Memory . . . 481
42.3 Memory Operation . . . 481
42.4 Files . . . 482
File types and storage memory . . . 482
File operation available . . . 483
File size . . . 485

CHAPTER 43 FUNCTIONS 486

43.1 Safety Operation Mode . . . 488

Checking the safety operation mode . . . 488
Switching the safety operation mode . . . 489
Operations restricted in SAFETY MODE . . . 491
43.2 Continuous RUN Prevention in TEST MODE . . . 492
Measuring the continuous RUN time in TEST MODE . . . 492
Setting method . . . 492
Checking the continuous RUN time in TEST MODE . . . 492
43.3 Safety Diagnostic Function . . . 493
43.4 Safety Data Identify Check . . . 494
43.5 Safety Communication Function . . . 494
43.6 Device/Label Memory Area Setting . . . 495
Default capacity . . . 495
Setting range of capacity of each area . . . 496
Setting method . . . 498
43.7 Data Logging Function . . . 500
Data to be collected . . . 500
Data collection conditions . . . 500
Trigger conditions . . . 500
Output settings . . . 500
Auto logging . . . 500
43.8 Self-Diagnostic Function . . . 501
How to check errors . . . 501
Operation setting in the CPU parameter (CPU Module Operation Setting at Error Detected) . . . 501
43.9 Error Clear . . . 502

CHAPTER 44 SAFETY DEVICES AND CONSTANTS 504

44.1 Safety Devices . . . 504
List of safety devices . . . 504
Safety user devices . . . 505
Safety system devices . . . 507
44.2 Constants . . . 508
44.3 How to Use Safety Devices Conveniently . . . 508
Safety global devices . . . 508
Safety local devices . . . 508

APPENDICES 510

Appendix 1 Error Codes . . . 510
Error code system . . . 510
Operation when an error occurs . . . 512
Safety output operation when error occurs . . . 512
How to clear errors . . . 512
List of error codes . . . 512
Appendix 2 Event List . . . 594
How to read the event list . . . 594
Event list . . . 596
Appendix 3 Troubleshooting by Symptom . . . 602
When the POWER LED of the power supply module turns off . . . 602
When the READY LED of the CPU module turns off . . . 602
When an error has occurred in a redundant function module . . . 603
When the specific extension base unit cannot be recognized . . . 603

When the specific Q series extension base unit cannot be recognized . . . 603
When data cannot be written to the programmable controller . . . 604
When data cannot be read from the programmable controller . . . 604
When the operating status of the CPU module cannot be changed . . . 604
When the Ethernet function cannot be used . . . 604
When the CC-Link IE Field Network Basic function cannot be used . . . 604
When the CPU module database access function cannot be used . . . 605
Appendix 4 List of Special Relay Areas . . . 606
Appendix 5 List of Special Register Areas . . . 635
Appendix 6 List of Safety Special Relay Areas . . . 680
Appendix 7 List of Safety Special Register Areas . . . 682
Appendix 8 Buffer Memory . . . 687
List of buffer memory addresses . . . 687
Details on buffer memory addresses . . . 690
Appendix 9 Processing Time . . . 694
Instruction execution time . . . 694
Program execution time . . . 694
END processing time . . . 696
Data logging function processing time . . . 707
Memory dump function processing time . . . 708
Real-time monitor function processing time . . . 708
Database function processing time . . . 709
Process control function processing time . . . 711
SFC program processing time . . . 711
SFC program switching . . . 713
Processing time by the redundant function . . . 714
Appendix 10 Parameter List . . . 721
System parameters . . . 721
CPU Parameters . . . 722
Module parameters . . . 723
Safety CPU parameters . . . 723
Memory card parameters . . . 723
Appendix 11 List of the Targets of Device/Label Access Service Processing Setting . . . 724
Communication functions via the MC protocol . . . 724
Communication function using an engineering tool . . . 725
Appendix 12 How to Use CPU Module Logging Configuration Tool . . . 726
System configuration . . . 726
Specifications . . . 729
Data logging procedure . . . 740
Menu items and setting methods . . . 741
Appendix 13 List of Available SQL Commands for CPU Module Database Access Function . . . 768
Data Definition . . . 768
Data Operation . . . 770
Clause . . . 772
Aggregate Function . . . 774
Arithmetic Function . . . 775
Appendix 14 Added and Enhanced Functions . . . 779

INDEX 783

REVISIONS . . . 787

WARRANTY 789
TRADEMARKS 792

RELEVANT MANUALS

Manual name [manual number]	Description	Available form
MELSEC iQ-R CPU Module User's Manual (Application) [SH-081264ENG] (this manual)	Memory, functions, devices, and parameters of the CPU module	Print book
		e-Manual PDF
MELSEC iQ-R CPU Module User's Manual (Startup) [SH-081263ENG]	Specifications, procedures before operation, and troubleshooting of the CPU module	Print book
		e-Manual PDF
MELSEC iQ-R Ethernet User's Manual (Application) [SH-081257ENG]	Functions, parameter settings, programming, troubleshooting, I/O signals, and buffer memory of Ethernet	Print book
		e-Manual PDF
MELSEC iQ-R Programming Manual (Program Design) [SH-081265ENG]	Program specifications, such as ladder programs, ST programs, FBD/LD programs, and SFC programs and labels	e-Manual PDF
MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks) [SH-081266ENG]	Instructions for the CPU module, dedicated instructions for the intelligent function modules, and standard functions/ function blocks	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
Backup mode	A mode for normal operation in a redundant system
Buffer memory	Memory in an intelligent function module for storing data such as setting values and monitored values. When integrated into the CPU module, this memory refers to a memory for storing data such as setting values and monitored values of the Ethernet function, and data used for data communication of the multiple CPU system function.
CC-Link IE Field Network-equipped master/local module	A generic term for the following modules when the CC-Link IE Field Network function is used: • RJ71GF11-T2 • RJ71EN71 • RnENCPU
Control CPU	A CPU module that controls connected I/O modules and intelligent function modules. The multiple CPU system allows the user to assign this control to any CPU module on a module-by-module basis.
Control system	A system that controls a redundant system and performs network communications in a redundant system
Control system execution program	A program that is executed only in the CPU module of the control system.
CPU module	A generic term for the MELSEC iQ-R series CPU module
CPU Module Logging Configuration Tool	Software to configure data logging settings and to manage collected data
Dedicated instruction	An instruction for using functions of the module
Device	A device (X, Y, M, D, or others) in a CPU module
Device supporting iQSS	A generic term for a device which supports iQ Sensor Solution. For iQ Sensor Solution, refer to the following. 📖 iQ Sensor Solution Reference Manual
Engineering tool	The product name of the software package for the MELSEC programmable controllers
Ethernet interface module with built-in CC-Link IE	Another term for the RJ71EN71
Ethernet-equipped module	A generic term for the following modules when the Ethernet function is used: • RJ71EN71 • CPU module
Global label	A label that is valid for all the program data when multiple program data are created in the project. There are two types of global label: a module specific label (module label), which is generated automatically by GX Works3, and an optional label, which can be created for any specified device.
GX LogViewer	Software to display data collected by data logging
I/O module	A generic term for the input module, output module, I/O combined module, and interrupt module
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.
Network module	A generic term for the following modules: • Ethernet interface module • CC-Link IE Controller Network module • Module on CC-Link IE Field Network • MELSECNET/H network module • MELSECNET/10 network module • RnENCPU (network part)
New control system	A system that has switched to control system from standby system after system switching
New standby system	A system that has switched to standby system from control system after system switching
Own system	The system that contains the CPU module that is used for descriptions.
POU	A unit that configures a program. Units are categorized and provided in accordance with functions. A program that is composed of POUs (program organization units) allows the lower-layer processing, when the program is multi-layered, to be divided into several units by processing and function, enabling the creation of programs based on each unit.
Process CPU	A generic term for the R08PCPU, R16PCPU, R32PCPU, and R120PCPU
Process CPU (process mode)	A Process CPU operating in process mode. Process control function blocks and the online module change function can be executed.
Process CPU (redundant mode)	A Process CPU operating in redundant mode. A redundant system is configured with this CPU module. Process control function blocks and the online module change function can be used even in this mode.

Term	Description
Program block	A group of POUs that configure a program
Program executed in both systems	A program that is executed in both CPU modules of the control system and the standby system
Programmable controller CPU	A generic term for the R04CPU, R04ENCPU, R08CPU, R08ENCPU, R16CPU, R16ENCPU, R32CPU, R32ENCPU, R120CPU, and R120ENCPU
RAS	The abbreviation for Reliability, Availability, and Serviceability. This term refers to the overall usability of automated equipment.
Redundant function module	Another term for the R6RFM. This module is used with the Process CPU (redundant mode) as a pair and configures a redundant system.
Redundant system	A system that consists of two basic systems that have a CPU module, a power supply module, a network module, or other modules for each of them. Even if an error occurs in one system, control is continued with the other system. For details, refer to the descriptions of the redundant system in the following manual. MELSEC iQ-R Module Configuration Manual
RnCPU	A generic term for the R04CPU, R08CPU, R16CPU, R32CPU, and R120CPU
RnENCPU	A generic term for the R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, and R120ENCPU
RnENCPU (CPU part)	A module on the left-hand side of the RnENCPU (MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup))
RnENCPU (network part)	A module on the right-hand side of the RnENCPU (MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup))
Safety CPU	A generic term for the R08SFCPU, R16SFCPU, R32SFCPU, and R120SFCPU. This module is used with a safety function module as a pair, and performs both standard control and safety control.
Safety function module	Another term for the R6SFM This module is used with a Safety CPU as a pair and performs safety control. The module can only be paired with the Safety CPU.
Separate mode	A mode for system maintenance in a redundant system. This mode can maintain a redundant system without stopping control while the system is running.
Signal flow	The execution status that the last time an operation of a program or an FB is executed in each step.
Standard/safety shared label	A label that can be used in both standard programs and safety programs. This label is used at data communications between a safety program and standard program.
Standby system	A backup system in a redundant system
Station sub ID number	The abbreviation for the ID number of a sensor connected to a CC-Link-compatible communication module
System A	A system that is set as system A to distinguish two systems, which are connected with two tracking cables. When the two systems start up at the same time, this system will be a control system. System switching does not affect the system A/B setting.
System B	A system that is set as system B to distinguish two systems, which are connected with two tracking cables. When the two systems start up at the same time, this system will be a standby system. System switching does not affect the system A/B setting.
The other system	The other system that has been connected to own system with tracking cables. When the system A is the host system, the system B is the other system, and vice versa.
Tracking cable	An optical fiber cable used to connect two redundant function modules in a redundant system
Windows® 7 or later	Windows operating system versions indicating Windows®7, Windows®8, Windows®8.1, and Windows®10.

The following terms are used to explain a safety programmable controller system using the Safety CPU.

Term	Description
Pair version	This is version information that determines the Safety CPU and safety function module combination.
Safety communications	Communication service that processes the send/receive of network layers for the safety defined in the safety predefined protocol
Safety control	Machine control by safety programs and safety data communications. When an error occurs, the machine in operation is securely stopped.
Safety cycle processing	A safety cycle processing is a execution processing for safety programs and safety input/output.
Safety cycle time	A interrupt time for executing a safety I/O and safety program
Safety device	A device that can be used in safety programs
Safety program	A program for performing safety control
Safety programmable controller	A generic term for the MELSEC iQ-R series modules that perform safety control (such as the Safety CPU, safety function module, CC-Link IE Field Network remote I/O module (with safety functions))
Standard communications	Communications other than safety communications, such as cyclic transmission and transient transmission of CC-Link IE Field Network
Standard control	Machine control by standard programs and standard data communications. Modules other than the safety programmable controller hold only standard control. (This term is used to distinguish from safety control.)

Term	Description
Standard CPU	A generic term for MELSEC iQ-R series CPU modules (other than Safety CPU) that perform standard control (This term is used to distinguish from the Safety CPU.)
Standard device	A device (X, Y, M, D, or others) in a CPU module. (Safety devices are excluded.) This device can be used only in standard programs. (This term is used to distinguish from a safety device.)
Standard program	A program that performs sequence control. (Safety programs are excluded.) (This term is used to distinguish from a safety program.)
Standard programmable controller	A generic term for MELSEC iQ-R series modules that perform standard control (This term is used to distinguish a programmable controller from a safety programmable controller.)

MEMO

PART 1 PROGRAMMING

This part consists of the following chapters.

1 RUNNING A PROGRAM

2 CPU MODULE OPERATING STATUS

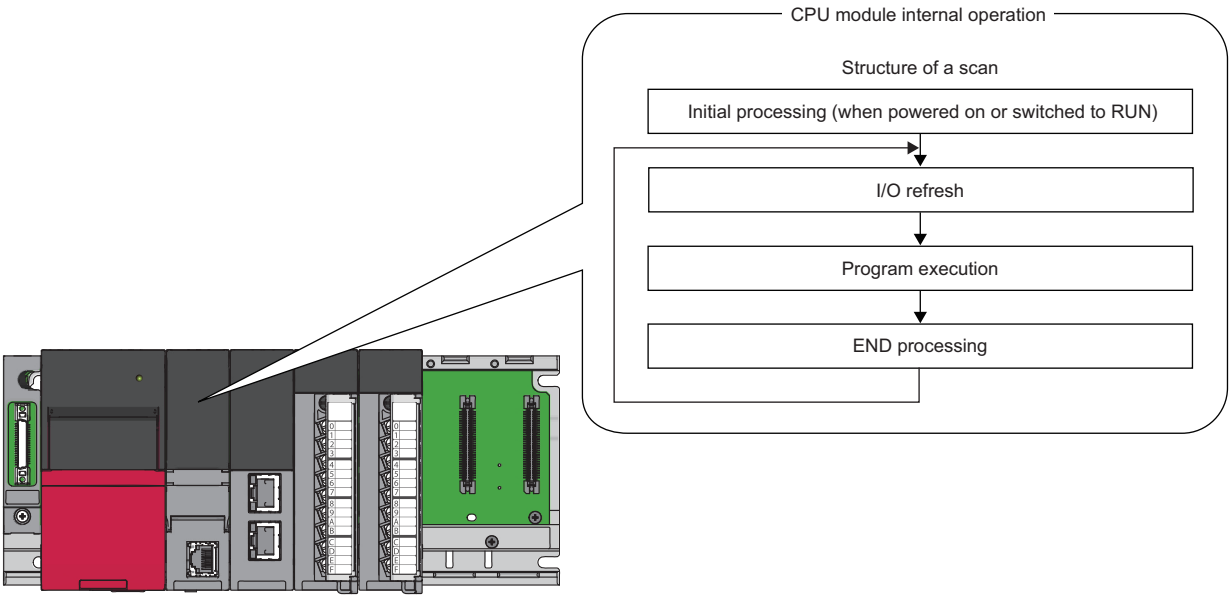
3 OPERATION PROCESSING AT MOMENTARY POWER FAILURE

4 MEMORY CONFIGURATION OF THE CPU MODULE

1 RUNNING A PROGRAM

1.1 Scan Configuration

The following shows the scan configuration of the CPU module.



Initial processing (when powered on or switched to RUN)

For the initial processing (when powered on or switched to RUN), the following processes are performed:*4

- Booting from an SD memory card
- Checking each parameter and program*1*3
- Checking the consistency of parameters for the multiple CPU system configuration*1
- Assigning the I/O number to the mounted module*1
- Initializing and setting the information of each module
- Initializing a device/label outside the latch range (Bit device: Off, Others: 0)
- Setting the device/initial label value*1*2

*1 The processing is performed in the initial processing (when switched to RUN).

*2 For details on the initial label value setting for the Process CPU, refer to the following.
☞ Page 306 LABEL INITIALIZATION FUNCTION

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

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

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

I/O refresh

The module performs the following before starting program operation.

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

Point

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

Program operation

According to the program settings, the module executes from step 0 through the END/FEND instruction for each program. This program is referred to as a main routine program. A main routine program can be divided into subroutine programs. (☞ Page 54 Subroutine program)

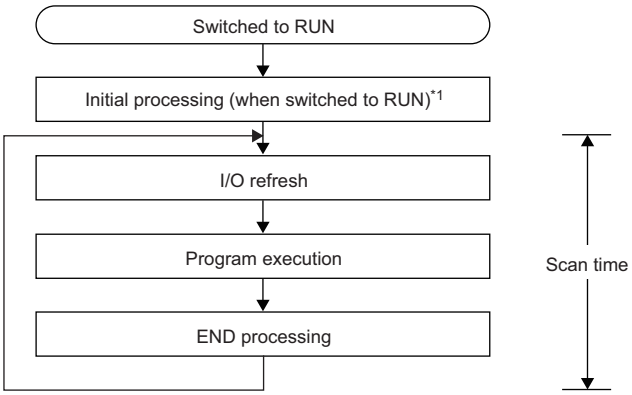
END processing

Performs the following processing.

- Network module link refresh
- Link refresh of CC-Link IE Field Network Basic
- Intelligent function module refresh
- Instruction end processing (including dedicated instruction for the module)
- Device latch processing
- The device latch processing is a kind of service processing such as read and write of devices, labels, and program access files (☞ Page 724 List of the Targets of Device/Label Access Service Processing Setting).
- Watchdog timer reset (☞ Page 89 Watchdog Timer Reset)
- Refresh between CPU modules (for the multiple CPU system configuration)
- Device collection by the data logging function (when the collection interval is set to "Each scanning cycle" or "Time specification")
- Self-diagnostics processing
- Sets a value to the special relay/special register (for those with the set timing specified as the END processing timing)

1.2 Scan Time

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



*1 The initial scan time includes this processing.

Initial scan time

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

How to check the initial scan time

See below.

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

Initial scan time monitoring

Monitors at the initial scan time monitoring time. (☞ Page 89 Scan Time Monitoring Time Setting)

■Precautions on the initial scan time monitoring time

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

1.3 Data Communication and I/O Processing

Data communication

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

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

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

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

I/O processing and response delay

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

Refresh mode

The CPU module performs I/O processing collectively at a specified timing. The following table lists the refresh timing which can be specified. (☞ Page 35 Refresh mode)

Refresh timing	Description
When END processing is performed	The refresh processing is performed in every END processing*1.
At the execution time of specified program	The refresh processing is performed before and after specified program execution. (☞ Page 53 Group setting for refresh)

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

Direct mode

The CPU module performs I/O processing when each instruction is executed in a program. To access input/output modules in the direct mode, use the direct access input or direct access output in a program. (☞ Page 37 Direct mode)

Differences between refresh mode and direct mode

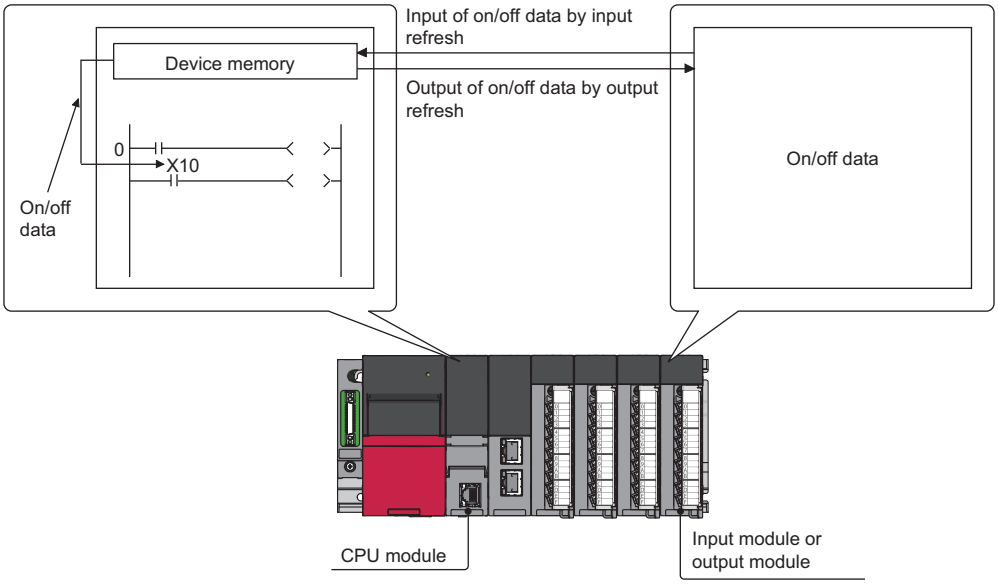
The direct mode directly accesses I/O modules at execution of an instruction. Therefore, data is input faster than when it is input in refresh mode. Processing time required for each instruction, however, takes longer. The following table lists the availability of the refresh mode and the direct mode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

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

Refresh mode

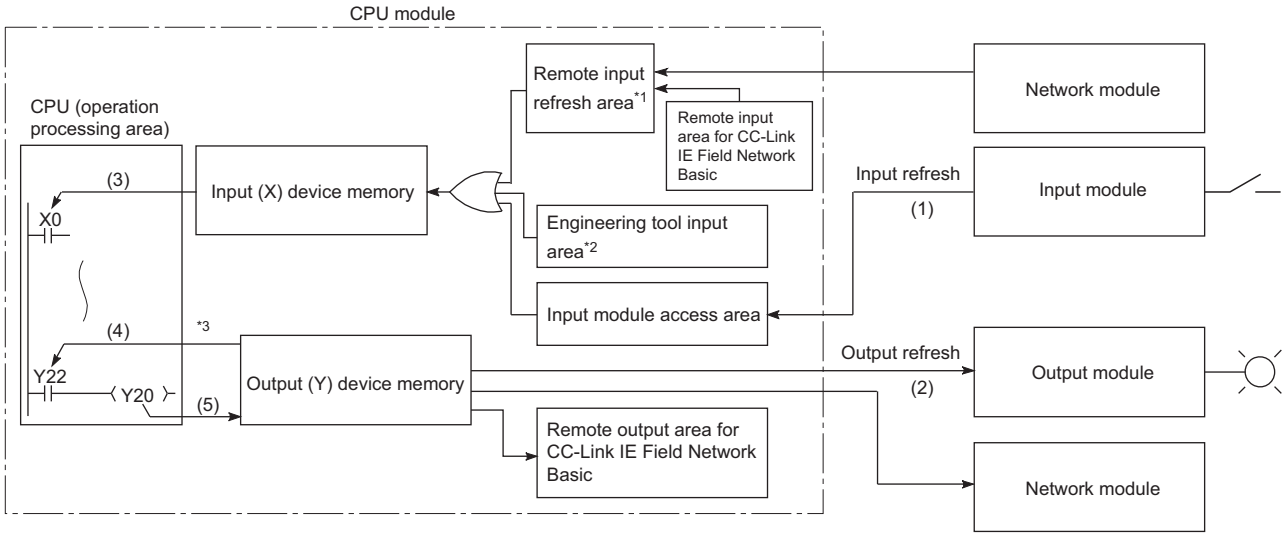
The CPU module performs I/O processing collectively at a specified timing.

The timing of the input refresh and output refresh follows the specified refresh timing setting.



Outline of the processing

The following describes the details of the refresh mode.



*1 The remote input refresh area indicates the area to be used when refresh is set to the input (X) in CC-Link IE Controller Network, CC-Link IE Field Network, CC-Link IE Field Network Basic, MELSECNET/H, or CC-Link.

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

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

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

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

■Input

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

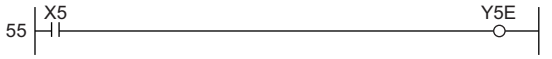
■Output

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

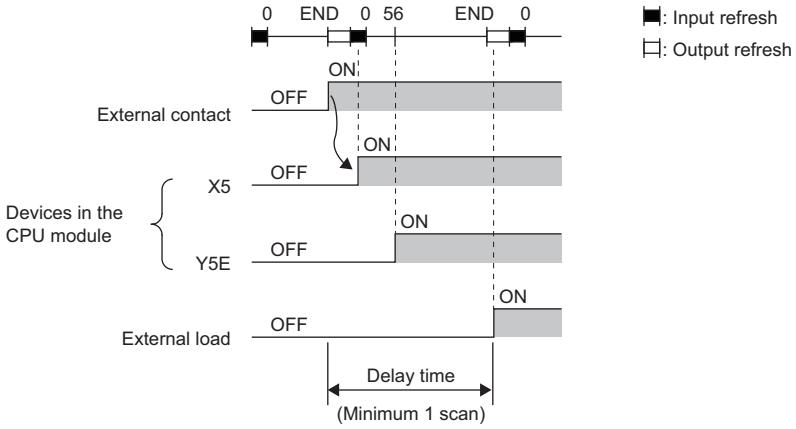
Response delay

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

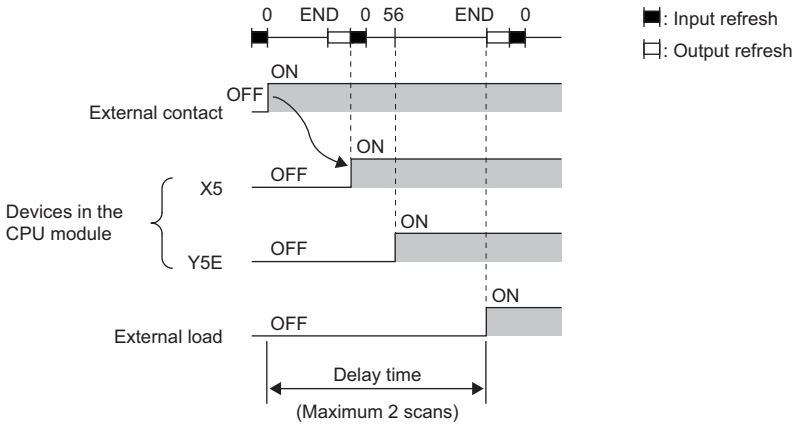
[Example] A program that turns on the output Y5E when the input X5 turns on



- Y5E turns on the earliest

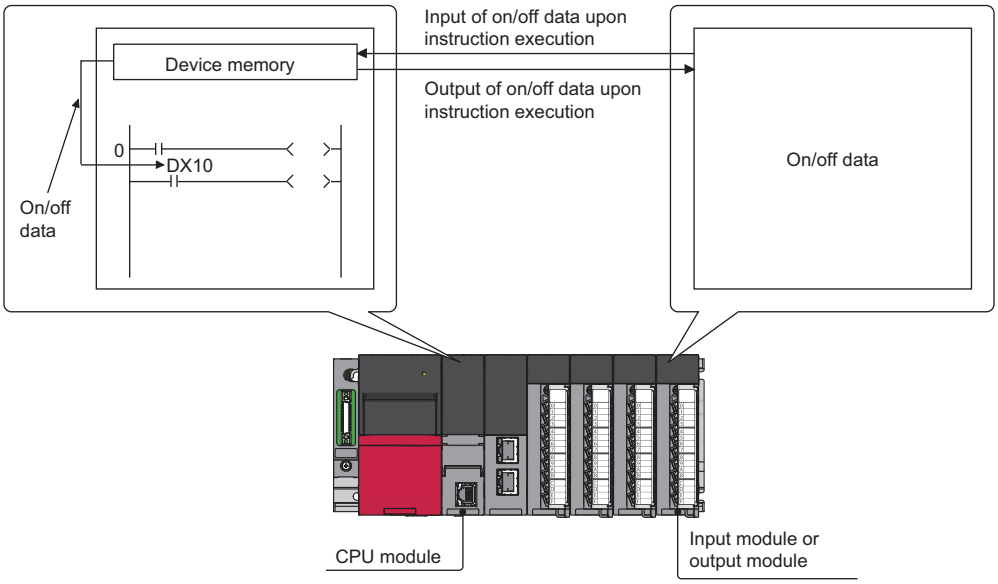


- Y5E turns on the latest



Direct mode

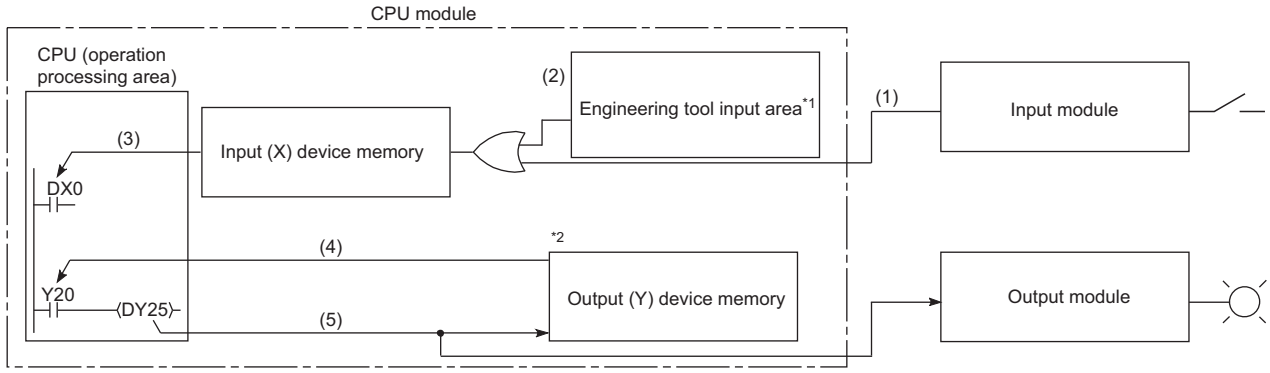
The CPU module performs I/O processing when each instruction is executed in a program.



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

Outline of the processing

The following describes the details of the direct mode.



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

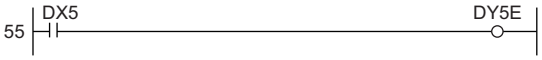
*2 Data in the output (Y) device memory can be turned on or off by the following:
- Test operation of an engineering tool
- Writing data from the network module
- Writing data from an external device using SLMP

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

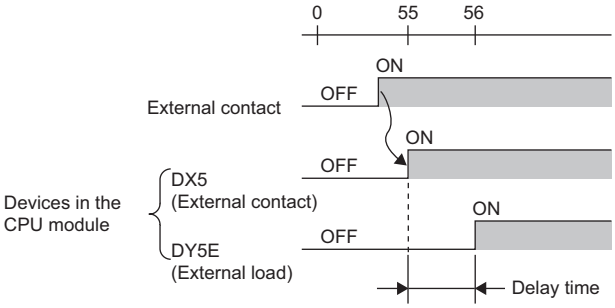
Response delay

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

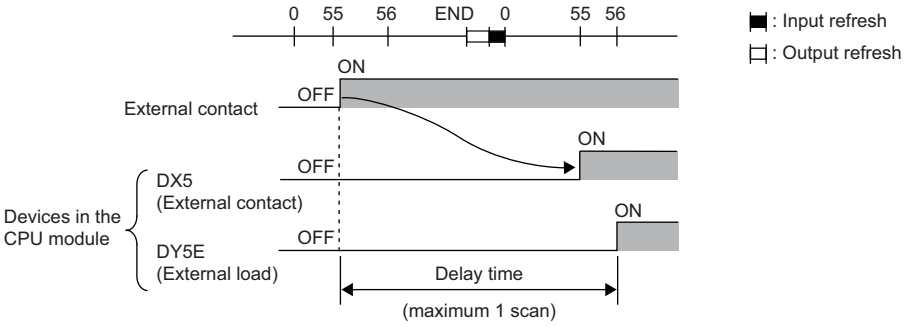
[Example] A program that turns on the output DY5E when the input DX5 turns on



- Y5E turns on the earliest

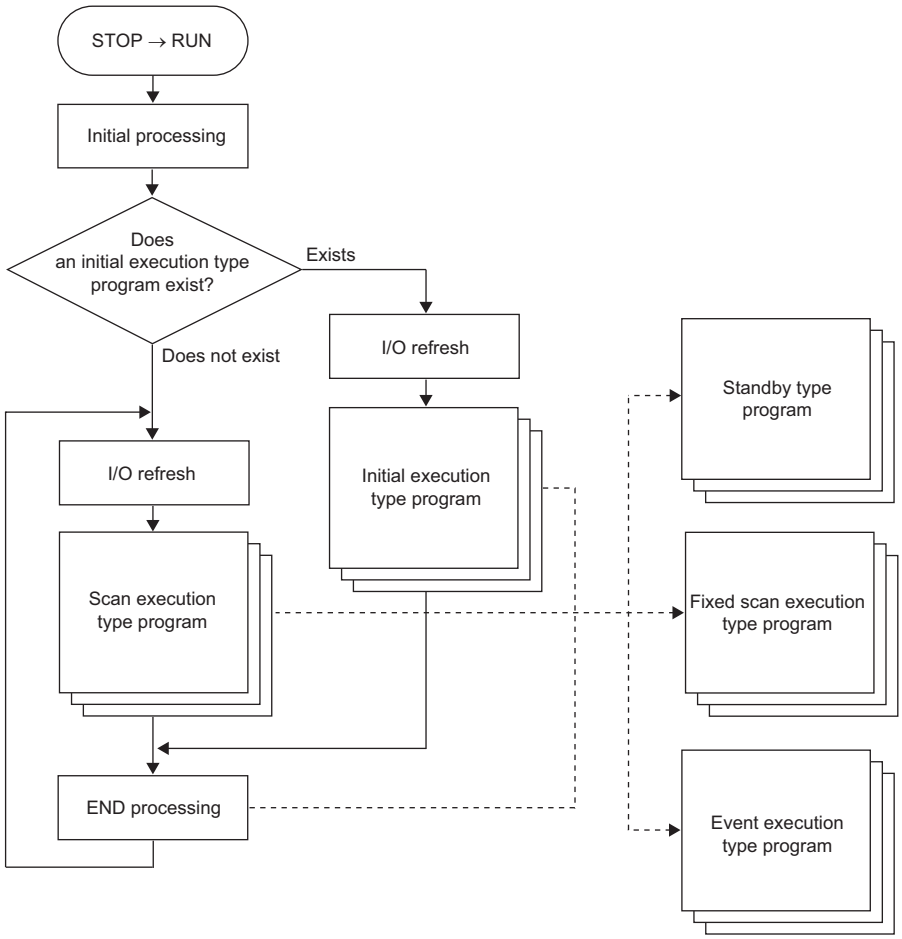


- Y5E turns on the latest



1.4 Program Flow

Programs are executed in order when the CPU module is switched to the RUN state according to the program execution type and execution sequence settings (☞ Page 41 Program Execution Type, ☞ Page 52 Execution type change).



Point

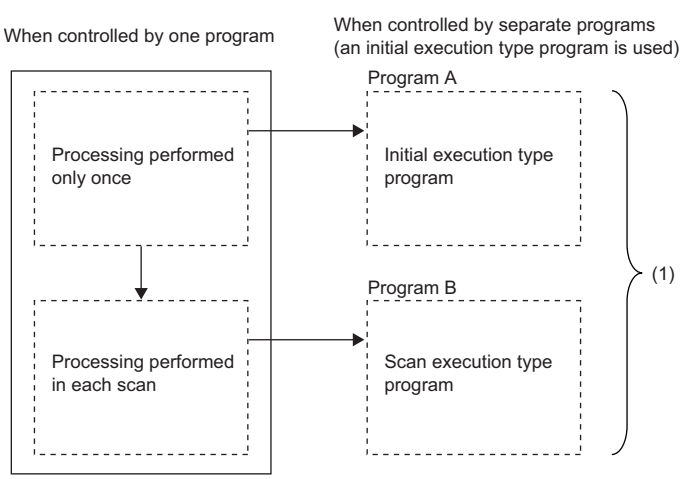
Programs with the same execution type are executed according to the execution sequence settings.

1.5 Program Execution Type

Sets the execution condition of the program. (☞ Page 52 Execution type change)

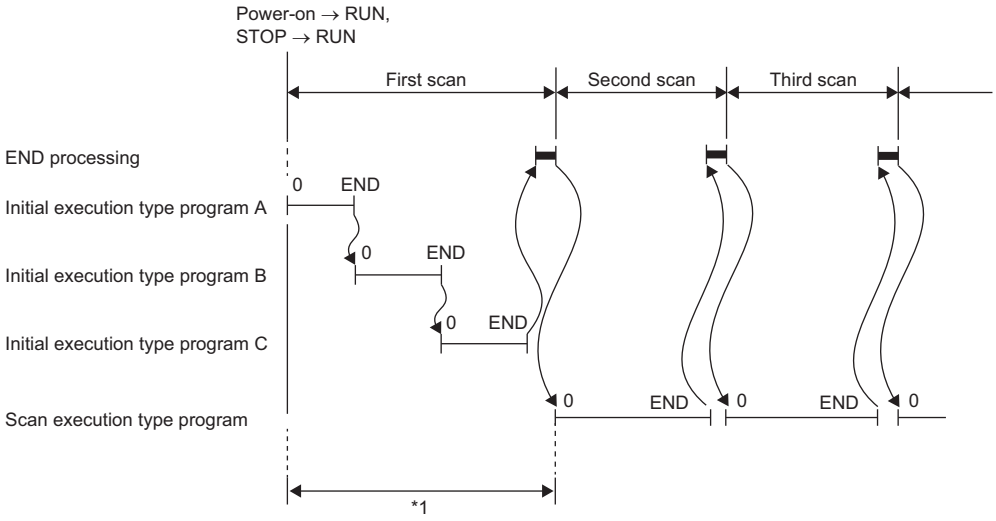
Initial execution type program

Executed only once when the CPU module has been powered off and on, or switched from the STOP state to the RUN state. Used for a program which does not need execution after executed once in the initial scan (e.g., initial processing for the intelligent function module).



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

Note that the execution time of the initial execution type program is the same as the initial scan time. (☞ Page 33 Initial scan time) When multiple initial execution type programs are executed, the execution time of the initial execution type program will be the total time until all the initial execution type programs finish execution.



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

Precautions

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

■Restrictions on program creation

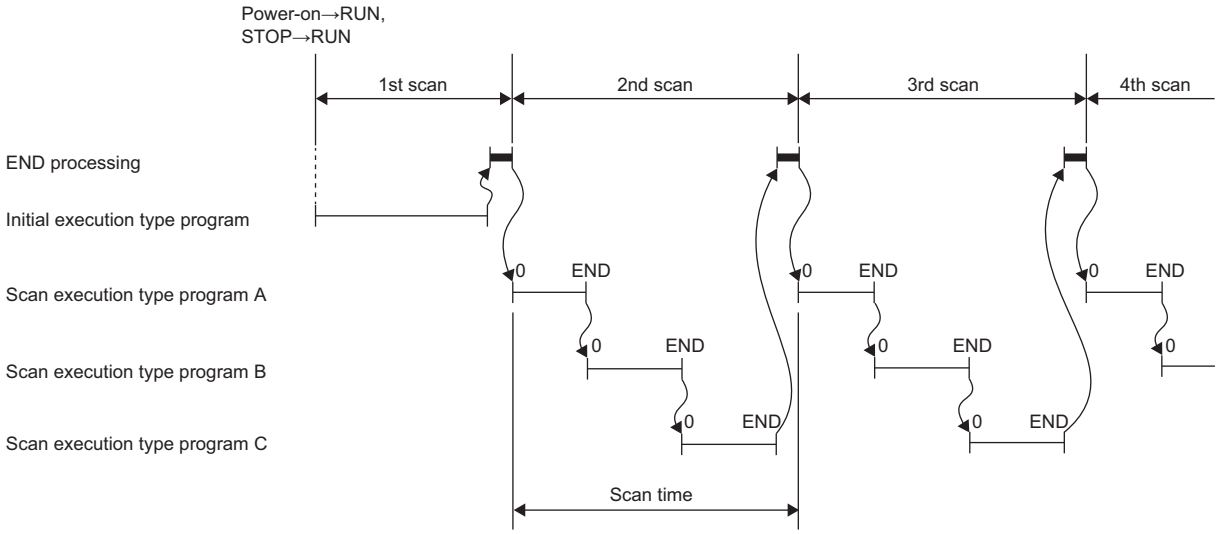
In the initial execution type program, do not use an instruction which requires several scans to complete (i.e., an instruction with a completion device).

Ex.

e.g. SEND and RECV instructions

Scan execution type program

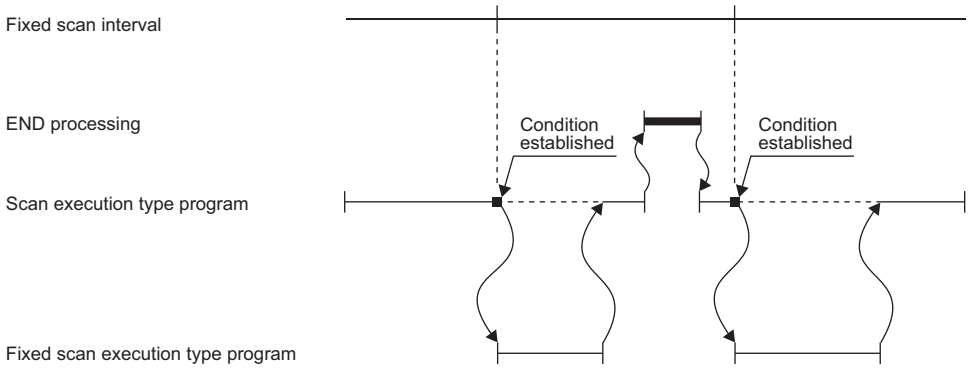
Executed only once per every scan starting from the scan following the scan in which the initial execution type program was executed.



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

Fixed scan execution type program

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



Point

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

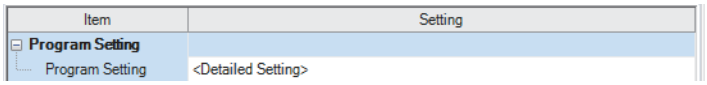
Fixed scan interval setting

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

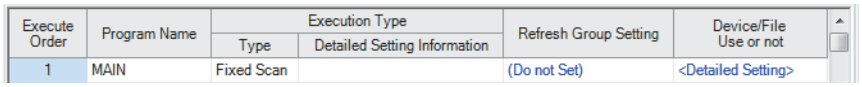
[CPU Parameter] ⇨ [Program Setting]

Operating procedure

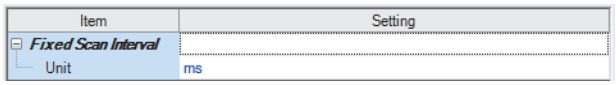
"Program Setting" window



"Detailed Setting" window



"Fixed Scan Execution Type Detailed Setting" window



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

Displayed items

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

Action when the execution condition is satisfied

Performs the following action.

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

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

■If there are multiple fixed scan execution type programs

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

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

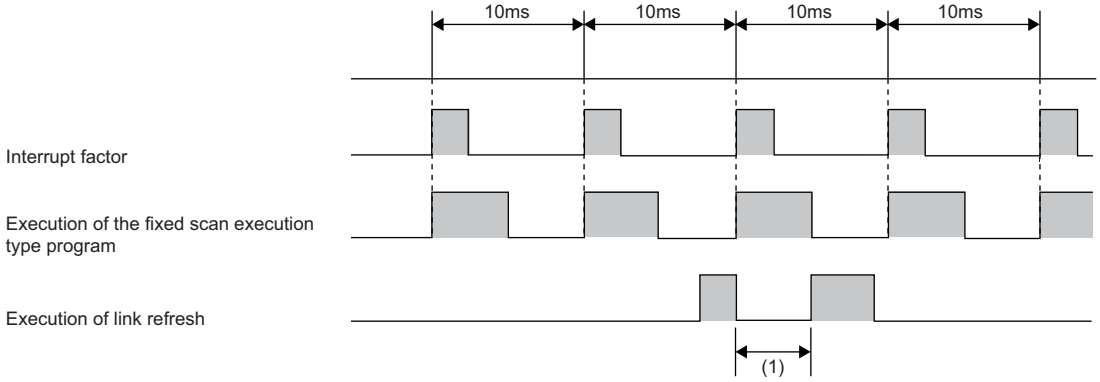
Operates according to the fixed scan execution mode setting.

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

Operates according to the fixed scan execution mode setting.

■If an interrupt factor occurs during link refresh

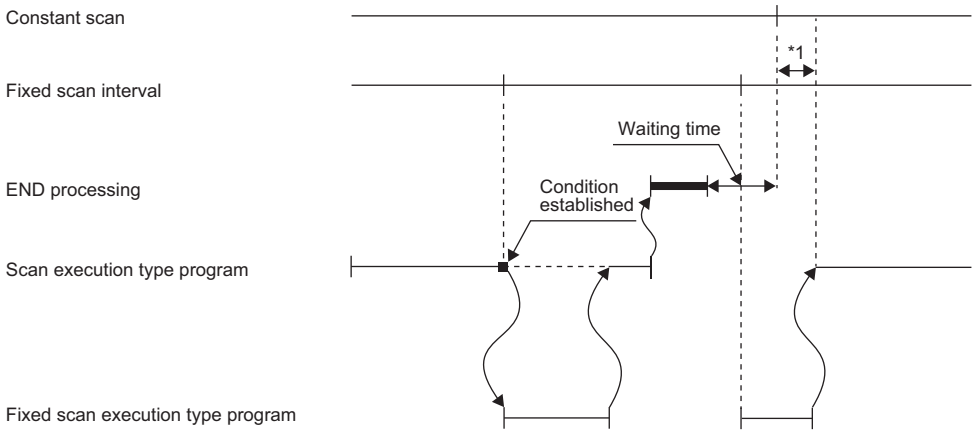
Suspends link refresh and executes the fixed scan execution type program. Even while station unit block guarantee is enabled for cyclic data during refresh of such links as CC-Link IE Field Network, if the fixed scan execution type program uses a device specified as the refresh target, station unit block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



(1) Suspends link refresh and executes the fixed scan execution type program.

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

Executes the fixed scan execution type program.



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

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

If an interrupt program (including an interrupt which triggers the event execution type program) is triggered while the fixed scan execution type program is being executed, the program operates in accordance with the interrupt priority. (☞ Page 106 Multiple Interrupt Function)

Processing when the fixed scan execution type program starts

The same processing as when the interrupt program starts. (☞ Page 66 Processing at interrupt program startup)

Fixed scan execution mode

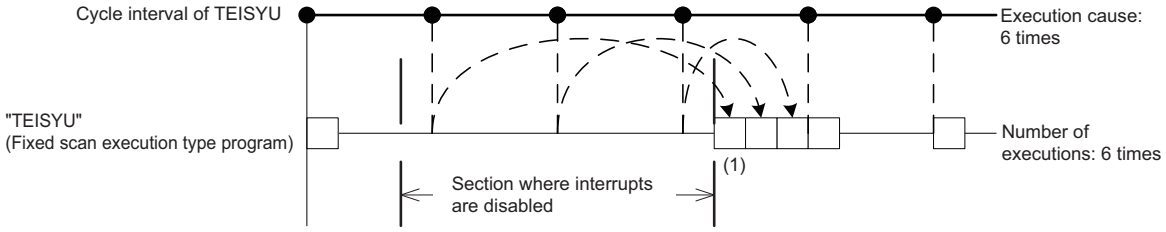
For fixed scan interrupts (I28 to I31, I48, I49) triggered by the fixed scan execution type program or the internal timer of the CPU module, this mode specifies the program execution operation which should be performed when more than one interrupt occurs (☞ Page 56 Operation upon occurrence of an interrupt factor) However, an execution cause which occurs while the interrupt is disabled by the DI instruction is excluded from the fixed scan execution mode target.

■Operation in the fixed scan execution mode

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

• Execution Count Takes Priority

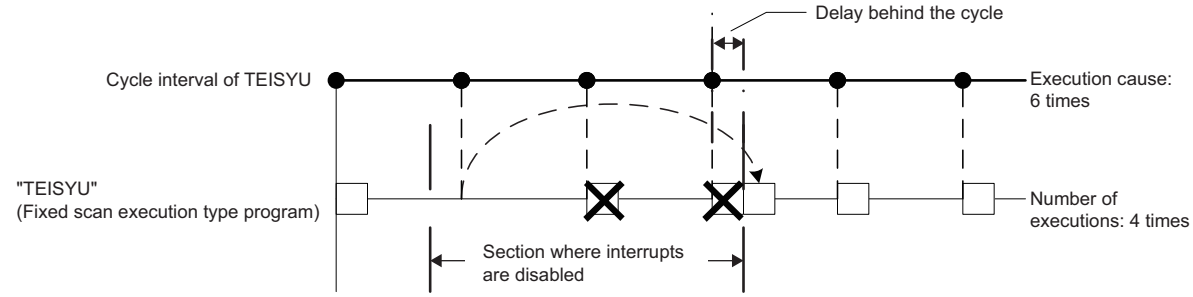
The program is executed for all the pending number of executions so that it can be executed the same number of times as execution causes.



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

• Precede Fixed Scan

When a pending execution exists, it is performed within the delay limit value behind a cycle. (☞ Page 46 Delay limit value behind a cycle) Note that, even if more than one pending execution exists, only one execution is performed.

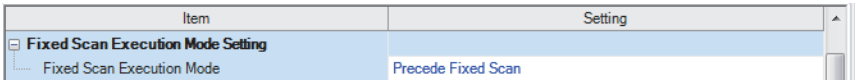


■Fixed scan execution mode setting

Use the fixed scan execution mode setting.

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

Window

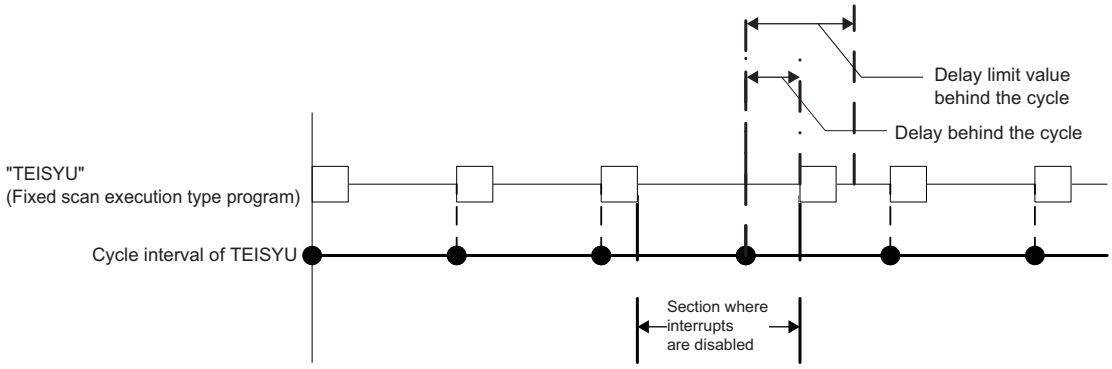


Displayed items

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

■Delay limit value behind a cycle

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



The delay limit value behind a cycle differs as follows.

• For the fixed scan execution type program

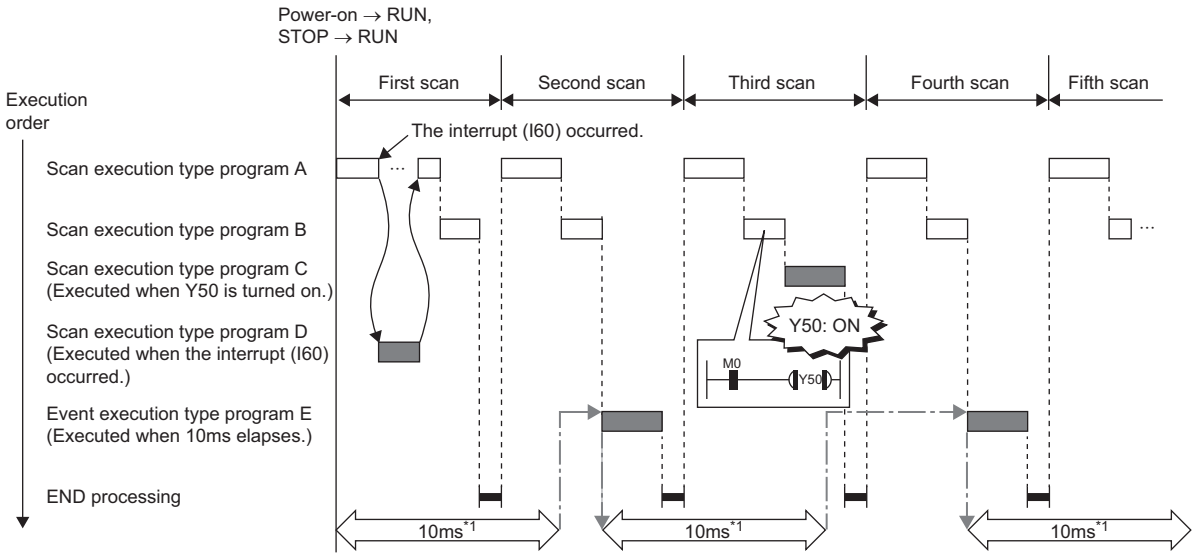
The greatest common divisor of the fixed scan intervals of all the fixed scan execution type programs. For example, if there are fixed scan execution type programs with 2ms, 4ms, and 12ms scan periods, the delay limit value behind a cycle is 2ms.

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

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

Event execution type program

This type of program starts execution when triggered by a specified event. (☞ Page 47 Trigger type) The program is executed at the execution turn specified in the program settings of the CPU parameters, and if execution conditions of the specified trigger are met when the execution turn of the event execution type program comes, the program is executed.



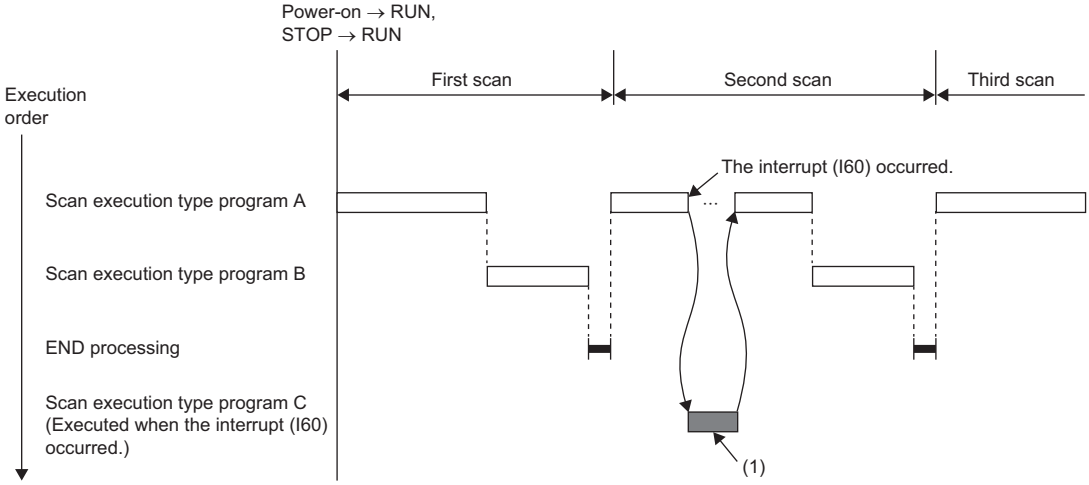
*1 Measurement of elapsed time is 10ms or more because it is determined depending on the scan time.

Trigger type

The following lists the triggers for the event execution type programs. (☞ Page 50 Trigger setting)

■Interrupt occurrence by the interrupt pointer (I)

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



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

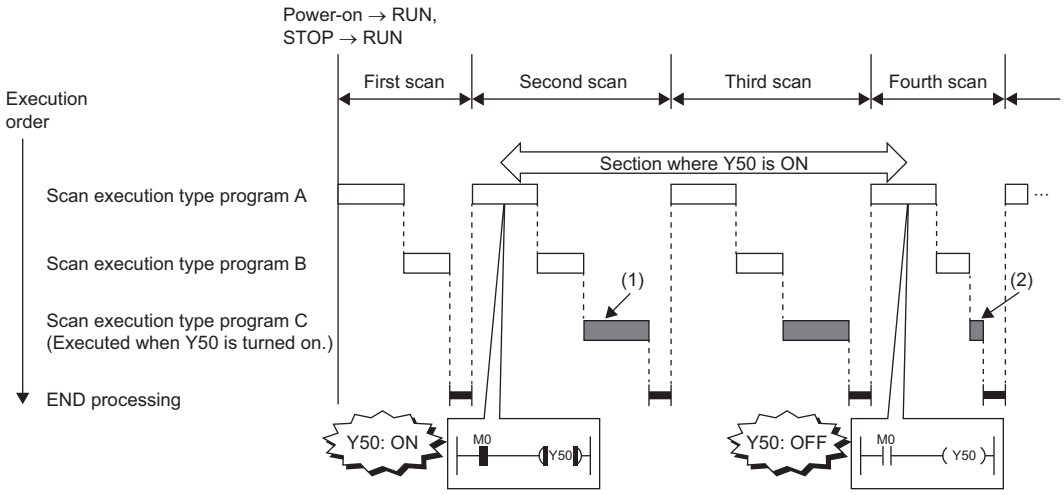
Interrupt pointers (I) which can be specified are I0 to I15, I28 to I31, I44, I45, I48, I49, I50 to I1023.

Point

Execution conditions for the event execution type program which is triggered by interrupt occurred by the interrupt pointer (I) are the same as those for general interrupt programs (☞ Page 56 Operation upon occurrence of an interrupt factor)

■Bit data ON (TRUE)

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



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

Applicable devices are as follows.

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

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

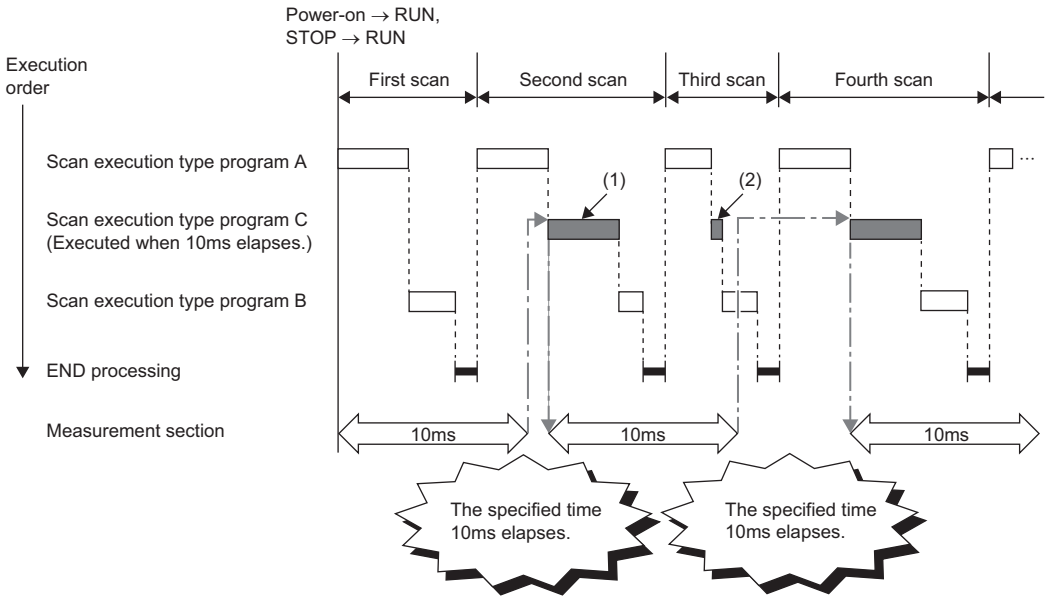
Point

To measure the timer and the retentive timer continuously even when the trigger execution condition of the event execution type program is not met, use the long timer (LT) and the long retentive timer (LST). The timer (T) and the retentive timer (ST) can be used in the event execution type program. However, they do not measure time when the trigger execution condition of the event execution type program is not met. Set "Clear Output and Current Value of Timer" to "Clear" when the long timer (LT) and the long retentive timer (LST) are used. (☞ Page 50 Trigger setting)

Note that if the above parameter is set, the use of the long timer (LT) and the long retentive timer (LST) is limited to the same as that of the timer (T) and the retentive timer (ST). (They do not measure time continuously when the trigger execution condition of the event execution type program is not met.)

■Passing time

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



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

Point

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

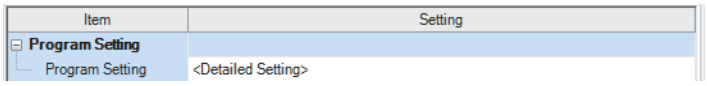
Trigger setting

Use the event execution type detail setting.

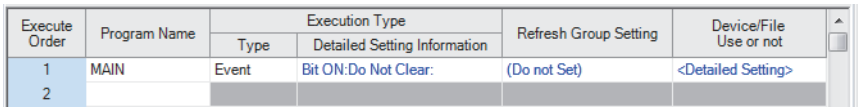
[CPU Parameter] ⇨ [Program Setting]

Operating procedure

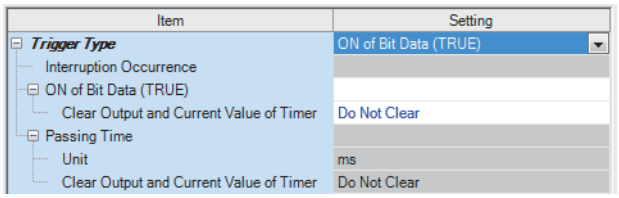
"Program Setting" window



"Detailed Setting" window



"Event Execution Type Detailed Setting" window



1. Click "Detailed Setting" on the "Program Setting" window.
2. Select the program name and set the execution type to "Event".
3. Click "Detailed Setting Information".
4. Sets the trigger type to execute the event execution type program.

Displayed items

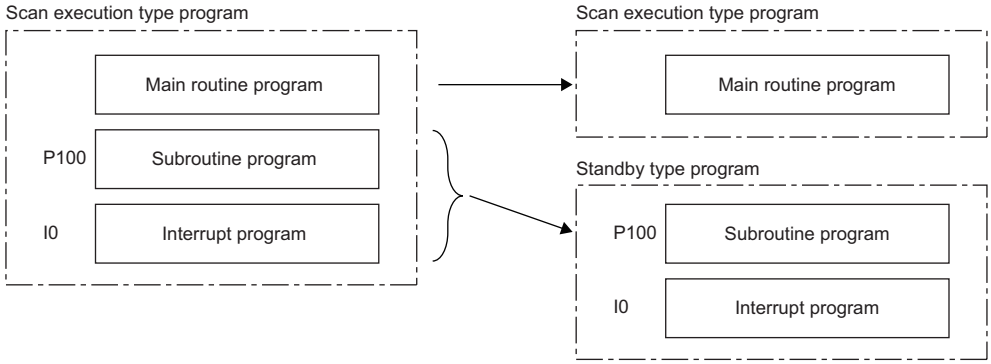
Item		Description	Setting range	Default
Interruption occurrence		Sets the interrupt pointer used as the trigger.	I0 to I15, I28 to I31, I44, I45, I48, I49, I50 to I1023	—
ON of Bit Data (TRUE)		Sets the device used as the trigger.	☞ Page 48 Bit data ON (TRUE)	—
Clear Output and Current Value of Timer		Sets that the current values of the output (Y), timer (T), and long timer (LT) used in this program are cleared at the execution turn of the event execution type program that comes after the specified bit data is OFF.	• Do Not Clear • Clear	Do Not Clear
Passing Time	Unit	Sets the time passed.	• When "ms" is selected: 1 to 65535ms (in units of 1ms) • When "s" is selected: 1 to 65535s (in units of 1s)	ms
	Clear Output and Current Value of Timer	Sets that the current values of the output (Y), timer (T), and long timer (LT) used in this program are cleared at the execution turn of the event execution type program that comes after the specified time passes.	• Do Not Clear • Clear	Do Not Clear

Standby type program

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

Librarization of programs

Used when making the subroutine program or interrupt program a standby type program and managing it separately from the main routine program. In a single standby type program, multiple subroutine programs and interrupt programs can be created.



Program arrangement change

Used when programs available for all systems are prepared and only those required are executed. For example, a program set in advance as the standby type with a parameter can be changed to scan execution type and executed. (☞ Page 52 Using an instruction)

Execution method

The standby type program is executed as follows.

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

Execution type change

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

Using parameter settings

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

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

Operating procedure

"Program Setting" window

Item	Setting
Program Setting	
Program Setting	<Detailed Setting>

"Detailed Setting" window

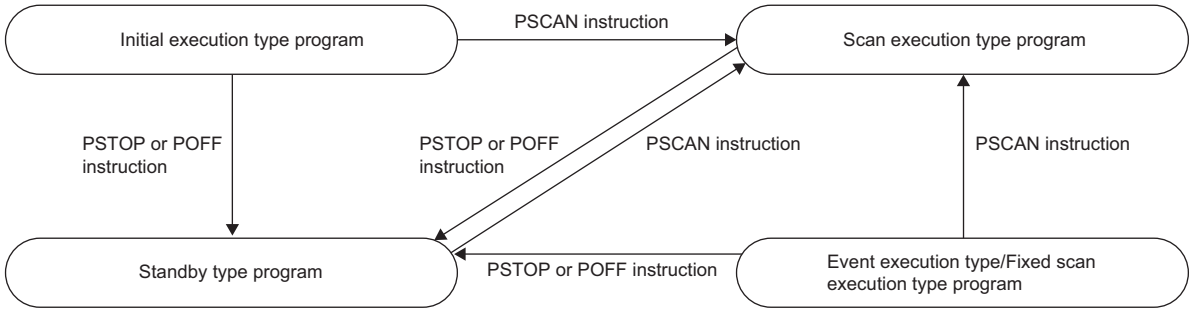
Execute Order	Program Name	Execution Type		Refresh Group Setting	Device/File Use or not
		Type	Detailed Setting Information		
1	MAIN	Scan		(Do not Set)	<Detailed Setting>
2	MAIN1	Fixed Scan		(Do not Set)	<Detailed Setting>
3	MAIN2	Standby		(Do not Set)	<Detailed Setting>

1. Click "Detailed Setting" on the "Program Setting" window.
2. For each program, select the execution type in "Type" of "Execution Type".

Using an instruction

Use the following instructions.

- PSCAN
- PSTOP
- POFF



■Change timing

The execution type change timing is shown below.

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

*1 Stops execution at END processing after instruction execution.

■Precautions

The following lists the precautions when changing the execution type.

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

MELSEC iQ-R Programming Manual (Program Design)

Group setting for refresh

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

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

Refresh Group Setting

A group number is set to each program.

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

Window

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

Displayed items

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

Point

If refresh corresponding to "Refresh Group Setting" is not set up, the relevant program is executed but refresh is not performed.

Refresh setting of each module

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

[Module Parameter] ⇨ [Refresh Setting]

Window

Item	Setting Value
Refresh by the Set Timing	
Refresh Timing	Set refresh timing.
Refresh Timing	At the execution time of specified program
Group[n](n:1-64)	1

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

Point

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

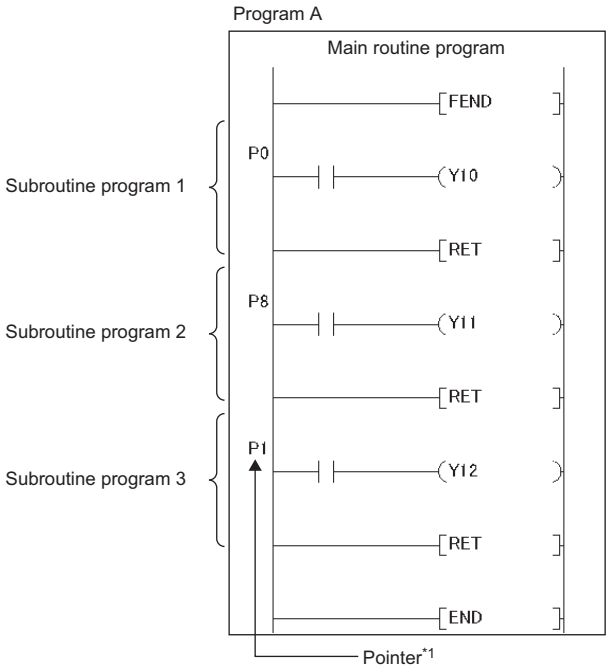
1.6 Program Type

This section describes programs which use pointers (P) and/or interrupt pointers (I).

Subroutine program

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

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



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

Point

- Making a program a standby type allows it to be managed as a separate program. (☞ Page 51 Standby type program) Use the ECALL instruction and other similar instructions to call program files.
- For details on nesting (nesting structure) of subroutine programs, refer to the nesting (N) (☞ Page 340 Nesting (N))

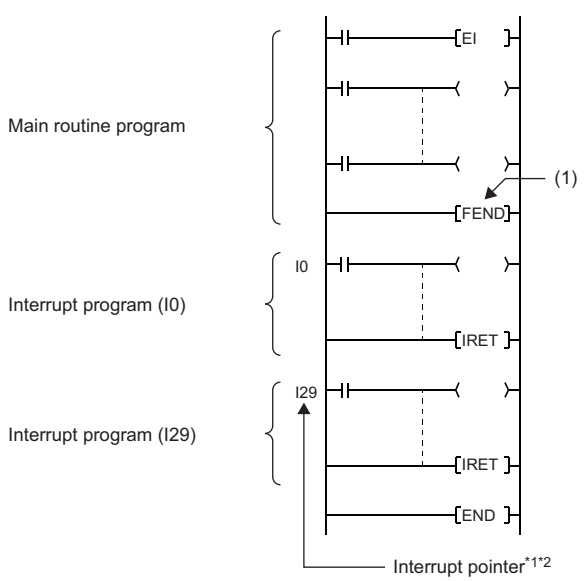
Precautions

The following lists the precautions when using the subroutine program.

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

Interrupt program

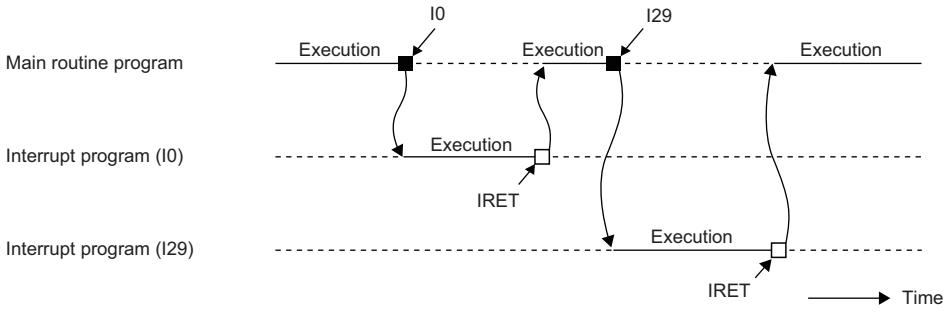
A program from an interrupt pointer (I) through the IRET instruction.



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

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

When an interrupt factor occurs, the interrupt program corresponding to the interrupt pointer number is executed (☞ Page 345 The priority for the interrupt pointer numbers and interrupt factors) However, before that, interrupt must be enabled by using the EI instruction.



I0: Interrupt factor for I0

I29: Interrupt factor for I29

Point

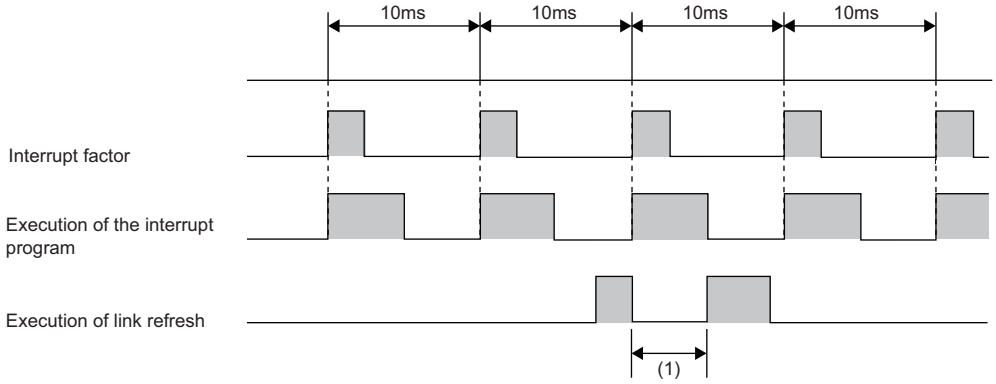
Making a program a standby type allows it to be managed as a separate program. (☞ Page 51 Standby type program)

Operation upon occurrence of an interrupt factor

The following shows the operation when an interrupt factor occurs.

■If an interrupt factor occurs during link refresh

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



(1) Suspends link refresh and executes the interrupt program.

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

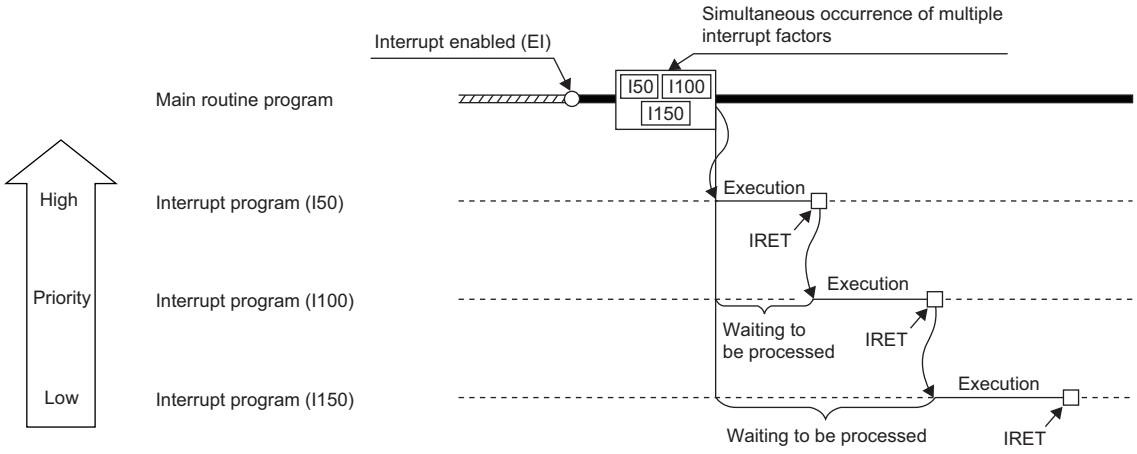
Executes the interrupt program for the interrupt factor.

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

If an interrupt such as a fixed scan execution type program (including an interrupt which triggers the event execution type program) is triggered while an interrupt program is being executed, the program operates in accordance with the interrupt priority. (☞ Page 106 Interrupt priority)

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

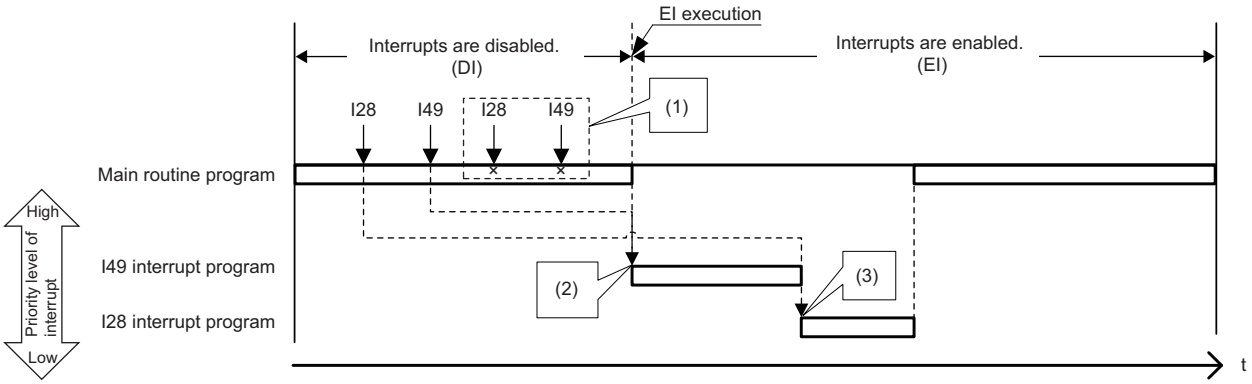
The interrupt programs are executed in the order of priority. If multiple interrupts with the same interrupt priority occur simultaneously, the interrupt programs are executed in the order of interrupt priority. (☞ Page 109 Multiple interrupt execution sequence)



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

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

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



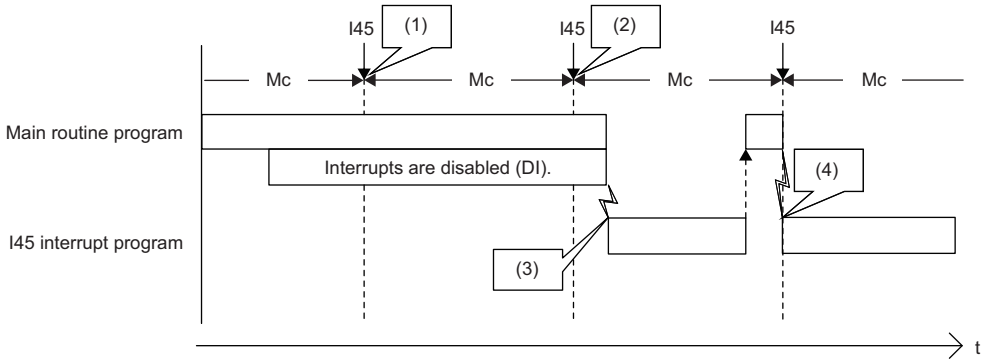
(1) The second and following interrupt factors that occur while interrupts have been disabled (DI) are not saved.

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

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

• For I45

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



Mc: Multiple CPU synchronization cycle

(1) Saved.

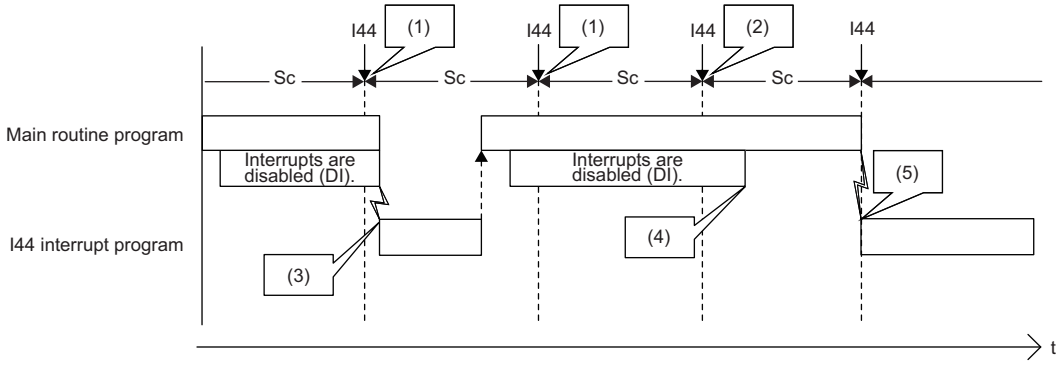
(2) The second and following interrupt factors that occur while interrupts have been disabled (DI) are not saved.

(3) The interrupt is executed.

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

• For I44

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



Sc: Inter-module synchronization cycle

(1) Saved.

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

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

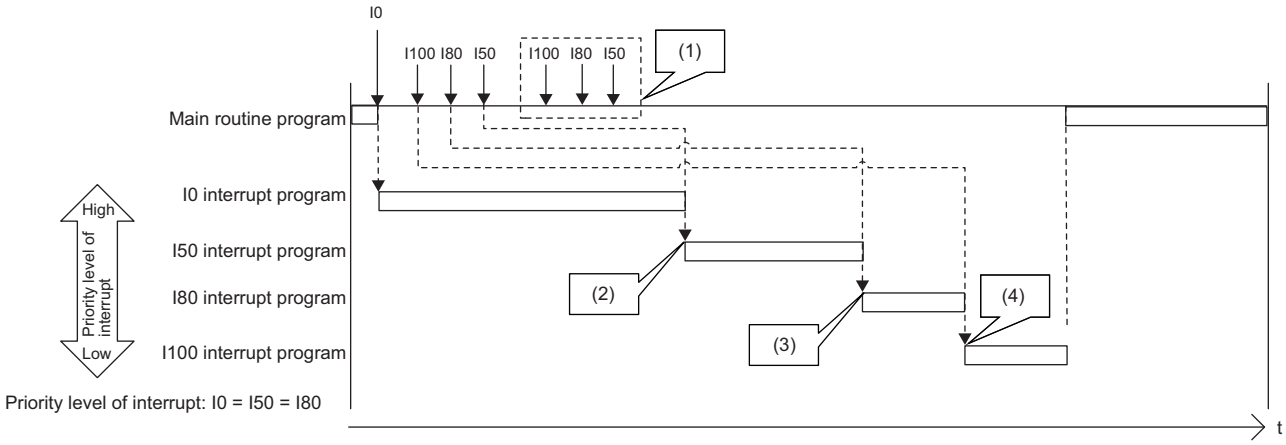
(4) I44 is not executed because "Interrupts are disabled (DI)." is in the second cycle of I44.

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

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

• For I0 to I15 and I50 to I1023

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



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

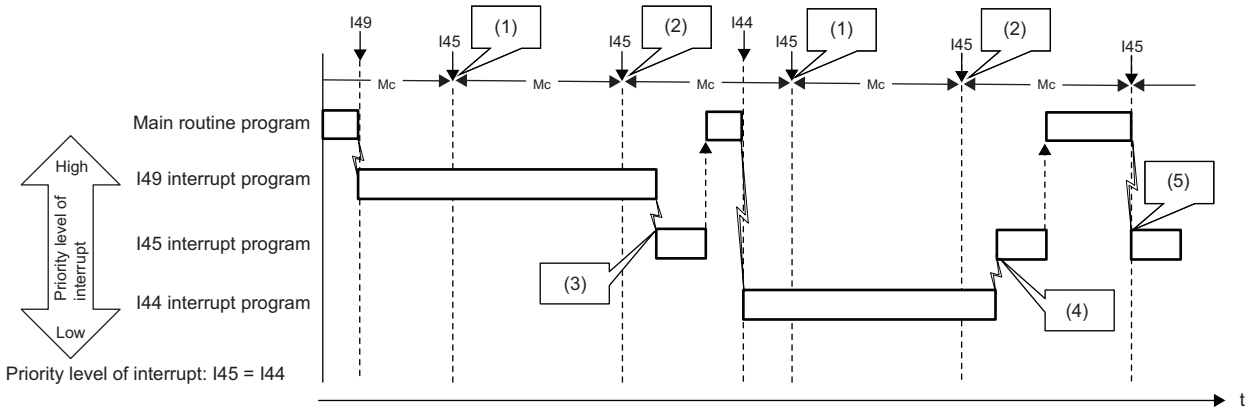
(2) After the interrupt in execution is completed, interrupts are executed in order from I50 because it has a higher priority. Although both I50 and I80 have the same priority level, I50 is executed ahead of I80 because I50 has a higher priority order.

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

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

• For I45

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



Mc: Multiple CPU synchronization cycle

(1) Saved.

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

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

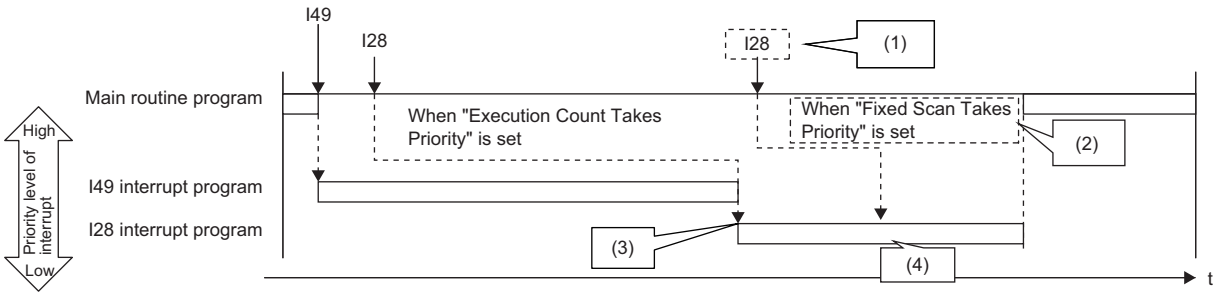
(4) Because both I44 and I45 have the same priority level, the interrupt is executed after I44 is completed.

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

• For I28 to I31, I48, and I49

The occurred interrupt factor is memorized, and the interrupt program corresponding to the factor will be executed after the running interrupt program finishes. If the same interrupt factor occurs multiple times, it will be memorized once but operation at the second and later occurrences depends on setting of the fixed scan execution mode (☞ Page 45 Fixed scan execution mode). When "Execution Count Takes Priority" is enabled, the interrupt program corresponding to the memorized interrupt factor will be executed after the running interrupt program finishes. When "Precede Fixed Scan" is enabled, the second and later occurrences will not be memorized.

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



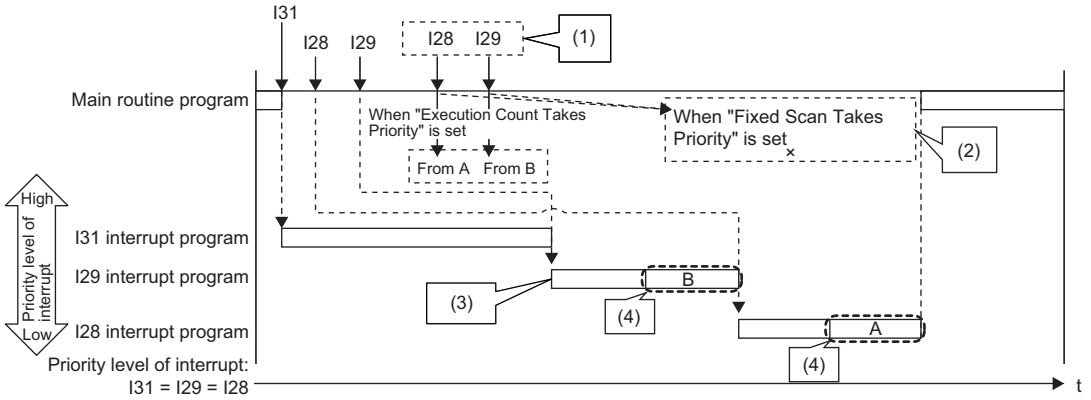
(1) The second and following interrupt factors that occur while interrupts are being executed operate according to the setting of the fixed scan execution mode.

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

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

(4) The second interrupt is executed.

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



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

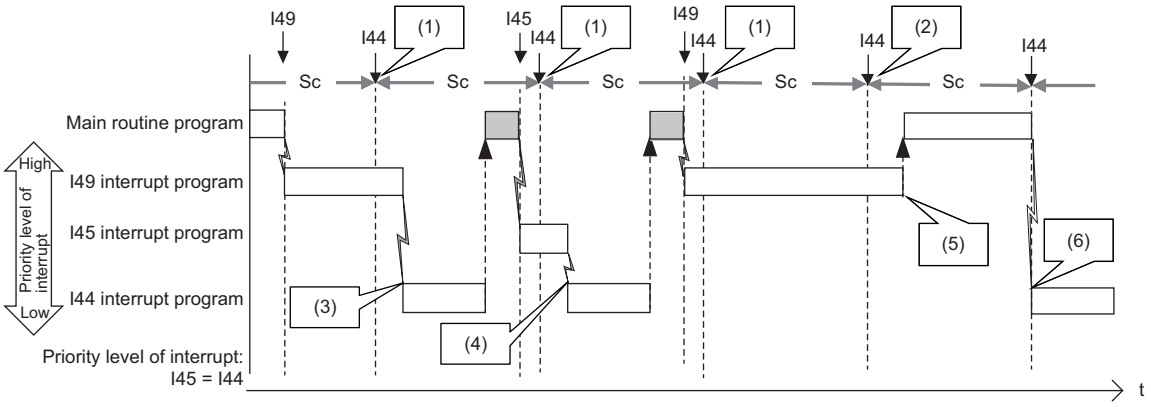
(2) The program is not executed for the second time.

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

(4) The second interrupt is executed.

• For I44

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



Sc: Inter-module synchronization cycle

(1) Saved.

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

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

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

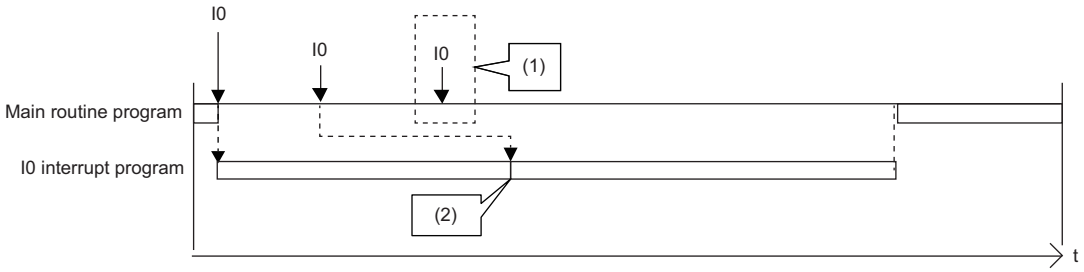
(5) I44 is not executed because the termination of I49 or I45 is in the second cycle for I44.

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

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

• For I0 to I15 and I50 to I1023

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

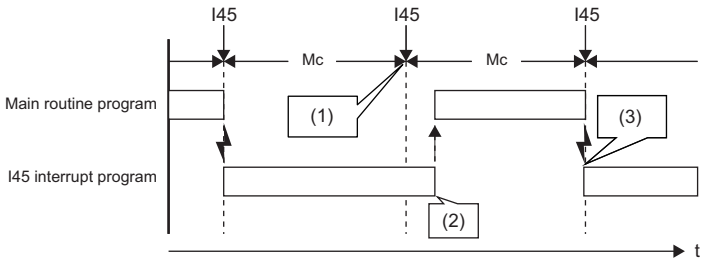


(1) The second and following interrupt factors that occur while interrupts are executed are not saved.

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

• For I45

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



Mc: Multiple CPU synchronization cycle

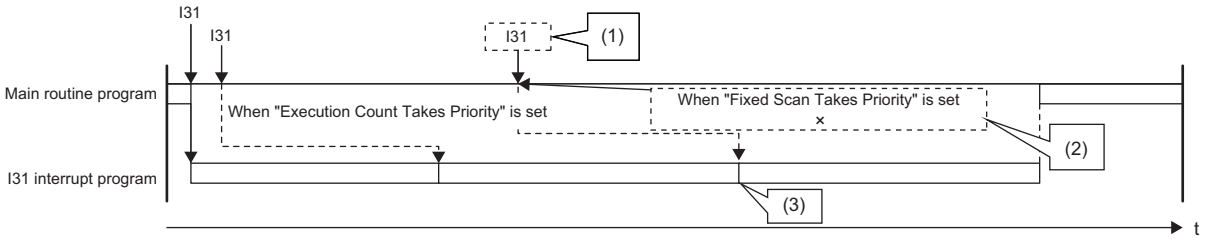
(1) The interrupt factor is discarded because the same interrupt (I45) is being executed.

(2) The interrupt is not executed.

(3) The interrupt is executed.

• For I28 to I31, I48, and I49

The occurred interrupt factor is memorized, and the interrupt program corresponding to the factor will be executed after the running interrupt program finishes. If the same interrupt factor occurs multiple times, it will be memorized once but operation at the second and later occurrences depends on setting of the fixed scan execution mode (☞ Page 45 Fixed scan execution mode). When "Execution Count Takes Priority" is enabled, the interrupt program corresponding to the memorized interrupt factor will be executed after the running interrupt program finishes. When "Precede Fixed Scan" is enabled, the second and later occurrences will not be memorized.



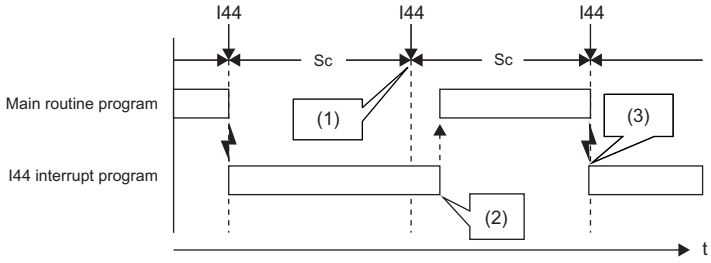
(1) The second and following interrupt factors that occur while interrupts are being executed operate according to the setting of the fixed scan execution mode.

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

(3) The second interrupt is executed.

• For I44

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



Sc: Inter-module synchronization cycle

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

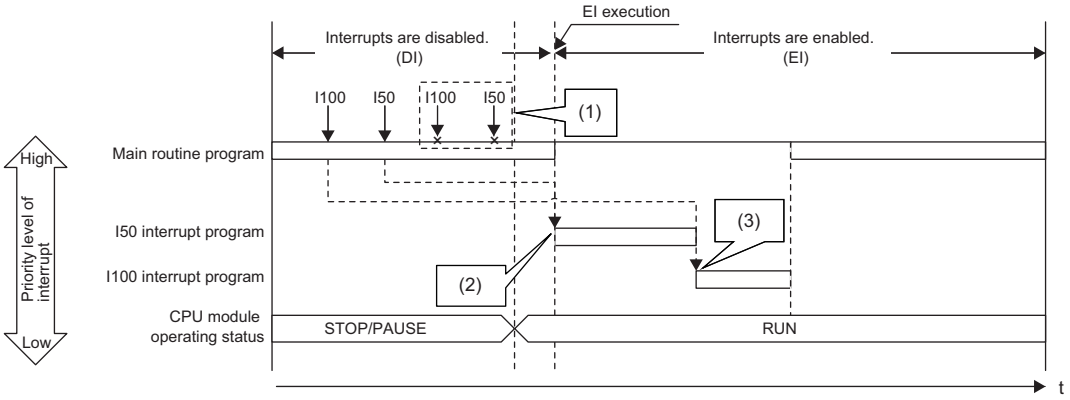
(2) The interrupt is not executed.

(3) The interrupt is executed.

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

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

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



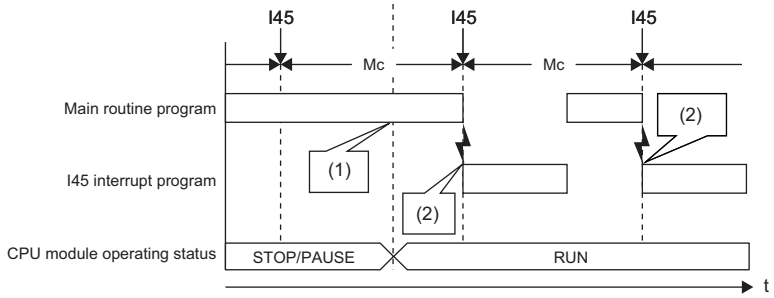
(1) The second and following interrupt factors that occur while the CPU module is in the STOP state are not saved.

(2) When interrupts are enabled by changing the operating status of the CPU module from STOP to RUN, interrupts are executed in order from I50 that has a higher priority level.

(3) I100 is executed. (I50 is not executed for the second time.)

• For I45

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



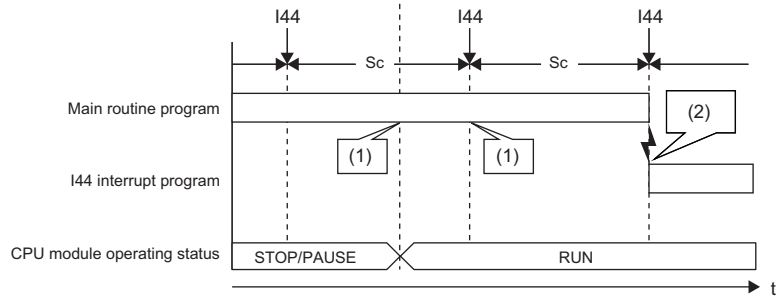
Mc: Multiple CPU synchronization cycle

(1) The interrupt is not executed.

(2) The interrupt is executed.

• For I44

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



Sc: Inter-module synchronization cycle

(1) The interrupt is not executed.

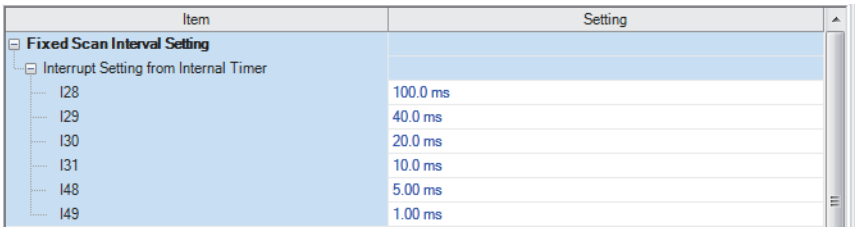
(2) The interrupt is executed.

Interrupt period setting

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

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

Window



Displayed items

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

Interrupt enabled during instruction execution

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

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

Window

Displayed items

Item	Description	Setting range	Default
Interrupt in Executing Instruction	Sets whether or not to enable or disable execution of an interrupt program during execution of an instruction. For the precautions with "Enable" set, refer to the precautions for an interrupt program. (☞ Page 69 Precautions)	• Disable • Enable	Disable

Processing at interrupt program startup

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

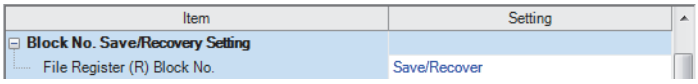
- Saving/restoring of the file register (R) block number
- Saving/restoring of the index register (Z, LZ)

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

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

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

Window



Displayed items

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

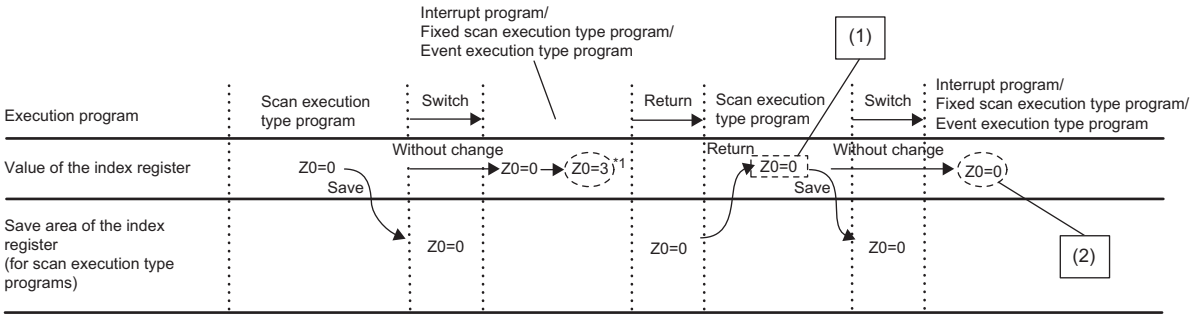
Point

When the block number of the file register (R) is not changed during execution of an interrupt program, enabling "Not Saved/Recovered" can reduce the startup time and the termination time of the interrupt program (☞ Page 695 Overhead time when executing the interrupt program). To change the block number of the file register (R) while "Not Saved/Recovered" is enabled, the block number of that file register (R) must be saved/restored by the program.

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

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

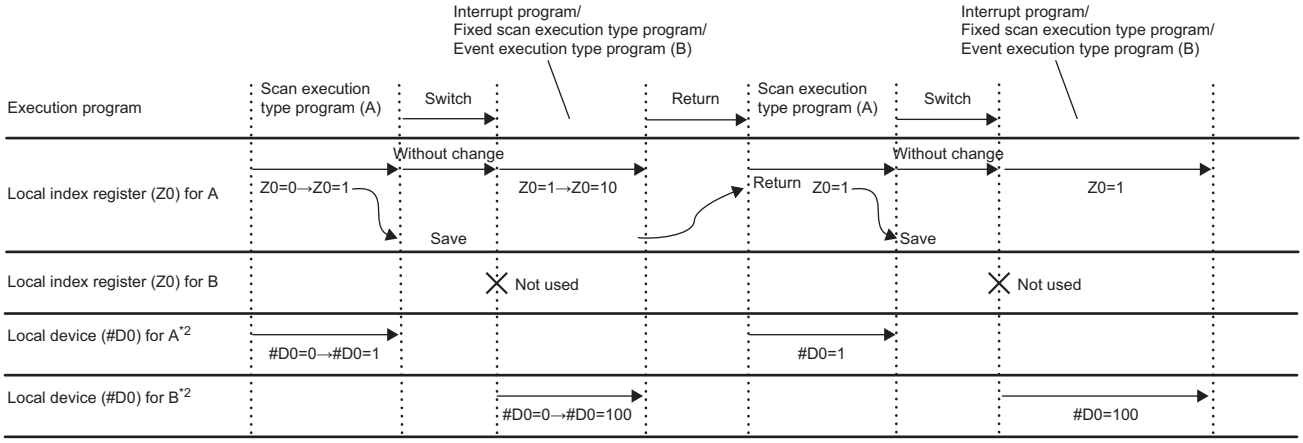
• Saving/restoring of the index register



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

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

• Operations of the local index register and local device

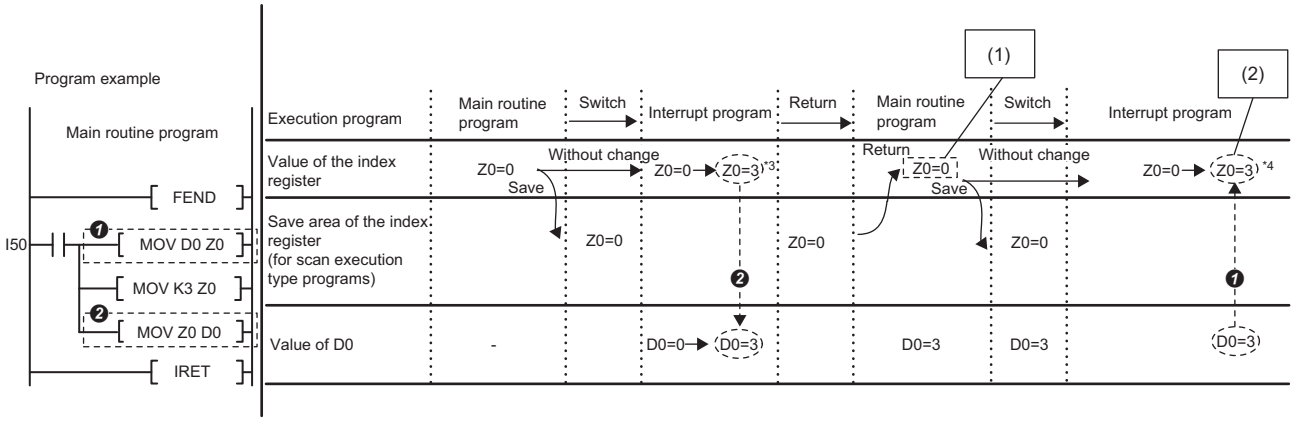


*1 The value of Z0 is changed by the interrupt program, fixed scan execution type program, or event execution type program.

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

Point

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



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

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

Precautions

The precautions for the interrupt program are mentioned below.

■Restrictions on program creation

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

■Data inconsistency

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

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

■If the interrupt accuracy does not increase

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

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

■Monitoring the interrupt program execution time

The execution time of the interrupt program can be monitored with the interrupt program in which the inter-module synchronous interrupt (I44) and multiple CPU synchronous interrupt (I45) are used (☞ Page 230 Error Detection Setting)

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

If the execution interval of the interrupt program is short, the start-up of the interrupt program may be delayed when the instruction with a long processing time is executed, the online ladder block change is performed, or a Q series module is accessed simultaneously from multiple CPU modules using instructions in the multiple CPU system configuration. Therefore, when the execution time is monitored for the interrupt program that uses the inter-module synchronous interrupt (I44) and multiple CPU synchronous interrupt (I45), an error may be detected in the CPU module (☞ Page 230 Error Detection Setting)

■Consideration when using the data logging function

To use occurrence of an interrupt as a data collection condition, the user must specify the fixed scan interval of the interrupt program, taking into consideration the processing time of the data logging function (☞ Page 215 Using together with interrupt programs)

■Interrupt processing with FB/FUN

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

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

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

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

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

■Extended interrupt program processing time

If interrupts during instruction execution are set to "Enable" in the interrupt enabled during instruction execution setting, and the interrupt program is executed when the following conditions are met, the interrupt program processing time will take longer. (☞ Page 65 Interrupt enabled during instruction execution)

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

2 CPU MODULE OPERATING STATUS

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

- RUN state
- STOP state
- PAUSE state

2.1 Operation Processing According to Operating Status of the CPU Module

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

Operation processing in RUN state

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

■Output at the time of entering RUN state

Depending on the setting of output (Y) at the time of change from STOP state to RUN state, either of the following is output.

(☞ Page 162 OUTPUT MODE AT OPERATING STATUS CHANGE (STOP TO RUN))

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

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

Operation processing in STOP state

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

■Output at the time of entering STOP state

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

Operation processing in PAUSE state

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

2.2 Operation Processing When Operating Status of the CPU Module Changed

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

CPU module operating status	CPU module processing			
	Program	External output	Device memory	
			Other than Y	Y
STOP to RUN state	Executes the program from the start.	Determined by the output mode setting when the CPU parameter is STOP to RUN. (☞ Page 162 OUTPUT MODE AT OPERATING STATUS CHANGE (STOP TO RUN))	Retains the device memory status immediately before the RUN state. However, if a device/label initial value has been set, this initial value is set. *1Clears local devices.	Determined by the output mode setting when the CPU parameter is STOP to RUN. (☞ Page 162 OUTPUT MODE AT OPERATING STATUS CHANGE (STOP TO RUN))
RUN to STOP state	Executes up to the END instruction and then stops.	Saves the output (Y) status immediately before the STOP state, and all points turn off.	Retains the device memory status immediately before the STOP state.	Saves the output (Y) status immediately before the STOP state, and all points turn off.
RUN → PAUSE	Stops operation after the execution of one scan.	Retains the output (Y) status immediately before the PAUSE state.	Retains the device memory status immediately before the PAUSE state.	Retains the output (Y) status immediately before the PAUSE state.
PAUSE → RUN	Executes the program from the start.	Retains the output (Y) status immediately before the RUN state.	Retains the device memory status immediately before the RUN state. Clears local devices.	Retains the output (Y) status immediately before the RUN state.
PAUSE → STOP	Operation remains stopped.	Saves the output (Y) status immediately before the STOP state, and all points turn off.	Retains the device memory status immediately before the STOP state.	Saves the output (Y) status immediately before the STOP state, and all points turn off.
STOP → PAUSE	Operation remains stopped.	Retains the output (Y) status immediately before the PAUSE state.	Retains the device memory status immediately before the PAUSE state.	Retains the output (Y) status immediately before the PAUSE state.

*1 For details on the initial label value setting for the Process CPU, refer to the following.
☞ Page 306 LABEL INITIALIZATION FUNCTION

3 OPERATION PROCESSING AT MOMENTARY POWER FAILURE

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

Momentary power failure not exceeding the allowable momentary power failure time

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

■When the momentary power failure is cleared

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

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

Even if the operation is suspended due to a momentary power failure, measurement of the watchdog timer continues. For example, assuming that the monitoring time of scan time is set to 200ms and the scan time is 190ms, a momentary power failure of 15ms causes a WDT error. (☞ Page 89 Scan Time Monitoring Time Setting)

Power failure exceeding the allowable momentary power failure time

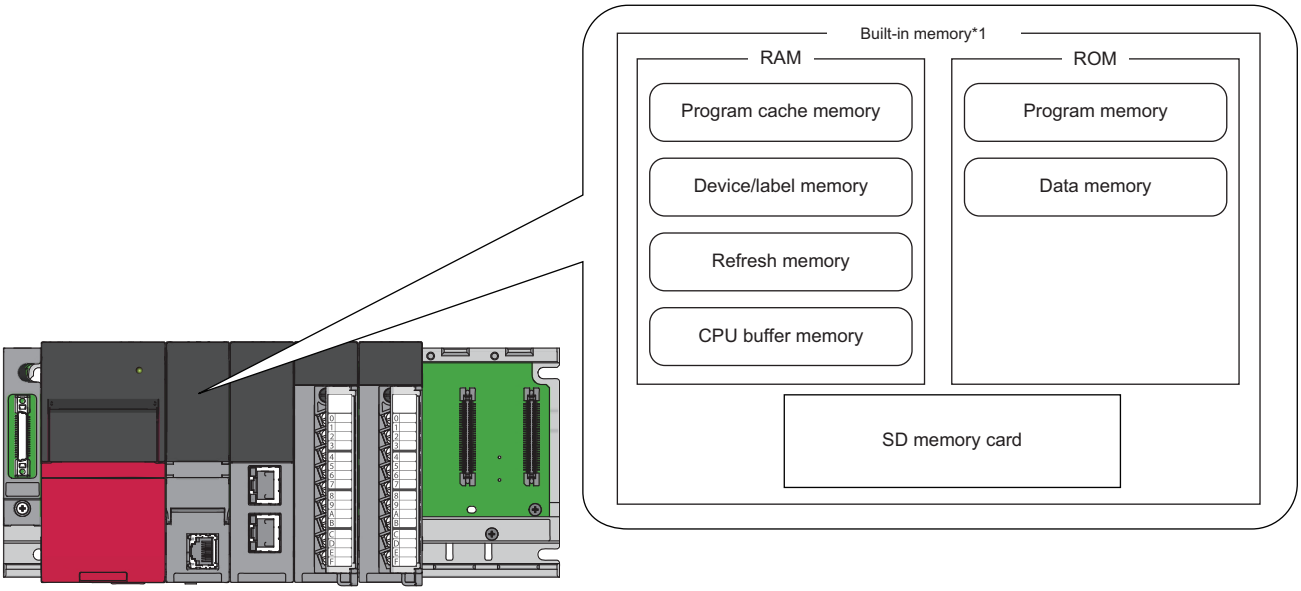
This case results in initial start, the same operation processing as when the CPU module is powered on or is reset.

MEMO

4 MEMORY CONFIGURATION OF THE CPU MODULE

4.1 Memory Configuration

The following shows the memory configuration of the CPU module.



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

Point

The usage of the memory can be checked from the engineering tool. For details, refer to the following.

GX Works3 Operating Manual

Program memory/program cache memory

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

*1 This memory is used for program operations.

- Powered-on
- Reset operation

Application

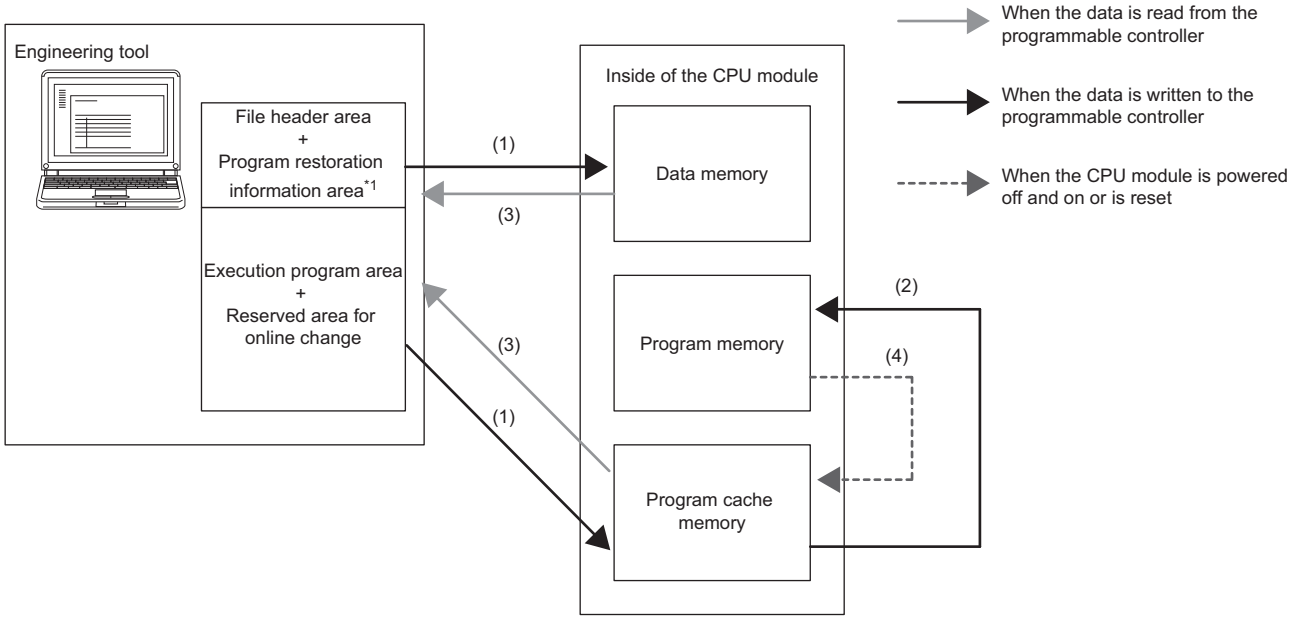
The program memory and program cache memory store execution programs of program files and FB (function block) files. Execution programs include instruction codes, statements, and notes.

Data to be allocated

Only the execution programs that exist in program files and FB files are allocated to the program memory.

Data allocation and procedure of read/write operations

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



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

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

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

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

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

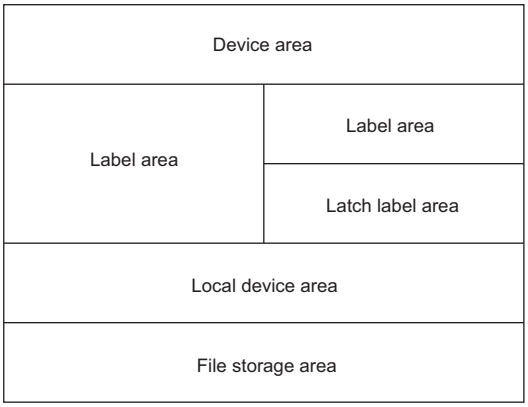
Point

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

GX Works3 Operating Manual

Device/label memory

The device/label memory has the following areas.



Data to be allocated

The following table lists the data allocated to each area.

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

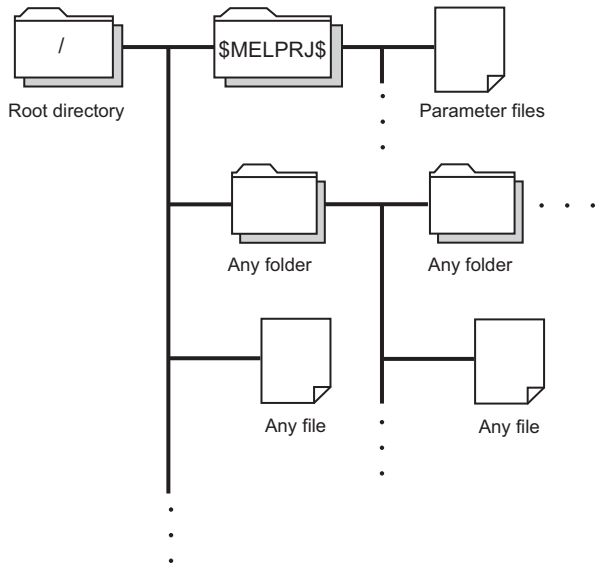
*1 File register files which are stored in the area for storing file register files can be written or read in file unit.

Point

Free space of areas can be checked in "Device/Label Memory Area Capacity Setting". (☞ Page 153 Device/Label Memory Area Setting).

Data memory

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



Point

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

Refresh memory

This memory is used to store data used to refresh communication with the intelligent function module. (Page 339 Refresh Data Register (RD))

CPU buffer memory

This memory is used by the Ethernet function or in data communication between multiple CPUs. (MELSEC iQ-R Ethernet User's Manual (Application))

SD memory card

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

4.2 File Size Unit in Memory

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

File size unit based on memory area

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

Point

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

File size unit of each SD memory card

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

4.3 Memory Operation

Using an engineering tool, each memory can be initialized and cleared to zero. For details on the operation method, refer to the following.

GX Works3 Operating Manual

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

Point

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

Precautions

Memory initialization during execution of another function

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

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

Clearing values during execution of another function

■CPU module data backup/restoration function

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

■iQ Sensor Solution data backup/restoration function

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

4.4 Files

This section lists the files used by the CPU module.

File types and storage memory

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

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

File type		CPU built-in memory			SD memory card	File name and extension
		Program memory	Device/label memory	Data memory		
		Drive 0	Drive 3	Drive 4	Drive 2	
Program		◎*4	×	◎*4	○	ANY_STRING.PRG
FB file		○*4	×	○*4	○	ANY_STRING.PFB
CPU parameter		×	×	◎	○	CPU.PRM
System parameter		×	×	◎	○	SYSTEM.PRM
Module parameter		×	×	○	○	UNIT.PRM
Module extension parameter		×	×	○	○	• UEXmmmnn.PRM*1 • UEXmmm00.PPR*5
Module-specific backup parameter*6		×	×	○	○	UBPmmmnn.BPR*1
Memory card parameter		×	×	×	○	MEMCARD.PRM
Device comment		×	×	○	○	ANY_STRING.DCM
Initial device value		×	×	○	○	ANY_STRING.DID
Global label setting file		×	×	○	○	GLBLINF.IFG
Initial label value file	Initial global label value file	×	×	○	○	GLBLINF.LID
	Initial local label value file	×	×	○	○	PROGRAM_NAME.LID
File register		×	○	×	○*3	ANY_STRING.QDR
Event history		×	×	○	○	EVENT.LOG
Device data storage file		×	×	○	○*3	DEVSTORE.QST
General-purpose data		×	×	○	○	ANY_STRING.CSV/BIN
Data logging setting file	Common setting file	×	×	×	○	LOGCOM.LCS
	Individual setting file	×	×	○	○	LOGnn.LIS*2
Memory dump setting file		×	×	○	○*3	MEMDUMP.DPS
Remote password		×	×	○	○	00000001.SYP
Faulty database check file		×	×	×	○	ErrorDB.txt
System file for backing up CPU module data		×	×	×	○	$BKUP_CPU_INF.BSC
Backup data file for backing up CPU module data		×	×	×	○	BKUP_CPU.BKD
Device/label data file for backing up CPU module data		×	×	×	○	BKUP_CPU_DEVLAB.BKD
System file for the iQ Sensor Solution data backup/restoration function		×	×	×	○	$BKUP_UNIT_INF.BSI
Backup data file for the iQ Sensor Solution data backup/restoration function		×	×	×	○	Depends on a connected device.QBR*7
ODBC server setting file		×	×	○	×	netserver.cfg
Database path file		×	×	×	○	dbmaintainpath.txt

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

*2 nn corresponds to the setting number and is 01 through 10.
*3 Can be stored but cannot operate as a function.
*4 When this file is stored in the built-in memory of the CPU module, it is divided into program memory and data memory and stored. (☞ Page 85 Configuration of a program file)
*5 Module extension parameter for the protocol setting, storing protocol setting information in the predefined protocol support function
*6 The module-specific backup parameter is a file for storing the save or restore data of the module to be replaced by the online module change function. (Process CPU only) For details, refer to the manual for the module used.
*7 This file name depends on the connection type of the iQ Sensor Solution data backup/restoration function. For the file name, refer to the following.
📖 iQ Sensor Solution Reference Manual

File operation available

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

○: Available, —: N/A

File type		Operation from engineering tool			Operation with SLMP and FTP server function			Operation via instruction[*1]	
		Write	Read	Delete	Write	Read	Delete	Write	Read
Program		○[*2]	○	○[*4]	○[*3*6]	○	○[*3*6]	—	—
FB file		○[*4]	○	○[*4]	○[*3*6]	○	○[*3*6]	—	—
Parameter		○[*4]	○	○[*4]	○[*3*6]	○	○[*3]	—	—
Device comment		○	○	○[*4]	○[*3*6]	○	○[*3*6]	—	—
Initial device value		○	○	○[*4]	○[*3*6]	○	○[*3*6]	—	—
Global label setting file		○[*4]	○	○[*4]	○[*3*6]	○	○[*3*6]	—	—
Initial label value file	Initial global label value file	○	○	○[*4]	○[*3]	○	○[*3*6]	—	—
	Initial local label value file	○	○	○[*4]	○[*3]	○	○[*3*6]	—	—
File register		○	○	○[*4]	○	○	○[*3]	○	○
Event history		—	—	—	○	○	○	—	—
Device data storage file		—	—	—	○[*3*6]	○	○[*3*6]	○	○
General-purpose data		○	○	○	○	○	○	○	○
Data logging setting file	Common setting file	○[*5]	○[*5]	○[*5]	○	○	○	—	—
	Individual setting file	○[*5]	○[*5]	○[*5]	○	○	○	—	—
Memory dump setting file		○	○	○	○	○	○	—	—
Remote password		○[*4]	○	○[*4]	○[*3*6]	○	○[*3*6]	—	—
Faulty database check file		○	○	○	○	○	○	○	○
Module-specific backup parameter		○	○	○[*3]	○	○	○[*3]	○	○
System file for backing up CPU module data		—	—	—	○	○	○	—	—
Backup data file for backing up CPU module data		○	○	○	○	○	○	—	—
Device/label data file for backing up CPU module data		○	○	○	○	○	○	—	—
System file for the iQ Sensor Solution data backup/restoration function		—	—	—	○	○	○	—	—
Backup data file for the iQ Sensor Solution data backup/restoration function		○	○	○	○	○	○	—	—
ODBC server setting file		—	—	—	○	○	○	—	—
Database path file		—	—	—	○	○	○	—	—

*1 Modification of data in files, such as read/write from/to file register and execution of the FWRITE/FREAD instruction

*2 Done during write to the programmable controller when the CPU module operation status is STOP, and during online program change when the status is RUN. In addition, when the operating status of the CPU module is RUN, only the program that is registered with the parameter can perform write operation.

*3 Available only when the CPU module operation status is STOP/PAUSE. A communication error occurs when operated in the RUN state.

*4 Available only when the CPU module operation status is STOP/PAUSE. When a program tries to perform memory operation while the CPU module is RUN, the operation is continued after the operation status is changed through remote STOP.

*5 Operation on CPU Module Logging Configuration Tool.

*6 When the operation target is the SD memory card, the operation can be performed even while the CPU module is RUN.

File size

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

File type		File size
Program		Approx. 4050 bytes minimum (only END instruction + 500 steps reserved for online program change)
FB file		Approx. 4900 bytes minimum (non-processing FB + 500 steps reserved for online program change)
CPU parameter		744 bytes minimum
System parameter		112 bytes minimum
Module parameter		The size differs depending on the module used. For example, the size is 1036 bytes at minimum when the Ethernet function is used.
Module extension parameter		The size differs depending on the module used. For example, the size is 65572 bytes when the predefined protocol support function is used with the Ethernet function.
Module-specific backup parameter		The size differs depending on the module used.
Memory card parameter		124 bytes minimum
Device comment		220 bytes minimum (when a device comment with 20 characters is set in a device)
Initial device value		Approx. 140 bytes minimum (When the initial value of a word device is set)
Global label setting file		Approx. 660 bytes minimum (When a word-type global label is set)
Initial label value file	Initial global label value file	Approx. 140 bytes minimum (When the initial value of a word type global label is set)
	Initial local label value file	
File register		2048 bytes minimum (when the setting capacity is 1K words)
Event history		1024 bytes minimum (when the setting capacity is 1K words (default setting is 131072 bytes))
Device data storage file		2048 bytes minimum (when the setting capacity is 1K words)
General-purpose data		The size differs depending on the size of the file to be written.
Data logging setting file	Common setting file	142 bytes (When the setting is configured so that one-word data is collected in a binary file at each scan in continuous logging)
	Individual setting file	1192 bytes (When the setting is configured so that one-word data is collected in a binary file at each scan in continuous logging only for setting No.1)
Memory dump setting file		1224 bytes (when the save file name is "MEMDUMP_00")
Remote password		224 bytes minimum
Faulty database check file		Total number of characters in the file path × 2 bytes + number of databases × 4 bytes
System file for backing up CPU module data		20 + 12 + ((N1 × 34) + (N2 × 34) + (N3 × 34)) + M + 8 bytes N1: No. of target drives N2: No. of target files N3: No. of target folders M: Total name size of target files/folders (bytes) (including ".", extensions, and no drive symbols)
Backup data file for backing up CPU module data		When the data logging setting has been registered: 30608 bytes When the data logging setting has not been registered: 30200 bytes
Device/label data file for backing up CPU module data		■When local devices are not set • 20 + 20 + (N1 × 142) + 16 + (N2 × 134) + (16 + (162 + (N3 × 18) + 1087104 + M1)) + 28 + M3 + M4 + M5 + 8 bytes[*1] ■When local devices are set • 20 + 20 + (N1 × 142) + 16 + (N2 × 134) + (16 + (162 + (N3 × 18) + 1087104 + M1)) + (16 + N1 × (8 + ((N4 × 18) + M2))) + 28 + M3 + M4 + M5 + 8 bytes[*1] N1: No. of programs N2: No. of file registers N3: No. of set device types (excluding SM, SD, FX, FY, FD, X, Y, and G/HG) N4: No. of local device settings (No. of device types) M1: Total size of set devices (bytes) (excluding SM, SD, FX, FY, FD, X, Y, and G/HG) M2: Total size of local devices (bytes) M3: Module label size (bytes) M4: Label area size (bytes) M5: Latch label area size (bytes)
System file for the iQ Sensor Solution data backup/restoration function		36 + 20 × N bytes (N: No. of devices supporting iQSS whose data has successfully been backed up)

File type	File size
Backup data file for the iQ Sensor Solution data backup/restoration function	Depends on the devices supporting iQSS.
ODBC server setting file	260 bytes (when the IP address of the CPU module is 192.168.3.39)
Database path file	Total number of characters in the file path × 2 bytes + number of databases × 4 bytes

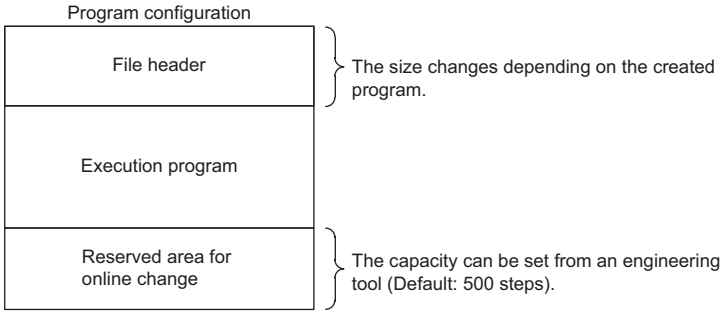
*1 Up to 3 bytes are added to the total file size so that the size will be a multiple of 4 bytes.

Configuration of a program file

The following figure shows the configuration of a program.

■Configuration of a program

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

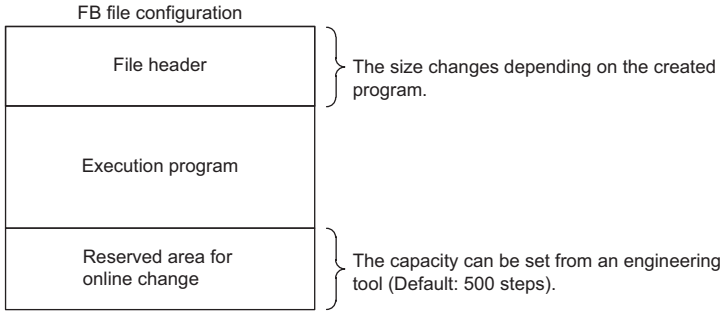


When the program is stored in the CPU module, execution program and reserved area for online change are allocated to the program memory. The file header is allocated to the data memory. The following table shows the destination memory to which each area is allocated when a program is stored in the CPU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

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

■FB file

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



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

MEMO

PART 2 FUNCTIONS

This part consists of the following chapters.

5 SCAN MONITORING FUNCTION

6 CLOCK FUNCTION

7 ONLINE CHANGE

8 INTERRUPT FUNCTION

9 PID CONTROL FUNCTION

10 PROCESS CONTROL FUNCTION

11 CONSTANT SCAN

12 DATABASE FUNCTION

13 REMOTE OPERATION

14 DEVICE/LABEL MEMORY AREA SETTING

15 INTERNAL BUFFER CAPACITY SETTING

16 DEVICE/LABEL INITIAL VALUE SETTINGS

17 OUTPUT MODE AT OPERATING STATUS CHANGE (STOP TO RUN)

18 LABEL ACCESS SETTING FROM EXTERNAL DEVICE

19 ROUTING SETTING

20 LATCH FUNCTION

21 MEMORY CARD FUNCTION

22 DEVICE/LABEL ACCESS SERVICE PROCESSING SETTING

23 DATA LOGGING FUNCTION

24 MEMORY DUMP FUNCTION

25 REAL-TIME MONITOR FUNCTION

26 RAS FUNCTIONS

27 CPU MODULE DATA BACKUP/RESTORATION FUNCTION

28 EXTERNAL INPUT/OUTPUT FORCED ON/OFF FUNCTION

29 MULTIPLE CPU SYSTEM FUNCTION

30 SECURITY FUNCTION

31 SEQUENCE SCAN SYNCHRONIZATION SAMPLING FUNCTION

32 LABEL INITIALIZATION FUNCTION

5 SCAN MONITORING FUNCTION

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

• When using a Process CPU (redundant mode), refer to the following as well.
☞ Page 387 FUNCTIONS

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

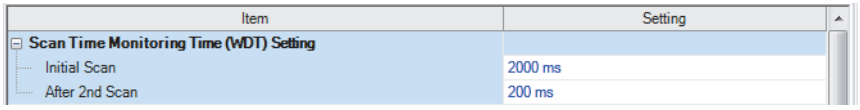
- Initial scan (first scan)
- Second and later scans

5.1 Scan Time Monitoring Time Setting

Sets the scan time monitoring time.

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

Window



Displayed items

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

5.2 Watchdog Timer Reset

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

5.3 Precautions

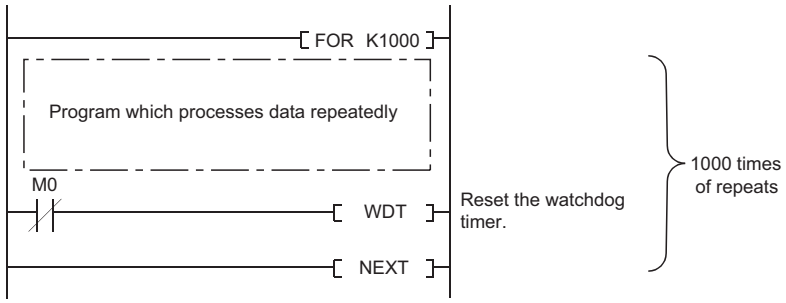
The following lists the precautions on the scan monitoring function.

Measurement error of watchdog timer

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

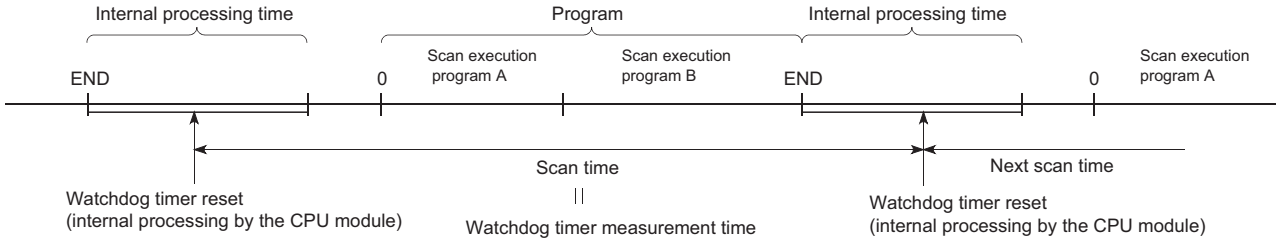
Resetting the watchdog timer when repeatedly executing a program

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



Scan time when the WDT reset instruction is used

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



6 CLOCK FUNCTION

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

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

☞ Page 387 FUNCTIONS

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

☞ Page 486 FUNCTIONS

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

6.1 Time Setting

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

Clock data

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

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

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

Changing the clock data

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

- Using the engineering tool
- Using SM/SD
- Using the instruction

Point

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

- The millisecond clock is reset to 0.
- Clock setting (Event code: 24000) is logged in the event history.

Using the engineering tool

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

Using SM/SD

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

Using the instruction

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

Reading the clock data

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

- Using SM/SD
- Using the instruction

Using SM/SD

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

Using the instruction

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

Precautions on the clock data

The following lists the precautions on the clock data.

When using the product for the first time

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

Modifying the clock data

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

Range of the clock data

The clock data must be written within the following range.

☞ Page 91 Clock data

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

Ex.

Operating status of the CPU module when impossible date is set

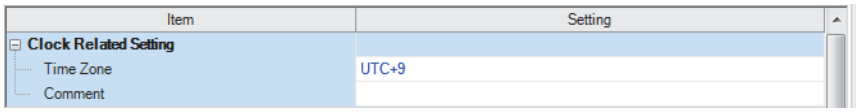
Date	Write operation to the CPU module	CPU module operating status
February 30	Executed	An error is not detected.
32th of month 13	Not executed	• When the DATEWR instruction is executed, "Operation error" (error code: 3405H) is detected. • When SM210 is on, SM211 turns on.

6.2 Setting Time Zone

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

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

Window



Displayed items

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

Point

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

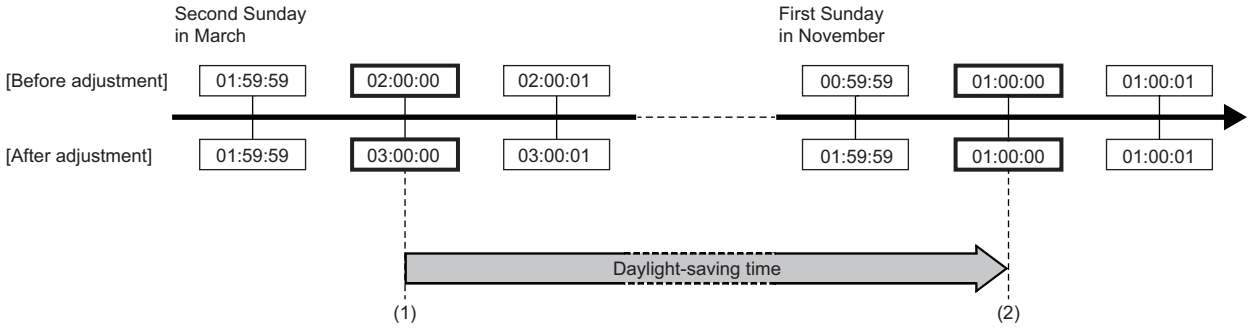
6.3 Daylight Saving Time Function

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

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

Ex.

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



Restriction

Before executing the daylight-saving time function, check the versions of the CPU module and engineering tool used. (☞ Page 779 Added and Enhanced Functions)

Daylight saving time setting

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

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

Window

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

Displayed items

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

Point

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

Restriction

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

Timing of daylight saving time adjustment

Daylight saving time is adjusted at the following times.

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

Daylight saving time function operation check

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

Special relays

SM217 (Daylight saving time status flag) can be used to check whether the date lies inside or outside the daylight saving time period. (☞ Page 606 List of Special Relay Areas)

Event history

The history for the start and end of daylight saving time can be checked in the event history for the date set in "Setting to Adjust Clock for Daylight Saving Time" (☞ Page 596 Event list)

Operation of other functions using clock data during daylight saving time

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

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

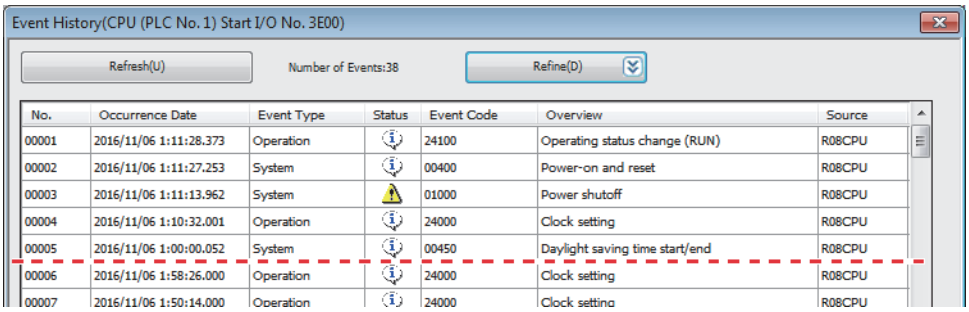
Precautions

Please be aware of the following daylight saving time function precaution.

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

Ex.

Event history



Since the time goes back one hour by ending the daylight saving time using No.00005, the even history will be (the date on which events occur of new event history) < (the date on which events occur of old event history) at the upper and bottom side of the broken line part.

6.4 System Clock

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

Special relay used for system clock

Special relay used for system clock are as follows (☞ Page 613 System clock)

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

Precautions

- SM409 to SM415 (System clock) start to measure the time before the CPU module is switched to RUN. Therefore, the time from the first scan after the CPU module has been switched to RUN until switching of the on/off state of the system clock may not match the measurement time of the system clock.

Special register used for system clock

Special register used for system clock are as follows (☞ Page 651 System clock)

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

7 ONLINE CHANGE

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

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

☞ Page 387 FUNCTIONS

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

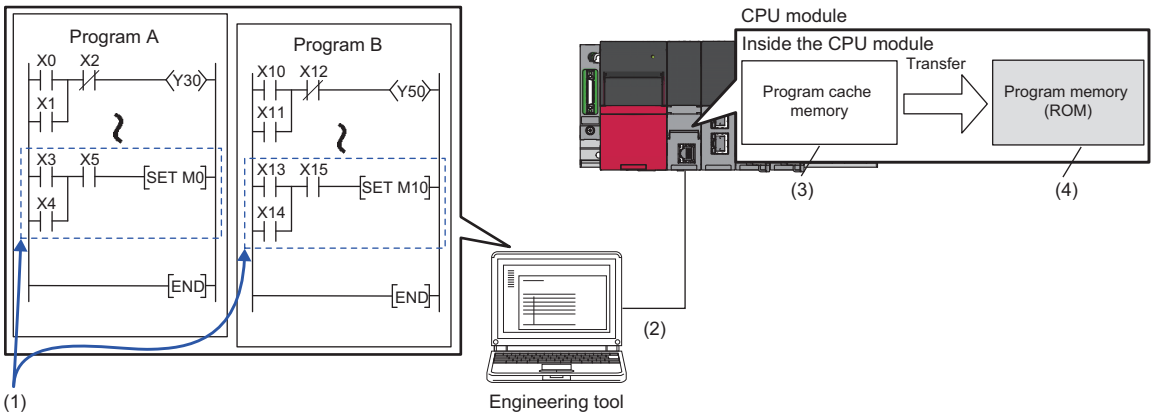
☞ Page 486 FUNCTIONS

The following table lists the types of the online change.

Type		Description	Reference
Online change	Online change (ladder block)	Changes and writes a part of the program and data online.	• ☞ Page 98 Online Ladder Block Change • 🕮 GX Works3 Operating Manual
	SFC program	Executes the online change from Zoom, changes, adds, and deletes blocks to an inactive block online.	• 🕮 GX Works3 Operating Manual • 🕮 MELSEC iQ-R Programming Manual (Program Design)
file batch online change		Writes data in file units online.	🕮 GX Works3 Operating Manual

7.1 Online Ladder Block Change

Writes the portion edited on the ladder edit window of the engineering tool to the CPU module in increments of ladders. Edited contents spanning multiple files or multiple portions can be written to the CPU module at once.



(1) Portion edited in the engineering tool
(2) Writes the changed ladder block to the running CPU module.
(3) Changes the program contents in the program cache memory.
(4) After changing the program contents in the program memory, automatically transfers them to the program memory.

Point

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

🕮 GX Works3 Operating Manual

Editable contents

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

Editable contents within a program block

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

Editable contents within FB definition

- To add a local label, the local label size cannot exceed the reserved area capacity[*1]. The same applies to the standard FBs, process control FBs, and module FBs because those FBs also have local labels. For details, refer to the technical bulletin, FA-A-0232.
- To add a local label, all the programs and FB files referencing the FB file which stores the FB definition to be modified must match both on the engineering tool and in the CPU module.
- For input and output labels (VAR_INPUT, VAR_OUTPUT, and VAR_IN_OUT) and public labels, to add, change, or delete subroutine-type FB and FUN interface information[*2], all the programs and FB files referencing the FB file storing the FB definition to modify must match both on the engineering tool and in the CPU module.

*1 This reserved area is for adding a local label or a local instance when a program is changed online. The area is fixed to 48 words for non-latch type local labels, and 16 words for latch type labels.

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

Editable contents within FUN definition

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

Editable contents for each program component

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

Range changeable in a single session

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

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

Reserved area for online change

Reserved area for online change can be set in a program file to address online ladder block change which causes a change in the program file size. (📖 GX Works3 Operating Manual)
In addition, if the changed program exceeds the program file capacity (including reserved area for online change) during online ladder block change, the reserved area for online change can be set again if there is space available in the program memory. If the data memory does not have enough space when the online ladder block change is performed, an error occurs. The free space of the data memory required for the execution of the online ladder block change depends on the size of program files to be changed or the usage of labels.

Target program of online ladder block change	Required data memory space	
	When labels are not used	When labels are used
For a ladder program of 1K steps	Approx. 20K bytes	Approx. 40K bytes
For a ladder program of 10K steps	Approx. 80K bytes	Approx. 140K bytes
For a ladder program of 100K steps	Approx. 550K bytes	Approx. 1000K bytes

Online ladder block change during the boot operation

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

Setting the initial value for registering/changing label definition

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

Initial value setting availability

Indicates whether or not the initial value can be set when adding or changing a label.
○: Available, △: Conditionally available, ×: Not available

Label type		Label addition	Label change
Program block	Global label	○	△*1
	Local label	○	△*1
FB definition	Local label	○*2	△*1
FUN definition	Local label	×*3	×*3

*1 By changing the data type, the initial value can be set only when a new area is assigned again. The initial value can be set for the following cases:
· Changing the data type
· Changing the latch setting
· Changing the constant setting (changing from constant to variable)
*2 A different initial value can be set for each instance.
*3 Since the local label in FUN definition is an undefined value, initialize it by a program in the FUN definition before using it in a program.

Point

When using GX Works3 version 1.000A

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

Precautions

This section describes the precautions on using online ladder block change.

Prohibited operation at online ladder block change and writing data

If the CPU module is powered off or is reset when the data is written to the built-in memory of the CPU module during the online ladder block change or write to the programmable controller, the operation does not complete normally. If doing so, write the data to the programmable controller again.

Operation from engineering tools

During online ladder block change and write to the programmable controller, the following operations cannot be performed from the engineering tool:

- Online ladder block change or write to the programmable controller
- Write to the programmable controller (excluding device, local device, global label, or local label data).
- Memory initialization

When deleting OUT instruction which is on

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

Program file not registered in program setting

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

Instructions which do not operate correctly

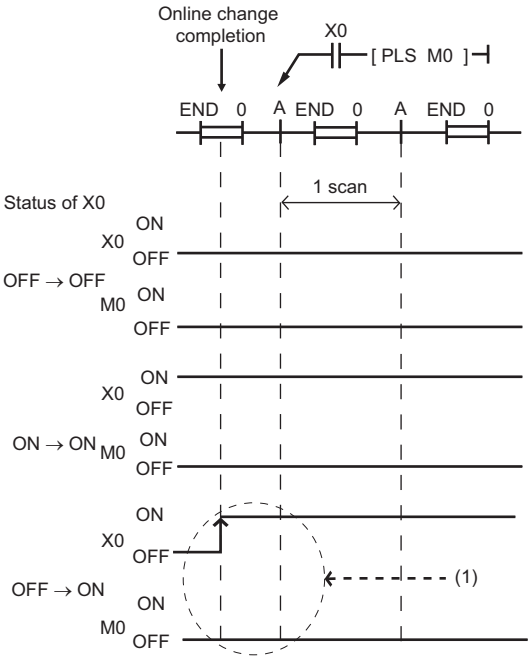
This section describes instructions which do not operate correctly during online ladder block change or write to the programmable controller. In the SFC program, these instructions do not operate correctly only when they are in active steps.

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

■Rising instruction

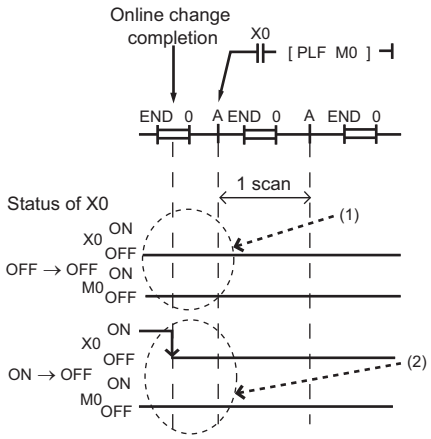
When a rising instruction exists within the range to be changed, the rising instruction will not be executed even if the execution condition (OFF to ON) is satisfied after completion of online ladder block change or write to the programmable controller.

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



■Falling instruction

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

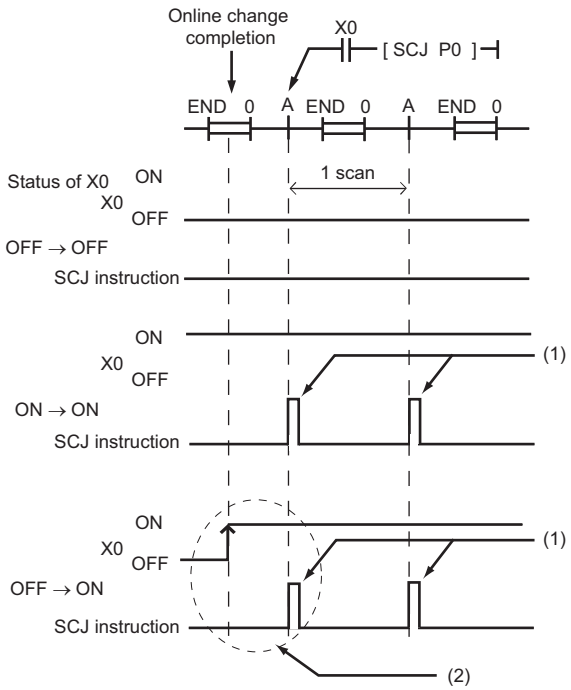


(1) The falling instruction will not be executed even if the execution condition is OFF to OFF.

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

■SCJ instruction

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

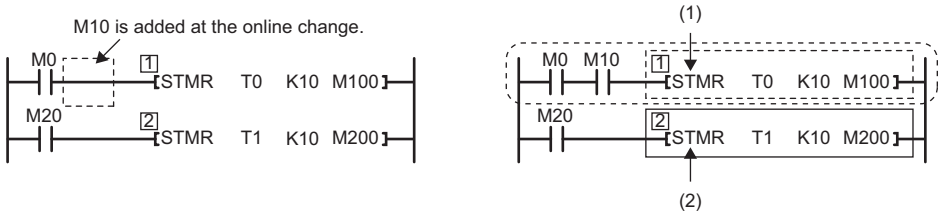


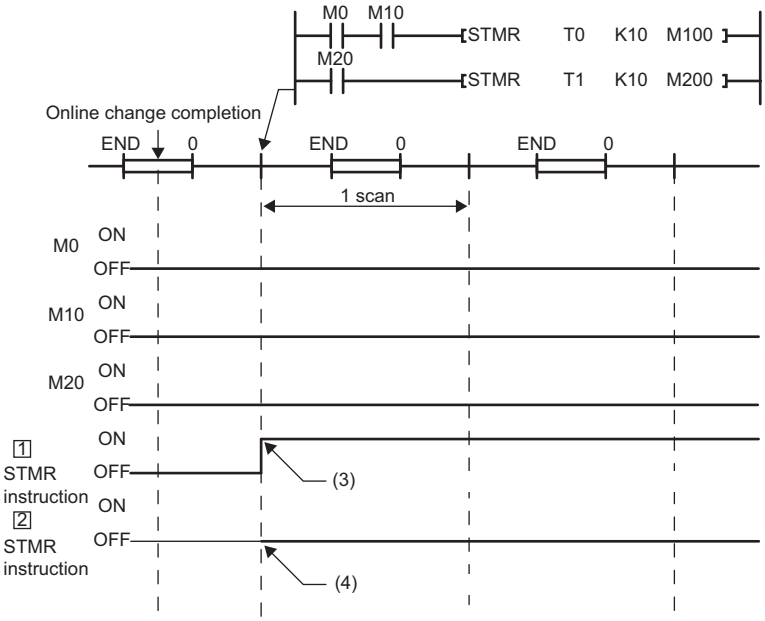
(1) Jumps to the specified pointer.

(2) Jumps to the specified pointer without waiting for a single scan.

■STMR instruction

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





(1) The STMR instruction functions because the STMR instruction is included in the ladder block where the online change has been executed.
(2) The STMR instruction does not function because the STMR instruction is not included in the ladder block where the online change is executed.
(3) The instruction functions even when M0 and M10 are off.
(4) The instruction does not function.

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

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

When the start-up of an interrupt program is delayed

When the online ladder block change is performed, the start-up of the interrupt program may be delayed. Therefore, when the execution time is monitored for the interrupt program that uses the inter-module synchronous interrupt (I44) and multiple CPU synchronous interrupt (I45), an error may be detected in the CPU module. (☞ Page 230 Error Detection Setting)

Scan monitoring function during online changes

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

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

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

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

8 INTERRUPT FUNCTION

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

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

☞ Page 387 FUNCTIONS

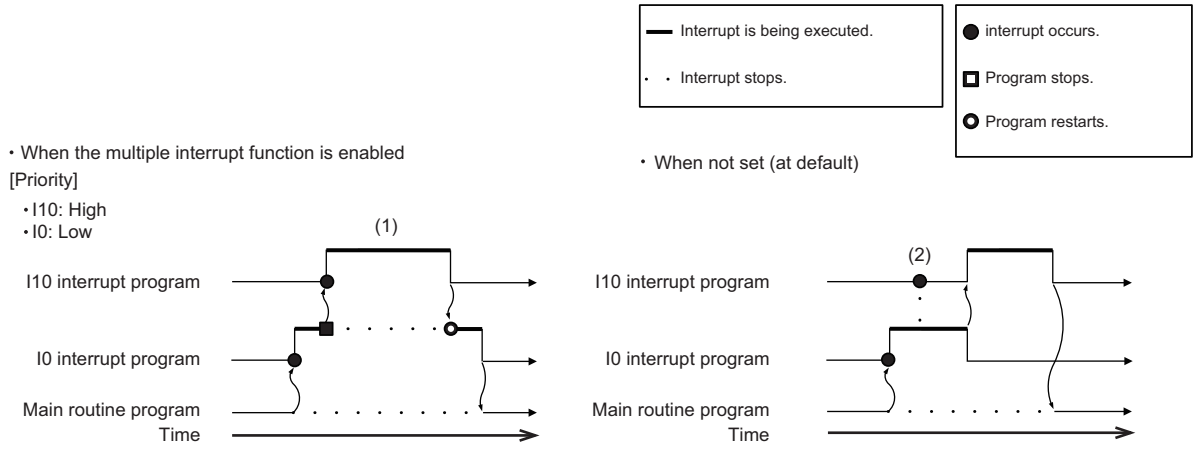
- When using a Safety CPU, refer to the following as well.

☞ Page 486 FUNCTIONS

This chapter describes the interrupt function.

8.1 Multiple Interrupt Function

When an interrupt occurs while an interrupt program triggered by another cause is running, stops the program if its priority is lower than that of the new interrupt, and runs the higher-priority program whenever its execution condition is satisfied.



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

Interrupt priority

If the interrupt priority of a program for which its execution condition has been satisfied is higher than that of the running program, the programs are executed in accordance with their interrupt priority. If the interrupt priority of the new program is the same or lower, it enters the waiting status until the running program finishes. The interrupt priorities 1 to 4 listed below cannot be changed, whereas the interrupt priorities 5 to 8 can. (☞ Page 107 Interrupt priority setting)

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

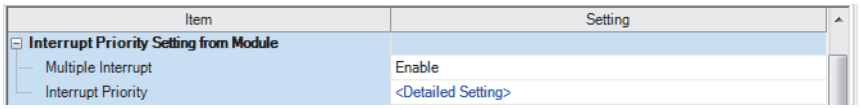
Interrupt priority setting

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

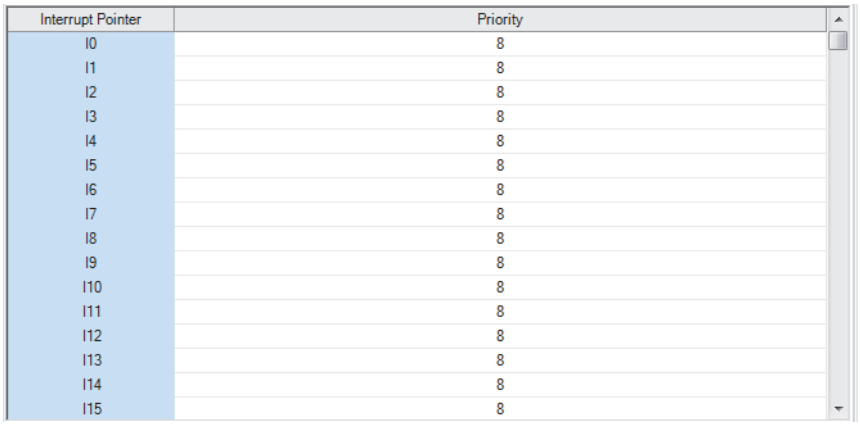
[CPU Parameter] ⇨ [Interrupt Settings] ⇨ [Interrupt Priority Setting from Module]

Operating procedure

"Interrupt Settings" window



"Detailed Setting" window



1. Set "Multiple Interrupt" to "Enable" on the "Interrupt Settings" window, and click "Detailed Setting".
2. Change the priority of each interrupt pointer.

Displayed items

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

Disabling/enabling interrupts with a specified or lower priority

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

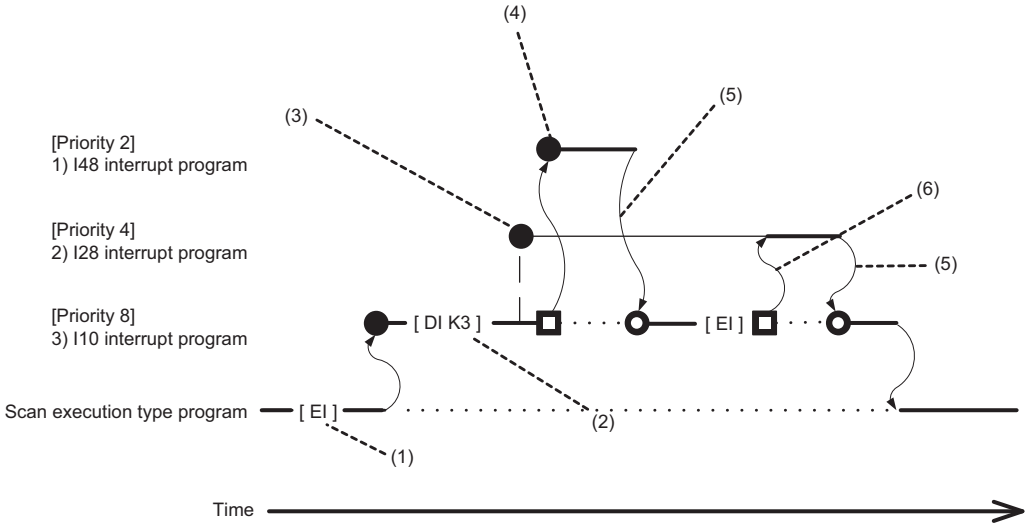
Ex.

Order of interrupt occurrence: ❸→❷→❶

Order of interrupt execution: ❸→❶→❷

Order of interrupt completion: ❸→❶→❷

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



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

Point

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

Multiple interrupt execution sequence

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

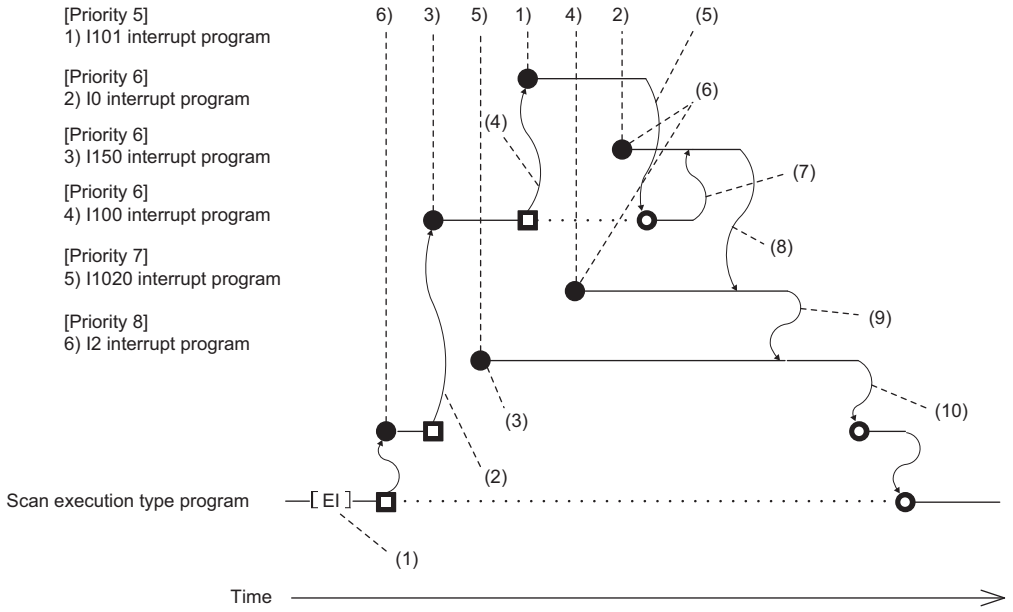
Ex.

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

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

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

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



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

Precautions

This section describes the precautions when using the multiple interrupt function.

Delay in interrupt programs

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

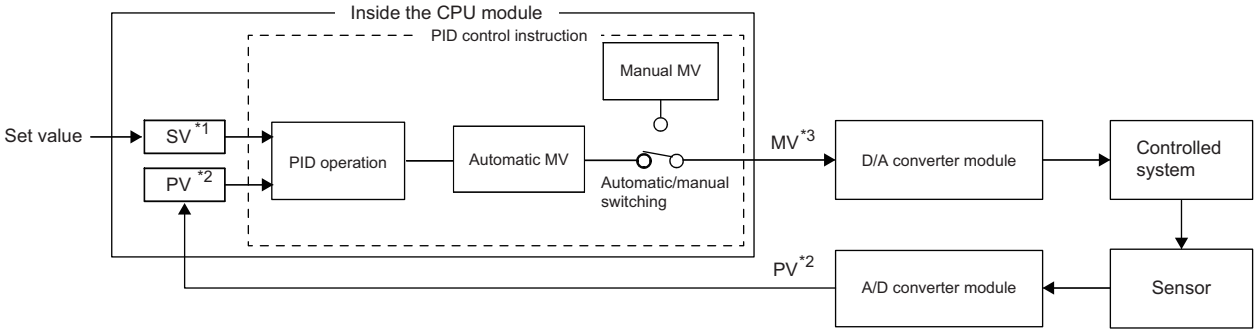
Devices and labels used by interrupt programs

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

9 PID CONTROL FUNCTION

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

PID control is applicable to process control in which factors such as flow rate, velocity, air flow volume, temperature, tension, mixing ratio must be controlled. The control for maintaining the control object at the preset value is shown in the diagram below: PID control via PID control instructions is implemented by combining the CPU module with the A/D converter module and D/A converter module.



*1 SV: Set value
*2 PV: Process value
*3 MV: Manipulated value

In the PID control, the PID operation is executed to compare the value set beforehand (SV) with the digital value (process value (PV)) in which the analog value measured by a sensor is read from the A/D converter module.
PID operation utilizes proportional operation (P), integral operation (I), and derivative operation (D) in combination, thereby calculating the manipulated value (MV) quickly and accurately so that a process value (PV) is to be equivalent to a set value (SV). A larger difference between the process value (PV) and set value (SV) results in a faster speed to reach the set value (SV) quickly by increasing the manipulated value (MV) and a smaller difference between the process value (PV) and set value (SV) results in a slower speed to reach the set value (SV) accurately by decreasing the manipulated value (MV). The calculated manipulated value (MV) is output to the D/A converter module.

Point

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

MEMO

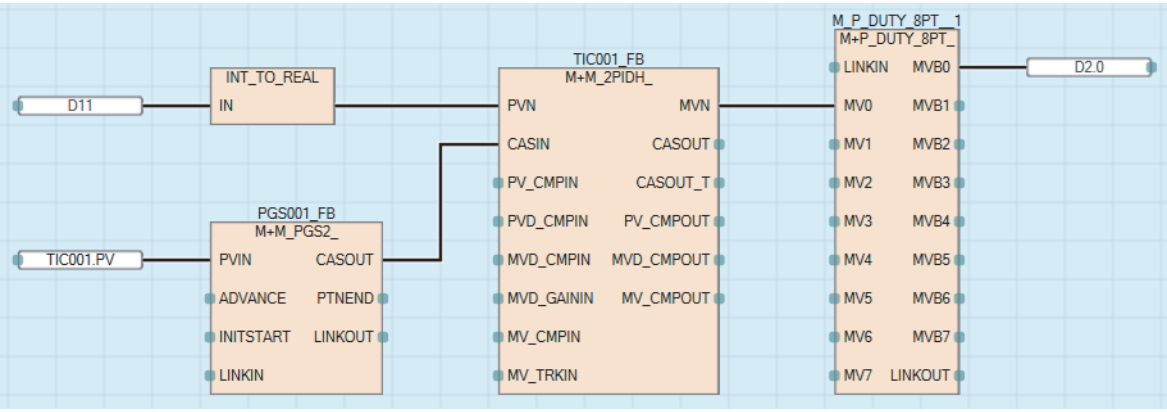
10 PROCESS CONTROL FUNCTION

This chapter describes the process control function.

10.1 Process Control by Using Process Control Function Blocks

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

A process control function block is the function block that is extended its functions for the process control. A process control program can be easily created by using process control function blocks.



Process control function blocks have features as follows.

- A program can be easily created by placing and connecting FB elements using various types of function block provided for the process control.
- Since the initial value of the function block can be set in the "FB Property" window of the engineering tool, the program for the initial value setting is not required.
- An argument of a function block can be specified using a label without considering the address of a device.
- The execution status of a tag FB can be checked and controlled by accessing the tag data from the faceplate of the engineering tool.

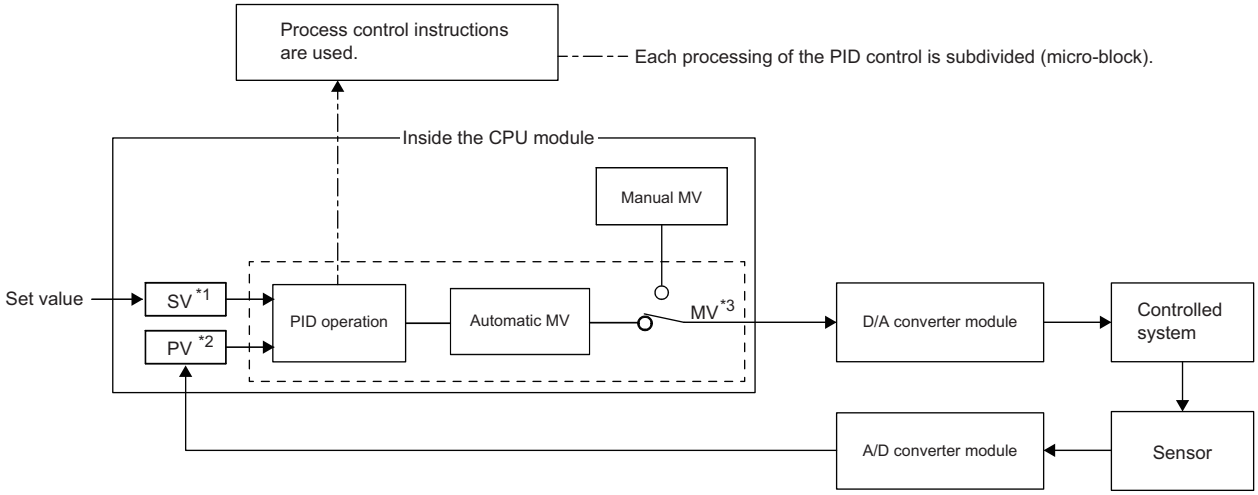
Point

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

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

10.2 Process Control by Using Process Control Instructions

This function performs various types of process control by using process control instructions, such as two-degree-of-freedom PID control, sample PI, and auto tuning, in combination.
Since each processing of the PID control is subdivided and multiple process control instructions are used in combination, this function performs more accurate and sensitive control than the PID control function.



*1 SV: Set value
*2 PV: Process value
*3 MV: Manipulated value

Process control instructions have the following features:

- Increased efficiency of system adjustment: Multiple process control instructions are combined to perform PID control, and operation can be checked for each instruction individually.
- Application to a wide range of control: Another process control instruction can be added to the existing control loop, which consists of multiple process control instructions, as an option.
- Configuration of safety system: Alarms are automatically detected in the system.
- Auto tuning: The auto tuning instruction automatically calculates PID constants that are most suitable for the control system.

Point

- When the process control function is used, the process control by the process control function block is recommended. For the process control function block, refer to the following.

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

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

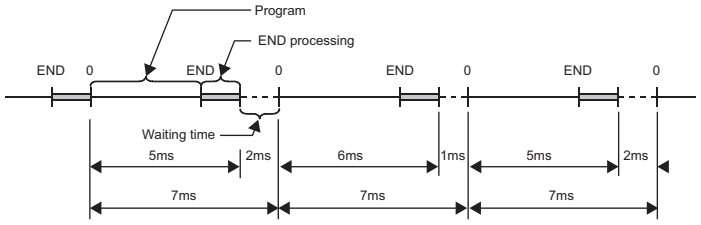
MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

11 CONSTANT SCAN

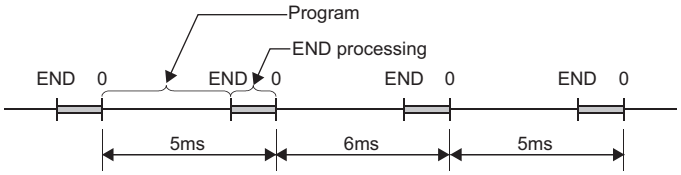
RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnSFCPU (Standard) RnSFCPU (Safety)

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

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



- When constant scan is not set

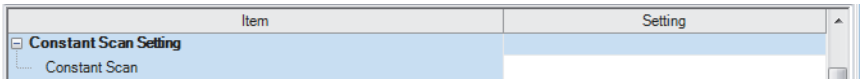


11.1 Setting Constant Scan

Constant scan can be set.

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

Window



Displayed items

Item	Description	Setting range	Default
Constant Scan	Sets the constant scan time.	0.2 to 2000.0ms (in unit of 0.1ms)	—

Point

The setting time for the constant scan must be a value that satisfies the relational expression below.
"Watchdog timer setting time" > "Constant scan setting time" > "Maximum scan time of the program"
If the maximum scan time of the program is longer than the constant scan setting time, it is assumed as an error to ignore the constant scan, and the scan time of the program is applied.

11.2 Accuracy of Constant Scan

The accuracy of the constant scan is 0.01ms. However, if processing, which should be executed during the waiting time from the completion of the END processing to the start of the next scan, is being executed, the constant scan cannot finish even if the constant scan time is reached. The constant scan may increase by the program execution time of the interrupt factor.
(☞ Page 116 Precautions)

11.3 Precautions

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

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

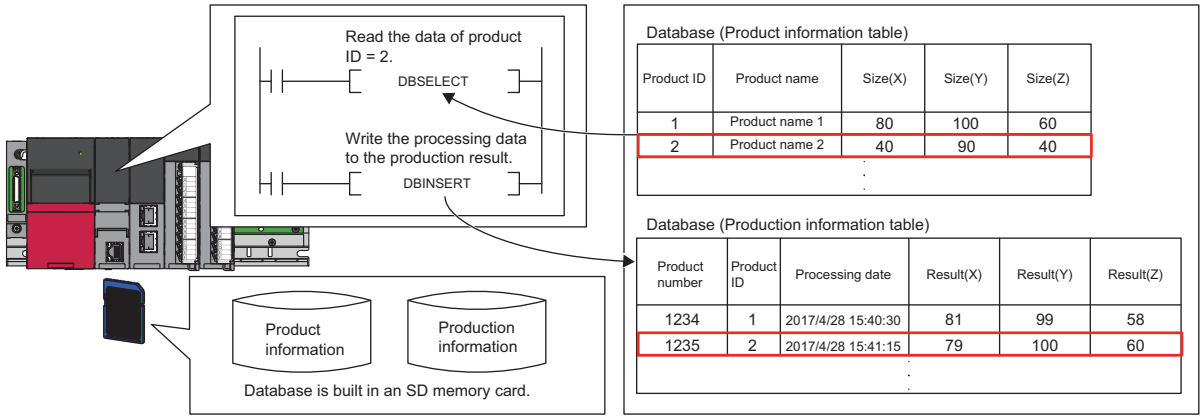
12 DATABASE FUNCTION

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

This function manages the data such as product information or production information as a database in an SD memory card of the CPU module.

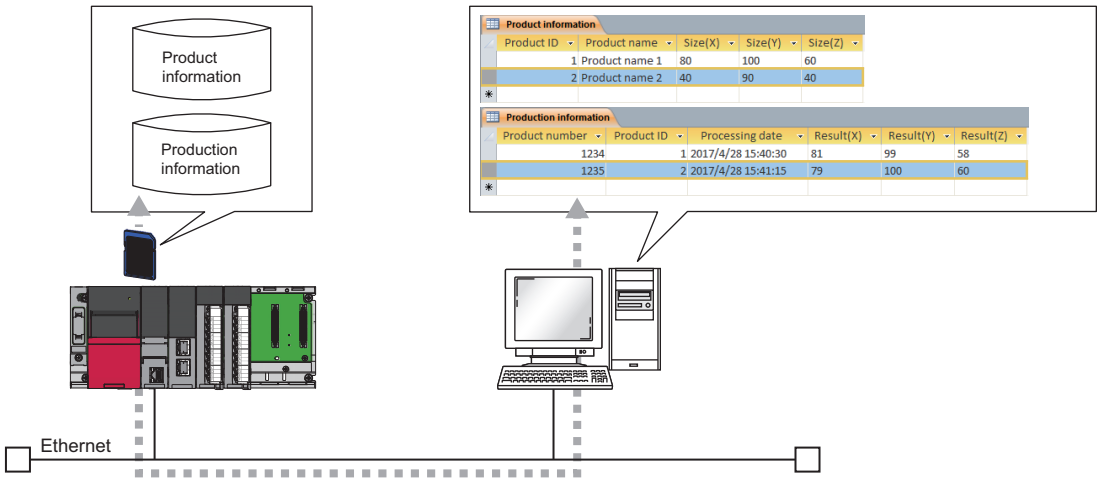
Database access instruction

The database access instructions allow the databases of product information and production information to be created, input, updated, edited, searched, transacted, and rolled back.



CPU module database access (from external device) function

The CPU module database access function enables operations such as table creation and record operation/search of the database built in an SD memory card inserted into the CPU module.



12.1 Specifications

Database specifications

The following table lists specifications of databases which the user can create.

Item	Description
The maximum number of fields	128 fields per table[*1]
The maximum number of records	No limitation (they can be created up to the capacity of the SD memory card)[*2*4]
The maximum number of tables	32 tables per database[*3]
The maximum number of databases	No limitation (they can be created up to the capacity of the SD memory card)
The maximum number of concurrently connectable databases (number of connections)	4
The maximum number of characters of a field name	32 single-byte alphabetical characters
The maximum number of characters of a table name	32 single-byte alphabetical characters
Corresponding data type	☞ Page 118 Data type

*1 For the programmable controller CPU with firmware version earlier than "28", up to 16 fields can be created per table.
*2 For the programmable controller CPU with firmware version earlier than "28", up to 100000 records can be created per table.
*3 For the programmable controller CPU with firmware version earlier than "28", up to 16 tables can be created per database.
*4 For rough standards of the access time depending on the database size, refer to the following.
☞ Page 709 Database function processing time

■Data type

The following table lists the data types that can be used in the table created by using the database access instructions and the corresponding data types for the CPU module database access function.

Data type that can be used in the table created by using the database access instruction	Range	Corresponding data type for the CPU module database access function
BOOL: Bit	0, 1	BOOLEAN: Bit
WORD: Unsigned 16-bit numerical value	0 to 65535	INT: Unsigned 16-bit numerical value
DWORD: Unsigned 32-bit numerical value	0 to 4294967295	BIGINT: Unsigned 32-bit numerical value
INT: Signed 16-bit numerical value	-32768 to 32767	INT: Signed 16-bit numerical value
DINT: Signed 32-bit numerical value	-2147483648 to 2147483647	BIGINT: Signed 32-bit numerical value
REAL: Single-precision real number	E±1.175495-38 to E±3.402823+38	REAL: Single-precision real number
LREAL: Double-precision real number	E±2.22507385850721-308 to E±1.79769313486231+308	DOUBLE PRECISION: Double-precision real number
STRING: String (Shift-JIS code)	Number of supported characters: 1 to 124	NLSCHAR: String (Unicode)
WSTRING: String (Unicode)	Number of supported characters: 1 to 124	NLSCHAR: String (Unicode)

The following table lists the data types that can be used in the table created by the CPU module database access function and the corresponding data types for the database access instructions.

Data type that can be used in the table created by the CPU module database access function	Range	Corresponding data type for the database access instruction
BOOLEAN: Bit	0, 1	BOOL: Bit
INT: Signed 16-bit numerical value	-32768 to 32767	INT: Signed 16-bit numerical value
BIGINT: Signed 32-bit numerical value	-2147483648 to 2147483647	DINT: Signed 32-bit numerical value
REAL: Single-precision real number	E±1.175495-38 to E±3.402823+38	REAL: Single-precision real number
DOUBLE PRECISION: Double-precision real number	E±2.22507385850721-308 to E±1.79769313486231+308	LREAL: Double-precision real number
NLSCHAR: String (Unicode)	Number of supported characters: 1 to 124	WSTRING: String (Unicode)

Point

- To input the data to the field of STRING using the CPU module database access function, use the data type of NLSCHAR. (Although the data is managed as Unicode in the database of the CPU module, it is converted to Shift-JIS code on the device of the CPU module.)
- To input the data in the data type of WORD or DWORD to the field using the CPU module database access function, the data must be within the range of WORD: 0 to 65535 or DWORD: 0 to 4294967295.

Available operations

The following table lists the operations that can be performed with the database functions.

○: Can be performed, ×: Cannot be performed

Operation		Description	Database access instruction		CPU module database access function	
			Availability and instruction symbol		Availability and application	
Database	Connection	Connects to the specified database.	○	DBOPEN(P)	○	Microsoft Access®, Excel®, user-created application
	Disconnection	Disconnects from the specified database.	○	DBCLOSE(P)	○	
	Import	Imports the specified data from the specified Unicode text file to create a database.	○	DBIMPORT(P)	×	—
	Export	Exports the specified data in the specified database into a Unicode text file.	○	DBEXPORT(P)	×	—
Table	Add	Adds a table to the database.	×	—	○	Microsoft Access®
Record	Access	Accesses the specified table in the database.	×	—	○	Microsoft Access®
	Add	Adds a record to the specified table of the database.	○	DBINSERT(P)	○	Microsoft Access®
	Update	Updates the value of the specified record on the specified table in the database.	○	DBUPDATE(P)	○	Microsoft Access®
	Acquire (Search)	Acquires the value of the specified record on the specified table in the database.	○	DBSELECT(P)	○	Excel®
	Delete	Deletes the specified record on the specified table in the database.	○	DBDELETE(P)	○	Microsoft Access®
Transaction	Start	Starts a transaction for the specified database.	○	DBTRANS(P)	○	Microsoft Access® (The transaction range depends on the application.)
	Commit	Commits a transaction for the specified database.	○	DBCOMMIT(P)	○	Microsoft Access® (The transaction range depends on the application.)
Rollback		Performs rollback on the specified database.	○	DBROLBAK(P)	○	Microsoft Access® (The transaction range depends on the application.)
Database operation by a desired SQL			×	—	○	User-created application

Point

- For details on the database access instructions, refer to the following.

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

- For the available SQL commands for the CPU module database access function, refer to the following.

Page 768 List of Available SQL Commands for CPU Module Database Access Function

Restriction

- The user cannot execute multiple database access instructions simultaneously. If they are executed simultaneously, instructions other than the first one will be completed with an error.
- Even though the operation can be performed for the database of the CPU module by the SQL command, an error occurs on the application side when it cannot be performed in the application. In this case, check the details of the error and perform the operation within the range of the specifications of the application again.

12.2 Database Access Instruction

Usage Procedure

This section describes the procedure to use the database function.

Creating databases

To construct a database on an SD memory card, create a Unicode text file which defines the configuration of the database and its tables, store the file on the SD memory card, and execute the DBIMPORT(P) instruction, specifying the created Unicode text file.

1. Create a Unicode text file which defines the configuration of a database and its tables. (☞ Page 121 Database Access Instruction)
2. Write the created Unicode text file to the SD memory card. For details on how to Unicode text files (write of user data), refer to the GX Works3 Operating Manual.
3. By executing the DBIMPORT(P) instruction with the created Unicode text file specified, the database folder is created on the same layer as the Unicode text file and the database is constructed. (☞ Page 124 Database Access Instruction)

Point

The DBEXPORT(P) instruction enables the user to check the contents of a database by exporting it to a Unicode text file. Also, when the user wants to add more than one record or field to a database, the user can change the configuration of the database by exporting it to a Unicode text file, editing the file, and executing the DBIMPORT(P) instruction to get the database back to the system.

Operating databases

First, the user must execute the DBOPEN(P) instruction to connect to the database which the user wants to work on. Also, after finishing database operation, the user must execute the DBCLOSE(P) instruction to disconnect from the database.

1. Execute the DBOPEN(P) instruction to connect to the database which the user wants to work on (connect to the database by specifying the folder which was created in Step 3 of the database creation procedure). When the execution is finished, a value including the "database ID number" is returned.

Point

Database ID numbers are necessary to perform database operation. Therefore, please make a note of the database ID number of each database.

2. Once the connection to the database is established, perform operation (addition, update, search, or deletion) of the database. To perform database operation, the user must specify the "database ID number" of the target database in each instruction.
3. When database operation is finished, execute the DBCLOSE(P) instruction to disconnect from the database.

Point

The user can use a transaction when the user wants to perform multiple operations for a database as a set and update the database at once. (☞ Page 124 Transactions for Databases)

Creating Unicode Text Files

When the DBIMPORT(P) instruction is executed, a database is created on an SD memory card, according to setting details in the tab-delimited format of Unicode text file. The user must create Unicode text files on an SD memory card.

Setting details of Unicode text file

The following table lists items which must be specified in a Unicode text file to create a database.

Item	Description
Database name	Specify a database name, using single-byte alpha-numeric characters. • Within 32 characters • Not case-sensitive • The following characters cannot be used: " # % * + , / : ; < = > ? [\ ] | ' { } & ~ $ @ ^
Table definition start tag	Specify the tag, <table>, using a single-byte lower-case characters. When defining more than one table, enclose each table with <table> and </table>.
Table name	Specify a table name, using single-byte alpha-numeric characters. • Within 32 characters • Not case-sensitive • The following characters cannot be used: " # % * + , / : ; < = > ? [\ ] | ' { } & ~ $ @ ^ • The maximum number of tables is 32.*3
Field name	Specify the name of each field of each record, using single-byte alpha-numeric characters, and in the tab-delimited format. • The maximum number of characters which can be used for each field name is 32. • Case-sensitive • The following characters cannot be used: " and ' • The maximum number of fields is 128.*3
Data type	Specify the data type of each field in the tab-delimited format. The data types which can be specified are as follows. They must be specified using upper-case characters. • BOOL • WORD • DWORD • INT • DINT • REAL • LREAL • STRING: Specify the number of characters after a colon (:) (For example, when the number of characters is 16, the user can specify like that: "STRING: 16"). The range of the number of characters is 1 to 124.*1*2 • WSTRING: Specify the number of characters after the colon (:) The range of the number of characters is 1 to 124.*1
Key constraint	Specify the key constraint on each field, using single-byte alphabetical characters. Between key constraints, a tab must be inserted as a delimiter. One of the following can be specified: 0: None 1: Primary key (used to uniquely identify each record) When a primary key is specified, its key name also must be specified after a colon (:) (e.g. "1: pk1"). Key names must be up to 16 single-byte alphabetical characters and are case sensitive. 2: NOT NULL constraint (When this setting is selected, setting values cannot be left unspecified by skipping specification of primary keys or foreign keys. (This value is used to set a restriction.) 3: Foreign key (used to refer to the field of another table.) When a foreign key is specified, a table name, which is to be associated with the key name, also must be specified after a colon (:), like that: "3: Key Name: Table Name" (e.g. "3: fk1: tb2"). Key names must be up to 16 single-byte alphabetical characters and are case sensitive.
Index	When the user don't want to specify any indexes, the user can specify 0 (single-byte numeric character). When an index is specified, the name of the index and a field name must be specified using a colon (:) between them, like that: "Index Name: Field Name" (e.g. "id1: field1"). • Index names must be up to 16 single-byte alphabetical characters and are case sensitive. • The maximum number of indexes is 128.*3 • For fields with a primary or foreign key constraint, indexes are automatically set, and it is necessary to specify indexes by this setting (if specified, the DBIMPORT(P) instruction is completed with an error).
Setting value	Specify the setting value of each field in the tab-delimited format. • Characters or numbers which will be set to each field must match with the specified data type (☞ Page 121 Creating Unicode Text Files, ☞ Page 122 Format of setting values in Unicode text) • Individual records are set by row.
Table definition end tag	Specify the tag, </table>, using a single-byte lower-case characters.

*1 The termination character, NULL, is not counted in the number of characters.

*2 The length of characters is represented by single-byte characters. For double-byte character strings, one double-byte character should be counted as two single-byte characters.

*3 For the programmable controller CPU with firmware version earlier than "28", the maximum number is 16.

Point

Specifications of the character code for Unicode text files are as follows:

- Character encoding schema: UTF-16 (Little-Endian)
- BOM: Yes

■Format of setting values in Unicode text

Item	Description
WORD, DWORD, INT, DINT	The value must be specified in the decimal format (e.g. 0, 1, 111, -111).
BOOL	The value must be 0 or 1.
REAL, LREAL	The value must be specified in the following exponential format (e.g. 1.0e-01, 1.0E+01): Mantissa + E[*1] + Exponent part[*2]

*1 "E" can be specified in lower case.

*2 The exponent part must always have a sign.

Configuration of Unicode text files

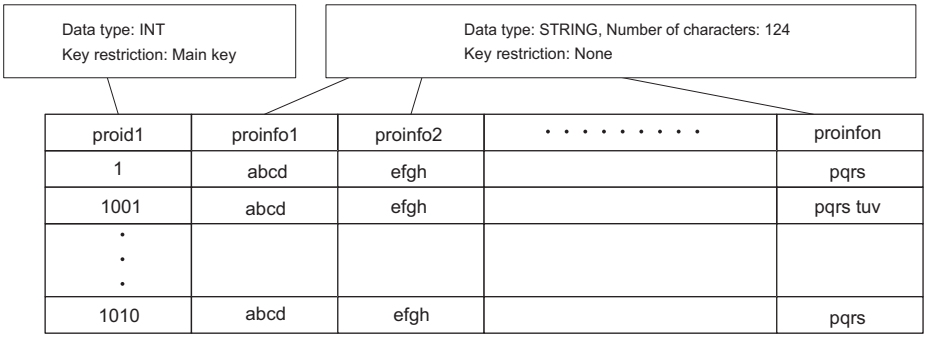
```
Database name
<table>
Table name 1
Field name 1 [Tab] Field name 2 [Tab] ... [Tab] Field name  n
Data type 1 [Tab] Data type 2 [Tab] ... [Tab] Data type n
Key restriction 1 [Tab] Key restriction 2 [Tab] ... [Tab] Key restriction n
Index 1 [Tab] Index 2 [Tab] ... [Tab] Index n
Set value 1 [Tab] Set value 2 [Tab] ... [Tab] Set value n
:
Set value k [Tab] Set value 2 [Tab] ... [Tab] Set value n
</table>
<table>
Table name 2
(The subsequent setting is the same as the setting of Table name 1.)
</table>
```

[Tab] in the figure represents a tab character and it is entered using the Tab key on the keyboard in practice. Line breaks, comments, tabs, and spaces are processed as follows.

Item	Description
Line break	• The line break code which can be recognized is CR+LF. • Lines consisting of only line breaks are ignored.
Comment	A row beginning with "//" is regarded as a comment. The maximum number of characters in a comment row is 2048 (including two characters for the line break code).
Tab	Tabs cannot be used for database names, table names, field names, key constrains, and setting values. Tabs are recognized as delimiters.
Space	• Spaces between tabs and printable characters are not recognized as a part of settings. • Spaces between printable characters are recognized as spaces.

Ex.

Example of the Unicode text file format (Database name: database1, Table name: product-info1)



proid1	proinfo1	proinfo2	· · · · · · · · ·	proinfon
1	abcd	efgh		pqrs
1001	abcd	efgh		pqrs tuv
· · ·				
1010	abcd	efgh		pqrs

```
//Database name
database1
<table>
//Table name
product-info1
//Field name
proid1 [Tab] proinfo1 [Tab] ... [Tab] proinfon
//Data type
INT [Tab] STRING: 124 [Tab] ... [Tab] STRING: 124
//Key restriction
1: pk1 [Tab] 0 [Tab] ... [Tab] 0
//Index
0
//Record
//First record
1 [Tab] abcd [Tab] efgh [Tab] ... [Tab] pqrs
//Second record
1001 [Tab] abcd [Tab] efgh [Tab] ... [Tab] pqrs tuv
:
:
//nth record
1010 [Tab] abcd [Tab] efgh [Tab] ... [Tab] pqrs
</table>
```

Point

Unicode text files can be created using a text editor or spreadsheet software program. The method for saving Unicode text files in Notepad or Excel is as follows:

- In the text editor (Notepad, an accessory of Windows® 7), select [Save As] from the [File] menu, and select "Unicode" as the character code to save the file.
- In the spreadsheet software program (Excel 2010), select [Save As] from the [File] menu, and select "Unicode" as the file type to save the file. When a Unicode text file is created using a spreadsheet software program, tabs are automatically inserted by saving the file. Therefore, it is not necessary to insert tabs into the file in the spreadsheet software program.

Transactions for Databases

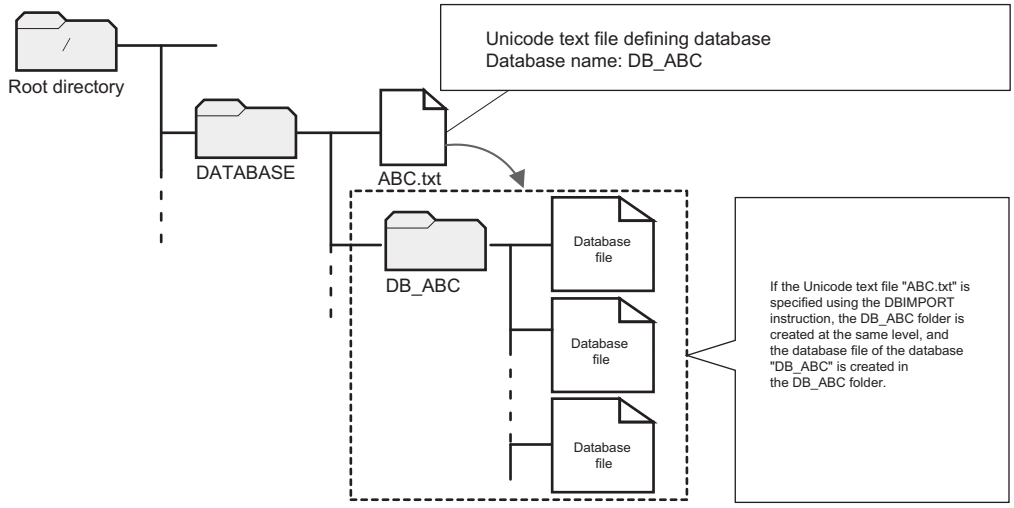
The user can use a transaction (the DBTRANS(P) instruction and the DBCOMMIT(P) instruction) when the user wants to perform multiple operations for a database as a set and update the database at once. The DBTRANS(P) instruction starts a transaction, and the DBCOMMIT(P) instruction groups the results of the following instructions together to update the target database at once.

- The target of a transaction must be a single database. If a transaction is executed for more than one database, transaction instructions for the second and later databases are completed with errors.
- When the CPU module is powered off or is reset in the period from the start of a transaction (execution of the DBTRANS(P) instruction) to the confirmation of the transaction (execution of the DBCOMMIT(P) instruction), the database is automatically rolled back to the status before the start of the transaction. However, if a database access instruction is completed with an error, the database is not roll backed to the previous status.
- To cancel the running transaction process and reset the database to the status before the start of the transaction, the user can perform rollback (DBROLBAK(P) instruction). Rollback is an operation which resets the target database to the previous status by executing the DBROLBAK(P) instruction in the period from the start of the transaction to the commitment of the transaction. When the DBCLOSE(P) instruction is executed without the DBCOMMIT(P) instruction or the DBROLBAK(P) instruction, the transaction is confirmed during execution of the DBCLOSE(P) instruction.

1. The DBOPEN(P) instruction is used to connect to a database.
2. The DBTRANS(P) instruction is used to start transaction.
3. Operations for the database are performed.
4. To confirm the transaction, the DBCOMMIT(P) instruction can be used. When it is not necessary to confirm the transaction, the DBROLBAK(P) instruction can be used to restore to the state before the start of the transaction (rollback).
5. The DBCLOSE(P) instruction is used to disconnect from the database.

Folder Configuration of Databases

The database folder for storing database files is generated at the same level in the folder hierarchy as the Unicode text file which is specified in the DBIMPORT(P) instruction. Also, the database folder is given the same names as the database. The length of a database folder path name must be up to 128 characters, including the drive name and a colon (:) prefixed to the path name.



Operation of files/folders in the database folder

Files and folders in the database folder are used for database access instructions. They should not be changed, deleted, and newly created. Otherwise, database access instructions may be completed with errors. In that case, delete the relevant database folder.

Timing of Database Update

Once execution of a database access instruction is completed, the target database is updated. However, during transaction, the database is not updated each time execution of an instruction is completed. Instead, all changes during the transaction are applied to the database at once when the DBCOMMIT(P) instruction is executed.

Precautions

This section describes precautions on using the database function.

Free space of SD memory card required for execution of database access instruction

To execute a database access instruction, the SD memory card must have at least 10MB of free space. When a database access instruction is performed and if the SD memory card doesn't have enough free space, the instruction is completed with an error.

When the SD memory card is write-protected

Before using database access instructions, check that the write protection of the SD memory card used is disabled. If a database access instruction is performed while the SD memory card is write protected, the instruction is completed with an error.

When the CPU module database access function is used

The database access instructions can be executed while SM1498 (CPU module database start-up flag) is on.

When the database is being operated by other function

Before creating a database by using the DBIMPORT(P) instruction, check that the same database is not being operated by other function.

If the DBIMPORT(P) instruction is executed specifying a name of a database in the SD memory card while the same database is operated by another function, the DBIMPORT(P) instruction may be completed with an error and the database operation in another function may fail. After the DBIMPORT(P) instruction is completed with an error, if the database operated by other function cannot be operated, the database may be broken. Power off and on or reset the CPU module, and build the database again by using the DBIMPORT(P) instruction.

Combination of the firmware version and database

For the combination of the firmware version of the programmable controller CPU that creates a database and that of the programmable controller CPU that can operate the database, refer to the following.

☞ Page 143 Combination of the firmware version and database

Removal of the SD memory card

Do not remove the SD memory card while connecting to a database by using the DBOPEN(P) instruction or executing a database access instruction. Otherwise, the database access instruction is completed with an error.

12.3 CPU Module Database Access (From External Device) Function

The CPU module database access function operates a database, that is built in an SD memory card inserted into the CPU module, from an application on a personal computer through the Ethernet port of the CPU module.
To use the CPU module database access function, install CPU Module Database Access Driver into a personal computer. (CPU Module Database Access Driver Installation Instructions)

Point

In the CPU module database access function, the CPU module operates as the ODBC server and the personal computer operates as the ODBC client.
ODBC (Open Database Connectivity) is the standard API that allows applications to connect with the database management system (DBMS) to acquire and write the data.

Restriction

Before using the CPU module database access function, check the version of the CPU module and the engineering tool used. (Page 779 Added and Enhanced Functions)

Usage procedure

This section describes the procedure for operating the database by using the CPU module database access function.

Procedure	Description
Construction of usage environment	Construct the environment to use the CPU module database access function.
Database access setting	Configure this setting to create or change a database of the CPU module.
Database operation	Operate a database of the CPU module from a personal computer.

Construction of usage environment

■Engineering tool

1. Module parameter setting of the CPU module

Set whether or not to use the CPU module database access function, login name, and password in the module parameter of the CPU module. (Page 128 Built-in database access setting)

2. Update of the built-in database access setting in the CPU module

Write the module parameter to the CPU module, and then power off and on or reset the CPU module.

■CPU Module Database Access Driver

1. Obtaining CPU Module Database Access Driver

For CPU Module Database Access Driver, please consult your local Mitsubishi representative.

2. Installing CPU Module Database Access Driver

Install CPU Module Database Access Driver into the personal computer. (CPU Module Database Access Driver Installation Instructions)

Point

The installer of CPU Module Database Access Driver has the 32-bit or the 64-bit version. Choose the installer depending on the supported bit of the application (Microsoft Access, Excel, user-created application) used for the CPU module database access function.
For example, when using Microsoft Access with 32-bit version, use the 32-bit installer of CPU Module Database Access Driver even though the 64-bit OS is used.

Database access setting

1. Creating a database

Create a database in an SD memory card of the CPU module. (☞ Page 129 Creating a database)

2. Adding a database to the ODBC data source

Add the created database of the CPU module to the ODBC data source of the personal computer. (☞ Page 130 Adding a database to the ODBC data source)

Database operation

1. Checking the start-up of the ODBC server

Check that SM1498 (CPU module database start-up flag) is on.

When SM1498 is on, the ODBC server of the CPU module is active. The ODBC server starts and SM1498 turns on when:

- An SD memory card storing the database folder is inserted to the CPU module, and the CPU module is powered off and on or is reset.
- An SD memory card storing the database folder is inserted to the CPU module powered-on when SD1498 (Start-up status of CPU module database) is 0H.

Whether or not the ODBC server has been started up can be checked in SD1498 (Start-up status of CPU module database).

Point

The active ODBC server stops when:

- The SD memory card is disabled (SD memory card forced disable) or the SD memory card is removed.
- The CPU module is powered off.
- The data memory is initialized.
- Free space of the SD memory card is less than 10MB.

2. Connecting to a database

Start an application (Microsoft Access, Excel, or the user-created application) on a personal computer and connect it to a database specified with the application. (☞ Page 132 Application example)

3. Operating a database

Operate an application on a personal computer to operate a database. (☞ Page 132 Application example)

Restriction

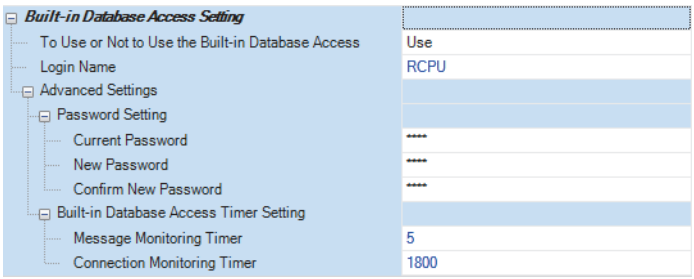
The SD memory card is accessed when the ODBC server is started up. This may lengthen the processing time of other functions that use the SD memory card.

Built-in database access setting

To use the CPU module database access function, set "To Use or Not to Use the Built-in Database Access" of the module parameter to "Use" with the engineering tool.

[Navigation window] ⇨ [Parameter] ⇨ CPU module ⇨ [Module Parameter] ⇨ [Application Settings] ⇨ [Built-in Database Access Setting]

Window



Displayed items

Item			Description	Setting range	Default
To Use or Not to Use the Built-in Database Access			Set whether to use the CPU module database access function.	• Use • Not Use	Not Use
Login Name			A user name to be used for the authentication at the access to a database	1 to 31 characters*1*2	RCPU
Advanced Settings	Password Setting	Current Password	A password to be used for the authentication at the access to a database	0 to 31 characters*1	RCPU
		New Password	A new password to be set	0 to 31 characters*1	—
		Confirm New Password	A password for confirmation. The password can be changed only when this password matches a new password.	0 to 31 characters*1	—
	Built-in Database Access Timer Setting	Message Monitoring Timer	A monitoring timer for until the first message is sent from the personal computer after the connection establishment.	1 to 16383s	5
		Connection Monitoring Timer	A monitoring timer for until the CPU module disconnects the personal computer after the CPU module recognizes that the personal computer stops accessing to the CPU module.	1 to 16383s	1800

*1 Single-byte alphabetical characters and special characters (excluding " # % * + , / : ; < = > ? [\ ] | ' { } & ~ $ @ ^) can be input.
*2 A space cannot be used as the first character.

Creating a database

This section describes the procedure for creating a database to be used with the CPU module database access function. A database can be created with the following methods.

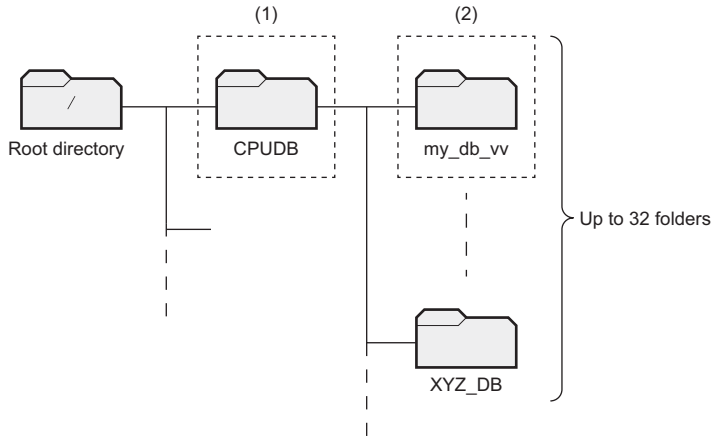
- Storing the database folder
- DBIMPORT(P) instruction (a database access instruction)

Storing the database folder

Store the database folder in a SD memory card.

1. For the database folder, please consult your local Mitsubishi representative.
2. Create the CPUDB folder directly under the root directory of the SD memory card from the personal computer. (1)
3. Store the database folder (default database name: my_db_vv[*1]) in the CPUDB folder. (2)

*1 "vv" indicates the version of the database folder.



- The name of the database folder stored is recognized as the database name. The maximum number of characters in the folder path to the database folder is 128. The characters that can be used for a database name is the same as the Unicode text file specified by using the DBIMPORT(P) instruction. (☞ Page 121 Setting details of Unicode text file)
- Only the CPUDB folder directly under the root directory can be used as the storage location of the database. In the CPUDB folder, store database folders only.
- Up to 32 folders can be stored in the CPUDB folder.
- If data other than the database folder or 33 database folders or more is stored in the CPUDB folder, starting up of the ODBC server is failed and an error code is stored in SD1498 (Start-up status of CPU module database).

■Precautions

Do not change, delete, or create a file and folder in the database folder from the personal computer. Otherwise, the CPU module database access function and database access instructions are completed with an error. If the above mentioned operations are performed from the personal computer, delete the database folder in the CPU module and create the new one.

DBIMPORT(P) instruction (a database access instruction)

Use the DBIMPORT(P) instruction and create a database in an SD memory card according to the setting of the Unicode text file.

- For details on the Unicode text file, refer to the following.

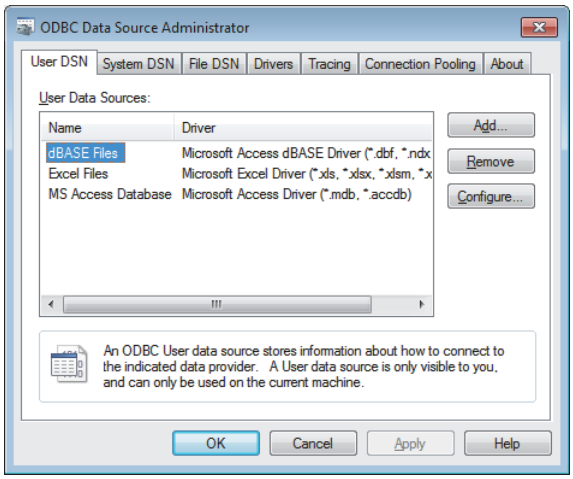
☞ Page 121 Setting details of Unicode text file

- For details on the DBIMPORT(P) instruction, refer to the following.

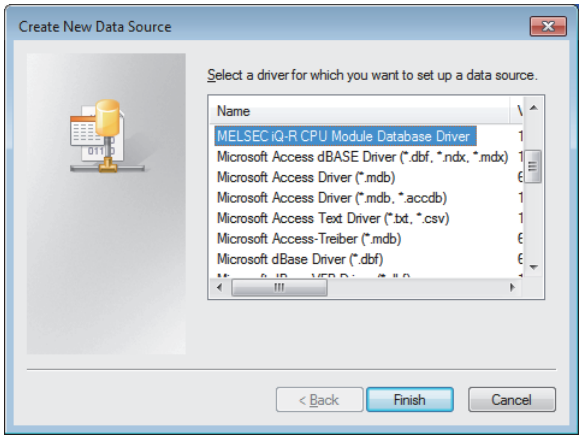
📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

Adding a database to the ODBC data source

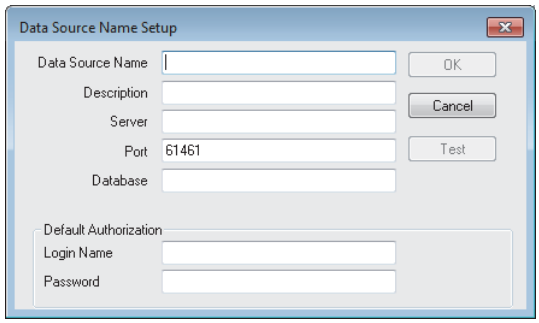
This section describes the procedure for adding a created database of the CPU module to the ODBC data source in the personal computer. (Examples of Windows 7 are shown below. Names of windows and menus may differ depending on the version of the OS.)



1. Select [Start] ⇨ [Control Panel] ⇨ [System and Security] ⇨ [Administrative Tools] ⇨ [Data Sources (ODBC)] on the personal computer and start the ODBC Data Source Administrator.
2. Click [User DSN] ⇨ the [Add] button to display the "Create New Data Source" window.



3. Select "MELSEC iQ-R CPU Module Database Driver", and click the [Finish] button.



4. Set the database of the CPU module on the "Data Source Name Setup" window. (☞ Page 131 Data Source Name Setup)

Point

When the 64-bit OS is used, the ODBC Data Source Administrator of a 32-bit or 64-bit version can be used. Choose the appropriate one depending on the supported bit of the application used with the CPU module database access function.

For the 64-bit OS, the ODBC Data Source Administrator can be started from the following. (When the start-up drive is C)

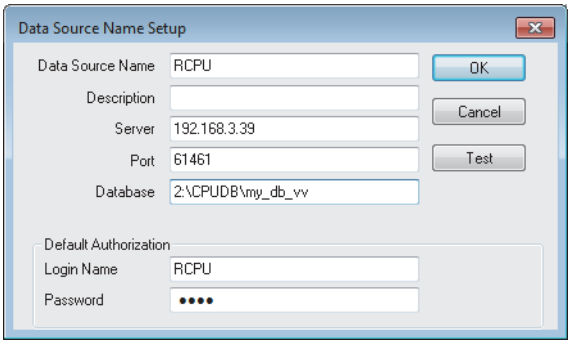
64-bit version: C:\WINDOWS\System32\odbcad32.exe

32-bit version: C:\WINDOWS\SysWOW64\odbcad32.exe

Data Source Name Setup

Set the database of the CPU module connected from the personal computer.

Window



Displayed items

Item		Description
Data Source Name		Set the identification name (any character string) for specifying the connection target database from the application. Input 1 to 32 characters in single-byte or double-byte alphanumeric characters. Characters except for * ! ( ) = \ ? , [ ] can be used.
Description		Input the comment. Input none, up to 1023 single-byte alphanumeric characters, or up to 512 double-byte characters.
Server		Set the IP address of the CPU module to be connected.
Port		Set 61461. (Default setting is 61461. Do not change the value.)
Database		Set the absolute path of the database to be connected. For example, input the following for the database folder, my_db_vv, stored in the CPUDB folder directly under the root directory of the SD memory card. 2:\CPUDB\my_db_vv ("2:\" indicates the root directory of the SD memory card. "vv" indicates the version of the database folder.)
Default Authorization	Login Name*1	Set the login user to access the database which is set. Input the user name which is set in the module parameter. (☞ Page 128 Built-in database access setting)
	Password*1	Input the password which is set in the module parameter. (☞ Page 128 Built-in database access setting)

*1 If they are not set, input them at login to connect to the database.

Point

Click the [Test] button to perform a connection test of the database of the CPU module in accordance with the setting.

Application example

This section describes the examples of database operations of the CPU module from applications in the personal computer using the CPU module database access function. (Names of windows and menus may differ depending on the version of the OS and application.)

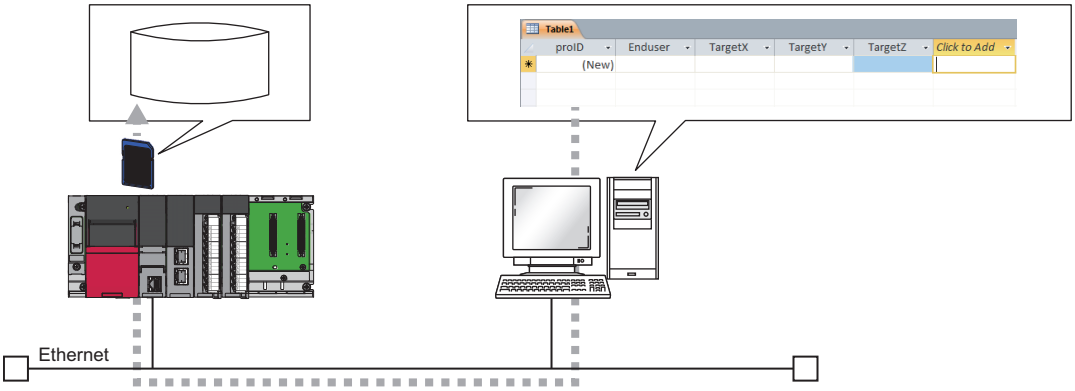
Item	Description	Application	Reference
Table creation	Create a table in a database of the CPU module from the personal computer.	Microsoft Access	Page 132 Creating a table by using an application
Record operation	Edit a database of the CPU module from the personal computer.	Microsoft Access	Page 135 Record operation from an application
Record search	Search a database of the CPU module from the personal computer.	Excel	Page 137 Record search from an application
Database operation with the user-created application	Create a database access application using the ODBC classes of Microsoft .NET Framework.	User-created application	Page 139 User-created application

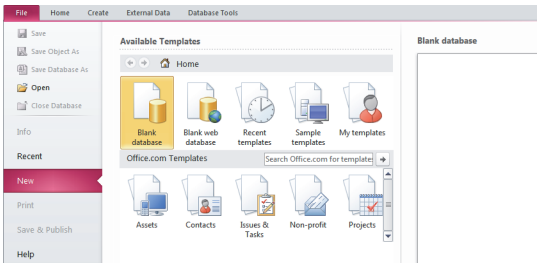
The following table lists the applications whose connection to the database has been checked.

Application	Version
Microsoft Access	Microsoft Office 2010, 2013, 2016
Excel	
User-created application	Application created by using Microsoft .NET Framework 4.5

Creating a table by using an application

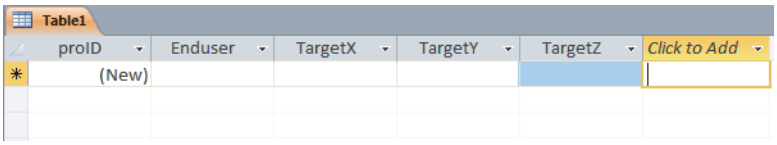
The following provides an example to create a recipe table in a database of the CPU module by using Microsoft Access.





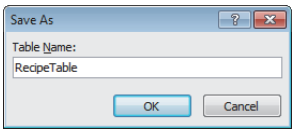
1. Start Microsoft Access and select "Blank database".

2. Add the following fields.

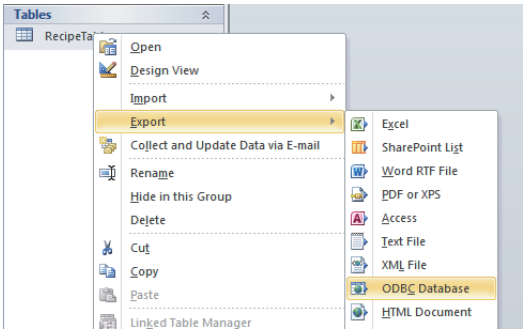


Field name	Data type	Data type for when the data is exported to the database of the CPU module
proID*1	AutoNumber type: Long integer	INT
Enduser	Numeric type: Long integer	INT
TargetX	Numeric type: Single-precision floating point type	REAL
TargetY	Numeric type: Single-precision floating point type	REAL
TargetZ	Numeric type: Single-precision floating point type	REAL

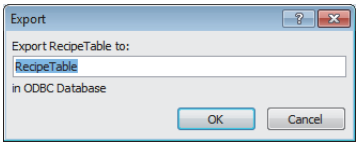
*1 Rename the ID field.



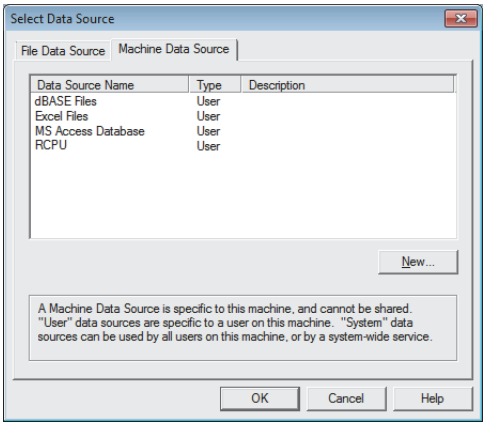
3. Name the table "RecipeTable" and save it. After saving, close the table.



4. Right-click on RecipeTable, and select [Export] ⇨ [ODBC Database].

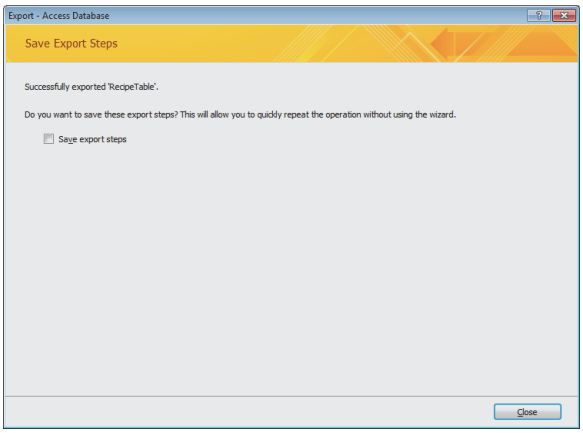


5. When the window to specify the export destination is displayed, click the [OK] button.



6. On the window for selecting the data source, click the [Machine Data Source] tab and select the data source of the CPU module.

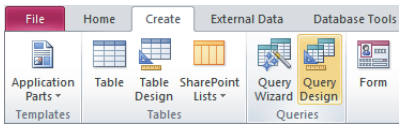
- When the default user is not set to the ODBC data source, input the user name and password.
- When no data source of the CPU module is displayed in the window, add a database of the CPU module to the ODBC data source. (☞ Page 130 Adding a database to the ODBC data source)



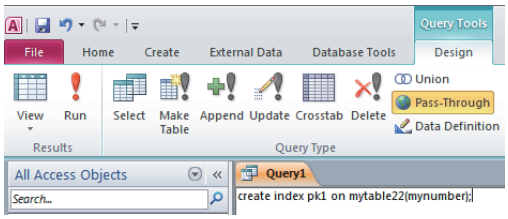
7. The result of export is displayed, and the table is added to the database of the CPU module.

Point

To set the details that cannot be set with Microsoft Access, send desired SQL commands to the database of the CPU module.



Clicking the [Create] tab ⇨ [Query Design] of Microsoft Access displays the "Show Table" window, so click the [Close] button. The [Query Tools] tab becomes enabled. Select [Pass-Through].



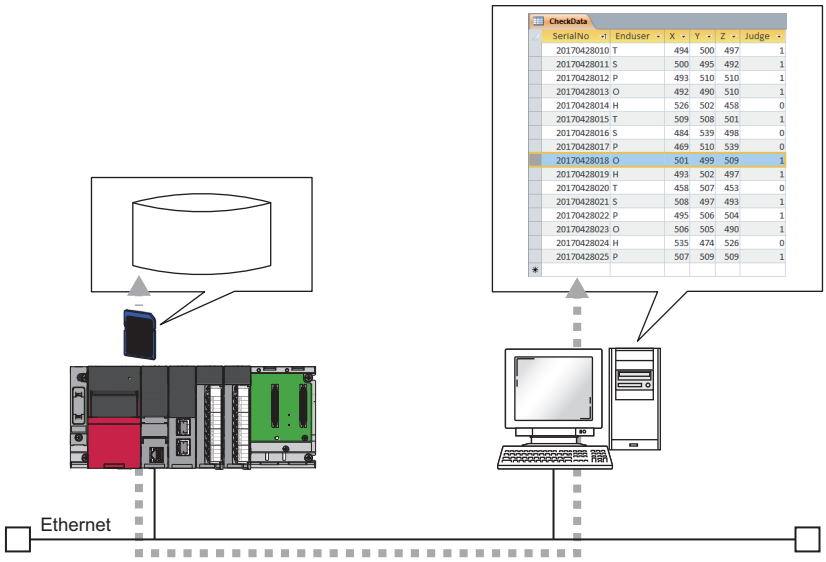
Selecting [Pass-Through] enables the query tab and allows input of SQL commands to be sent to the database of the CPU module directly.

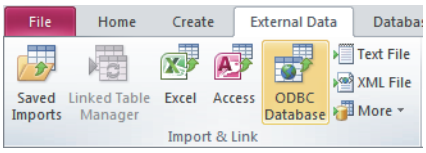
For available SQL commands for the CPU module database access function, refer to the following.

☞ Page 768 List of Available SQL Commands for CPU Module Database Access Function

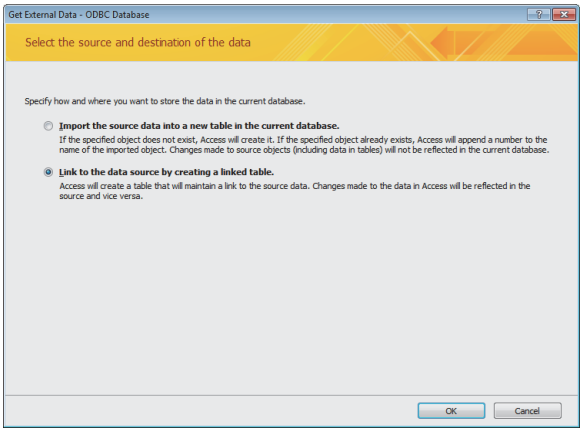
Record operation from an application

The following provides an example of record operation such as synchronizing, writing, and deleting of the data by connecting to the database of the CPU module by using Microsoft Access.

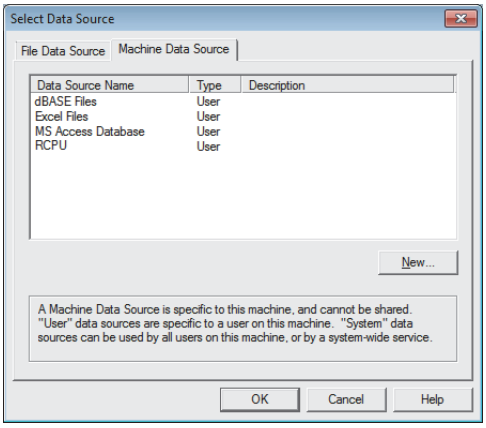




1. Start Microsoft Access and select the [External Data] tab ⇨ [ODBC Database].

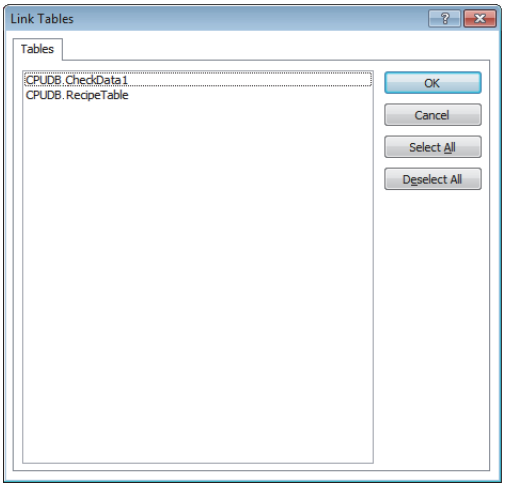


2. When the window to select the method for storing the data in the database, select "Link to the data source by creating a linked table", and click the [OK] button.

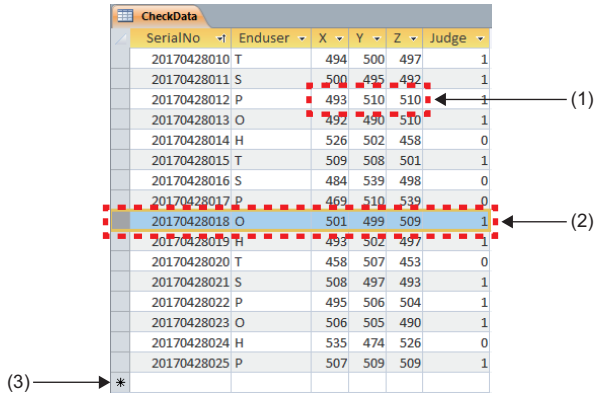


3. On the window for selecting the data source, click the [Machine Data Source] tab and select the data source of the CPU module.

- When the default user is not set to the ODBC data source, input the user name and password.
- When no data source of the CPU module is displayed in the window, add a database of the CPU module to the ODBC data source. (☞ Page 130 Adding a database to the ODBC data source)



4. On the window for selecting table, select a table to be operated and click the [OK] button to display the contents of the table.



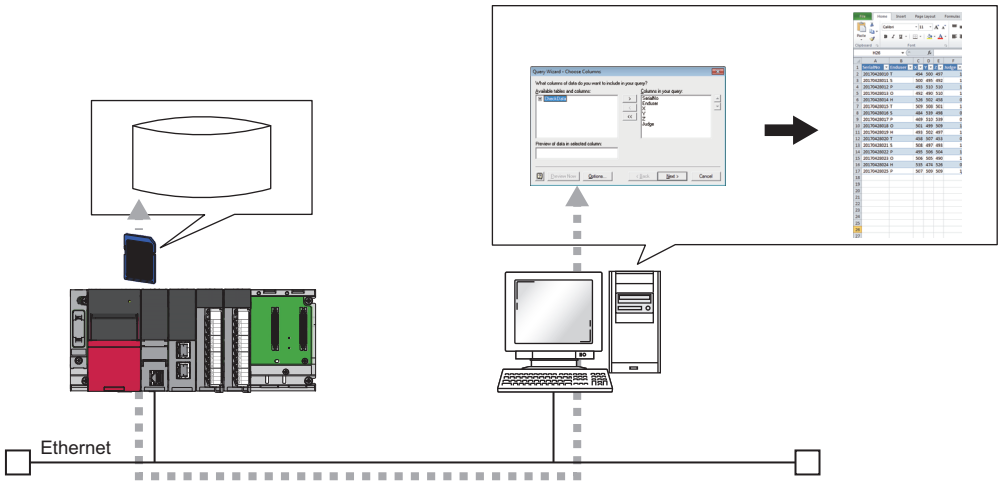
SerialNo	Enduser	X	Y	Z	Judge
20170428010	T	494	500	497	1
20170428011	S	500	495	492	1
20170428012	P	493	510	510	1
20170428013	O	492	490	510	1
20170428014	H	526	502	458	0
20170428015	T	509	508	501	1
20170428016	S	484	539	498	0
20170428017	P	469	510	539	0
20170428018	O	501	499	509	1
20170428019	H	493	502	497	1
20170428020	T	458	507	453	0
20170428021	S	508	497	493	1
20170428022	P	495	506	504	1
20170428023	O	506	505	490	1
20170428024	H	535	474	526	0
20170428025	P	507	509	509	1

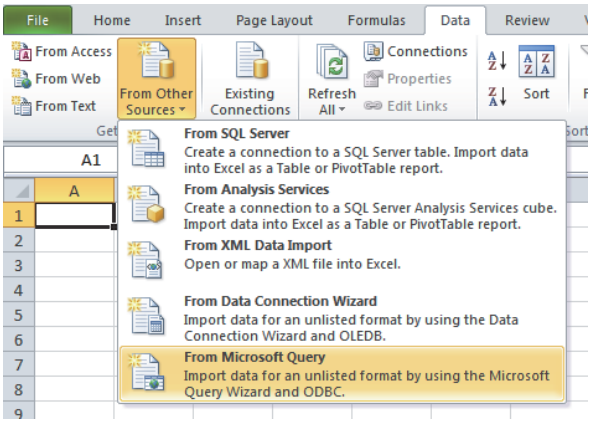
(1) Select a cell to change the value.
(2) Select a row to delete the record.
(3) Add a new record.

5. When the value is changed or the record is deleted, the database of the CPU module is changed according to the operation.

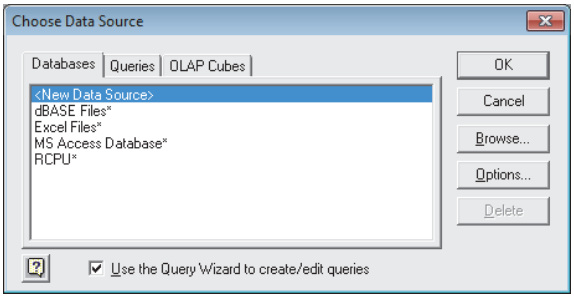
Record search from an application

The following provides an example to search the record that matches the specified condition from the production data stored in the database by connecting to the database of the CPU module by using Excel.



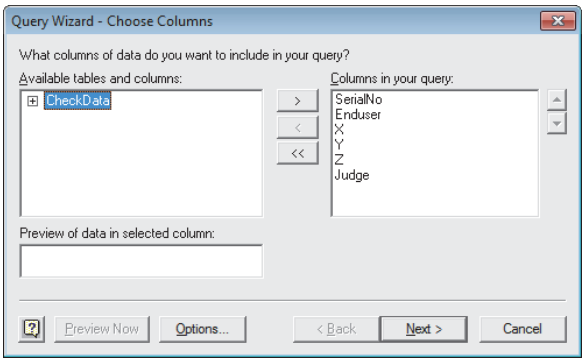


1. Start Excel, and select the [Data] tab ⇨ [From Other Sources] ⇨ [From Microsoft Query].

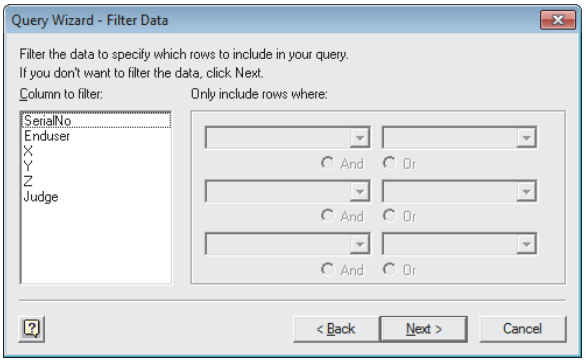


2. On the window for selecting the data source, select the data source of the CPU module.

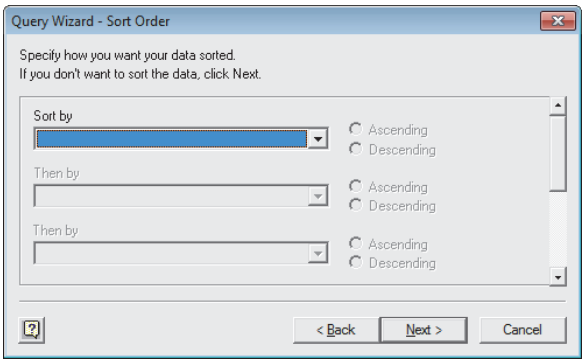
- When the default user is not set to the ODBC data source, input the user name and password.
- When no data source of the CPU module is displayed in the window, add a database of the CPU module to the ODBC data source. (☞ Page 130 Adding a database to the ODBC data source)



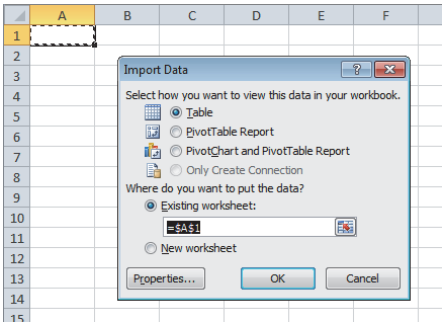
3. On the window for selecting columns of the query wizard, select the field to be output.



4. On the window for filtering data of the query wizard, set the search condition.



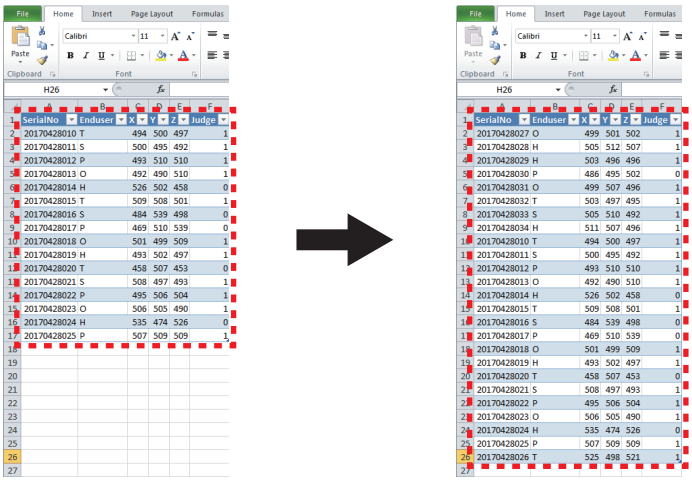
5. On the window for setting the sort order of the query wizard, set the condition to sort the output data.



6. Set the output destination. Then, the search result is output.

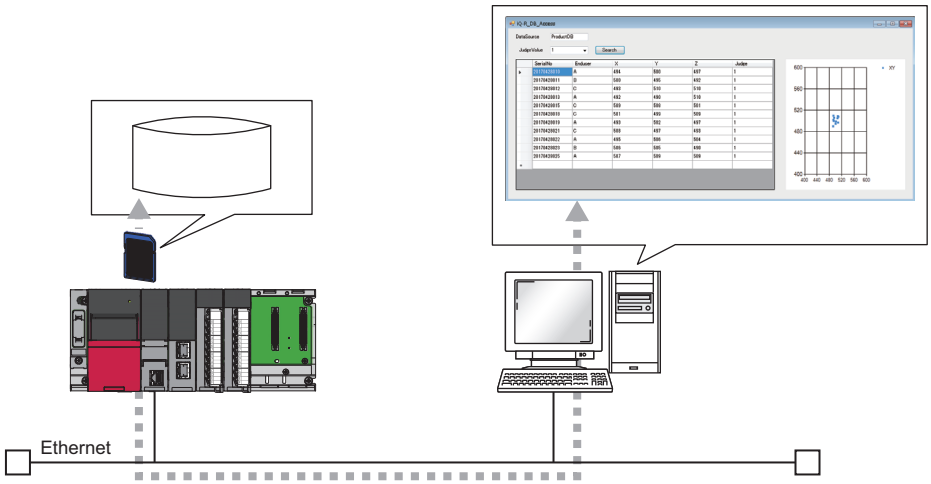
Point

When the file used in the above procedure is saved, the search condition is also saved.
Click the [Data] tab ⇨ [Refresh All] in Excel to acquire the data in the same condition from the database of the CPU module once more.



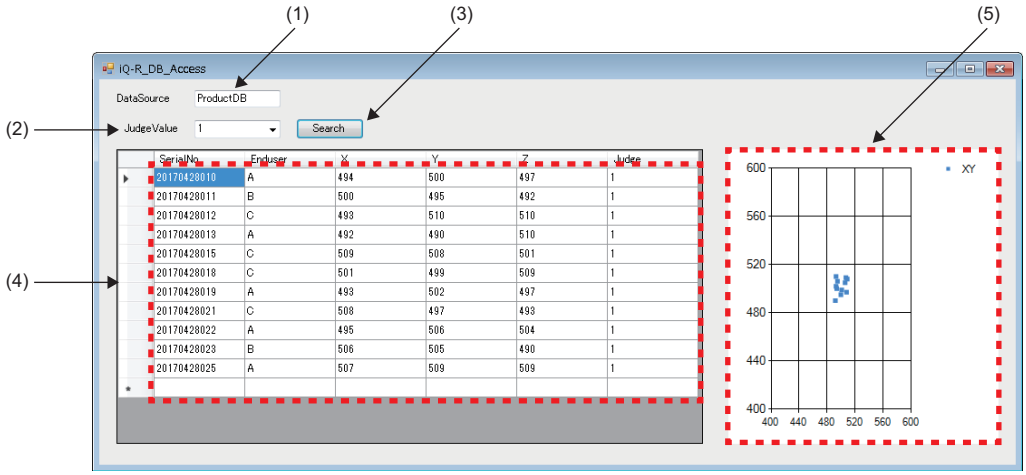
User-created application

The following describes a sample program for accessing the database of the CPU module with the SQL command using the ODBC class of Microsoft .NET Framework.



■Overview of the sample program

This sample program performs the condition search in the database of the CPU module using the value of the Judge field as its key, and displays the acquired search result in a list and graph (scatter plot).



(1)Specify the database source name to be connected.
(2)Specify the Judge field value to be searched.
(3)Click the [Search] button to search the record with the same value as the one specified by "JudgeValue" in the database specified by "DataSource".
(4)The acquired record is displayed in the list.
(5)The X value and Y value of the acquired record are displayed in a graph (scatter plot).

■Database configuration of the sample program

The following describes the database configuration to be searched by the sample program.

Item	Description
Server name (IP address of CPU module)	192.168.3.39
Database folder path	2:\Database\SampleDB
Database name	SampleDB
Table name to be searched	CheckData

The following table lists the records of CheckData table.

SerialNo	Enduser	X	Y	Z	Judge
20170428010	A	494	500	497	1
20170428011	B	500	495	492	1
20170428012	C	493	510	510	1
20170428013	A	492	490	510	1
20170428014	B	526	502	458	0
20170428015	C	509	508	501	1
20170428016	A	484	539	498	0
20170428017	B	469	510	539	0
20170428018	C	501	499	509	1
20170428019	A	493	502	497	1
20170428020	B	458	507	453	0
20170428021	C	508	497	493	1
20170428022	A	495	506	504	1
20170428023	B	506	505	490	1
20170428024	C	535	474	526	0
20170428025	A	507	509	509	1

■Source code of the sample program

The following describes the source code of the sample program.

- Development environment: Visual Studio 2015
- Programming language: C#

```
namespace iQ_R_DB_Access
{
    public partial class FrmMain : Form
    {
        public FrmMain()
        {
            InitializeComponent();

            // Range setting of X-/Y-axis on a graph
            chart1.ChartAreas[0].AxisX.Minimum = 400;
            chart1.ChartAreas[0].AxisX.Maximum = 600;
            chart1.ChartAreas[0].AxisY.Minimum = 400;
            chart1.ChartAreas[0].AxisY.Maximum = 600;

            // Initial value (all) set to the judge value of search target
            cmbJudge.SelectedIndex = 2;
        }

        /// <summary>
        /// Operation when the Search button is clicked
        /// Records that match the judge value specified by the combo box are acquired and displayed in a list and a scatter plot of X and Y field values.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnSearch_Click(object sender, EventArgs e)
        {
            // The number of fields (the number of fields on the CheckData table)
            int len = 6;

            // Acquiring the data source name
            string strConnect = txtDataSourceName.Text.ToString();
            // ODBC connection information management class
            OdbcConnectionStringBuilder o = new OdbcConnectionStringBuilder();
            o.Dsn = strConnect;
            // Creating the connection object
            OdbcConnection cn = new OdbcConnection(o.ConnectionString);

            // SQL statement
            string strQuery = GetSQL();
            // Creating the command object
            OdbcCommand cmd = new OdbcCommand(strQuery, cn);


            // Initializing the list
            dtRecord.Rows.Clear();
            // Initializing the graph
            chart1.Series[0].Points.Clear();

            try
            {
                // Connecting to DB
                cn.Open();

                // Creating the data reader object
                OdbcDataReader reader = cmd.ExecuteReader();
```

```
            // Processing the search results one record at a time
            for (int recordnum = 0; reader.Read(); recordnum++)
            {
                // Adding a blank row to the list
                dtRecord.Rows.Add();

                Setting the number of rows to be inserted into the list
                recordnum = dtRecord.Rows.Count-2;
                Storing acquired records into the list one field at a time
                for (int i = 0; i < len; i++)
                {
                    // Adding search results into the list
                    dtRecord.Rows[recordnum].Cells[i].Value = reader.GetValue(i);
                }

                // Plotting values of X and Y fields on the graph
                chart1.Series[0].Points.AddXY(Convert.ToInt32(dtRecord.Rows[recordnum].Cells[2].Value),
                                    Convert.ToInt32(dtRecord.Rows[recordnum].Cells[3].Value));
            }
        }
        catch (OdbcException ex)
        {
            MessageBox.Show(ex.ToString());
        }
        finally
        {
            // Disconnecting from DB
            cn.Close();
        }
    }

    /// <summary>
    /// Generating the SQL statement from the value in the combo box
    /// </summary>
    public string GetSQL()
    {
        string strSQL = "";

        switch (cmbJudge.SelectedIndex)
        {
            case 0:
                // The search target is records of Judge = 0.
                strSQL = "SELECT SerialNo, Enduser, X, Y, Z, Judge FROM CheckData WHERE Judge=0";
                break;
            case 1:
                // The search target is records of Judge = 1.
                strSQL = "SELECT SerialNo, Enduser, X, Y, Z, Judge FROM CheckData WHERE Judge=1";
                break;
            case 2:
                // Acquiring all the records
                strSQL = "SELECT SerialNo, Enduser, X, Y, Z, Judge FROM CheckData";
                break;

            default:
                strSQL = "SELECT * FROM CheckData";
                break;
        }
        return strSQL;
    }
  }
}
```

Precautions

This section describes the precautions for using the CPU module database access function.

Database creation

When creating a database in the CPUDB folder, use characters that can be specified only. If the characters that cannot be specified are used to create the database, the table cannot be accessed by the database access instructions.
For the characters that can be specified, refer to the following.
☞ Page 121 Setting details of Unicode text file

Combination of the firmware version and database

The following table lists the combination of the firmware version of the programmable controller CPU that creates the database and that of the programmable controller CPU where the operation of the database is available.
○: Available, ×: Not available

Database	Availability	
	The programmable controller CPU with firmware version earlier than "28"	The programmable controller CPU with firmware version "28" or later
Database created with the programmable controller CPU with firmware version earlier than "28"	○	○*1
Database created with the programmable controller CPU with firmware version "28" or later	×	○

*1 If the database that has been operated with the programmable controller CPU with firmware version "28" or later is operated with the programmable controller CPU with firmware version earlier than "28", the operation is completed with an error.

Number of connections that can be operated simultaneously

Up to four connections can be operated with the CPU module database access function simultaneously.
However, if an application such as Microsoft Access uses several connections, four applications may not connect with the database of the CPU module simultaneously. If an attempt was made to connect with the database with all the connections used, an error occurs in the application and the database cannot be connected. In this case, end unused applications connected with the database and connect again.

Several simultaneous accesses

When several applications access the same database in the CPU module, the database processing becomes slow.

■Access to the same table

When the database access instructions or the CPU module database access function access the same table, the function executed first locks the table, and an error may occur in the function executed later. For the database access instruction, an error code is stored in the completion status.

■Definition change of the same database

When the database definition is changed with "CREATE TABLE" or "DROP TABLE" in a single database from several applications using the CPU module database access function, the application executed first locks the database and an error may occur in the function executed later.

Completion with an error during database access

Do not power off or reset the CPU module during the access to a database of the CPU module. Otherwise, the change is not reflected on the database that is being executed.

Files created with this function

Do not create, change, and delete the ODBC server setting file (netserver.cfg), database path file (dbmaintainpath.txt), and error database check file (ErrorDB.txt) created with the CPU module database access function. Otherwise, the CPU module database access function or database access instruction is completed with an error.
If the above mentioned operations are performed, delete the database folder in the CPU module and create the new one.

Removal of the SD memory card

To remove the SD memory card while the ODBC server is active, execute the SD memory card forced disable and check that CARD READY LED is off before removing the SD memory card. (Failure to do so may result in a failure of databases.)
Since databases stop after the execution of the SD memory card forced disable, databases of the CPU module cannot be accessed. (The error code is stored in SD1498.)
To access the databases of the CPU module again, insert the SD memory card and power off and on or reset the CPU module.

Free space of the SD memory card

To use the CPU module database access function, the SD memory card must have at least 20MB of free space.
When the free space of the SD memory card becomes 20MB or less while the CPU module database access function is used, SM1497 (Memory card free space flag for CPU module database) turns on. When the free space of the SD memory card becomes 10MB or less, the ODBC server of the CPU module stops and the error code is stored in SD1498 (Start-up status of CPU module database).
When the ODBC server stops because the SD memory card does not have enough free space, secure the free space of the SD memory card using the online operation and power off and on or reset the CPU module. Powering off and on or resetting the CPU module starts the ODBC server.

When the SD memory card is write-protected

To use the CPU module database access function, the user must ensure that write protection of the SD memory card is off.
When the SD memory card is write-protected, the CPU module built-in database access function cannot update the database and the error code is stored in SD1498.

When the load of the CPU module is high

When the CPU module database access function is used with high circuit load to the Ethernet port of the CPU module by other functions or high access load to the SD memory card, a timeout error may occur in the application on the personal computer side. In this case, eliminate the cause of the error and execute this function.

Errors

■Errors detected by the CPU module

The CPU module starts the ODBC server based on the setting of the module parameter. When the CPU module fails to start the ODBC server, the error code is stored in SD1498 (Start-up status of CPU module database). For the error code to be stored, refer to the following.

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

■Errors detected by the personal computer

When an error occurs during the access to the database of the CPU module, the CPU module returns the error message to the personal computer. In this case, check the error message displayed on the personal computer and eliminate the cause of the error. The error causes are as follows.

- The built-in database access setting of the module parameter is disabled.
- The ODBC server of the CPU module is not started. (SM1498 is not on.)
- The communication between the Ethernet port of the CPU module and the personal computer is not established.
- There is no database to be accessed.
- Multiple accesses are simultaneously performed to an access target database.
- A broken database is being accessed.
- The database is accessed when the circuit load of the Ethernet port of the CPU module or the access load to the SD memory card are high.

13 REMOTE OPERATION

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

• When using a Process CPU (redundant mode), refer to the following as well.
☞ Page 387 FUNCTIONS
• When using a Safety CPU, refer to the following as well.
☞ Page 486 FUNCTIONS

The operating status of the CPU module can be changed using an engineering tool or program, or dedicated instructions from the module. The following types of remote operation are available:

- Remote RUN/STOP
- Remote PAUSE
- Remote RESET

13.1 Remote RUN/STOP

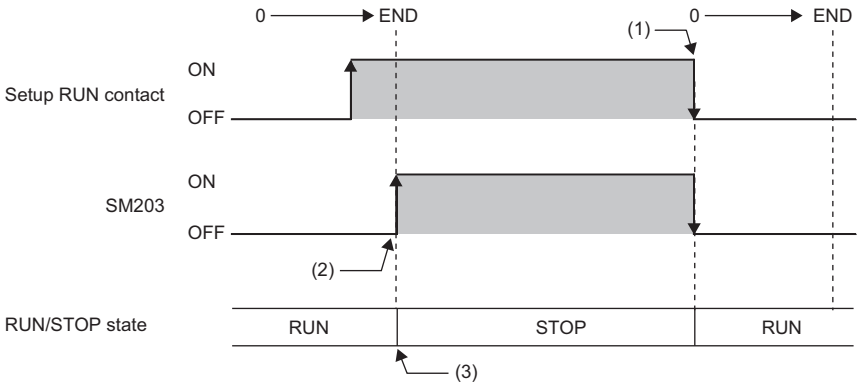
Externally changes the CPU module status to RUN or STOP with the RUN/STOP/RESET switch of the CPU module set to RUN. This function is useful to RUN or STOP a CPU module located in an unaccessible place, or to RUN or STOP a CPU module located on the control panel by using an external signal.

Executing method

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

Using a contact

Set RUN contact in the RUN-PAUSE contact setting. (☞ Page 148 Setting RUN-PAUSE Contacts) When the END processing for the scan where the RUN contact is turned on is executed, SM203 (STOP contact) turns on to change the CPU module status to STOP and stop the operation. When the RUN contact is turned off, the CPU module exits the STOP state and resumes the program operation from the step 0.



(1) Turns on when the END processing for the scan where the set RUN contact is turned on is executed.
(2) When the RUN contact is turned off, the CPU module status changes to RUN and the program operation resumed from the step 0.
(3) When the RUN contact is turned on, the CPU module executes the operation until the END instruction, and enters the STOP state.

Using an engineering tool

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

By an external device using SLMP

Perform remote RUN/STOP with the SLMP command. (📖 SLMP Reference Manual)

Using the module dedicated instructions

Perform remote RUN/STOP with the dedicated instructions of a network module. (📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks))

Precautions

This section describes the precautions on using remote RUN/STOP.

- When remote RUN is performed during execution of the data logging function, it may fail In that case, wait for a while and retry remote RUN. If remote RUN still cannot be executed, check whether remote RUN is acceptable and retry remote RUN (☞ Page 214 About remote operation)
- When the RUN contact, which is specified in the RUN-PAUSE contact setting, is turned off during execution of the data logging function, it may take time to return to the RUN state.

13.2 Remote PAUSE

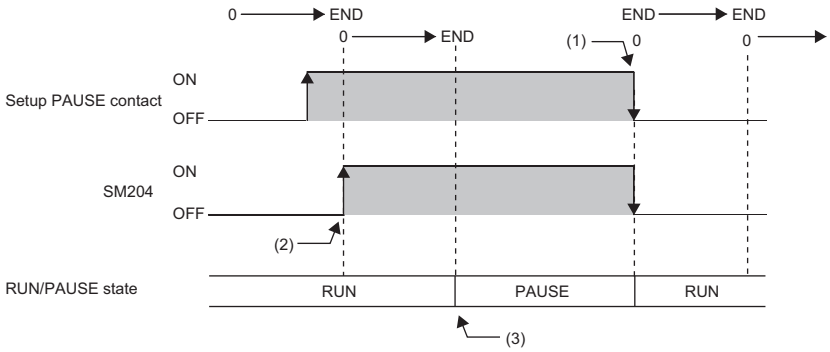
Externally changes the CPU module status to PAUSE with the RUN/STOP/RESET switch of the CPU module set to RUN. Use this in a process control and other situations to keep the CPU module status in RUN even when changing the status of the output (Y) to STOP.

Executing method

The following methods are available to execute remote PAUSE:

Using a contact

Set PAUSE contact in "RUN-PAUSE Contact Setting". (☞ Page 148 Setting RUN-PAUSE Contacts) When the END processing for the scan where the PAUSE contact is turned on is executed, SM204 (PAUSE contact) turns on. When the CPU module executes until the END instruction for the scan next to that where the PAUSE contact is turned on, its status changes to PAUSE, and the operation stops. When the PAUSE contact is turned off, the CPU module exits the PAUSE status and the program operation resumed from the step 0.



(1) When the PAUSE contact is turned off, the CPU module exits the PAUSE state and resumes the program operation from the step 0.
(2) Turns on when the END processing for the scan where the set PAUSE contact is turned on is executed.
(3) When the CPU module executes until the END instruction for the scan next to that where the PAUSE contact is turned on, its status changes to PAUSE, and the operation stops.

Using an engineering tool

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

By an external device using SLMP

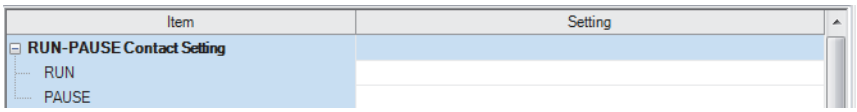
Perform remote PAUSE with the SLMP command. (SLMP Reference Manual)

13.3 Setting RUN-PAUSE Contacts

RUN-PAUSE contacts can be set. RUN-PAUSE contacts are used to perform remote RUN or STOP, or remote PAUSE using a contact.

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [RUN-PAUSE Contact Setting]

Window



Displayed items

Item	Description	Setting range	Default
RUN	Set the contact that controls RUN for the CPU module.	X0 to X2FFF	—
PAUSE	Set the contact that controls PAUSE for the CPU module.	X0 to X2FFF	—

Point

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

13.4 Remote RESET

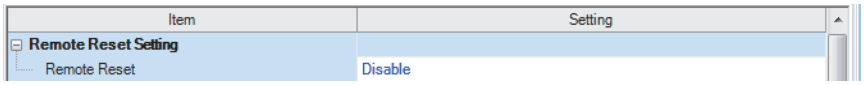
Externally resets a CPU module in the STOP state (including that stopped due to an error). Even when the RUN/STOP/RESET switch of a CPU module is set to RUN, the CPU module can be reset in the STOP state.

Enabling remote RESET

Performing remote RESET requires setting for enabling remote reset.

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Remote Reset Setting]

Window



Displayed items

Item	Description	Setting range	Default
Remote RESET	Set whether or not to enable the remote RESET.	• Disable • Enable	Disable

Executing method

Using an engineering tool

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

By an external device using SLMP

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

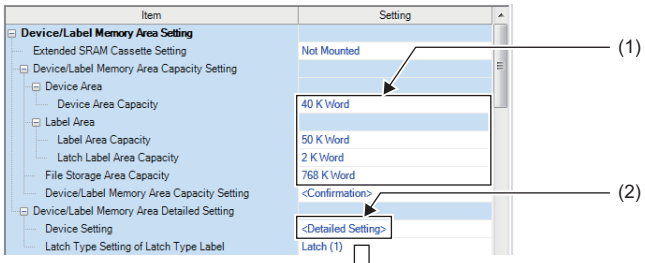
14 DEVICE/LABEL MEMORY AREA SETTING

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

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

☞ Page 486 FUNCTIONS

The capacity of each area on the device/label memory can be specified (☞ Page 77 Device/label memory)



Item	Symbol	Device	
		Points	Range
Input	*X*	12K	0 to 2FFF
Output	Y	12K	0 to 2FFF
Internal Relay	M	12K	0 to 12287
Link Relay	B	8K	0 to 1FFF
Link Special Relay	SB	2K	0 to 7FF
Annunciator	F	2K	0 to 2047
Edge Relay	V	2K	0 to 2047
Step Relay	S	0	
Timer	T	1K	0 to 1023
Long Timer	LT	1K	0 to 1023
Retentive Timer	ST	0	
Long Retentive Timer	LST	0	
Counter	C	512	0 to 511
Long Counter	LC	512	0 to 511
Data Register	D	18K	0 to 18431
Link Register	W	8K	0 to 1FFF
Link Special Register	SW	2K	0 to 7FF
Latch Relay	L	8K	0 to 8191
Total Device			38.4K Word
Total Word Device			34.5K Word
Total Bit Device			62.0K Bit

(1) The capacity of each area can be changed (☞ Page 153 Device/Label Memory Area Setting)
(2) The number of points of user devices can be changed (☞ Page 154 Device Setting)

14.1 Default Capacity of Each Area

The default capacity of each area is as follows.

Item	R04CPU, R04ENCPU	R08CPU, R08ENCPU, R08PCPU	R16CPU, R16ENCPU, R16PCPU	R32CPU, R32ENCPU, R32PCPU	R120CPU, R120ENCPU, R120PCPU
Device Area	40K words	40K words	40K words	40K words	40K words
Label Area	30K words	40K words	50K words	90K words	110K words
Latch Label Area	2K words	2K words	2K words	4K words	4K words
Local Device Area	0K words	0K words	0K words	0K words	0K words
File Storage Area	128K words	512K words	768K words	1024K words	1536K words

Point

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

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

14.2 The Setting Range of the Capacity of Each Area

The setting range of the capacity of each area on the device/label memory is as follows.*1

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

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

R04CPU, R04ENCPU

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

R08CPU, R08ENCPU, R08PCPU

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

R16CPU, R16ENCPU, R16PCPU

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

R32CPU, R32ENCPU, R32PCPU

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

R120CPU, R120ENCPU, R120PCPU

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

Point

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

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

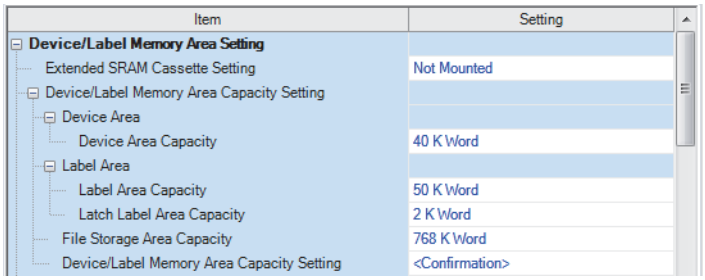
14.3 Device/Label Memory Area Setting

The capacity of each data area allocated within the device/label memory can be changed. (☞ Page 77 Device/label memory)

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

Operating procedure

"Device/Label Memory Area Setting" window



1. Set whether to use an extended SRAM cassette or not in "Extended SRAM Cassette Setting".
2. Set the capacity of each area in "Device/Label Memory Area Capacity Setting".

Displayed items

Item			Description	Setting range	Default
Extended SRAM Cassette Setting			When an extended SRAM cassette is used, the capacity of the cassette must be selected.	• Not Mounted • 1MB • 2MB • 4MB • 8MB • 16MB*1	Not Mounted
Device/Label Memory Area Capacity Setting	Device Area	Device Area Capacity	Sets the capacity of the device area to be used for global devices.	☞ Page 151 The Setting Range of the Capacity of Each Area	☞ Page 150 Default Capacity of Each Area
	Label Area	Label Area Capacity	Sets the capacity of the label area to be used for global labels and local labels.		
		Latch Label Area Capacity	Sets the capacity of the latch label area to be used for latch-type labels.		
	File Storage Area Capacity*2		Sets the capacity of the file storage area to be used for storing register files and others.		
	Device/Label Memory Area Capacity Setting		Shows the device/label memory configuration.		

*1 If using the items with the programmable controller CPU, check the versions of the CPU module and engineering tool. (☞ Page 779 Added and Enhanced Functions)

*2 When using 16MB extended SRAM cassettes (NZ2MC-16MBS), precautions apply when setting the file register capacity. (☞ Page 338 File register setting)

Point

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

14.4 Device Setting

The number of points of each user device can be changed (☞ Page 314 User Device)

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

Window

Item	Symbol	Device	
		Points	Range
Input	***X***	12K	0 to 2FFF
Output	Y	12K	0 to 2FFF
Internal Relay	M	12K	0 to 12287
Link Relay	B	8K	0 to 1FFF
Link Special Relay	SB	2K	0 to 7FF
Annunciator	F	2K	0 to 2047
Edge Relay	V	2K	0 to 2047
Step Relay	S	0	
Timer	T	1K	0 to 1023
Long Timer	LT	1K	0 to 1023
Retentive Timer	ST	0	
Long Retentive Time	LST	0	
Counter	C	512	0 to 511
Long Counter	LC	512	0 to 511
Data Register	D	18K	0 to 18431
Link Register	W	8K	0 to 1FFF
Link Special Registe	SW	2K	0 to 7FF
Latch Relay	L	8K	0 to 8191
Total Device			38.4K Word
Total Word Device			34.5K Word
Total Bit Device			62.0K Bit

Point

Specify each item so that the total number of points for each user device does not exceed the capacity of the device area. (☞ Page 153 Device/Label Memory Area Setting)

Range of use of device points

The following table lists the range of use of device points to be set in the device setting.

Type	Device name	Symbol	Range of use*1	Increment of setting
Bit	Input	X	X0 to X2FFF	—
Bit	Output	Y	Y0 to Y2FFF	—
Bit	Internal relay	M	M0 to M161882111	64 points
Bit	Link relay	B	B0 to B9A61FFF	64 points
Bit	Annunciator	F	F0 to F32767	64 points
Bit	Link special relay	SB	SB0 to SB9A61FFF	64 points
Bit	Edge relay	V	V0 to V32767	64 points
Bit	Step relay*2	S	S0 to S16383	1024 points
Word	Timer	T	T0 to T8993439	32 points
Word	Retentive timer	ST	ST0 to ST8993439	32 points
Word	Long timer	LT	LT0 to LT2529407	1 points
Word	Long retentive timer	LST	LST0 to LST2529407	1 points
Word	Counter	C	C0 to C8993439	32 points
Word	Long counter	LC	LC0 to LC4761215	32 points
Word	Data register	D	D0 to D10117631	4 points
Word	Link register	W	W0 to W9A61FF	4 points
Word	Link special register	SW	SW to SW9A61FF	4 points
Bit	Latch relay	L	L0 to L32767	64 points

*1 This is the maximum number of points for the R120CPU with an extended SRAM cassette (16MB) (NZ2MC-16MBS). The number of points varies depending on the model of the CPU module used, whether to use an extended SRAM cassette, and the type of its cassette.

*2 If using the items with the programmable controller CPU or Process CPU, check the versions of the CPU module and engineering tool. (☞ Page 779 Added and Enhanced Functions)

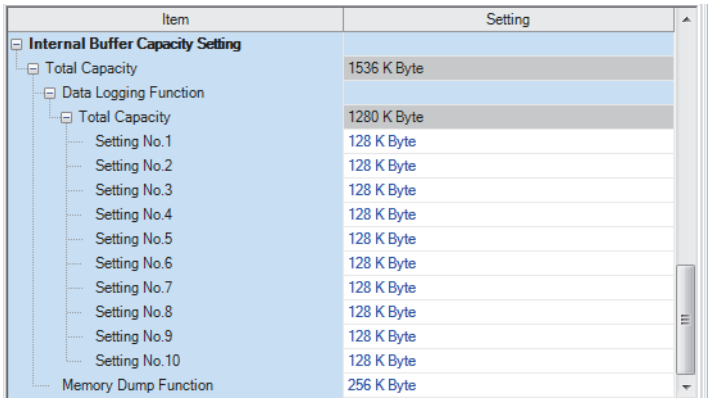
15 INTERNAL BUFFER CAPACITY SETTING

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

Configure the capacity of an area (internal buffer) that the system consumes to temporarily store the result of data logging and the collection result of memory dump. Adjusting the internal buffer capacity allows an increase in the number of collected data and reduces the risk of processing overflow.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Internal Buffer Capacity Setting]

Window



Displayed items

Item		Description	Setting range	Default
Total Capacity		Shows the total of the internal buffer capacity set in the data logging function and the memory dump function.	60 to 3072K bytes	1536K bytes
Data Logging Function	Total Capacity	Shows the total of the internal buffer capacity used for the data logging function.	—	1280K bytes
	Setting No.1 to 10	The internal buffer capacity used for each Setting No. of Data Logging Function.	• Each setting range: 32 to 3040K bytes (in 1K bytes)*1 • Total setting range: 32 to 3040K bytes	128K bytes
Memory dump function		Set the internal buffer capacity used for the memory dump function.	32 to 3040K bytes (in 1K bytes)*2	256K bytes

*1 Leaving this field blank allows the setting to be unused (0K bytes).
*2 A reduced capacity of the internal buffer decreases the extension of scan time, but it takes time to complete.

Point

The internal buffer is also consumed in the real-time monitor function. Set the total of the internal buffer to no more than 3072K bytes including the internal buffer to be consumed in the real-time monitor function as well.
The internal buffer for the real-time monitor function can be set with GX LogViewer. (GX LogViewer Version 1 Operating Manual)

Restriction

The internal buffer capacity setting cannot be set in the CPU module used or the function that is not supported by the engineering tool.
Check whether the engineering tool supports the function with availability for each model in the function list. (MELSEC iQ-R CPU Module User's Manual (Startup))

MEMO

16 DEVICE/LABEL INITIAL VALUE SETTINGS

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

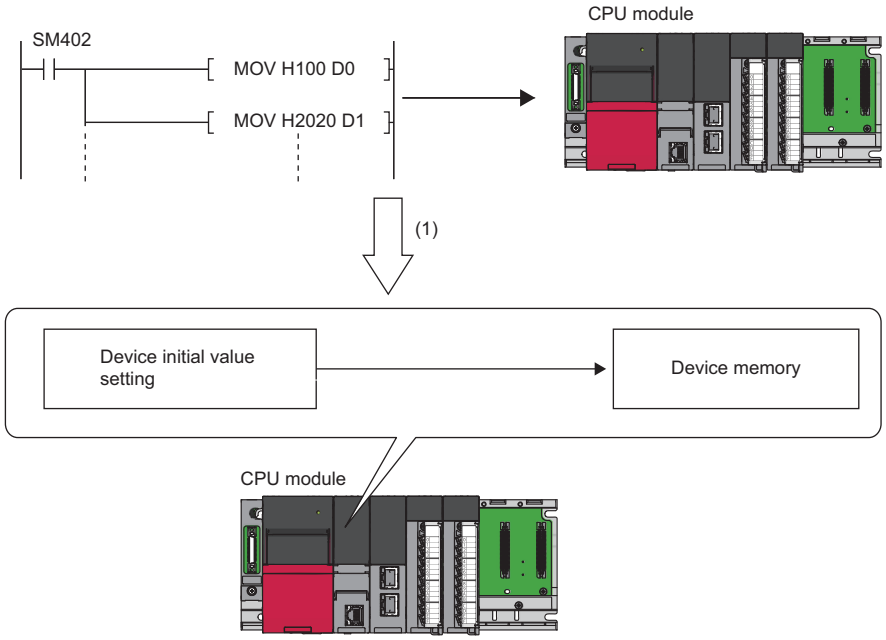
- When using a Process CPU, refer to the following as well.

☞ Page 306 LABEL INITIALIZATION FUNCTION

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

☞ Page 486 FUNCTIONS

Directly sets the initial value of a device or label used by the program (i.e., not via the program) to the buffer memory of the device, label, or intelligent function module.



(1) If initial device values are used, a program to set data to the devices becomes unnecessary.

16.1 Setting Initial Device/Label Values

This section describes the settings required to use initial device/label values.

Setting initial device values

This section describes the settings of initial device values.

Setting procedure

The procedure for using initial device values is as follows.

1. First, the user must create an initial device value file. To set initial values to a local device, create an initial device value file with the same name as the program which sets these values, and specify the range of the values. To set initial values to a global device, create an initial device value file (with any name) which sets these initial values, and specify the range of the values.
2. On the device memory, set up initial device value data within the range specified in the initial device value file.

(📖 GX Works3 Operating Manual)

3. In the "Device Memory Register Diversion", select the device memory which was set up in Step 2. Setting "Device Memory Register Diversion" enables data set up on the device memory to be used as initial device values for the device which is specified in the initial device value file.

(📖 GX Works3 Operating Manual)

4. Configure CPU parameters. (☞ Page 159 Initial value setting)
5. Write the set initial device value file and the CPU parameters to the CPU module.

(📖 GX Works3 Operating Manual)

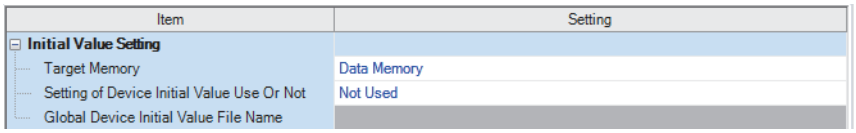
6. The data in the specified initial device value file is automatically set to the specified device or buffer memory of the intelligent function module when the CPU module is powered off and on, reset, or the status changes from STOP to RUN.

Initial value setting

Configure the initial value setting.

[CPU Parameter] ⇨ [File Setting] ⇨ [Initial Value Setting]

Window



Displayed items

Item	Description	Setting range	Default
Target Memory	Sets the storage memory for the initial device value file, initial global label value file, and initial local label value file.	• Memory card • Data Memory	Data Memory
Setting of Device Initial Value Use Or Not	Selects whether or not to use initial device values.	• Not Used • Use	Not Used
Global Device Initial Value File Name	Sets the name of the initial global device value file.*1	60 characters or less	—

*1 If noting is specified, initial global device values are not applied.

Applicable range of initial device value files

The applicable range of initial device value files is as follows.

Target device	Applicable range
Global device	Initial device values set up in the initial value file of the global device are used.
Buffer memory	
Local device	Initial device values set up in the initial value file of the local device (Program Name.DID) are used.
File register	• When "Use File Register of Each Program" is set, initial device values set up in the initial value file of the local device (Program Name.DID) are used. If there is a program which doesn't have its file register (a file register with the same name as the program), it is not assumed as an error but the initial device values are not used. • When a common file register is used across all programs, initial device values set up in the initial value file of the global device are used.

Number of initial device value settings and maximum range of one range

Up to 1000 ranges can be set in one initial device value file. Up to 8000 data points can be set in one range.

Setting initial label values

This section describes the settings of initial label values. For labels assigned to a device, the initial value setting follows the initial device value.

Setting procedure

The procedure for using initial label values is as follows.

1. Set up initial label values using a global label and local label.
2. Configure CPU parameters (☞ Page 159 Initial value setting).
3. The CPU parameters, programs, initial global label value file, and initial local label value file are written to the CPU module.

(📖 GX Works3 Operating Manual)

4. The data in the specified initial global label value file and initial local label value file is automatically set to the specified label when the CPU module is powered off and on, reset, or the status changes from STOP to RUN.

16.2 Applicable Devices/Labels

For details on devices/labels to which initial device/label values can be set, refer to the following.

GX Works3 Operating Manual

16.3 Precautions

This section describes the precautions when using the initial device/label value setting.

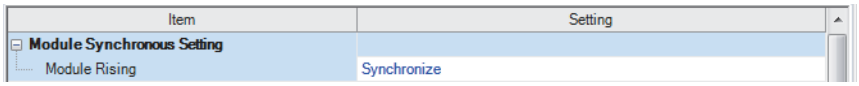
- When initial device values or initial label values are overlapped with the latch range, these initial values take precedence over the latch range. Therefore, even if a device or label is already latched, its values are changed to the specified initial values.
- The initial device value and initial label value are also set when the CPU module status changes from STOP to RUN.*1 Therefore, the initial device value and initial label value cannot be used for areas that are not desirable to be set when the CPU module status changes from STOP to RUN (because they should be set when the power is turned off and on, and changed by the program). By using the MOV instruction, for example, create a program to set an initial value to the specified device or label. In addition, if it is a module access device, make sure to write the initial value in the buffer memory by using the TO instruction.

*1 For details on the initial label value setting for the Process CPU, refer to the following.
Page 306 LABEL INITIALIZATION FUNCTION

- To make the CPU module hold a device only when its values fall into the range of device initial setting (such as module access device and link direct device), the start-up time between the device and the CPU module must be synchronized using the CPU parameter, "Module Synchronous Setting".

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Module Synchronous Setting]

Window



17 OUTPUT MODE AT OPERATING STATUS CHANGE (STOP TO RUN)

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

• When using a Safety CPU, refer to the following as well.
☞ Page 486 FUNCTIONS

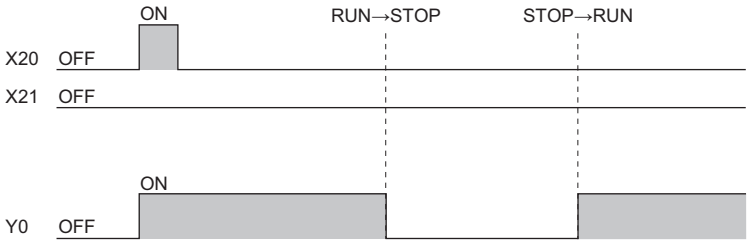
When the operating status changes from RUN to STOP, for example, the CPU module internally stores the status of the outputs (Y) to turn them all off.

17.1 Operation When the Operating Status Is Changed from STOP to RUN

Selects whether or not to resume from the previous output status when the CPU module status is changed from STOP to RUN by using a holding circuit, for example.

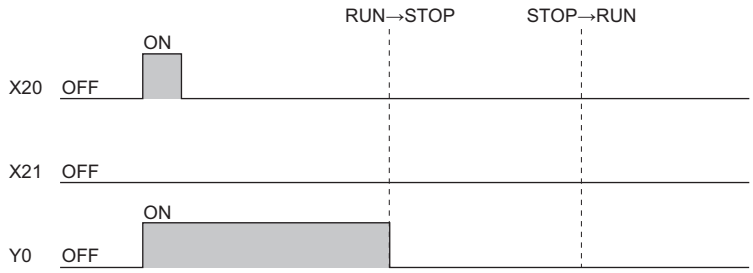
Outputting the output (Y) status before STOP

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



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

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

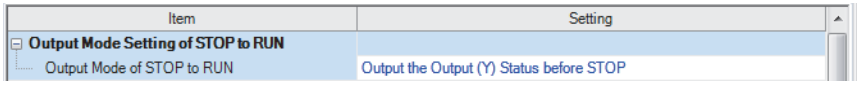


17.2 Setting the Output Mode

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

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

Window



Displayed items

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

18 LABEL ACCESS SETTING FROM EXTERNAL DEVICE

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

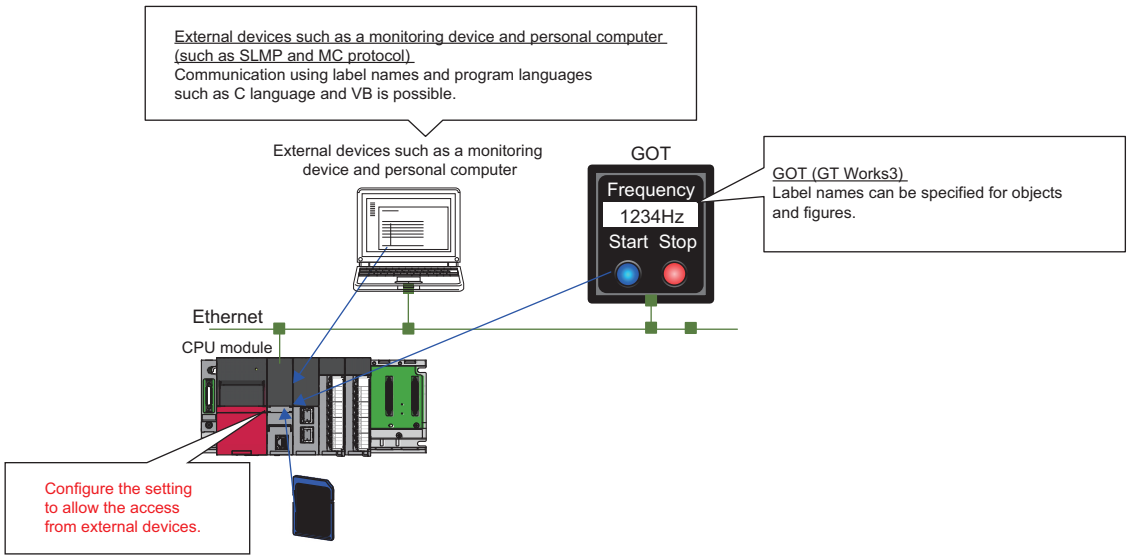
- When using a Process CPU (redundant mode), refer to the following as well.

☞ Page 387 FUNCTIONS

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

☞ Page 486 FUNCTIONS

This setting enables external devices to communicate with the CPU module through GOT, SLMP, or other method by specifying global label names.

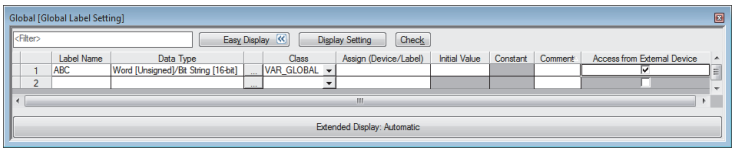


18.1 Configuration Procedure

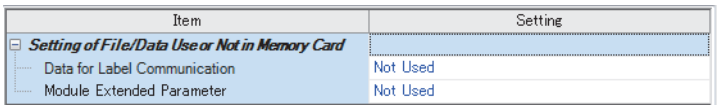
This section describes the configuration procedure to enable access by specifying the global label from external devices.

Operating procedure

"Global Label Setting" window



"Setting of File/Data Use or Not in Memory Card windows



1. Set the label in "Global Label Setting" and select the "Access from External Device" checkbox.
2. Check the capacity of the label communication data.
3. Enable "Data for Label Communication" to use the label communication data stored in the SD memory card (or store the label communication data in the SD memory card).

[Memory Card Parameters] ⇨ [Setting of File/Data Use or Not in Memory Card] ⇨ [Data for Label Communication]

4. Write parameters, global label setting, and global label assignment information files to the CPU module.

Point

Data storage to the data memory, not using the SD memory card, does not need the setting of procedure 3.

18.2 Label Communication Data Storage Location and Operation at the Time of Each Setting Operation

This section describes the storage location for label communication data and the accessibility according to the setting of whether to use the file/data in a memory card.

○: Access from external device possible (Communication memory indicated in parentheses) ×: Access from external device not possible (Generates communication error)

Label communication data storage location	When "Enable" is set in the setting for the usage status of the file/data in a memory card	When "Disable" is set in the setting for the usage status of the file/data in a memory card
Data Memory	×*1	○ (Data memory)
SD memory card	○ (SD memory card)	×
Data memory and SD memory card	○ (SD memory card)	○ (Data memory)

*1 An error occurs in the CPU module.

18.3 File Operation

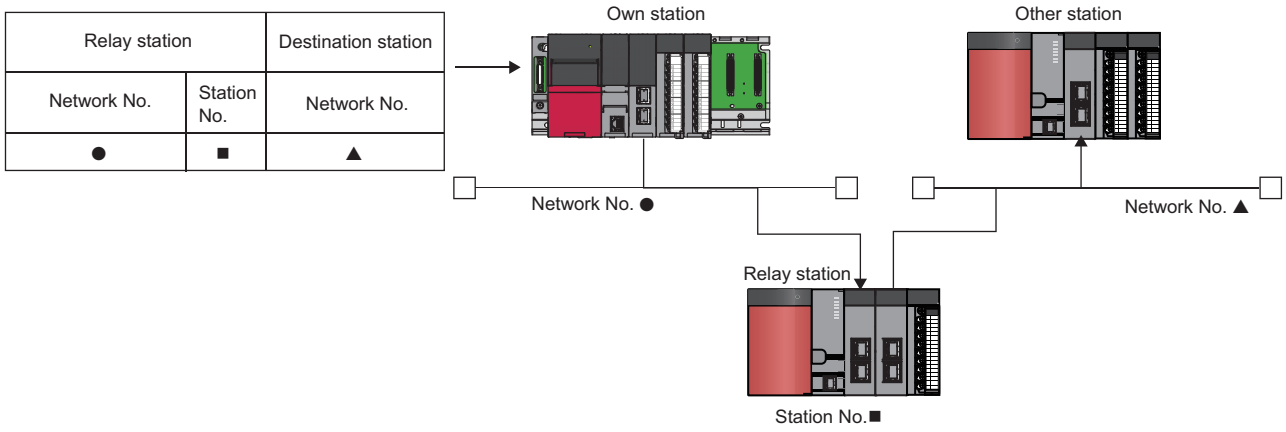
The label communication data can be written from an engineering tool. Reading and deleting*1 are not possible.

*1 Deleting an entire folder is possible.

19 ROUTING SETTING

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

The user can configure any communication route to perform transient transmission to stations in a different network. This setting can be used when the system has a network module which does not support dynamic routing or when it is necessary to clearly specify a communication route.

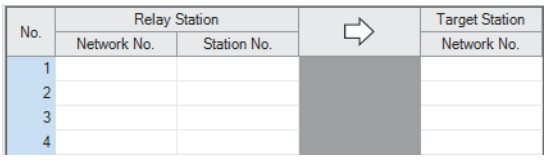


19.1 Setting Method

The user must specify the following: The network number and the station number of the own network (relay station) which will be pass through to another network, and the network number of the final arrival network (destination station). The maximum 238 routing settings can be specified.

[CPU Parameter] ⇨ [Routing Setting]

Window



Displayed items

Item		Description	Setting range	Default
Relay Station	Network Number	Sets the network number of the fist relay station to pass through to the destination station.	1 to 239	—
	Station No.	Sets the station number of the fist relay station to pass through to the destination station.	0 to 120	—
Target Station	Network Number	Sets the network number of the final arrival network.	1 to 239	—

Point

The S(P).RTWRITE instruction can be used to temporary change or add a routing setting during operation (setting made by the S(P).RTWRITE instruction is cleared when the CPU module is powered off or reset). Also, the S(P).RTREAD instruction can be used to read setting details of the parameters. For details on these instructions, refer to the following.

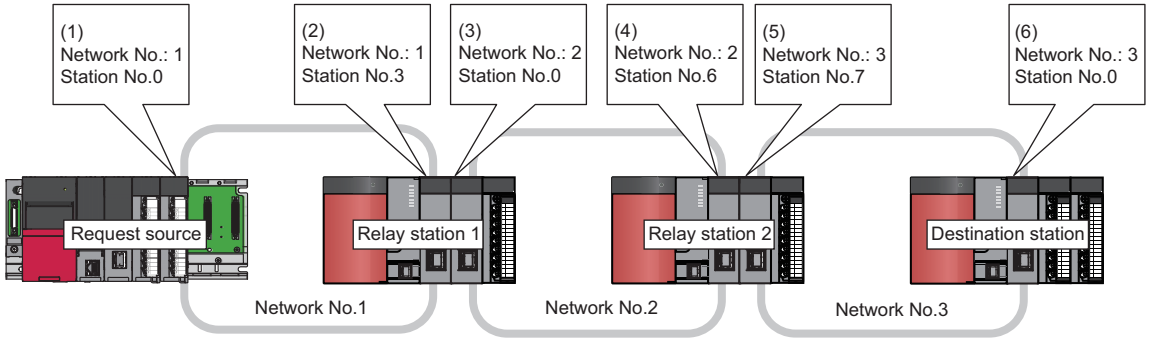
MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

19.2 Setting Example

The following is an example of the routing setting.

Ex.

Transient transmission from the request source (Network No.1) to the target (Network No.3) via Network No.2.



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

19.3 Precautions

The precautions on the routing setting are as follows:

- For the multiple CPU system configuration, the same routing setting must be used for all CPUs.

20 LATCH FUNCTION

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

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

☞ Page 387 FUNCTIONS

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

☞ Page 486 FUNCTIONS

The data in each device/label of the CPU module is cleared to its default value in the following cases:

- When the CPU module is powered off and on
- When the CPU module is reset
- When a power failure exceeding the allowable momentary power failure time occurs

The data in each device/label with latch setting is retained even under the above described conditions. For this reason, control can be continued with the retained data even if the power of the CPU module is turned off or a power failure exceeding the allowable momentary power failure time occurs while the data is managed by a sequential control.

20.1 Types of Latch

The following types of latch are available:

- Latch (1): Data in this range can be cleared by a latch clear operation
- Latch (2): Data in this range cannot be cleared by a latch clear operation

20.2 Applicable Devices and Labels

This section describes the devices/labels that can be latched.

Applicable devices

The devices below can be latched. However, local devices cannot.

Device	Specification method	Applicable latch type
Internal relay (M)	Specify the latch range.	Latch (1) or Latch (2)
Link relay (B)	Specify the latch range.	Latch (1) or Latch (2)
Annunciator (F)	Specify the latch range.	Latch (1) or Latch (2)
Edge relay (V)	Specify the latch range.	Latch (1) or Latch (2)
Timer (T)/Long timer (LT)/Retentive timer (ST)/Long retentive timer (LST)	Specify the latch range.	Latch (1) or Latch (2)
Counter (C) and Long counter (LC)	Specify the latch range.	Latch (1) or Latch (2)
Data register (D)	Specify the latch range.	Latch (1) or Latch (2)
Link register (W)	Specify the latch range.	Latch (1) or Latch (2)
Latch relay (L)	Specify the number of points (latch is performed the same number of times as the specified number of points).	Latch (2) only
File register (R, ZR)	Specify the number of points (latch is performed the same number of times as the specified number of points). Specify the latch range.*1	Latch (2) only

*1 When "Use Common File Register in All Programs" is enabled, any of the number of points is registered within the range for Latch (2). Also, the range can be specified in "Latch (2)" of "Latch Range Setting" (values out of the specified range are cleared to 0).

Point

- When "Use File Register of Each Program" is enabled, the latch range for file registers cannot be specified (the whole range for file registers is retained).
- When the file register file which should be used is changed by the QDRSET instruction, the latch range setting for file registers is disabled. After the change, the whole range for file registers is retained regardless of the latch range setting.

Applicable labels

The labels below can be latched.

Label type	Class
Global label	VAR_GLOBAL_RETAIN
Local label of a program block	VAR_RETAIN
Local label of a function block	VAR_RETAIN
	VAR_OUTPUT_RETAIN
	VAR_PUBLIC_RETAIN

20.3 Latch Setting

This section describes the latch setting.

Setting latch on devices

Multiple latch ranges can be set for a device type. A total of 32 latch ranges between latch (1) and latch (2) can be set. However, the ranges of latch (1) and latch (2) must not overlap.

Setting a latch range

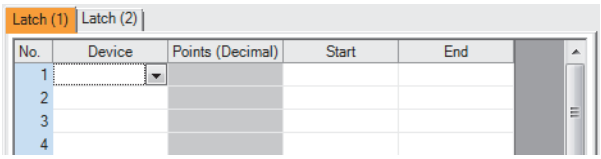
Set the device to latch, its range, and the latch type.

Operating procedure

"Device Setting" window

Item	Symbol	Latch (1)	Latch (2)
Input	X		
Output	Y		
Internal Relay	M	No Setting	No Setting
Link Relay	B	No Setting	No Setting
Link Special Relay	SB		
Annunciator	F	No Setting	No Setting
Edge Relay	V	No Setting	No Setting
Step Relay	S		
Timer	T	No Setting	No Setting
Long Timer	LT	No Setting	No Setting
Retentive Timer	ST	No Setting	No Setting
Long Retentive Time	LST	No Setting	No Setting
Counter	C	No Setting	No Setting
Long Counter	LC	No Setting	No Setting
Data Register	D	No Setting	No Setting
Link Register	W	No Setting	No Setting
Link Special Registe	SW		
Latch Relay	L		No Setting
Total Device			
Total Word Device			
Total Bit Device			

"Latch Range Setting" window



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

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

2. In the "Device Setting" window, select the type of latch for the target device. The "Latch Range Setting" window is displayed.

3. Check the tab for the latch type, select the device to set and set the latch range (Start, End).

Setting the latch interval

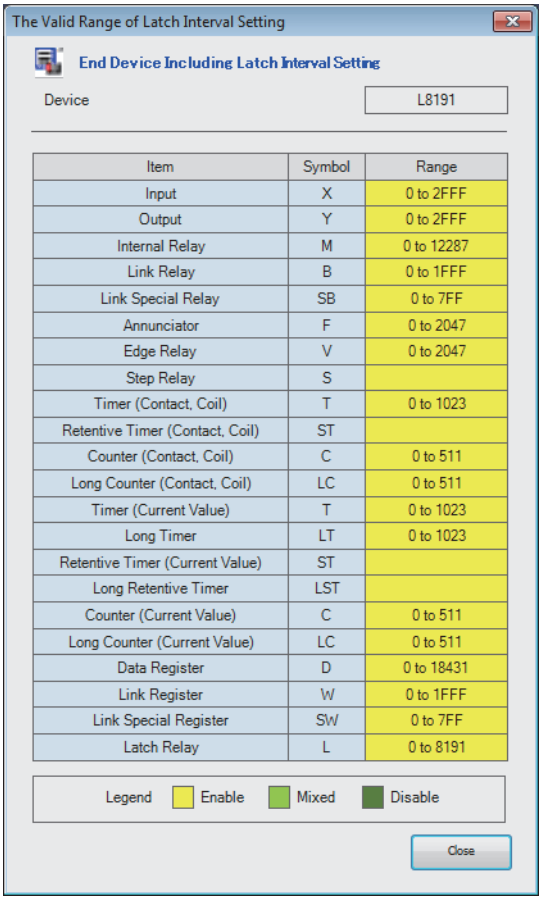
The user can specify the operation which should be performed at a latch interval (☞ Page 172 Timing of the latch processing) within the effective range of the latch timing setting[*1].

*1 The effective range of the latch timing means the range of devices which is enabled on the "The Valid Range of Latch Interval Setting" window. For other ranges (ranges of devices which are not enabled on the "The Valid Range of Latch Interval Setting" window), this setting is not applied, the END processing is accelerated, and latching is performed in real time.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device Latch Interval Setting]

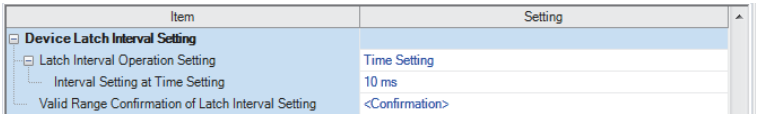
Operating procedure

"The Valid Range of Latch Interval Setting" window



1. On the "Valid Range Confirmation of Latch Interval Setting" window, check the effective range for the latch timing setting. Specify the latch interval operation within the effective range.

"Device Latch Interval Setting" window



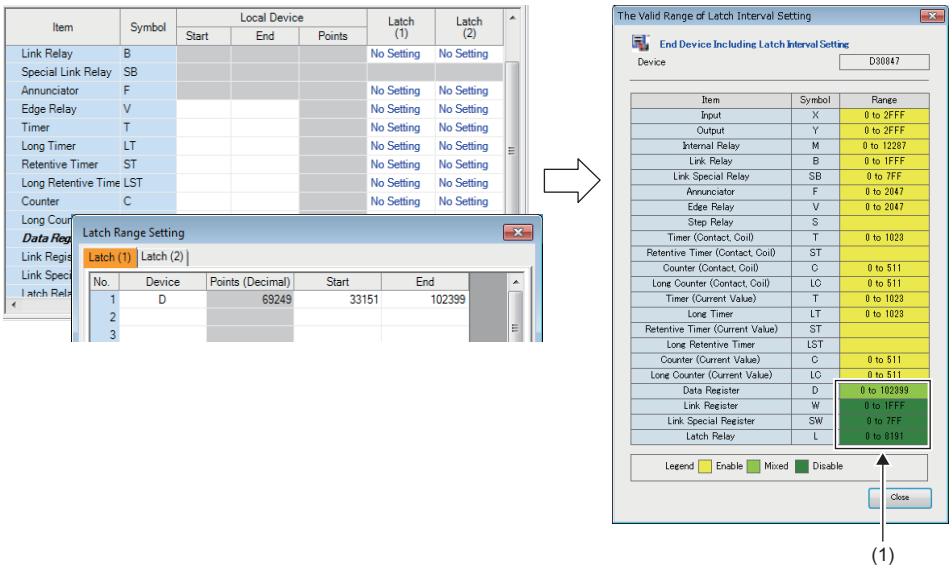
2. In "Latch Interval Operation Setting", select the timing of the latch processing. When "Time Setting" is selected, specify the time interval of latching.

Displayed items

Item	Description	Setting range	Default
Latch Interval Operation Setting	Sets the timing of the latch processing.	• Per Scan • Time Setting	Time Setting
Interval Setting at Time Setting	Sets the time interval of latching (when "Time Setting" is selected).	1 to 2000ms (in units of 1ms)	10ms
Valid Range Confirmation of Latch Interval Setting	Shows the effective range for the latch timing setting.	—	—

Point

For device latching, increasing the device range in the device setting of CPU parameters eliminates the latch processing from the END processing for the devices and enables real-time latching. For example, assume the following configuration for R04CPU: (1) 0K word is specified for the file storage area and 168K words for the device area in "Device/Label Memory Area Capacity Setting", (2) 100K points is specified for the data register (D) on the "Device Setting" window, and (3) D33151 and later are specified as the latch target devices. In this configuration, D33152 and later data registers (D) can be latched in real time as well as the devices (W, SW, and L) which are out of the effective range.



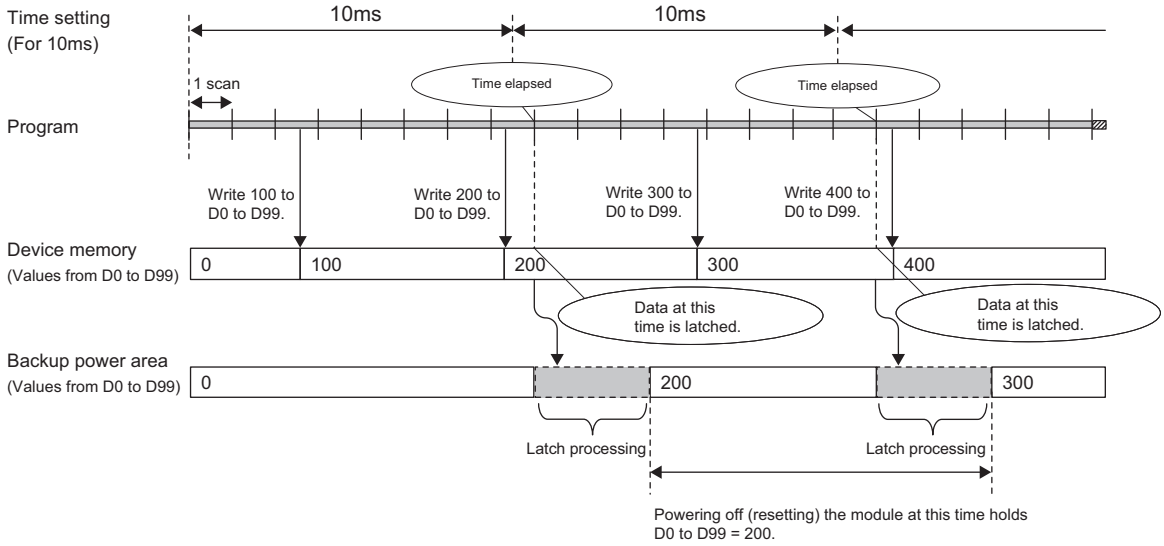
(1) D33152 or later areas of the data register (D) and areas of the devices (W, SW, and L) in disabled ranges can be latched in real time.

■Timing of the latch processing

The timing of the latch processing is determined based on the effective range of the latch timing setting and the operation setting for the specified latch interval (☞ Page 170 Setting the latch interval)

• When set to "Time Setting"

Latch processing is started in the END processing executed after the set time.

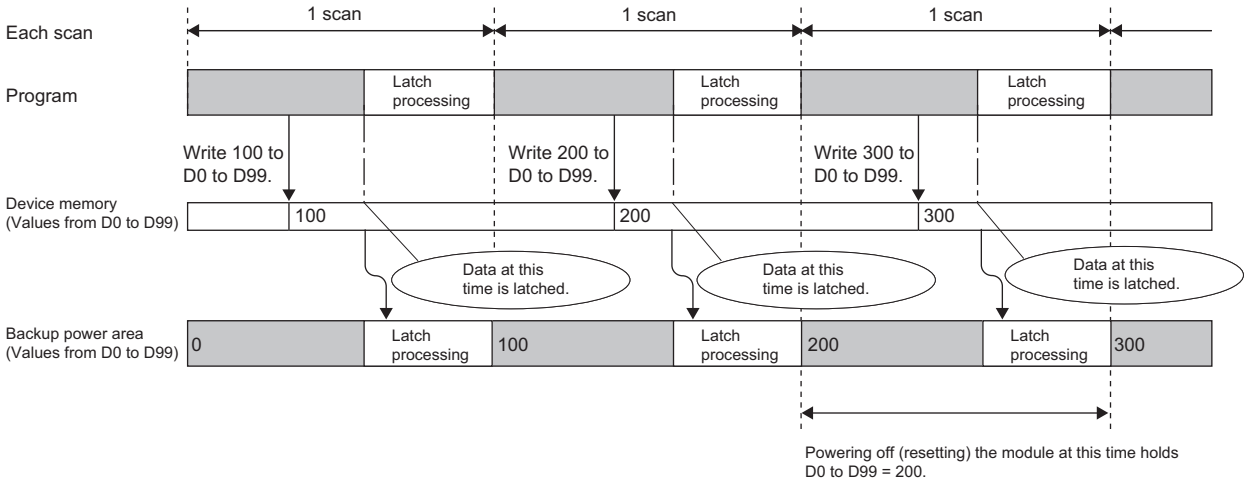


Point

- The more the latch points, the more it takes to complete the latch processing. If the set time elapses while latch processing is still executing, the next one is executed in the END processing that follows the completion of the previous latch processing. Set a latch interval longer than the latch processing time so that the specified time does not elapse during latch processing.
- If the latch interval is shorter than the scan time, latch timing occurs more than once within one scan. However, latch processing is executed once during the END processing.

• When set to "Per Scan"

Latch processing is executed in the END processing of every scan. This setting allows to always retain the device of the previous scan because the devices are latched at every scan. However, it may increase the scan time.

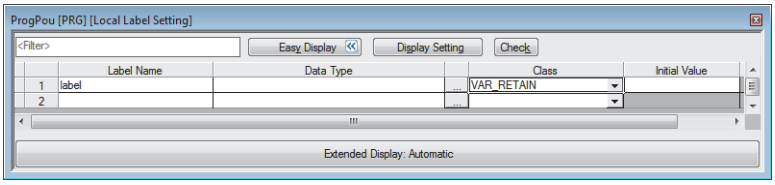


Setting latch on labels

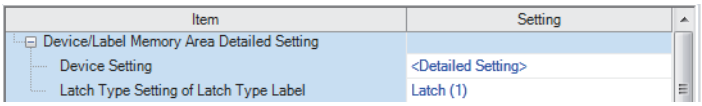
This section describes latch setting on labels.

Operating procedure

Label edit window



"Device/Label Memory Area Detailed Setting" window



1. In the label edit window, specify "RETAIN" for label attribute.

2. There are two types of latch for labels: latch (1) and latch (2). Select one. The selected latch type is applied to labels of all latch attributes.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Detailed Setting] ⇨ [Latch Type Setting of Latch Type Label]

20.4 Clearing Latch Range Data

Latch range data is cleared by either of the operations below. (☞ Page 80 Memory Operation)

- Latch clear: Performed from the engineering tool. (📖 GX Works3 Operating Manual)
- Latch clear by program: Execute RST instruction for latched devices, or clear by transferring K0 by using the MOV or FMOV instruction.

20.5 Precautions

This section describes the precautions when using the latch function.

- The data in a device/label within the latch range is retained by using the battery on the CPU module unit. Therefore, data cannot be retained during power failure if this battery runs out.
- If the latch range and number of device points are modified by using parameters, latch is performed on the modified latch range. However, if the value of the parameter setting the latch range is different between the previous and current operations because the CPU module was powered off and on or is reset, and the latch range has been increased, the device range in the increased portion is not latched.

21 MEMORY CARD FUNCTION

This section describes functions that use the SD memory card.

21.1 SD Memory Card Forced Disable

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

Disables the SD memory card without turning the power off and on even while a function using the SD memory card, such as the data logging function is running.

How to forcibly disable the SD memory card

The SD memory card is forcibly disabled by the following operations.

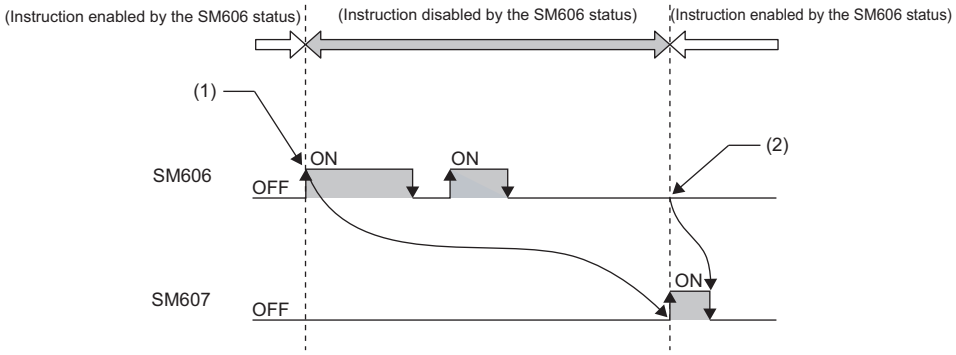
Operation of SD memory card access control switch

1. Press the SD memory card access control switch for one second or more.
2. The CARD READY LED flashes and then turns off.*1
3. Remove the SD memory card.

*1 If the SD memory card was being accessed by a function, CARD READY LED goes out after access by the function is finished. Therefore, the time required for the LED to go out varies depending on the function.

Operation of special relay

1. Turn on SM606 (SD memory card forced disable instruction). Operation by turning SM606 on and off is disabled until SM607 turns on after SM606 is turned on. However, it is enabled and operates according to ON/OFF state of SM606 after the status of SM607 changes.



(1) On: Forcibly stopping the use of the SD memory card
(2) Off: The use of the SD memory card is allowed.

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

Operation of functions accessing the SD memory card

Disabling the access to SD memory card affects the operation of functions that access the SD memory card. This section describes the operation of the functions listed below when SD memory card forced disable is executed during an access to the SD memory card or when they access the SD memory card after it is forcibly disabled.

Running function	When SD memory card forced disable is executed during access to the SD memory card	When SD memory card is accessed after it is disabled
Boot Operation	—	—
• Access to the device comment and label in the SD memory card • Operation for device/label initial value at the time of STOP → RUN	The SD memory card is disabled after the running function finishes.	An error occurs in the CPU module.*1
Access to SD memory card by an engineering tool, SLMP, and FTP function	An error response is returned.	An error response is returned.
FREAD/FWRITE instruction	The SD memory card is disabled after the execution of the instruction finishes.	The instruction is completed with an error.
CPU module data backup/restoration function	At the completion of the backup/restoration processing to a file, the SD memory card is disabled, the backup/restoration function is completed with an error, and then an error cause is stored in the special register.	An error cause is stored in the special register.
iQ Sensor Solution data backup/restoration function	The SD memory card is disabled at the completion of the backup/restoration processing.	An error is returned.
File transfer function (FTP client)	At the completion of the file transfer processing executed to a file, the SD memory card is disabled, the file transfer function instruction is completed with an error, and then an error cause is stored in the completion status.	The file transfer function instruction is completed with an error and an error cause is stored in the completion status.

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

Canceling SD memory card forced disable

The SD memory card forced disable is canceled by the following operation:

1. Install the SD memory card again.*1
2. Power off and on, or reset the CPU module.

*1 The CARD READY LED flashes and goes on.

Precautions

This section describes the precautions on using SD memory card forced disable.

- When the SD memory card is forcibly disabled by operating both the SD memory card access control switch and SM606, the operation executed first is valid and the operation executed afterwards, invalid. For example, turning SM606 on and off without removing the SD memory card after forcibly disabling it by the SD memory card access control switch cancels the SD memory card forced disable. However, if SM606 is turned on after forcibly disabling the SD memory card by the SD memory card access control switch and removing the SD memory card, the removed SD memory card will not be available even when it is re-inserted. To enable the SD memory card, SM606 must be turned off after re-insertion.
- If this function is executed while a file is written from an external device to the SD memory card, the writing may fail. Write the file again after enabling the disabled SD memory card.

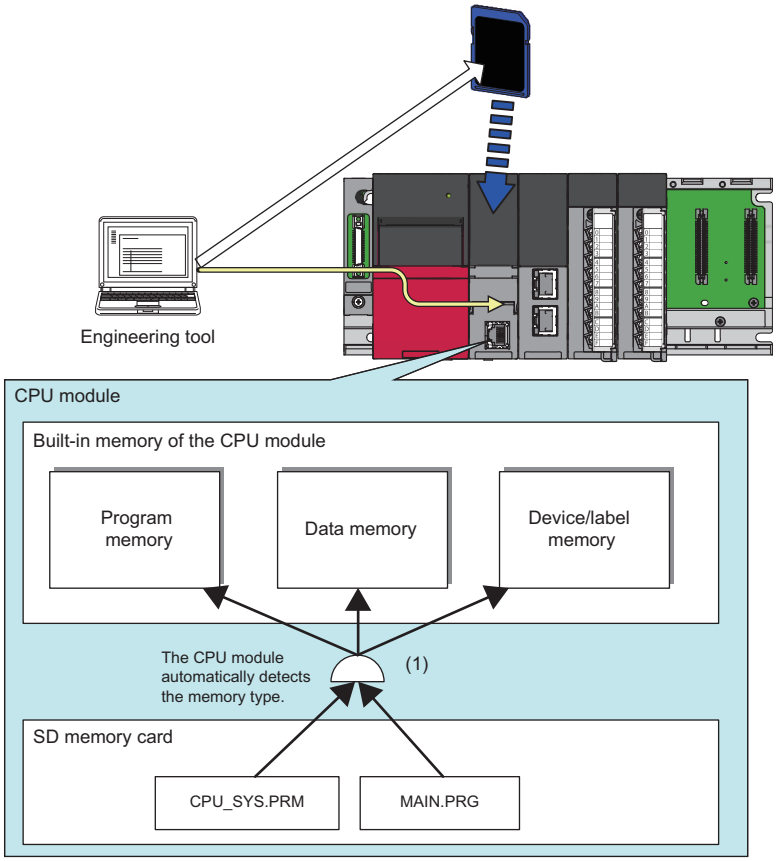
21.2 Boot Operation

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

• When using a Process CPU (redundant mode), refer to the following as well.
☞ Page 387 FUNCTIONS

The files stored on the SD memory card are transferred to the storage memory which is automatically determined by the CPU module when the CPU module is powered off and on or is reset.

(1) The boot operation is executed according to the boot file setting when the CPU module is powered on or is reset.



Boot operation procedure

The boot operation procedure is as follows.

1. Configure the boot setting.
2. Attach an SD memory card.
3. Write the boot setting and boot file to the SD memory card.
4. Power off and on, or reset the CPU module.

Specifiable file types

The files which can be booted are as follows.

- System parameter
- CPU parameter
- Module parameter
- Module extension parameter
- Module extension parameter for protocol setting
- Remote password
- Global label setting file
- Initial global label value file
- Program file
- Initial local label value file
- FB/FUN file
- Device comment line
- Initial device value file

Allowed maximum number of boot files

The maximum number of boot file settings which can be specified is 512. However, because more than one file is bootable for a single setting, the maximum number of boot files is the same as the number of files which can be stored in the transfer destination memory.

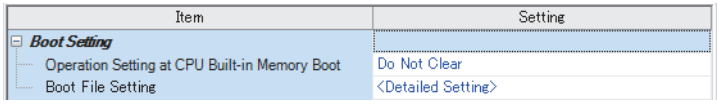
Configuring the boot setting

Configure the necessary settings for the boot operation.

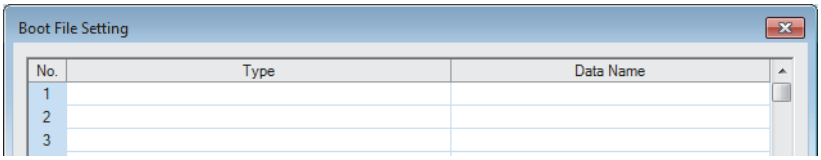
[Memory Card Parameters] ⇨ [Boot Setting]

Operating procedure

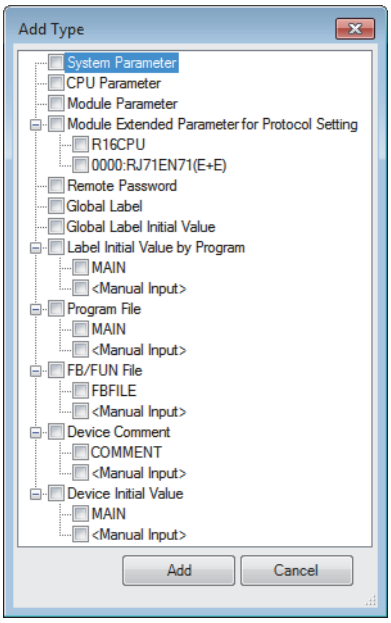
"Boot Setting" window



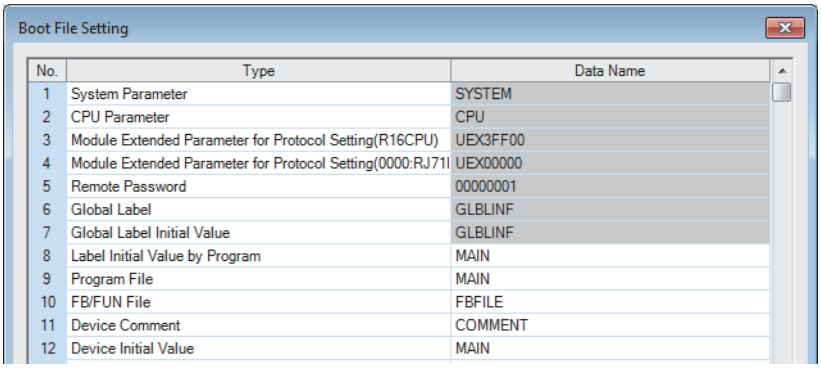
"Boot File Setting" window



"Add Type" window



"Boot File Setting" window



1. Click "Detailed Setting" on the "Boot File Setting" window.
2. Click the "Type" column. The maximum number of boot files that can be specified is the same as the number of files that can be stored in the storage memory.
3. Select type for the boot file. (Multiple selection possible)
4. Set the data name (file name).

Displayed items

Item	Description	Setting range	Default
Operation Setting at CPU Built-in Memory Boot	Sets whether or not to clear the CPU built-in memory upon file transfer from the SD memory card.	• Do Not Clear • Clear	Do Not Clear
Boot File Setting	Sets the file used for boot operation from the SD memory card.	—	—

Operation when security functions are enabled

This section describes the operation when security functions are enabled.

When a security key is set

When a security key is set to the boot target program file and the security of the program file does not match with that of the CPU module, a boot error occurs. Also, when no security key is written to the CPU module, a boot error occurs as well.

Security key of boot target program file	Security key of CPU module	Security key match/mismatch	Boot program execution
Set	Written	Match	Execute
	Written	Not match	Not execute (boot error)
	Not written	—	Not execute (boot error)

When a file password is set

If a file password is set on both the source boot file and destination file, the file can be transferred only when the passwords match. Furthermore, the file transfer does not work if a file password is set only on either one.

Transferring boot file		Transferred boot file		Password match/ mismatch	Transfer
File	File password setting	File	File password setting		
Existing	Set	Existing	Set	Match	Yes
				Mismatch	No
			Not set	—	No
		Not set	—		Yes
	Not set	Existing	Set		No
			Not set		Yes
		Not set	—		Yes

Precautions

This section describes the precautions when using the boot operation.

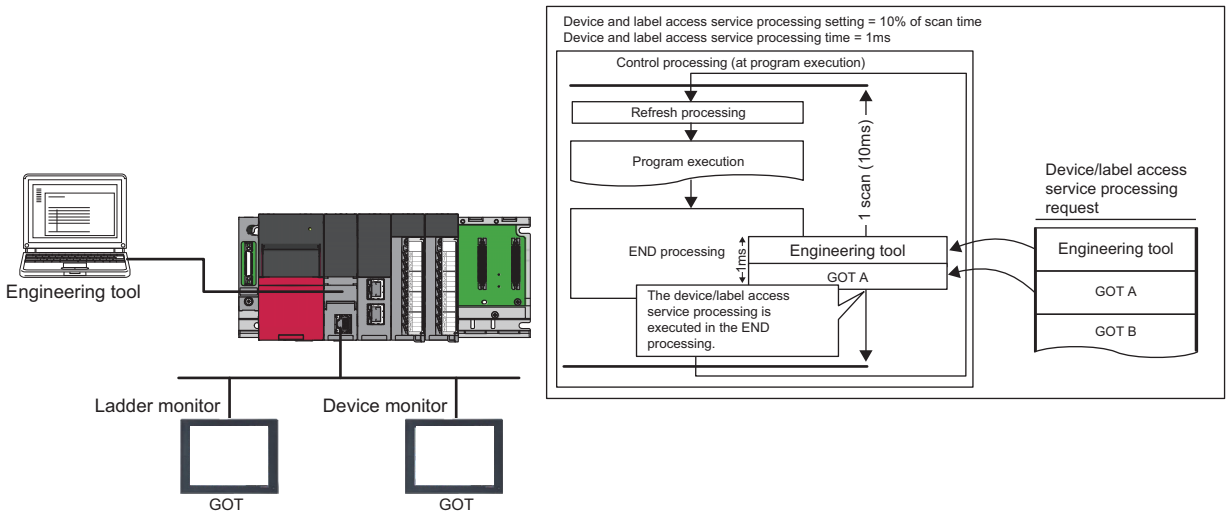
- When a parameter file is set as boot file, it overwrites the parameter file inside the transfer destination CPU module. In addition, if a parameter file is not set as boot file even when stored on the SD memory card, the CPU module operates in accordance with the settings in the parameter file inside it.
- If a program in the program memory is changed online during boot operation, the change is also reflected in the transfer source program on the SD memory card.
- Note the model of the program written on the SD memory card (program specified in the boot file settings) and the model of the CPU module must be the same.
- For the boot file setting, check that the setting is consistent with the program/label setting. When the file is read from the programmable controller after the boot operation without consistency, conversion may not function properly due to lack of consistency in project data.

22 DEVICE/LABEL ACCESS SERVICE PROCESSING SETTING

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

• When using a Safety CPU, refer to the following as well.
☞ Page 486 FUNCTIONS

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



Point

For functions to which the device/label access service processing setting can be applied, refer to the list targeted for device/label access service processing setting. (☞ Page 724 List of the Targets of Device/Label Access Service Processing Setting)

22.1 Operations Enabled by Setting Details

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

Item	Scan performance		Device/label access service process performance		Inter-program monitoring*5	Application
	Increase*1	Stability*2	Response time*3	Stability*4		
Execute the process as scan time proceeds	Medium	Medium	Medium	Medium	None	This setting is useful to execute the device/label access service processing in a way commensurate with the system size. It allows the system to be designed without considering the device/label access service processing time because it is determined as a function of the scan time.
Set processing time	Medium	High	Medium	High	None	This setting is useful to give priority to the device/label access service processing. It allows for stable communication because the CPU module can always process a constant amount of the device/label access service processing without affecting the scan time.
Set processing counts	Large	Low	Medium	High	None	This setting is useful to stably execute the device/label access service processing in a system where requests come from multiple peripherals. It provides stable communication in a system where multiple peripherals exist because the CPU module can execute the device/label access service processing based on the number of request sources.
Execute END Processing between Programs	Large	Medium	Quick	High	Yes	This setting is useful to give priority to the device/label access service processing in a system with a large number of programs. It improves the response performance as it enables the device/label access service processing to be executed multiple times during a single scan.

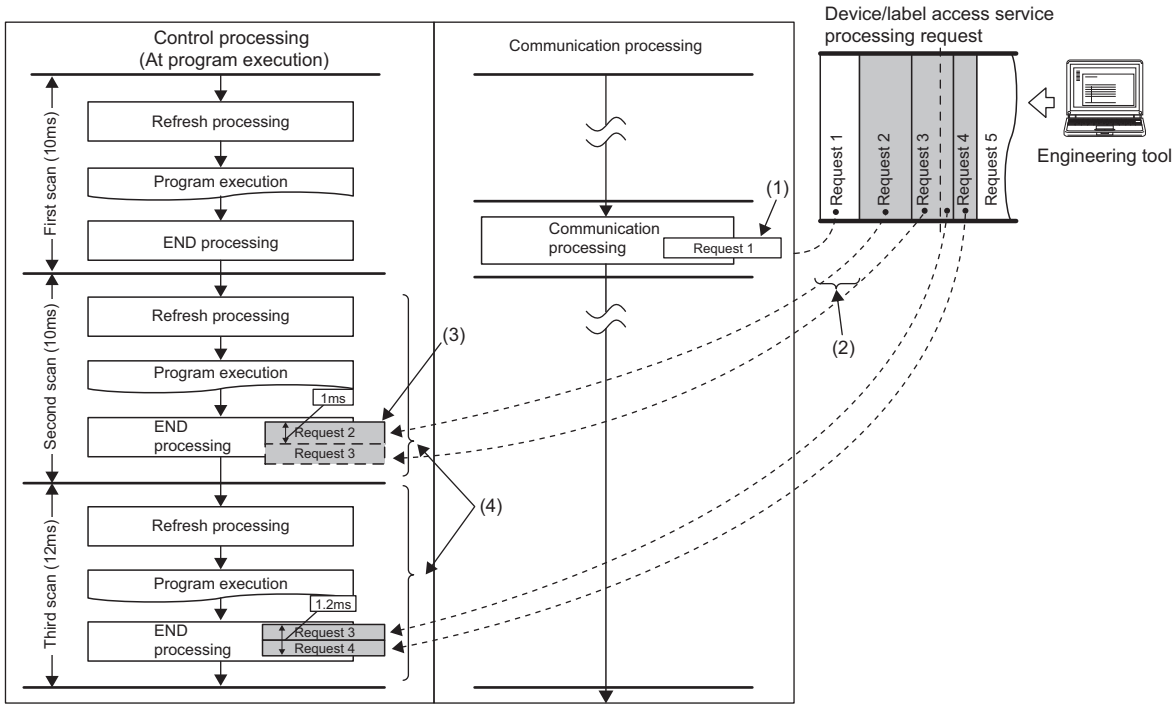
*1 Indicates how much the scan time will be extended at its maximum due to the device/label access service processing.
*2 Indicates how much the scan time will be varied due to the device/label access service processing.
*3 Indicates how long the response time will take after a request for the device/label access service processing is received from an engineering tool or the like.
*4 Indicates how much the response time will be varied depending on the contents of requests for the device/label access service processing from engineering tools or the like.
*5 When this function is enabled, monitoring is performed between execution of programs, and therefore values of ongoing operation may be read out (☞ Page 186 When "Execute END Processing between Programs" is enabled)

Execute the Process as Scan Time Proceeds

This setting is useful to execute the device/label access service processing in a way commensurate with the system size. It allows the system to be designed without considering the device/label access service processing time because it is determined as a function of the scan time.

Ex.

When "Scan time ratio = 10%" is set



(1) Synchronization with program is unnecessary.

(2) Multiple requests are processed until the specified device/label access service processing ratio (10%) is exceeded. When the specified device/label access service processing ratio is exceeded, the remaining requests are processed in the END processing of the next scan. Also, for scan which has the device/label access service processing time shorter than 0.1ms, it is assumed as "The device/label access service processing time per scan = 0.1ms".

(3) Operations such as access to device, which are synchronized with the program, are processed in the END processing.

(4) The maximum device/label access service processing time available varies because the scan time (program execution time) varies.

Point

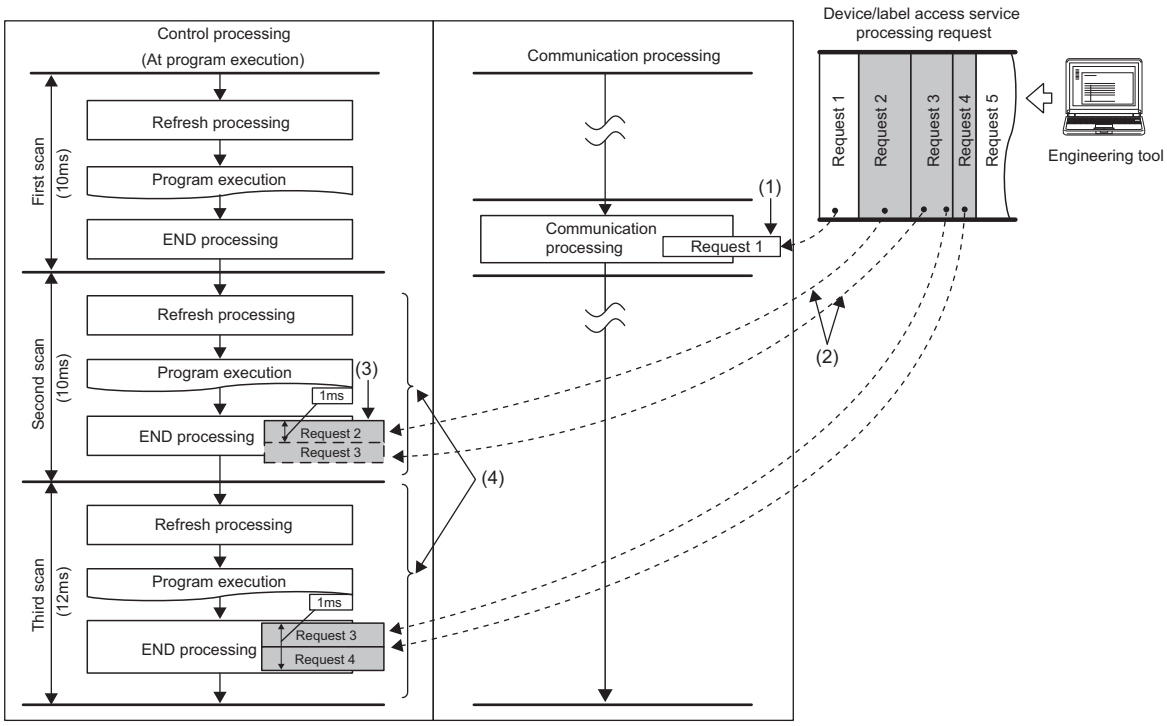
For operations such as access to devices, which are synchronized with the program, adjust the time by this setting because they are processed during the END processing. If no request data for the device/label access service processing exists, the scan time is shortened by the specified ratio as the CPU module proceeds to the next scan without waiting for requests.

Set Processing Time

This setting is useful to give priority to the device/label access service processing. It allows for stable communication because the CPU module can always process a constant amount of the device/label access service processing without affecting the scan time.

Ex.

When "Processing time = 1ms" is set



(1) Synchronization with program is unnecessary.

(2) Multiple requests are processed until the specified processing time (1ms) is exceeded. If the specified processing time is exceeded, the remaining requests are processed in the END processing of the next scan.

(3) Operations such as access to device, which are synchronized with the program, are processed in the END processing.

(4) The maximum device/label access service processing time available is the same even when the scan time (program execution time) varies.

Point

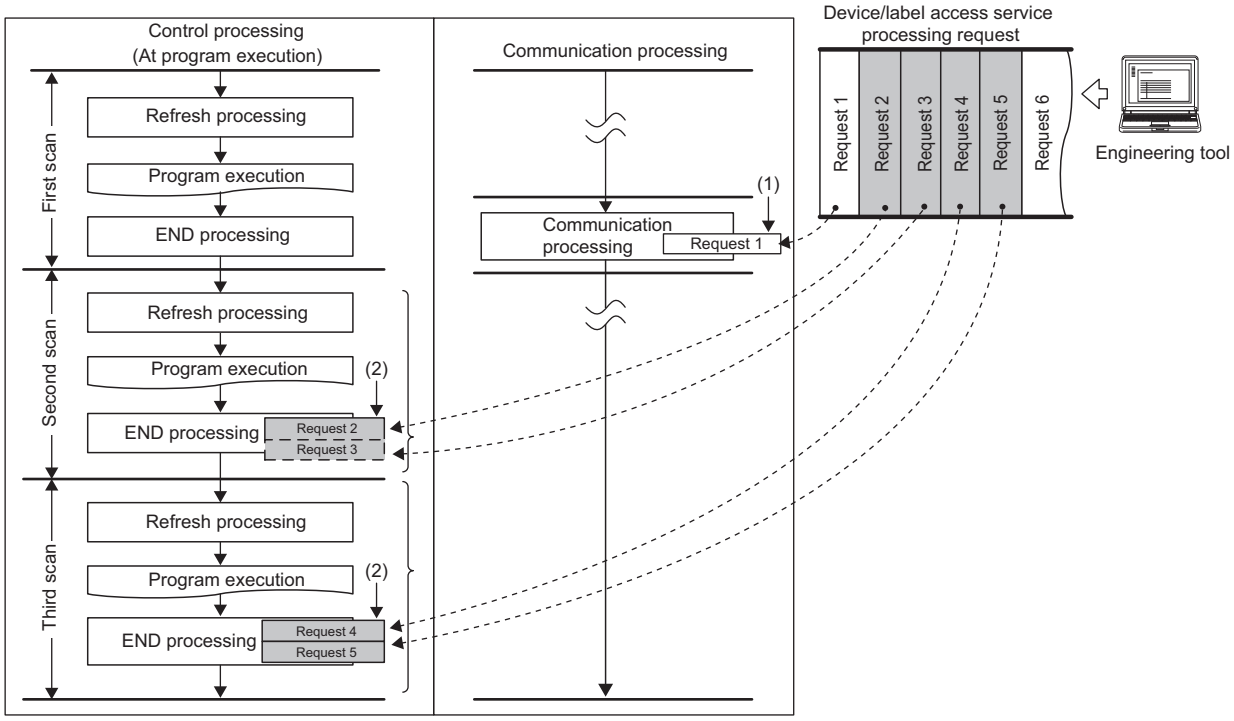
For operations such as access to devices, which are synchronized with the program, adjust the time by this setting because they are processed during the END processing. If no request data for the device/label access service processing exists, the scan time is shortened by the specified ratio as the CPU module proceeds to the next scan without waiting for requests.

Set Processing Counts

This setting is useful to stably execute the device/label access service processing in a system where requests come from multiple peripherals. It provides stable communication in a system where multiple peripherals exist because the CPU module can execute the device/label access service processing based on the number of request sources.

Ex.

When "Processing counts = 2" is set



(1) Synchronization with program is unnecessary.

(2) Two requests are processed in one END processing independently of the requested processing time.

Point

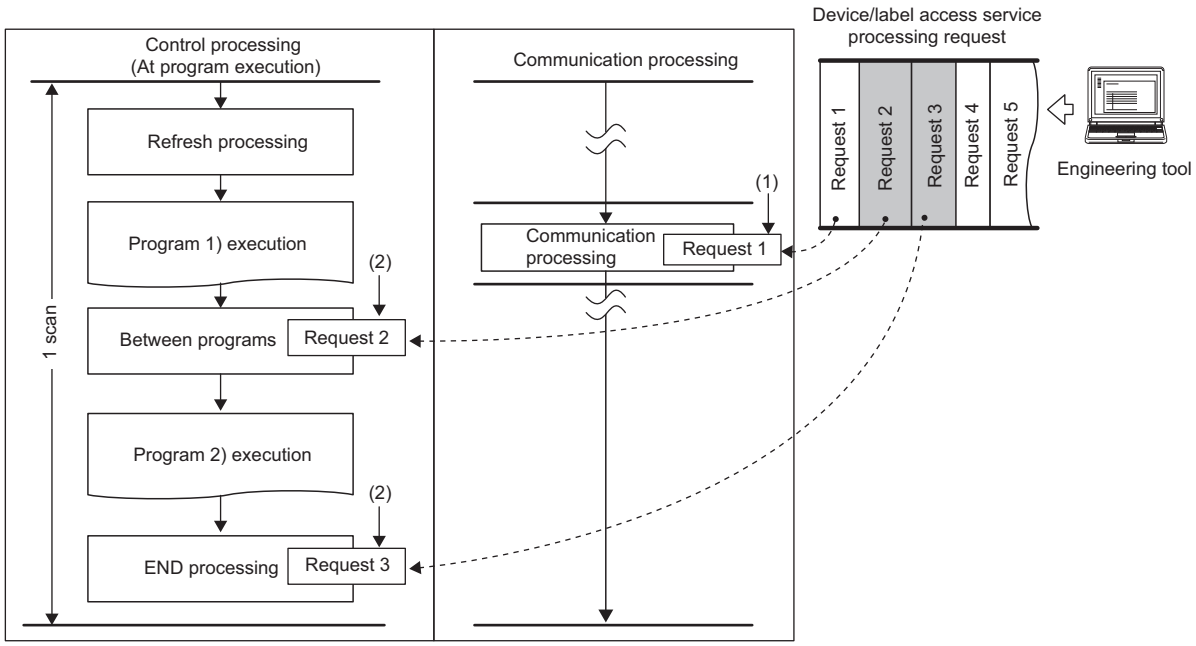
For operations such as access to devices, which are synchronized with the program, adjust the number of executions by this setting because they are processed during the END processing. If no request data for the device/label access service processing exists, the CPU module proceeds to the next scan without waiting for requests.

Execute END Processing between Programs

This setting is useful to give priority to the device/label access service processing in a system with a large number of programs. Because operations such as access to devices are performed between program executions and during the END processing, requests as many as the number of programs can be processed during a single scan. As a result, it can reduce the response for the device/label access service processing.

Ex.

When "Execute END Processing between Programs" is enabled



(1) Synchronization with program is unnecessary.

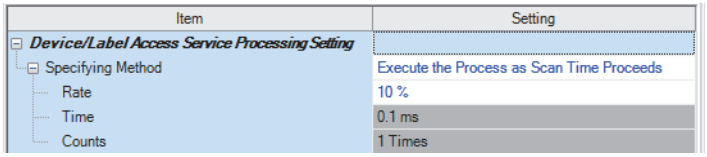
(2) Requests for operations such as access to devices are processed between program executions and during the END processing.

22.2 Setting Method

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

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

Window



Displayed items

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

22.3 Precautions

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

Functions that may prolong the scan time

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

- Online change
- Device/buffer memory batch monitor
- Data logging function
- File register write or read*1

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

When constant scan is set

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

When "Execute END Processing between Programs" is enabled

The device/label access service processing, such as access to devices, is performed between program executions and/or during the END processing. Therefore, when monitoring is performed in the situation where a device value is processed across programs, values of ongoing operation may be read out.*1 Also, when "Use File Register of Each Program" is enabled and monitoring is performed for the file registers, unintended data may be obtained. (GX Works3 Operating Manual)

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

23 DATA LOGGING FUNCTION

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

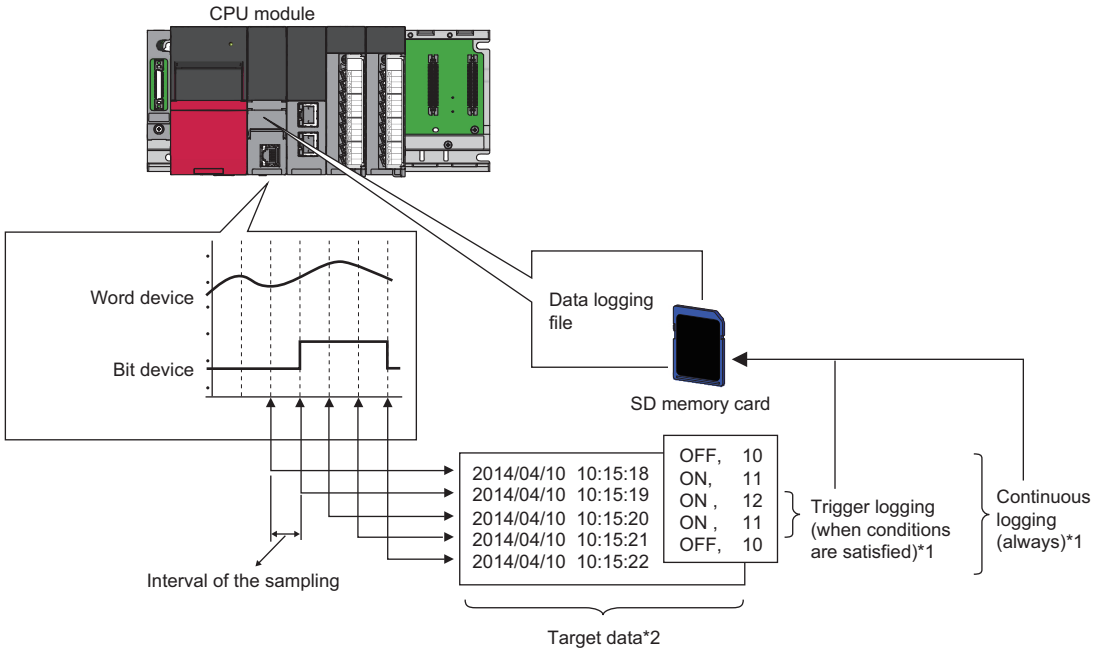
- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 387 FUNCTIONS

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

☞ Page 486 FUNCTIONS

This function collects data at a specified interval or a desired timing and stores them as a file on the SD memory card. The entire data logging function allows the entry of up to 10 data logging settings.



*1 For details on logging types, refer to the following. (☞ Page 194 Logging Type)

*2 For details on the data to be collected, refer to the following. (☞ Page 188 Data to Be Collected)

Point

For system configuration, specifications, the procedure for operating and configuring this function in the CPU Module Logging Configuration Tool, which are not covered here, refer to the CPU Module Logging Configuration Tool. (☞ Page 726 How to Use CPU Module Logging Configuration Tool)

23.1 Data to Be Collected

This section describes the data to be collected by data logging.

Number of data points

The data logging function can collect up to 1280 data records. (10 settings × 128 records)*1

*1 Duplicate data records are counted as distinct.

Data type

The following table shows the number of data records for each data type.

Data type	Data number of points
Bit	1
Word (signed)	1
Double word (signed)	2
Word (unsigned)	1
Double word (unsigned)	2
Single-precision real number	2
Double-precision real number	4
Time	2
String*1	Specified size/2*2
Numeric string	Specified size/2*2

*1 Outputs the entered character code.

*2 The specified size can be 1 to 256. If the specified size is an odd number, the number of data records is rounded to the next higher integer. Example: The number of data records is 3 if the specified size is 5.

Data to be collected

The data for the following devices can be specified to be collected.

Type	Device*1
Bit device*2	X, DX, Y, DY, M, L, F, SM, V, B, SB, T (contact)*4, T (coil)*4, ST (contact)*4, ST (coil)*4, C (contact)*4, C (coil)*4, LT (contact)*4, LT (coil)*4, LST (contact)*4, LST (coil)*4, LC (contact)*4, LC (coil)*4, FX, FY, Jn\X, Jn\Y, Jn\SB, Jn\B, BLn\S*5
Word device*3	T (current value), ST (current value), C (current value), D, SD, W, SW, RD, R, ZR, Z, FD, Un\G, Jn\W, Jn\SW, U3En\G, U3En\HG
Double-word device	LT (current value), LST (current value), LC (current value), LZ

*1 A local device, index modified device, and indirectly specified device cannot be specified.

*2 For bit devices, digit specification is allowed for K1 to K8 only.

*3 For word devices, bit specification is allowed.

*4 To specify these devices with the CPU Module Logging Configuration Tool, use T (contact): TS, T (coil): TC, ST (contact): STS, ST (coil): STC, C (contact): CS, C (coil): CC, LT (contact): LTS, LT (coil): LTC, LST (contact): LSTS, LST (coil): LSTC, LC (contact): LCS, and LC (coil): LCC.

*5 If collecting the items with the programmable controller CPU or Process CPU, check the versions of the CPU module and engineering tool. (☞ Page 779 Added and Enhanced Functions)
Note that the items cannot be collected with the Safety CPU.

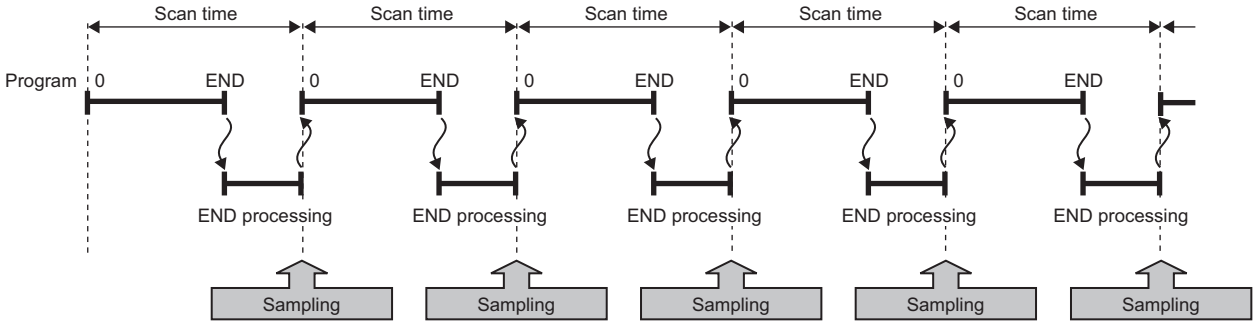
23.2 Data Collection Conditions

This section describes the timing when data is collected and the conditions under which data is collected.

Data collection conditions		Description
Each scan		Collects data during the END processing of each scan.
Time specification	Data collection at specified time interval	Collects data at specified time interval.
	Data collection during the END processing after specified time interval	Collects data during the END processing after specified time interval.
Interrupt occurrence		Collects data at specified time interval after the execution of an interrupt program.
Condition specification	Device specification	Collects data when the monitored data meets the specified condition during the END processing.
	Step No. specification	Collects data when the specified condition is met immediately before the execution of the specified step.

Each scan

Collects data during the END processing of each scan.



Time specification

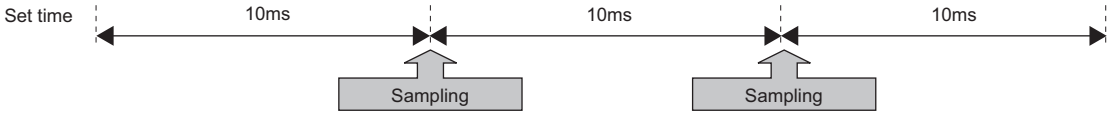
Lets the user specify the collection time interval. The user can also specify the timing of data collection using advanced settings.

Data collection at specified time interval

Collects data at specified time interval.

Ex.

10 milliseconds (10ms)

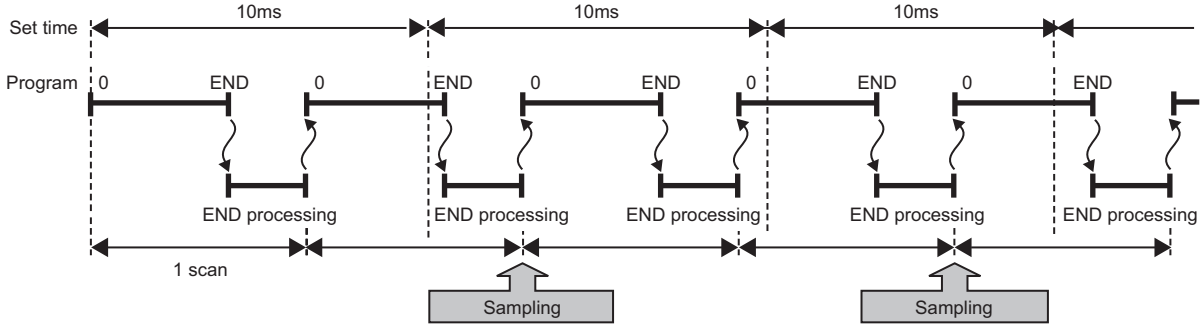


Data collection during the END processing after specified time interval

This option causes data collection to be perfumed at the timing of the END processing rather than during the course of program execution. Ensure that the "Scan time" is less than "Time specification." If the scan time is longer than the specified time and the collection interval or the collection timing occurs more than once during the same scan, data is collected only once during the END processing. Data collection is performed on a scan by scan basis, which is the same behavior as when "Each scan" is used.

Ex.

10 milliseconds (10ms)

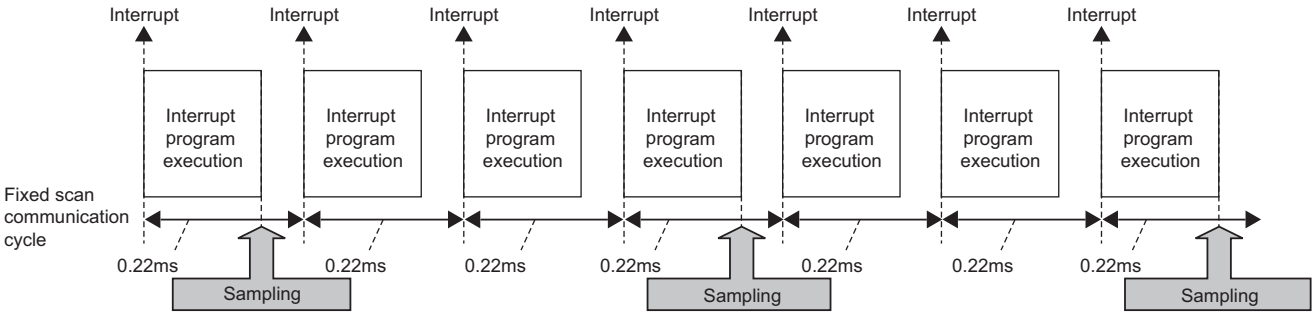


Interrupt occurrence

Collects data at specified time interval after the execution of an interrupt program. To specify when to collect data, use "Interrupt cycle specified sampling intervals"× "Interval." Interrupt pointers which can be specified are I28 to I31, I44, I45, I48, and I49.

Ex.

When multiple CPU synchronous interrupt (I45) is used (multiple CPU synchronous interrupt is set to 0.22 milliseconds (0.22ms) and interval to 3):



Point

Since data collection is performed at the timing of the execution of the interrupt program, data is collected only when all of the following conditions are met:

- The specified interrupt meets the operating condition.
- The specified interrupt pointer exists in the program.
- The current state is an EI state and the interrupt mask of the interrupt pointer has been reset.

Condition specification

This option lets the user set the data collection timing by specifying the device data conditions and step number. The AND condition using a combination of "Device specification" and "Step No. specification" results in the collection of data at the time when both conditions are established.

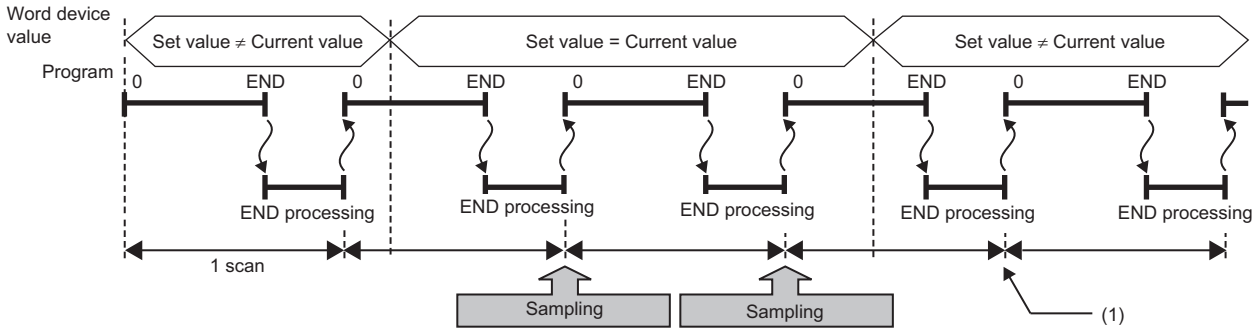
Device specification

Collects data when the monitored data meets the specified condition during the END processing.

■To collect data continuously while the conditions are met

The following conditional formula causes the data logging function to collect data continuously while the conditions are met:

- =: When the current value of the monitored data is equal to the comparison value
- ≠: When the current value of the monitored data is not equal to the comparison value
- ≥: When the current value of the monitored data is equal to or larger than the comparison value
- >: When the current value of the monitored data is larger than the comparison value
- ≤: When the current value of the monitored data is equal to or smaller than the comparison value
- <: When the current value of the monitored data is smaller than the comparison value

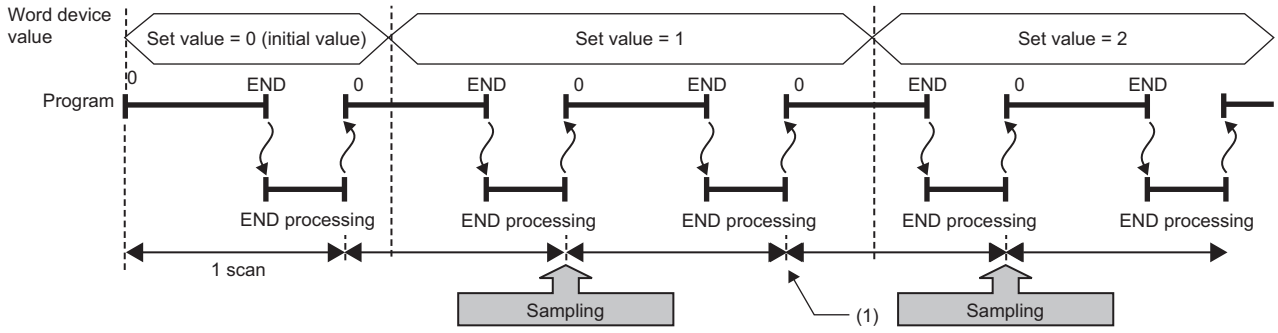


(1) During the END processing, the data logging function does not collect data because the conditions are not met.

■To collect data only when the state changes

The following conditional formula causes the data logging function to collect data only during the END processing for the scans where the conditional formula is met. It does not collect data for any single scan where the conditional formula is not met during the END processing (even if the conditional formula is met before the END processing is initiated).

- ↑: When the specified data turns off and on
- ↓: When the specified data turns on and off
- At change: When the current value of the specified data changes



(1) The data logging function does not collect data because there has been no change in state since the last scan.

■Specifying the monitored data

For monitored data, the following devices can be specified. The data types that can be selected include bit/word (unsigned), word (signed), double word (unsigned), and double word (signed).

Type	Device[*1]
Bit device[*2]	X, Y, M, L, F, SM, V, B, SB, T (contact)[*4], ST (contact)[*4], C (contact)[*4], LT (contact)[*4], LST (contact)[*4], LC (contact)[*4], FX, FY, Jn\X, Jn\Y, Jn\SB, Jn\B, BLn\S[*5]
Word device[*3]	T (current value), ST (current value), C (current value), D, SD, W, SW, RD, R, ZR, Z, FD, Un\G, Jn\W, Jn\SW, U3En\G, U3En\HG
Double-word device	LT (current value), LST (current value), LC (current value), LZ

*1 A local device, index modified device, and indirectly specified device cannot be specified.
*2 For bit devices, digit specification is allowed for K1 to K8 only.
*3 For word devices, bit specification is allowed.
*4 To specify these devices with the CPU Module Logging Configuration Tool, use T (contact): TS, ST (contact): STS, C (contact): CS, LT (contact): LTS, LST (contact): LSTS, and LC (contact): LCS.
*5 If specifying the items with the programmable controller CPU or Process CPU, check the versions of the CPU module and engineering tool. (☞ Page 779 Added and Enhanced Functions)
Note that the items cannot be specified with the Safety CPU.

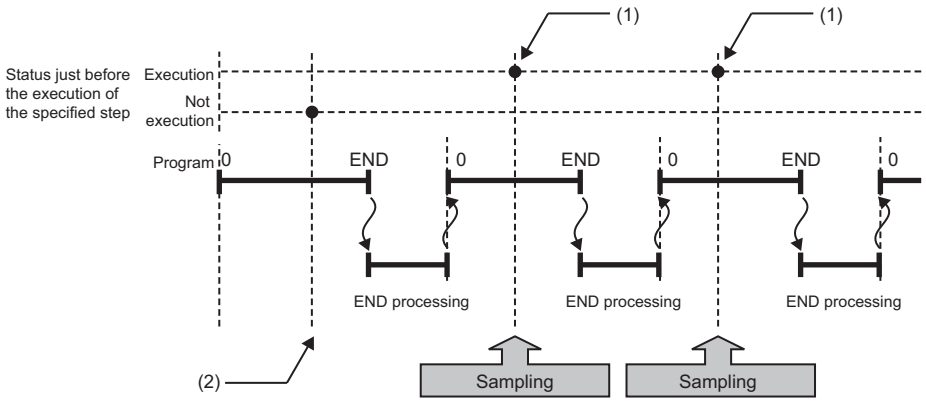
Step No. specification

Collects data when the specified condition is met immediately before the execution of the specified step.

■To collect data continuously while the execution conditions are met

The following execution conditions cause the data logging function to collect data continuously while the execution condition are met:

- Always: Executes the specified step regardless of the state immediately before the execution of it.
- In the specified condition satisfied: Executes the specified step if the state immediately before the execution is a running state.
- In the specified condition not satisfied: Executes the specified step if the state immediately before the execution is not a running state.

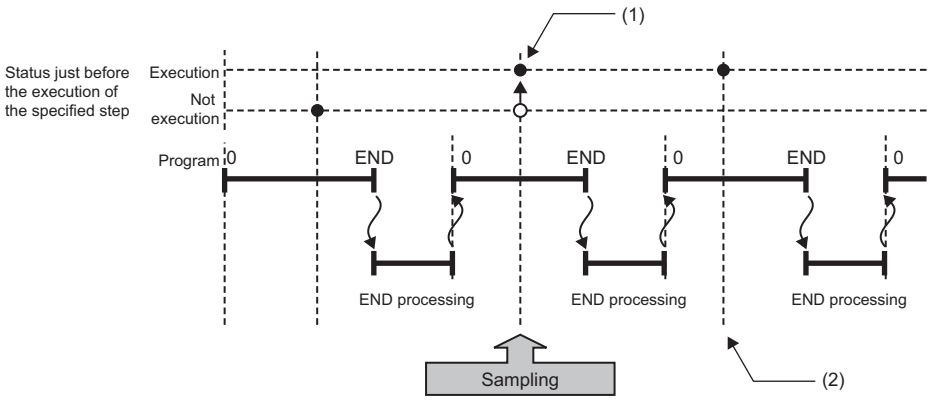


(1) The data logging function collects data because the state immediately before the execution of the specified step is a running state.
(2) It does not collect data because the state immediately before the execution of the specified step is not a running state.

■To collect data only when the execution conditions are met

The following execution conditions cause the data logging function to collect data only when the execution conditions are met:

- On the rising edge of the specified condition: Executes the specified step if the state changes from non-running to running immediately before the execution of it.
- On the falling edge of the specified condition: Executes the specified step if the state changes from running to non-running immediately before the execution of it.



(1) The data logging function collects data because the state changes from running to non-running immediately before the execution of the specified step.
(2) The data logging function does not collect data because there has been no change in state since the last scan.

Point

If the specified step is contained between FOR and NEXT loop of instructions, the data logging function collects the data for only the first iteration of the loop where the specified conditions are met.

Restriction

Since the step number cannot be checked in the following programs, the step number cannot be specified.

- Program having multiple parts
- ST program
- FB program
- SFC program

23.3 Logging Type

The following table describes available methods of data collection:

Logging type	Data collection method	Application
Continuous logging	Continuously collects specified data at specified interval or timing.	Allows the user to continuously monitor the content of specified data.
Trigger logging	Collects specified data at specified interval or timing and extracts a specified number of data records before and after the satisfaction of a trigger condition.	Allows the user to monitor the content of specified data before and after the satisfaction of a trigger condition.

Operating procedure for continuous logging

In continuous logging, the CPU module stores specified data in its internal buffer at a specified collection interval or timing and, at the time of a file save operation, it saves the data in a data logging file residing in the storage memory.

1. Write the settings into data memory or SD memory card using the CPU Module Logging Configuration Tool.
2. Instruct the CPU Module Logging Configuration Tool to start data logging. The data logging settings are registered and continuous logging begins. (The special relay (data logging start) turns on.)
3. Data collection finishes upon reaching "Number of files to be saved" specified as part of the "Stop" setting configured in "Operation when the number of files exceeds the set value."
4. Specify the desired file in the storage memory to read the results of data logging.

Point

In continuous logging, data logging files are continuously created, thus allowing the user to read the results of data logging any time without having to wait for the completion of collection.

To stop continuous logging

The user can completely stop data logging by instructing the CPU Module Logging Configuration Tool to stop data logging and unregister the data logging settings stored in the CPU module. (The special relay (data logging start) turns off.)

To suspend/resume continuous logging

The user can suspend data logging with the data logging settings remaining intact by doing either of the following:

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

To resume continuous logging from suspension, do either of the following:

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

Operating procedure for trigger logging

In trigger logging, the CPU module stores specified data in its internal buffer at a specified collection interval or timing; it extracts a specified number of data records before and after the satisfaction of a trigger condition and saves the extracted data in a data logging file residing in the storage memory. Note that data collection is performed not only at the specified collection interval or timing but also when a trigger condition is met. In addition, once a trigger condition is met, any subsequent trigger conditions are ignored.

1. Write the settings into data memory or SD memory card using the CPU Module Logging Configuration Tool.
2. Instruct the CPU Module Logging Configuration Tool to start data logging. The data logging settings are registered and trigger logging begins. (The special relay (data logging start) turns on.)
3. Wait until the trigger condition is met. (Trigger standby)
4. The data specified in CPU Module Logging Configuration Tool is collected. (Trigger condition met)
5. Data collection is completed by collecting as much data as the number of records specified in the CPU Module Logging Configuration Tool and writing the collected data into the storage memory.
6. Specify the desired file in the storage memory to read the results of data logging.

To stop trigger logging

The user can completely stop data logging by instructing the CPU Module Logging Configuration Tool to stop data logging and unregister the data logging settings stored in the CPU module. (The special relay (data logging start) turns off.)

To suspend/resume trigger logging

The user can suspend data logging with the data logging settings remaining intact by doing either of the following:

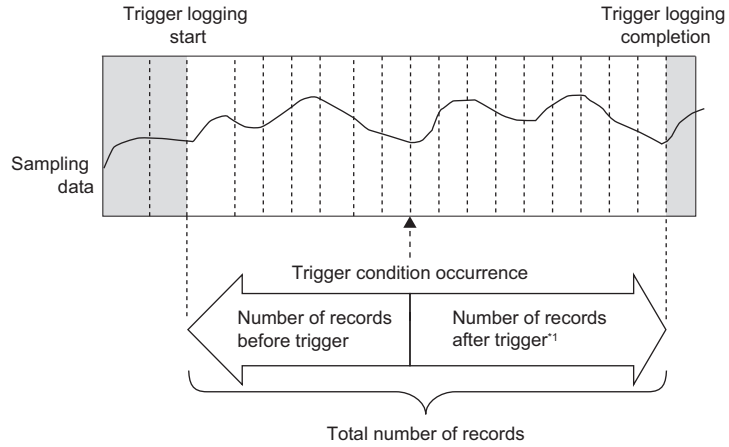
- Instruct the CPU Module Logging Configuration Tool to suspend data logging (the special relay (data logging start) turns off).
- Turn off and on the special relay (Data logging suspend/resume flag).

To resume trigger logging from suspension, do either of the following:

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

Number of records

Specify the number of records to be collected before and after the satisfaction of a trigger condition. (☞ Page 751 Number of records)



*1 This number includes the record exactly at the time when the trigger condition is met.

Point

After starting data logging, if the trigger condition is met before data collection of the specified number of records (before trigger) is completed, the number of sampled records will be less than the specified.

23.4 Trigger Condition

The following table lists the conditions to be used as a trigger.

Trigger condition		Description
Condition specification	Device change specification	A trigger occurs when the monitored data meets the specified condition.
	Step No. specification	A trigger occurs when the specified condition is met immediately before the execution of the specified step.
When trigger instruction executed		A trigger occurs when the LOGTRG instruction is executed.

Condition specification

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

Device change specification

A trigger occurs when the monitored data meets the specified condition.

- ↑: When the specified data turns off and on
- ↓: When the specified data turns on and off
- =: When the monitored data is equal to the comparison value, regardless of whether or not its current value is equal.
- ≠: When the monitored data is not equal to the comparison value, regardless of whether or not its current value is equal.
- ≥: When the monitored data is equal to or larger than the comparison value, regardless of whether or not its current value is equal.
- >: When the monitored data is larger than the comparison value, regardless of whether or not its current value is equal.
- ≤: When the monitored data is equal to or smaller than the comparison value, regardless of whether or not its current value is equal.
- <: When the monitored data is smaller than the comparison value, regardless of whether or not its current value is equal.
- At change: When the current value of the specified data changes

■Specifying the monitored data

For the device change specification, monitored data can be configured to be collected from the devices listed in the following table. The data types that can be selected include bit/word (unsigned), word (signed), double word (unsigned), and double word (signed). If double word (unsigned) or double word (signed) is specified, a trigger occurs only when as much data as one double word is written. No trigger occurs when only the upper or lower word of a double word is written.

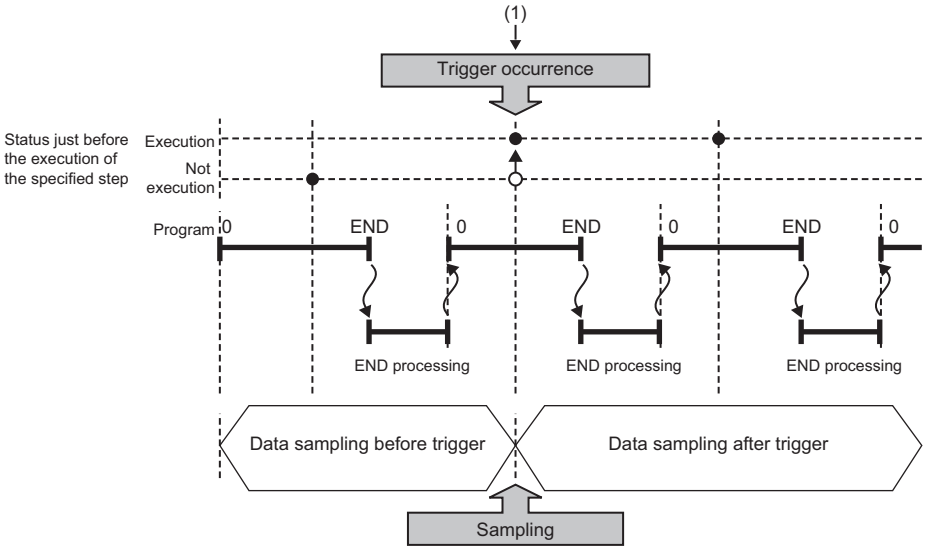
Type	Device*1
Bit device*2	X, Y, M, L, F, SM, V, B, SB, T (contact)*4, ST (contact)*4, C (contact)*4, LT (contact)*4, LST (contact)*4, LC (contact)*4, FX, FY
Word device*3	T (Current value), ST (Current value), C (Current value), D, SD, W, SW, RD, R, ZR, FD
Double-word device	LT (Current value), LST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

*1 A local device, index modified device, and indirectly specified device cannot be specified.
*2 For bit devices, digit specification is not supported.
*3 For word devices, bit specification is allowed.
*4 To specify these devices with the CPU Module Logging Configuration Tool, use T (contact): TS, ST (contact): STS, C (contact): CS, LT (contact): LTS, LST (contact): LSTS, and LC (contact): LCS.

Step No. specification

A trigger occurs when the specified condition is met immediately before the execution of the specified step.

- Always: Executes the specified step regardless of the state immediately before the execution of it.
- In the specified condition satisfied: Executes the specified step if the state immediately before the execution is a running state.
- In the specified condition not satisfied: Executes the specified step if the state immediately before the execution is not a running state.
- On the rising edge of the specified condition: Executes the specified step if the state changes from non-running to running immediately before the execution of it.
- On the falling edge of the specified condition: Executes the specified step if the state changes from running to non-running immediately before the execution of it.



(1) The state changes from non-running to running immediately before the execution of the specified step.

Point

If the specified step is contained between FOR and NEXT loop of instructions, only the first iteration of the loop where the specified conditions are met is handled as a trigger.

Restriction

Since the step number cannot be checked in the following programs, the step number cannot be specified.

- Program having multiple parts
- ST program
- FB program
- SFC program

When trigger instruction executed

A trigger occurs when the LOGTRG instruction is executed. (MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks))

23.5 States of the Data Logging Function

The data logging function has various states that can be classified into data logging and storage. The data logging state and storage state can be checked by CPU Module Logging Configuration Tool. (☞ Page 762 Logging status and operation)

Data logging states

The following table lists all the possible data logging states.

Data logging states	Description
Stop	No data logging settings are registered and data collection is inactive.
Stop (Execute other functions after completing collection)	Transition from "Collection completed" to "Stop" has occurred due to the execution of another function*1.
Stop (Execute other functions after error)	Transition from "Error" to "Stop" has occurred due to the execution of another function*1.
Waiting RUN Not collected	Data collection has not yet begun because the CPU module is not in the RUN mode.
Waiting start Not collected	Data collection is inactive, waiting for the start command.
Pause	Data logging is suspended and data collection is inactive.
Waiting to establish collection conditions Not collected	Waiting for the first collection timing after the start command.
Collecting	Continuous logging is active and collecting data.
Waiting trigger Collecting before trigger	Trigger logging is active and collecting data, waiting until the trigger condition is met.
Collecting after trigger	Trigger logging is active and collecting data after the trigger condition is met.
Collection completed	• Continuous logging: Data collection has finished upon reaching "Number of files to be saved" specified as part of the "Stop" setting configured in "Operation when the number of files exceeds the set value." • Trigger logging: Has finished collecting as much data as the specified number of records.
Error	Data logging has failed due to the occurrence of an error.

*1 The execution of another function includes:
· Execution of data logging with the same trigger conditions (trigger conditions = the specified conditions)
· Auto logging
· Online change

Storage states

The following table lists all the possible storage states.

Storage states	Description
Unsaved	Has not yet stored the collected data into the SD memory card.
Saving in progress	Has begun but not yet finished storing the collected data into the SD memory card.
Save completed	Has finished storing the collected data as much as the specified number of records into the SD memory card.*1

*1 If the data logging function has not yet collected and stored as much data as the specified number of records (i.e., either data logging has been stopped or suspended before collecting or storing as the specified number of records or the CPU module has been stopped), it completes the storage operation by storing all the data that has been collected into the internal buffer. It does not store data, however, before the trigger condition is met.

LED status

Whether the data logging function is active or not can be checked by the LED of the CPU module.

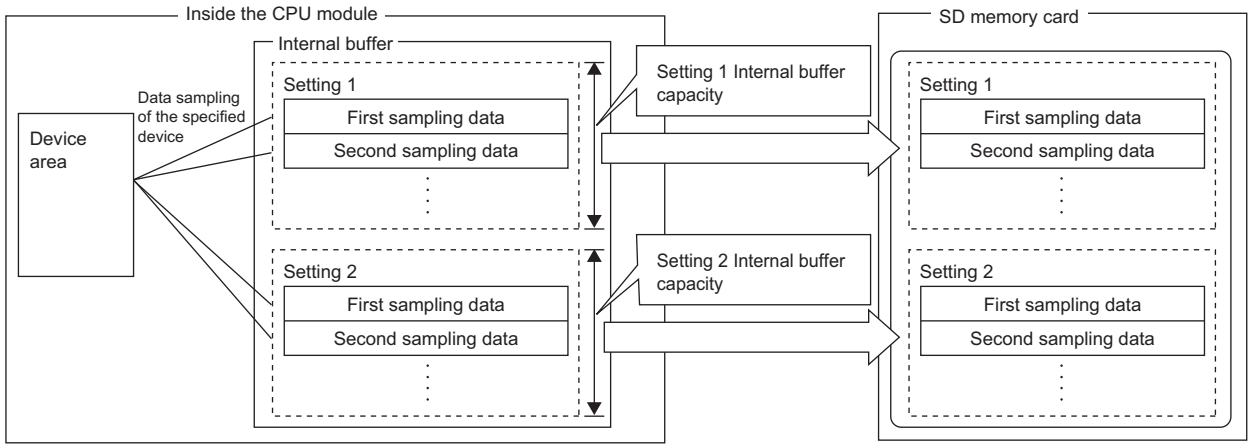
States of the Data Logging Function		LED status
At registration	The CPU module is now populated with an SD card that contains the settings to enable the auto logging function after the user instructs the CPU Module Logging Configuration Tool to start data logging.	• CARD READY LED: On • CARD ACCESS LED: Off • FUNCTION LED: On
Execution in progress	All of the registered data logging sessions are in the state "Waiting RUN Not collected", "Waiting start Not collected", "Pause", "Waiting to establish collection conditions Not collected", or "Waiting trigger Collecting before trigger".	
	One or more of the registered data logging sessions are in the state "Collecting" (including the data being saved) or "Collecting after trigger" (including the data being saved).	• CARD READY LED: On • CARD ACCESS LED: Turns on when the SD memory card is accessed • FUNCTION LED: Flashes slowly (every one second)
At completion	All of the registered data logging sessions have finished (or failed due to an error).	• CARD READY LED: On • CARD ACCESS LED: Off • FUNCTION LED: Flashes at normal rate (every 200 milliseconds)

Point

When the external input/output forced on/off function is executed (in registration), the LED indicator follows the status of the external input/output forced on/off function. The display of the FUNCTION LED switches regardless of "LED Display Setting" in "RAS Setting" of the CPU parameter. After the external input/output forced on/off function has been executed (after the registration is canceled), the LED indicator follows the status of the data logging function. (☞ Page 232 LED display setting)

23.6 Steps Until the Collected Data Is Saved

The collected data is temporarily stored in the specified internal buffer. (☞ Page 203 Internal Buffer) The data stored in the internal buffer is stored into the SD memory card at the time of a file save operation.



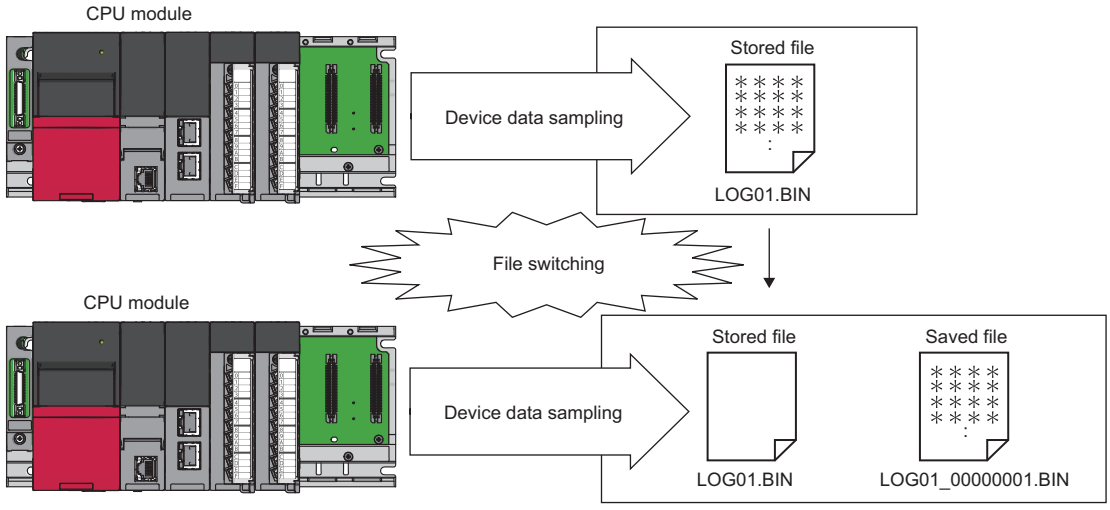
Switching to a storage file

The data collected by data logging is temporarily stored in a stack file that resides in the SD memory card. The stack file can be switched to a storage file to free the space in the SD memory card.

How file switching (log rotation) works

File switching works as follows:

1. The CPU module writes collected data into a stack file (such as LOG01.BIN).
2. It changes the file name when the storage file switching condition is met.*1*2
3. It creates a new stack file.
4. It continues to write collected data into the newly created stack file.



*1 The file name format can be customized. (☞ Page 754 Save)

*2 The file number of the most recent storage file is stored in the special register (Latest storage file number).

File switching condition

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

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

*1 File switching occurs before the file grows beyond the specified size.
However, file switching occurs regardless of the setting when:

- the number of records has reached 65500;
- the file size has reached 16MB;
- the CPU module is stopped or suspended/resumed: or
- data logging is started and there is an existing stack file.

Point

Trigger logging does not require the configuration of these settings because the stack file is automatically switched to a storage file after as much data as the specified number of records is written into the stack file. SM1218 (logging data storage file switching in progress) can be used to check if storage file switching is in progress.

Storage file

The CPU module creates a subfolder ("storage file container folder") under the file storage folder and writes storage files to that storage file container folder. One storage file container folder can contain up to 256 storage files. When the files contained in the current storage file container folder reach the maximum number, the CPU module creates a new storage file container folder at the time of next storage file switching and begins writing storage files to that new folder. The number of files that can be contained in one file storage folder is configurable within the range of 1 to 65535.

Point

The base folder name of a storage file container folder is an eight-digit (hexadecimal) number. This number matches the lowest of the serial numbers of the files contained in the directory. Date and time stamps can be appended to the folder name.

Action to take when the maximum number of storage files is exceeded

Either "Overwrite" or "Stop"*1 can be selected as the action to take when the maximum number of storage files is exceeded.

*1 This settings is not configurable for trigger logging.

■When "Overwrite" is selected

When the storage file switching condition is met after the specified maximum number of storage files is exceeded, the CPU module deletes the file with the lowest serial number and creates a new file that has a serial number incremented by one from the highest serial number, allowing data logging to continue. In addition, if deleting the file with the lowest serial number results in an empty folder, the CPU module deletes that folder as well.

■When "Stop" is selected

As described in the following table, the action differs depending on when the specified maximum number of storage files is exceeded.

Occurrence timing	Occurrence condition	Behavior
When data logging is started	There exist more storage files than the specified maximum number when data logging is started.	• If an attempt is made to register the data logging settings from within the CPU Module Logging Configuration Tool, an error is generated, resulting in failure to run data logging. • If an attempt is made to register[*1] the data logging settings from outside the CPU Module Logging Configuration Tool, the special relay (data logging error) turns on and the special register (data logging error cause) stores the cause of the error, resulting in failure to run data logging.
While data logging is running	The specified maximum number of storage files is reached due to file switching upon the satisfaction of the storage file switching condition.	Data logging stops and enters into the completion state. The special relay (Data logging end) turns on to indicate that data logging is completed.

*1 When an attempt is made to register the data logging settings again, the CPU module enters into the data logging completed state. The special relay (Data logging end) turns on to indicate that data logging is completed.

23.7 Internal Buffer

The internal buffer is a system area used to temporarily store collected data.

Internal buffer capacity setting

This setting can be specified using an engineering tool (☞ Page 156 INTERNAL BUFFER CAPACITY SETTING). For trigger logging, increasing the internal buffer capacity allows for a larger number of data records collected before a trigger and also helps to prevent processing overflow. If the free space in the internal buffer is still insufficient after increasing the internal buffer capacity, use the following workarounds:

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

Amount of internal buffer consumed

This value can be calculated by multiplying "Number of data points" by 2 bytes. Note, however, that additional space is consumed by columns configured for output, as indicated below:

- Date/time column: 10 bytes
- Data collection interval column: 8 bytes
- Execution step No. column: 10 bytes
- Execution program No. column: 2 bytes
- Index column: 4 bytes

Ex.

When data logging is configured to collect as much data as one setting x 128 records and output all of the columns (i.e., maximum allowable configuration):

128 × 2 + (10 + 8 + 10 + 2 + 4) = 290 bytes

23.8 Data Logging File

This section describes data logging files.

Storage format of data logging files

The following storage formats are available for data logging files. For details on the output format, format specifications, and output contents of each file, refer to the data output format. (☞ Page 731 Data output type)

File format	Application
Unicode text file format	This is a file format which can be open in generic-purpose application programs such as Excel and Notepad. GX LogViewer is also available for displaying data.
Binary file format	Comparing the Unicode text file format, the size of files is small and therefore quicker access to files is provided. GX LogViewer is also available for displaying data.

23.9 Missing Data

The term "missing data" means that some of the collected data is missing, resulting in data discontinuity.

Conditions under which missing data occurs

Missing data occurs under the following conditions:

Item	Description
Processing overflow	Processing overflow has occurred due to failure to keep up with the specified collection interval/timing.*1
Operations for the CPU module	The CPU module has been stopped and run with "Operation at transition to RUN" set to "Auto Start."
	The CPU module has been turned off and on with "Operation at transition to RUN" set to "Auto Start."
	The CPU module has been reset and run with "Operation at transition to RUN" set to "Auto Start."
Operation from engineering tools, CPU Module Logging Configuration Tool, and external devices via protocols such as FTP, SLMP, and MC	• When the CPU module is suspended and restarted, and operation for displaying the logging state is performed from CPU Module Logging Configuration Tool • File read*2, write, delete, or verification

*1 Failure to collect data at the specified collection interval/timing due to the execution of a long-running instruction (such as FMOV) does not cause processing overflow or missing data.

*2 The following operation also is included:
· Online operation which displays data by operation such as read from the programmable controller performed from an engineering tool (retrieval and display of a list of files on the CPU module)
· View of the event history (retrieval of the event history from the CPU module)

Processing overflow

In normal cases when the usage of the internal buffer reaches the specified maximum capacity, the CPU module overwrites the data stored in the storage memory on a first-in first-out basis. If the internal buffer becomes full before all of the data stored in it is saved to the storage memory, however, the CPU module does not overwrite the existing data and stops storing data in the internal buffer, thus resulting in missing data. This situation is referred to as processing overflow. Upon the occurrence overflow, the special register (Number of processing overflow occurrences) stores the number of times when processing overflow occurred.

23.10 Errors Generated During Data Logging

No diagnostic error occurs if an error occurs during data logging, the SM applicable to the special relay (data logging error) setting No. turns on, and the error cause is stored in the SD applicable to the special register (data logging error cause) setting No.

23.11 Special Relay and Special Register Used by the Data Logging Function

For details on the special relays and special registers used by the data logging function, refer to the following:

- Special relay: Special relay related with the data logging function (☞ Page 622 Data logging function)
- Special register: Special register related with the data logging function (☞ Page 661 Data logging function)

23.12 Other Functions

This section describes some useful functions available in addition to the basic features of the data logging function.

Setting the behavior at the time of a transition to RUN

This function configures the behavior of data logging that occurs when the user performs the following operations (transition to RUN) after the data logging setting are registered. (☞ Page 756 Logging operation)

- Turning off and on the CPU module and switching to the RUN mode
- Resetting and running the CPU module
- Stopping and running the CPU module

Point

The behavior at the time of a transition to RUN can be set individually for each setting number (1 to 10).

Behavior at the time of a transition to RUN

The behavior at the time of a transition to RUN can be set to either of the following:

■Auto start

After the user performs one of the operations listed above, data logging automatically starts when the operating status of the CPU module changes from STOP to RUN.

Point

To start data logging, first, the user must instruct the CPU Module Logging Configuration Tool to start data logging.

■Start by user operation

After the user performs one of the operations listed above, the data logging state is switched to "Waiting start Not collected" when the operating status of the CPU module changes from STOP to RUN. To start data logging again, the user must instruct the CPU Module Logging Configuration Tool to start data logging.

Data logging behavior that occurs after operating status of CPU module has changed

Data logging does not continue when the operating state of the CPU module changes from RUN to STOP or PAUSE after it has been started. The data logging state changes to "Waiting RUN Not collected" and data collection is stopped.

Auto logging

When inserting an SD memory card, which holds data logging setting, into the CPU module, the data logging automatically starts based on the data logging setting information on the SD memory card.

How to use auto logging

This section describes how to use auto logging.

1. Prepare an SD memory card that contains data logging settings as well as common settings (auto logging common settings) in which auto logging is enabled.
2. Insert the SD memory card prepared in step 1 into the CPU module while it is running.

Conditions for auto logging completion

Auto logging completes when a completion condition is met as described in the following table. These completion conditions can also be configured in combination with each other. When they are configured in combination, auto logging completes as soon as one of the conditions is met.

Completion condition	Description
Data logging stop	Choose one of the following: • When all data loggings stop • When any of the data loggings stops
Complete with timer	Auto logging is stopped when the specified time has elapsed after the start of data logging. Elapsed time setting range: 1 second to 86400 seconds (unit: second)

■When "Data logging stop" is selected

Auto logging is assumed to be completed if all the settings or any of auto logging stops[*1].

*1 Here the term "stop" means one of the following:
· Continuous logging: When the "Number of files to be saved" setting configured in the save setting has been exceeded and data logging has been completed.
· Trigger logging: When as much data as the number of records specified in "Number of records" has been collected, the collected data has been written to the SD memory card, and data logging has been completed.
· When the user has instructed the CPU Module Logging Configuration Tool to stop data logging.

Point

- If "When any of the data loggings stops" is selected, the behavior of any other data logging sessions than stopped is the same as the behavior that occurs upon the elapse of the time configured using "Complete with timer."
- When completing auto logging for continuous logging, do not select "Overwrite" for the operation at the time when the number of files exceeds the save setting because doing so results in failure to stop data logging; instead, select "Stop."

■When "Complete with timer" is selected

When the operating time since the start of data logging reaches the specified time, the CPU module completes auto logging by moving all the data collected so far from the internal buffer to the SD memory card. If trigger logging has not yet collected as much data as the number of records specified "Number of logging lines", however, the CPU module does not store any data including the collected data.

Point

The timer is cleared to zero when auto logging is suspended and resumed by stopping and running the CPU module or turning off and on the CPU module or resetting it before the completion of auto logging. If auto logging is suspended by the turning off and on of the special relay (Data logging suspend/resume flag), the timer continues to run even while data logging is suspended.

Conditions under which auto logging does not start

Doing any of the following operations once auto logging is completed does not start auto logging:

- Turn off and on the power
- Reset
- STOP to RUN state

Behavior of auto logging that occurs when entering into the RUN mode

If some, not all, of data logging sessions configured using auto logging have been completed when entering into the RUN mode, only the uncompleted data logging sessions are started.

Behavior of auto logging that occurs when an error is generated

When an error is generated, auto logging behaves as follows:

■When registration has failed at the start of auto logging

If auto logging cannot be started due to failure of registration at the start of auto logging, it behaves in the same way as when it is completed.

■When an error has occurred during the execution of auto logging

An error that occurs during the execution of auto logging does not constitute the auto logging completion condition since it does not prevent data logging from being resumed. If such an error is resulting from an online change during the execution of auto logging, however, it constitutes the auto logging completion condition because it prevents data logging from being resumed.*1

*1 Applies only when "When all data loggings stop" is selected as part of the auto logging completion condition.

SD memory card replacement

SD memory cards can be replaced using the SD memory card forced disable function even while data logging is in progress. (☞ Page 174 SD Memory Card Forced Disable) This function works by disabling data writes to the SD memory card while allowing data collection to continue. (Data collection continues in accordance with the settings registered when data logging is started.)

Point

If SD memory card replacement causes processing overflow, make adjustments by changing the collection interval, internal buffer capacity, or other settings.

Behavior during SD memory card replacement

If the internal buffer becomes full during the time between SD memory card replacement and the resumption of data writes to the SD memory card, processing overflow occurs resulting in missing data.

Storage file numbers after SD memory card replacement

The numbering of the first storage file created after SD memory card replacement differs depending on the storage file switching condition, as described in the following table.

Storage file switching condition	Storage file numbers after SD memory card replacement
Overwrite	Numbering continues from the number of the last storage file contained in the replaced SD memory card.
Stop	Numbering begins at 00000001.

Point

If the new SD memory card contains a "LOGGING" folder and its subfolders, folder deletion takes time, possibly resulting in missing data. Ensure that the new SD memory card does not contain a "LOGGING" folder.

Logging state during SD memory card replacement

SD memory cards can be replaced without depending on the current data logging state. SD memory card replacement results in the deletion of the LOGGING folder if the data logging state is other than below:

- Stop
- Stop (Execute other functions after completing collection)
- Stop (Execute other functions after error)

Operations during SD memory card replacement

If one of the following operations is performed during the time between the removal and installation of SD memory cards, any data collected during that time will not be stored in the new SD memory card.

- Stop and run*1
- Power off and on*1
- Reset*1
- Suspend data logging
- Stop data logging

*1 An error is generated if data logging was previously running based on the setting file contained in the replaced SD memory card.

Operations after SD memory card replacement

If the SD memory card was replaced while data logging was running based on the data logging setting file contained in the SD memory card, the data logging setting file contained in the new SD memory card is used when data logging is started by one of the following operations. If the new SD memory card does not contain the data logging setting file, data logging is not started.

- STOP to RUN state
- Power on and run
- Reset and run
- Data logging start

Stack file remaining in the replaced SD memory card

Replacing an SD memory card that contains a stack file may result in the storage file remaining in the replaced SD memory card along with storage files. If the stack file is remaining in the replaced SD memory card, recover the latest data contained in the stack file by doing the following:

- Retrieve the data from the stack file and combine the data with a storage file.
- Save the stack file as a storage file.

23.13 SD Memory Card Life When the Data Logging Function Is Used

An SD memory card has a life (restriction on writing data). The following shows the calculation method of an SD memory card life when the data logging function is used. Note that the actual life of the card varies depending on the use conditions and environment. Therefore, use the calculated life as a rough standard for the replacement of the card.

Calculation formula of SD memory card life

SD memory card life (year) = Total size of data that can be written (G bytes) ÷ Size of data to be written per year (G bytes/year)

Total size of data that can be written

Capacity × Number of writes[*1]

*1 For the capacity of applicable SD memory cards and the number of writes, refer to the following.
MELSEC iQ-R Module Configuration Manual

Size of data to be written per year

The size of data to be written per year is obtained by the following formula.

Size of data to be written per year (G bytes/year) = ((DS1[*1] + 6144) × DN1 + ··· + (DSn[*1] + 6144) × DNn + (DCS1[*1] + 6144) × DCN1 + ··· + (DCSn[*1] + 6144) × DCNn) ÷ 1073741824

*1 Round up DSn and DCSn to a multiple of 512.

DSn, DNn, DCSn, and DCNn are obtained as follows.

■Data logging data size per record (DSn)

Binary file output format: Refer to the data. (☞ Page 736 Binary file output format) Unicode text file format: Refer to the data row. (☞ Page 731 Unicode text file output type)

■Number of records for data logging per year (DNn)

Continuous logging: DNn = 60 × 60 × 24 × 365 ÷ Collection interval and timing (seconds)[*1] × Operating rate[*2]

Trigger logging: DNn = Total number of records[*3]

*1 The value that is determined depending on the condition set in "Sampling" when "Continuous logging" is selected for the logging type. (When the value is determined in milliseconds, convert the value into seconds.)

*2 Calculate the ratio using the operating time per year of the CPU module. For example, if the operating time per year is 5000 hours, the operating rate is calculated as follows: 5000 ÷ (24 × 365) = 0.57.

*3 The value set in "Number of logging lines" when "Trigger logging" is selected for the logging type.

■Header size of data logging (DCSn)

Binary file output format: Refer to the header. (☞ Page 736 Binary file output format) Unicode text file format: Refer to the file information row to device comment row. (☞ Page 731 Unicode text file output type)

■Number of file switching times for the data logging per year (DCNn)

Calculate this number with an estimated number according to the save setting of the data logging and system operations. For example, when 1000 records are set in "Number of records" of "File switching timing" in the save setting and "Each scanning cycle" is specified for "Sampling interval" in the sampling setting, the time interval of the file switching is obtained by multiplying the scan time by 1000. Therefore, the number of file switching times for the data logging per year is obtained by the following formula: 60 × 60 × 24 × 365 ÷ (Scan time (second) × 1000)

23.14 Precautions to Take When Using the Data Logging Function

This section describes precautions to take when using the data logging function.

Mutual exclusion of the data logging function

This section describes the mutual exclusion of the data logging function.

■When another function is executed during the execution of the data logging function

The following table lists the cases when another function is executed during the execution of the data logging function.

Function that has been already executed	Function to be executed later	Behavior
Data logging function	Data logging function	When the data logging is started using the CPU Module Logging Configuration Tool to the same data logging setting number via another route, the data logging to be executed later cannot be executed. However, the data logging to be executed later can be executed to a data logging setting number different from the data logging setting number currently being executed.
		The data conditions cannot be specified as the trigger conditions for multiple trigger conditions at the same time.
		For the execution of multiple data loggings, the data logging settings stored in different target memory areas cannot be executed at the same time.
	Auto logging	The auto logging cannot be executed during the execution of the data logging. (Even though an SD memory card where the auto logging setting is written is inserted, the auto logging does not start.)
	Online change	When a step number is set in the condition specification of the collection or trigger condition, the data logging function and online change may not be executed at the same time.
	CPU module data backup function	The CPU module data backup function cannot be executed while a logging setting file is being written/deleted or a logging setting is being registered/cleared.
	CPU module data restoration function	The CPU module data restoration function cannot be executed while a logging setting file is being written/read/deleted or a logging setting is being registered/cleared.
	iQ Sensor Solution data backup/restoration function	The iQ Sensor Solution data backup/restoration function cannot be executed while a logging setting file is being written/deleted or a logging setting is being registered/cleared.
	Function specified in the internal buffer capacity setting*1	If the internal buffer capacity setting is changed to execute the subsequent function, attempting to start the subsequent function results in an error. The data logging continues to function normally.
	Function not specified in the internal buffer capacity setting	If the condition "Total capacity that is set in the internal buffer capacity setting + Internal buffer capacity that is set in other than the internal buffer capacity setting > 3072K bytes" is satisfied, attempting to start the subsequent function results in an error. The data logging continues to function normally.
		If the internal buffer capacity setting is changed to execute the subsequent function, attempting to start the subsequent function results in an error. The data logging continues to function normally.
Auto logging	Data logging function	Another data logging cannot be executed during the execution of the auto logging.

*1 The data logging function is not included here.

■When the data logging function is executed during the execution of another function

The following table lists the cases when the data logging function is executed during the execution of another function.

Function that has been already executed	Function to be executed later	Behavior
Online change	Data logging function	When a step number is set in the condition specification of the collection or trigger condition, the data logging function and online change may not be executed at the same time.
CPU module data backup function		While the CPU module data backup function is being executed, a logging setting file cannot be written/deleted or a logging setting cannot be registered/cleared.
CPU module data restoration function		While the CPU module data restoration function is being executed, a logging setting file cannot be written/read/deleted or a logging setting cannot be registered/cleared.
iQ Sensor Solution data backup/ restoration function		While the iQ Sensor Solution data backup/restoration function is being executed, a logging setting file cannot be written/deleted or a logging setting cannot be registered/cleared.
Function specified in the internal buffer capacity setting*1		If the internal buffer capacity setting is changed to execute data logging, attempting to start data logging results in an error. The function already in execution continues to function normally.
Function not specified in the internal buffer capacity setting		If the internal buffer capacity setting is changed to execute data logging, attempting to start data logging results in an error. The function already in execution continues to function normally.

*1 The data logging function is not included here.

Locations from which data logging can be performed

Data logging cannot be performed from multiple locations to the same setting number. The CPU module supports data logging performed concurrently at a maximum of 10 locations assigned to setting numbers 1 to 10.

Retention and clearance of data logging settings

The data logging settings registered in the CPU module are latched and thus survive across a power cycle (power off and on) or reset of the CPU module in normal cases. In the following cases, however, the data logging status is cleared to the unregistered state and therefore the setting data must be written again:

- The CPU module is turned off and on or is reset without an SD memory card that contains the data logging setting file.
- The replaced SD memory card does not contain the data logging setting file and the CPU module is turned off and on or is reset.*1

*1 If the data logging setting file contained in the replacement (new) SD memory card is different from that contained in the replaced (old) SD memory card, register the data logging setting file contained in the replacement SD memory card.

Behavior that occurs when trigger logging is resumed

If data logging is stopped or collection is suspended before the completion of trigger logging and subsequently data logging is run again, data collection begins from the initial state before trigger logging, rather than continuing from the last time.

Stopping/suspending data logging using CPU Module Logging Configuration Tool

After data logging is stopped or suspended from within the CPU Module Logging Configuration Tool, all the data in the internal buffer is written into the target memory. If a small number of records or a small file size is specified as part of the storage file switching condition, writes to the target memory may take a longer time.

Behavior against the failure to register data logging setting files

When an attempt to register multiple data logging setting files at the same time is made and fails for some of them, the CPU module behaves differently depending on whether the registration is via the CPU Module Logging Configuration Tool and program or via auto logging:

- If the registration is via the CPU Module Logging Configuration Tool and program, the CPU module runs data logging for the setting files that have been successfully registered.
- If the registration is via auto logging, the CPU module does not run any data logging session.

Behavior upon change of the internal buffer capacity

When the internal buffer capacity is changed during execution of the data logging function, please note that:

- If the internal buffer capacity for the setting number of the running data logging is left empty to disable the capacity, an error occurs when the data logging is stopped and restarted (write to the programmable controller does not cause an error).
- If the internal buffer capacity of the setting number of the running data logging is changed to a smaller value, data may be lost when the data logging is stopped and restarted.

Trigger condition should not be met during data logging registration

Ensure that the trigger condition is not met during data logging registration. If the trigger condition is met, the data logging settings cannot be registered.

When file register is specified as specification device in condition specification

If a file register is specified as the specification device in the condition specification for the collection and trigger condition settings, do not change the file register file name or the block number of the file register once the data logging settings are registered. Doing so may possibly result in failure to successfully collect data logging results.

When the data collection condition is set to "Time specification"

If "Time specification" is selected for the data collection condition, data collection is performed as an interrupt processing operation and therefore special care should be taken when configuring the data collection interval, the data collection timing, and the data logging processing time per scan. In the following cases, scans may take a longer time, possibly resulting in a WDT error:

- The collection interval and the collection timing are so short that data logging is frequently performed during a single scan.
- The data to be collected is so much that the data logging processing time per scan is long.

Numbering of the storage files used during data logging

If one or more numbered storage files already exist in the specified file storage folder and a new file is written, the new file is given a file name that uses a number incremented by one from the highest number among the existing files. If the file storage folder has no storage files but one or more storage file container folders exist, the new file is stored under the folder with the lowest number and it is given the same number as the folder. However, if there are 258 or more folders under the said conditions, a new folder is created and the file is given the same number as the new folder.

Behavior that occurs while collected data is stored in the target memory

If one of the following operations is performed while collected data is stored in the target memory, any unsaved data is cleared and not reflected to the results:

- Turning power of the CPU module off and on
- Reset

If one of the following operations is performed, unsaved data continues to be stored in the target memory:

- Stopping the CPU module
- Stopping/suspending data logging from within the CPU Module Logging Configuration Tool
- Issuing the LOGTRGR instruction

Creating files and folders

Under the "LOGGING" folder that contains data logging setting files and data logging files, do not attempt to create files or folders using a personal computer or other device. Doing so may result in deletion of files and folders.

When collection is performed at the specified time

When "Time specification" is selected for the data collection condition and the collection at the specified time is selected rather than data collection during the END processing, check the collection interval by referring to the information reported in the collection interval column. Do not relay on the information reported in the date/time column because it may be incorrect due to clock accuracy error.

Changing the clock data

Whatever changes, such as advancing or reverting the clock, are made to the clock data of the CPU module during data logging, the CPU module performs data collection at the specified collection interval/timing, but the date/time column in the output file reports the changed clock data.

Events that are not recognized as a trigger condition

For trigger logging, the following events are not recognized as a trigger condition:

- A second trigger condition is met after the first trigger condition is met.
- The data condition specified as part of the trigger condition is met within the I49 interrupt program.

Point

If a data condition is specified as part of the trigger condition, any trigger condition met during the execution of the I49 interrupt program will not be recognized as a trigger condition. In this case, using I48 instead of I49 allows for avoiding the timing when the condition is not met.

Access to the SD memory card

If data logging is performed with a small setting of the data collection interval/timing or with a large number of records to be collected, access (read/write) to the SD memory card occurs so frequently that a delay occurs in completing the access. To avoid such a delay, use the following workarounds:

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

Time-out error in a communication or dedicated instruction from a peripheral

Under the following conditions, a time-out error may occur in a communication or dedicated instruction issued from a peripheral to the CPU module while the data logging function is running:

■Multiple data logging sessions are started concurrently

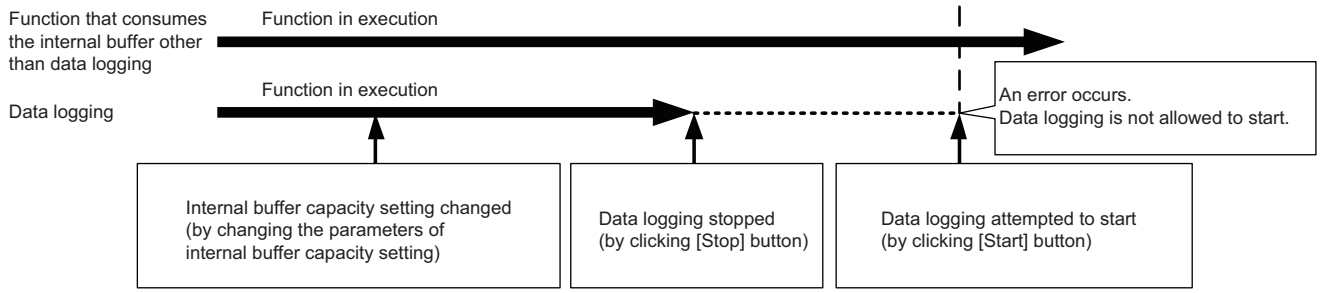
To fix a time-out error, review and correct the device/label access service processing settings, try to start the data logging sessions one after another, or take other appropriate measures.

■Multiple data logging sessions' settings are registered to be started when the CPU module is stopped and run

To fix a time-out error, increase the time-out time period of the affected peripheral or reduce the number of data logging sessions or take other appropriate measures.

Behavior at parameter change when functions consuming the internal buffer are active

If the internal buffer capacity setting is changed during the execution of functions that consume the internal buffer, attempting to start data logging results in an error, where the data logging fails to start.



Changing the operating status of the CPU module

The operating state of the CPU module should not be changed until the completion of the following operations and registrations:

- Save of the data in the internal buffer by changing the state of the CPU module from RUN to STOP or instructing the CPU Module Logging Configuration Tool to stop or suspend the CPU module
- Registering multiple data logging sessions' settings*1
- Registering data logging settings with any unused folders remaining in the storage memory*1

*1 Data logging settings are also registered when the CPU module is stopped and run.

Point

- To shorten the time required to register multiple data logging sessions' settings, reduce the number of data logging sessions.
- To shorten the time required to register data logging settings with any unused folders remaining in the storage memory, delete the unused folders before registration.

Unicode text file

To view the data, use a software application with Unicode support. The data may be displayed incorrectly in a software application with no Unicode support. To open a Unicode text file with multiple lines of comments in Microsoft® Excel, be sure to drag and drop the file on the Microsoft® Excel window; otherwise, the content of the file may be displayed incorrectly in Microsoft® Excel.

File Operation during Execution of Data Logging

This section describes file operation during execution of data logging.

Target file	File operation	Behavior
Data logging setting file	Write	During execution of data logging, it is not possible to write/read data to/from the data logging setting file being used.
	Delete	
	Initialize	During execution of data logging, it is not possible to initialize the memory storing the data logging setting file being executed.
	Folder delete	Folders cannot be deleted from the $MELPRJ$ folder in which the data logging setting file is stored.
Data Logging File	Write	During execution of data logging, it is not possible to perform write/read data to/from and delete folders from the data logging setting file being used.
	Delete	
	Folder delete	
	Initialize	During execution of data logging, it is not possible to initialize the memory storing the data logging setting file being executed.

About remote operation

When remote RUN is performed while the data logging function is in the following execution status, the remote RUN may fail. In that case, wait for a while and retry remote RUN. If remote RUN still cannot be executed, check whether remote RUN is acceptable and retry remote RUN.

Execution state of data logging function	The situation to accept remote RUN
Data saving into memory card in progress	No special relay (Data logging data saving into memory card in progress) is on.
Registration of the data logging setting from CPU Module Logging Configuration Tool in progress	The special relay (data logging preparation) and the special relay (data logging start) corresponding to the setting number of the data logging setting, which is being registered in the way shown in the left column, are on.
Registration of the data logging setting by auto logging in progress	M1200 (Auto logging setting file and registration status) is on.

RUN operation through switching operation or the RUN contact

During execution of data logging, when the status of the CPU module is switched from STOP to RUN with the RUN/STOP/RESET switch, or when the RUN contact that is specified in the RUN-PAUSE contact setting turns off, it may take time to return to the RUN state.

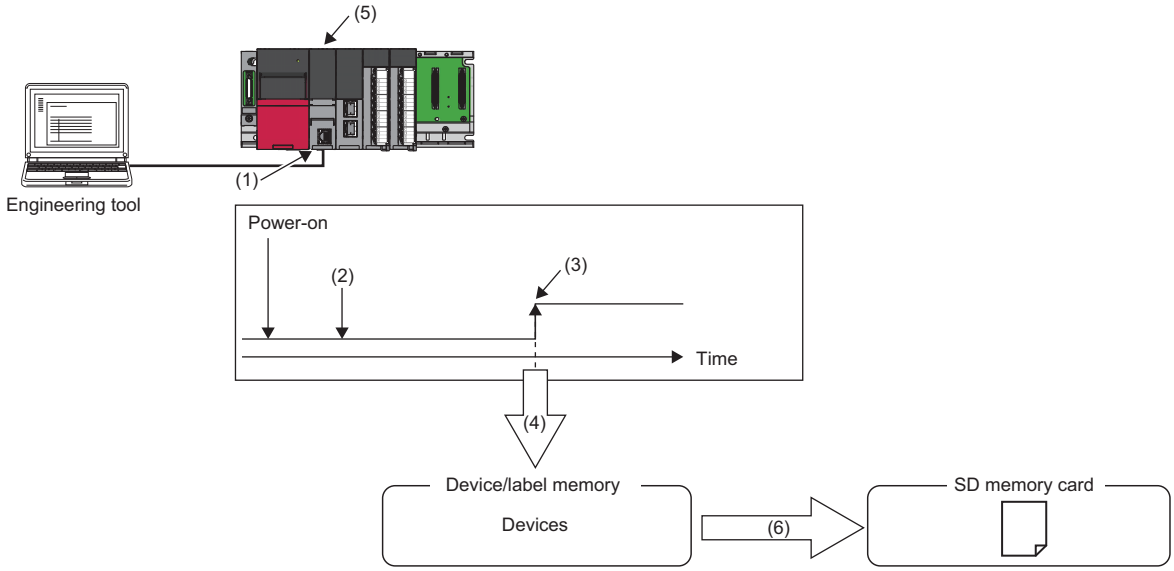
Using together with interrupt programs

When occurrence of an interrupt is specified as the data collection condition for data logging, the processing time of the data logging function is added and, consequently, the processing time of the interrupt program is increased. Therefore, when I49, an interrupt by timer, is specified with a shorter interval (such as 0.05ms) in the internal timer for an interrupt program and occurrence of I49 is specified as the data collection condition for data logging, the processing time of the interrupt program may be too long to fit in the interval specified in the fixed scan interval setting, which causes an END instruction executed error (WDT error) due to continuous execution of the interrupt program. For the method of reducing the processing time of interrupt programs, refer to Saving/restoring of the file register (R) block number. (☞ Page 66 Saving/restoring of the file register (R) block number)

24 MEMORY DUMP FUNCTION

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

This function stores device values of the CPU module at any given timing. Checking data at the desired timing through the function facilitates the analysis of troubles, the occurrence of which depends on a particular condition.



(1) Perform memory dump settings.
(2) Enters a wait state for the trigger after the memory dump setting file has been written.
(3) Establishment of the trigger condition
(4) Start of data collection
(5) The LED on the front of the CPU module allows memory dump status to be checked. (☞ Page 222 LED status)
(6) The memory dump file is stored in the SD memory card. (☞ Page 221 Memory Dump File)

Restriction

Before executing the function, check the versions of the CPU module and engineering tool used. (☞ Page 779 Added and Enhanced Functions)

24.1 Object Data

This section describes the data to be collected by memory dump.

Data to be collected

Of the devices listed below, all devices that are within the range specified in the device settings are subject to the collection.

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

*1 Includes local devices as well. Note that to the memory dump file, the program names of collected local devices are output.
*2 All file registers that exist in the device/label memory are collected.
*3 If collecting the items with the programmable controller CPU, check the versions of the CPU module and engineering tool. (☞ Page 779 Added and Enhanced Functions)

24.2 Trigger Condition

The following table lists the conditions to be used as a trigger. Set the trigger condition in the memory dump settings. (GX Works3 Operating Manual)

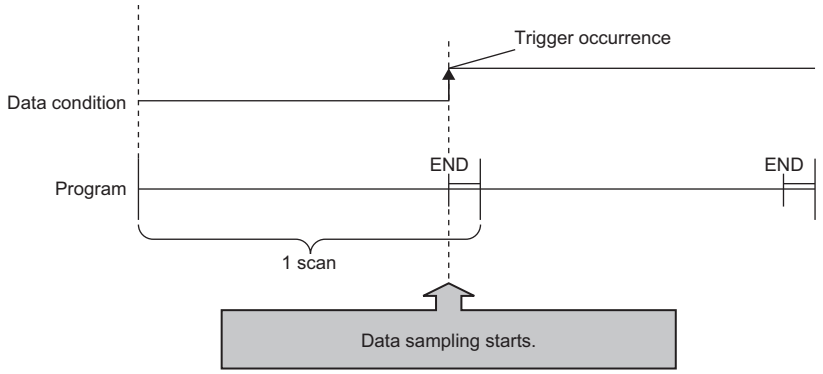
Trigger condition	Description
Device specification	Data are collected when the specified monitoring target data (bit data) turns on during the END processing.
Error code specification	Data are collected when an error occurs in the specified CPU module.

Point

- On the occurrence of consecutive triggers, if the status of data save due to the previous occurrence of trigger is "Save complete" of collected data, the next trigger is recognized as a trigger again. Note that events other than the above are not recognized as a trigger condition.
- A trigger can be generated with trigger conditions combined. (Page 219 Combining trigger conditions)

Device specification

Data are collected when the specified monitoring target data (bit data) turns on during the END processing. Even though the value of the monitoring target data changes during a single scan, if the value during the END processing is same as that during the last END processing, it is not recognized as a trigger. Completion bits (including error completion bits) used in dedicated instructions cannot be used as a trigger.



For monitoring data, the following devices can be specified.

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

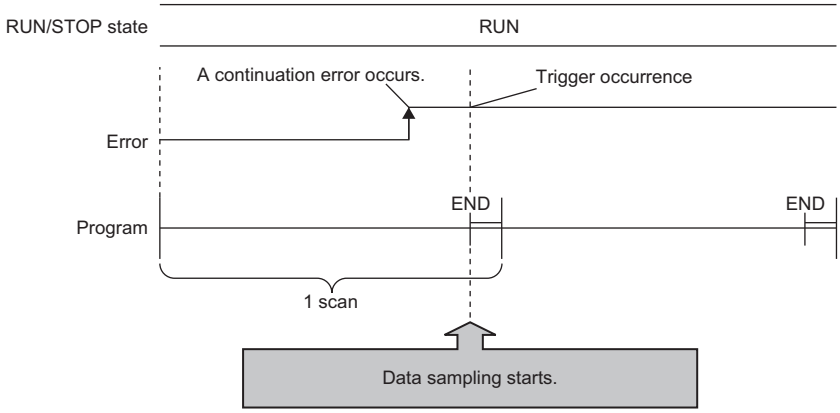
*1 A local device, index modified device, and indirectly specified device cannot be specified.
*2 Word devices allow bit specification only.
*3 In the engineering tool, specify TS as T (contact), STS as ST (contact), CS as C (contact), LTS as LT (contact), LSTS as LST (contact), and LCS as LC (contact).

Error code specification

With a specified error code of the CPU module as a trigger, data is to be collected. The occurrence timing of trigger varies depending on the error type: continuation error or stop error.

At the occurrence of a continuation error

The occurrence timing of trigger is at the time of END processing of the scan where an error has occurred.



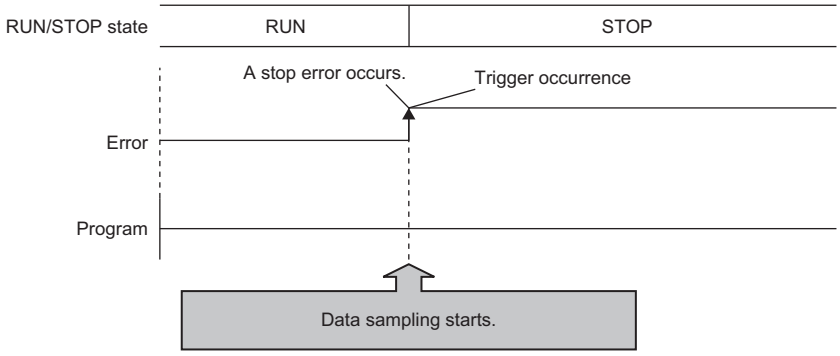
In the following situations, the occurrence of a continuation error that is specified as a trigger condition is not recognized as a trigger, with no data collection:

- After the occurrence of a continuation error that is specified as a trigger condition, the same error, or a continuation error, has occurred again.
- In the situation that a continuation error of 15 or more has occurred, a continuation error that is specified as a trigger condition has occurred.

Being recognized as a trigger requires the error to be cleared.

At the occurrence of a stop error

The occurrence timing of trigger is at the time of occurrence of an error.

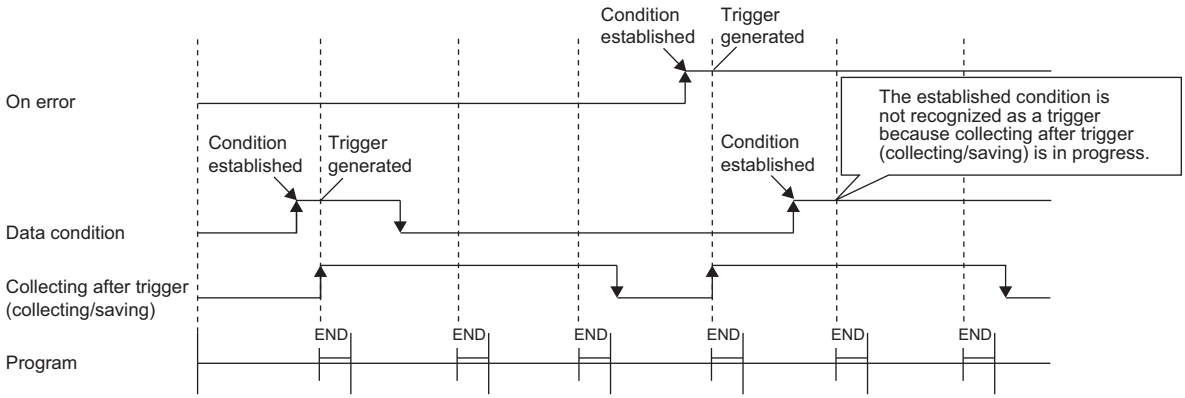


Point

To specify an annunciator number as a trigger condition, employ device specification to specify any desired annunciator number.

Combining trigger conditions

A trigger can be generated with trigger conditions combined. This combination is based on an OR condition. The establishment of a condition, either device specification or error code specification, results in data collection.



24.3 Procedure for Memory Dump

This section describes the procedure for memory dump. Note that each operation of the memory dump function is performed with the engineering tool.

[Debug] ⇨ [Memory Dump]

For how to view and operate the window, refer to the following.

GX Works3 Operating Manual

1. Configure the memory dump settings by the menu operation in the engineering tool.
2. Writing the memory dump setting file results in a wait state for the trigger. Whether the CPU module is in a RUN state, STOP state (a stop error also included[*1]), or PAUSE state, a wait state for the trigger results.

*1 Limited to where the trigger condition is device specification.

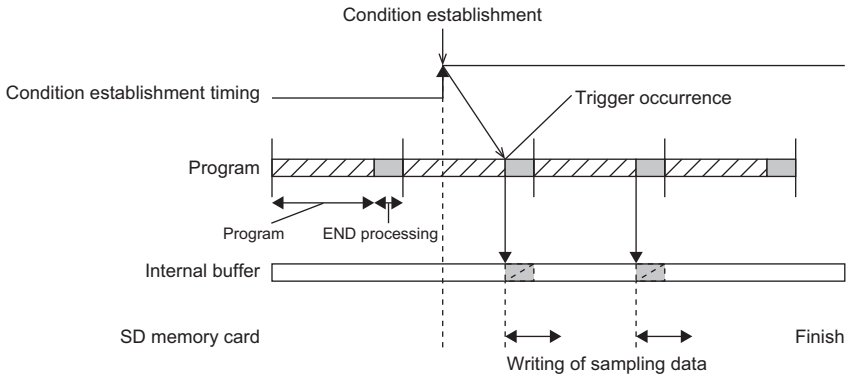
Point

- If the memory dump setting file is stored, putting the CPU module back into operation (powering off and on, or resetting) results in a wait state for the trigger.
- The engineering tool allows the memory dump status to be checked. The event history also allows checking whether or not in a wait state for the trigger.

3. Establishment of the trigger condition initiates data collection, saving the memory dump file to the SD memory card.
4. The contents of the memory dump file (collected device data) can be checked with the engineering tool.

24.4 Flow of Data Collection

Collected data is stored in the internal buffer, where the data is partitioned at END processing and saved in the SD memory card.



The size of data to be collected per END processing is determined according to the capacity setting of the internal buffer and the individual area size of devices to be collected. Data is collected and partitioned into the groups per END processing, as in the table below. If the capacity of an individual area exceeds the internal buffer capacity, collected data in the individual area is further partitioned. The first 1K byte in the internal buffer is used as the system area, and thus the collection size per END processing is determined to a value of the internal buffer capacity minus 1K byte.

Partition area		Remarks
Device area		If the total value of devices exceeds the internal buffer capacity, the amount by which the internal buffer capacity is exceeded is collected at the next END processing in order from the top device, not collected at the current END processing.
Local device (at each program)		If the total value of local devices of programs concerned exceeds the internal buffer capacity, the amount by which the internal buffer capacity is exceeded is collected at the next END processing in order from the top device, not collected at the current END processing.
File register (at each file)		—
Refresh memory	Refresh memory area	—

Effect on the scan time

The greater the number of collected points, the longer the scan time. For the increase in scan time due to the collected points, refer to the processing time. (☞ Page 704 Memory dump function)

24.5 Memory Dump File

This file stores data that is collected through memory dump (collection result by memory dump). Data collected by one execution is saved in one file. The memory dump file is saved in a binary format and stored under the "MEMDUMP" folder.

Save file name

The file name can be arbitrarily set within a range of 64 characters (extension and period included) together with an auto-assigned number (00 to 99). Specify the save file name in the memory dump settings. (GX Works3 Operating Manual)

Ex.

MEMDUMP_00*1

*1 Between a specified file name (MEMDUMP) and an auto-assigned number (00), the single-byte underscore (_) is added.

When the memory dump function is registered, the debug folder (DEBUG (fixed)) and the memory dump folder (MEMDUMP (fixed)) are created in the SD memory card. The memory dump file (result file) is stored in the memory dump folder. One folder can contain a maximum of 100 files. If any file does not exist in creating a save file, the file with the number 00 is created. If any file already exists in creating a save file, the behavior is as follows:

Number of files	Behavior
For less than 100	Creates a file*2 assigning the number obtained by adding 1 to the number of the file where the creation date and time is the latest.
For 100 (maximum)	Deletes the file where the creation date and time is the oldest and creates a new file using the deleted number as it is.

*2 If the corresponding file number is 99, a file with file number 00 is created.

24.6 States of the Memory Dump Function

The state of the memory dump function is reflected in the memory dump status. The engineering tool allows the memory dump status to be checked. (GX Works3 Operating Manual)

Memory dump status

The following table lists the memory dump status.

Memory dump status	Description
Trigger-wait not collected	A state that data is not yet collected and establishment of the trigger condition is being waited
Collecting after trigger	A state that collection of the data after trigger is in progress (includes a state that collected data is being saved in the target memory)
Collection completed	A state that collection of a specified data is completed
Error	A state that a memory dump error occurs and memory dump fails

LED status

Whether the memory dump function is active or not can be checked by the LED of the CPU module.

LED status	Description
Trigger-wait not collected	• CARD READY LED: On • CARD ACCESS LED: Off • FUNCTION LED: On
Collecting after trigger	• CARD READY LED: On • CARD ACCESS LED: Turns on when the SD memory card is accessed • FUNCTION LED: Flashes slowly (every one second)
Collection completed	• CARD READY LED: On • CARD ACCESS LED: Off • FUNCTION LED: Flashes at normal rate (every 200ms)

Point

To let the FUNCTION LED indicate the status while the memory dump function is used, setting "LED Display Setting" in "RAS Setting" of the CPU parameter is required. When the external input/output forced on/off function is executed (in registration), the LED display follows the status of the external input/output forced on/off function. The display of the FUNCTION LED switches regardless of the LED display setting. After the external input/output forced on/off function has been executed (after the registration is canceled), the LED display follows the status of the memory dump function. (Page 232 LED display setting)

24.7 Sizes of Files Used for the Memory Dump Function

This section shows the sizes of files used for the memory dump function.

Capacity of the memory dump setting file

The capacity of the memory dump setting file varies depending on the length of the save file name. The following formula is used for the calculation:

Capacity of memory dump setting file = (((Number of characters of save file name[*1] × 2 bytes + 1201 bytes (fixed)) + 3) ÷ 4)[*2] × 4

*1 Except for the period and extension.

*2 The remainder is discarded.

Capacity of the memory dump file

The capacity of the memory dump file is given by the total of the following items:

Capacity of memory dump file = Volume of header + Volume of data of program file name + Volume of device data + Volume of local device data + Volume of data of file register file name + Volume of file register data

Volume of header

The volume of header is given by:
Volume of header = 1088 bytes (fixed)

Volume of data of program file name

The volume of data of program file name is given by the total of the following items. Note that this data is always created in the memory dump file regardless of the settings of CPU parameters.
Volume of data of program file name = 16 bytes (fixed) + (Number of programs × (2 bytes (length of program file name) + 130 bytes (program file name))

Volume of device data

The volume of device data is given by the total of the following items. Note that this data is always created in the memory dump file regardless of the settings of CPU parameters.
Volume of device data = 682 bytes (fixed) + Volume of collected device data

■Volume of collected device data

The volume of collected device data is given by the following:
Volume of collected device data = (Total number of points of bit devices ÷ 8) + (Total number of points of word devices × 2) + (Total number of points of word devices × 4)

Volume of local device data

The volume of local device data is given by the total of the following items. Note that this data is not created in the memory dump file unless local devices are set in the CPU parameters.
Volume of local device data = 16 bytes (fixed) + (Number of programs × 4 bytes) + (Number of programs × Volume of local device contents)

■Volume of local device contents

The volume of local device contents is given by the following:
Volume of local device contents = 418 bytes (fixed) + Volume of collected local device data

■Volume of collected local device data

The volume of collected local device data is given by the following:
Volume of collected local device data = (Total number of points of bit devices ÷ 8) + (Total number of points of word devices × 2) + (Total number of points of word devices × 4)

Volume of data of file register file name

The volume of data of file register file name is given by the total of the following items. Note that this data is not created in the memory dump file unless a file register file exists in the device/label memory.
Volume of data of file register file name = 16 bytes (fixed) + (Number of file register files × (2 bytes (length of file register file name) + 130 bytes (file register file name)

Volume of file register data

The volume of file register data is given by the total of the following items. Note that this data is not created in the memory dump file unless a file register file exists in the device/label memory.
Volume of file register data = Number of file register files × (148 bytes (fixed) + (Number of file registers × 2))

24.8 Special Relay and Special Register Used in the Memory Dump Function

For details on the special relay and special register used in the memory dump function, refer to the following:

- Special relay: Special relay relating to the memory dump function (☞ Page 626 Memory dump function)
- Special register: Special register relating to the memory dump function (☞ Page 667 Memory dump function)

24.9 Precautions for the Memory Dump Function

This section describes precautions to take when using the memory dump function.

Mutual exclusion of the memory dump function

The mutual exclusion of the memory dump function is as follows.

■When another function is executed

The following table lists the cases when another function is executed during the execution of the memory dump function[*1].

Function that has been already executed	Function to be executed later	Behavior
Memory dump function	CPU module data backup function	The CPU module data backup function cannot be executed while memory dump is being registered/cleared.
	CPU module data restoration function	The CPU module data restoration function cannot be executed while a memory dump file or memory dump setting file is being read or memory dump is being registered/cleared.
	iQ Sensor Solution data backup/restoration function	The iQ Sensor Solution data backup/restoration function cannot be executed while memory dump is being registered/cleared.
	Function specified in the internal buffer capacity setting[*2]	If the internal buffer capacity setting is changed to execute the subsequent function, attempting to start the subsequent function results in an error. The memory dump continues to function normally.
	Function not specified in the internal buffer capacity setting	If the condition "Total capacity that is set in the internal buffer capacity setting + Internal buffer capacity that is set in other than the internal buffer capacity setting > 3072K bytes" is satisfied, attempting to start the subsequent function results in an error. The memory dump continues to function normally.
		If the internal buffer capacity setting is changed to execute the subsequent function, attempting to start the subsequent function results in an error. The memory dump continues to function normally.

*1 Indicates the state that the memory dump status is "Collecting after trigger" or the save status is "Saving in progress".

*2 The memory dump function is not included here.

■When the memory dump function is executed during the execution of another function

The following table lists the cases when the memory dump function is executed during the execution of another function.

Function that has been already executed	Function to be executed later	Behavior
CPU module data backup function	Memory dump function	While the CPU module data backup function is being executed, memory dump cannot be registered/cleared.
CPU module data restoration function		While the CPU module data restoration function is being executed, a memory dump file or memory dump setting file cannot be read or memory dump cannot be registered/cleared.
iQ Sensor Solution data backup/ restoration function		While the iQ Sensor Solution data backup/restoration function is being executed, memory dump cannot be registered/cleared.
Function specified in the internal buffer capacity setting[*1]		If the internal buffer capacity setting is changed to execute memory dump, attempting to start memory dump results in an error. The function already in execution continues to function normally.
Function not specified in the internal buffer capacity setting		If the internal buffer capacity setting is changed to execute memory dump, attempting to start memory dump results in an error. The function already in execution continues to function normally.

*1 The memory dump function is not included here.

■When the memory dump function is executed

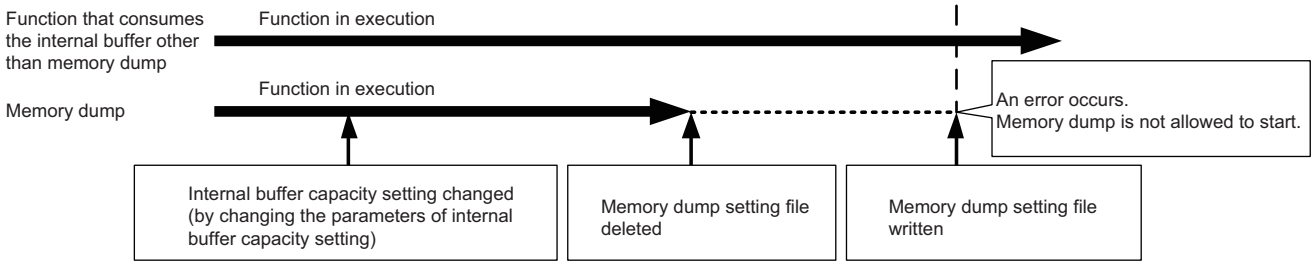
The following table shows the cases where the file operation related to the memory dump function is executed while the memory dump function is in execution.*1

Target file	File operation	Behavior
Memory dump setting file	Write	Settings that are subsequently written during the execution of the memory dump function are reflected after the completion of save, not reflected immediately.
	Delete	If the memory dump setting file is subsequently deleted during the execution of the memory dump function, the memory dump settings are cleared after the completion of save.
	Initialize	Initialization fails on the memory dump setting file during the execution of the memory dump function.
	Folder delete	Folder delete fails on the folder in which the memory dump setting file is stored.
Memory dump file	Write, read, delete, initialize, and folder delete	Write, read, delete, initialize, and folder delete fails on the memory dump file during the execution of the memory dump function.

*1 Indicates the state that the memory dump status is "Collecting after trigger" or the save status is "Saving in progress".

Behavior at parameter change when functions consuming the internal buffer are active

If the internal buffer capacity setting is changed during the execution of functions that consume the internal buffer, attempting to start memory dump results in an error, where the memory dump fails to start.



Operation on each individual file

Write, read, and delete are possible on each file. In addition, folder/file all delete and folder delete are possible on the memory or folder in which files are stored. The following table shows whether each operation is possible or not depending on the execution status of memory dump.

○: Operation possible, ×: Operation not possible

File type	Operation to be performed							
	Read		Write/delete		Folder/file all deletion		Folder delete	
	Not during execution*1	During execution*1	Not during execution*1	During execution*1	Not during execution*1	During execution*1	Not during execution*1	During execution*1
Memory dump setting file	○	○	○	○	○	×	×	×
Memory dump file	○	×	○	×	○	×	○	×

*1 Indicates the state that the memory dump status is "Collecting after trigger".

Where to carry out memory dump

Concurrent execution from multiple sources is not allowed. In the CPU module, execution at a time from only one source is possible.

Trigger condition during the registration of memory dump

If a trigger condition is established during the registration of memory dump, the memory dump settings can be registered. A second establishment of the trigger condition is recognized as a trigger condition there.

When file register is specified as specification device in the condition specification

After registering memory dump, do not change the file name of the file registers and the block number of the file registers. Doing so may result in a failure to successfully collect the memory dump results.

Creating files and folders

Under the "MEMDUMP" folder containing memory dump files, do not create any files or folders using a personal computer or other device. Doing so may result in deletion of files and folders.

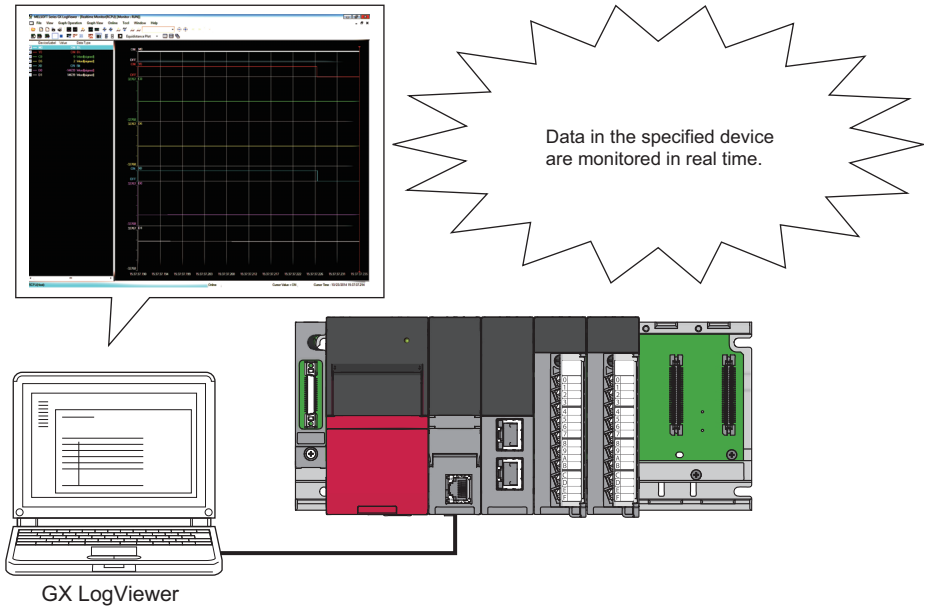
Access to the SD memory card

The SD memory card is so frequently accessed that a delay occurs in completing the access to the SD memory card (read/write).

25 REAL-TIME MONITOR FUNCTION

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

This function monitors the contents of a specified device of the CPU module in real time with a specified interval or a desired timing. The function can be set with GX LogViewer, where the value changes of a specified device can be shown graphically. Saving the set data and displayed graphs makes it possible to simplify the settings and check the graphs at a later time.



Restriction

- Before executing the function, check the versions of the CPU module and GX LogViewer used. (☞ Page 779 Added and Enhanced Functions)

For details on the function, refer to the following.

📖 GX LogViewer Version 1 Operating Manual

MEMO

26 RAS FUNCTIONS

26.1 Self-Diagnostics Function

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

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

☞ Page 387 FUNCTIONS

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

☞ Page 486 FUNCTIONS

Checks if a problem exists with the CPU module.

Self-diagnostics timing

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

Check method of error

This section describes the check methods when error occurs.

Check method using the special relay and special register

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

Check method using LED

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

Check method using the engineering tool

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

■Existing errors

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

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

■Error history

Occurred errors is logged in the event history (☞ Page 236 Event History Function)

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

CPU module operation upon error detection setting

Configure each CPU Module Operation setting when an error is detected.

Mode when an error is detected

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

■Mode for stopping the operation of CPU module

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

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

Point

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

■Mode for continuing the operation of CPU module

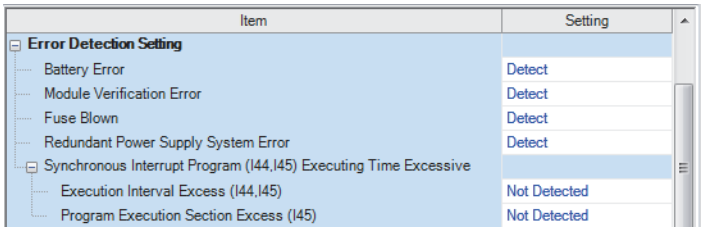
If an error has been detected, the program other than programs (instruction) where an error was generated is executed.

Error Detection Setting

Sets whether or not to detect errors.

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

Window



Displayed items

Item		Description	Setting range	Default
Battery Error		Sets whether or not to detect the battery error.	• Detect • Not Detected	Detect
Module Verify Error*1		Sets whether or not to detect the module verification error.		
Fuse Blown		Sets whether or not to detect the fuse blown of the controlled module.		
Redundant Power Supply System Error*2		Sets whether or not to detect the power supply module error (power off or power supply failure in either system) in the redundant power supply system.		
Synchronous Interrupt Program (I44, I45) Executing Time Excessive	Execution Interval Exceed (I44, I45)	Sets whether or not to detect the excessive execution time of synchronous interrupt program (I44, I45).		Not Detected
	Program execution section excess (I45)	Sets whether or not to detect the program execution section excess errors of synchronous interrupt program (I45).		

*1 When "Not Detected" is enabled and the operating CPU module is removed, a verify module error is not detected, but a stop error may occur if an program tries to access the removed CPU module. Also, the removed CPU module will not be accessible even when it is re-installed. Therefore, if access is tried to the re-installed CPU module through the STOP and RUN operation, it may result in a stop error.

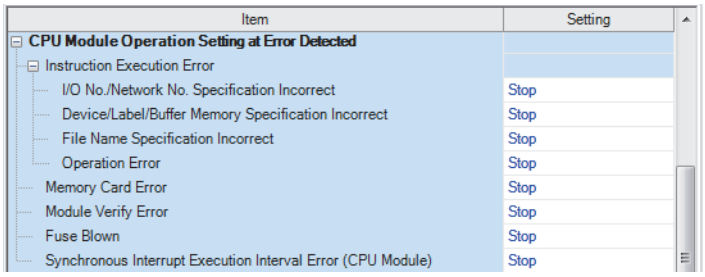
*2 When "Not Detected" is set, values are stored in the special relay and special register (SM150 to SM154/SD150 to SD154) that indicate the status of the redundant power supply module.

CPU module operation upon error detection setting

Sets the CPU module operation upon error detection.

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

Window



Displayed items

Item		Description	Setting range	Default
Instruction Execution Error	I/O numbers/Network numbers specification error	Sets the CPU module operation when I/O numbers/Network numbers specification error has been detected.	• Stop • Continue	Stop
	Device/Label/Buffer memory specification error	Sets the CPU module operation when Device/Label/Buffer memory specification error has been detected.		
	File Name Specification Incorrect	Sets the CPU module operation upon a file name specification error.		
	Operation Error	Sets the CPU module operation upon operation error.		
Memory Card Error		Sets the CPU module operation upon a memory card error.		
Module Verify Error*1		Sets the CPU module operation upon a module verification error.		
Fuse Blown*1		Sets the CPU module operation upon fuse blown error of controlled module.		
Synchronous Interrupt Execution Interval Error (CPU Module)		Sets the CPU module operation upon a detection of a synchronization signal error on the CPU module.		

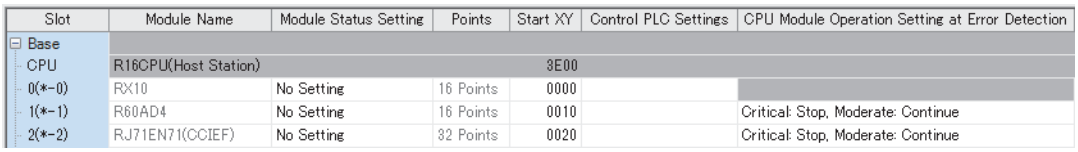
*1 For the Process CPU, setting "Direct change setting" to "Enable" in "Online module change setting" causes a continuation error regardless of this setting. For the online module change function setting, refer to the following.
MELSEC iQ-R Online Module Change Manual

CPU Module Operation Setting

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

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

Window



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

Displayed items

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

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

Stop Setting

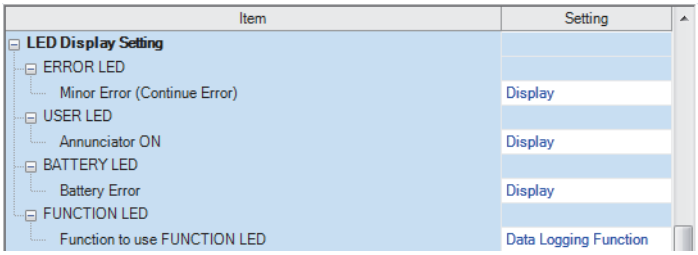
For the multiple CPU system configuration, it can be specified whether or not to stop all CPU modules when a major or moderate error occurs on a CPU module. (☞ Page 276 Stop setting)

LED display setting

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

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

Window



Item	Setting
LED Display Setting	
ERROR LED	
Minor Error (Continue Error)	Display
USER LED	
Annunciator ON	Display
BATTERY LED	
Battery Error	Display
FUNCTION LED	
Function to use FUNCTION LED	Data Logging Function

Displayed items

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

*1 To specify whether to display or hide the FUNCTION LED, verify the versions of the CPU module and the engineering tool. (☞ Page 779 Added and Enhanced Functions)

*2 When the external input/output forced on/off function is executed (in registration), the LED indicator follows the status of the external input/output forced on/off function. The status of the FUNCTION LED switches regardless of the LED Display Setting. After the external input/output forced on/off function has been executed (after the registration is canceled), the LED indicator follows the setting of "Function to use FUNCTION LED". (☞ Page 269 FUNCTION LED)

26.2 Error Clear

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

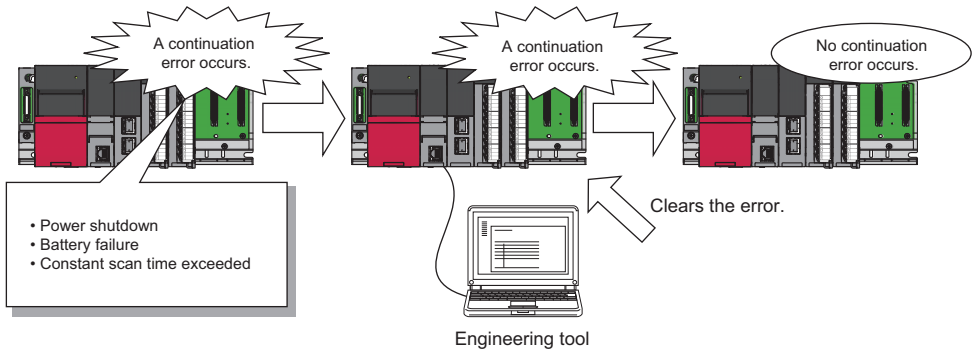
- When using the Process CPU (redundant mode), refer to the following as well.

☞ Page 387 FUNCTIONS

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

☞ Page 486 FUNCTIONS

This function clears all the existing continuation errors at once.



Errors that can be cleared

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

Error code	Error name
1000H	Power shutoff
1010H	Power shutoff (either of the redundant power supply modules)
1020H	Failure (either of the redundant power supply modules)
1030H	Invalid power supply module
1031H	Power supply module configuration error
1080H	ROM write count error
1090H	Battery error
1100H	Memory card access error
1120H	SNTP clock setting error
1124H	Default gateway/gateway IP address error
1128H	Own node port number error
1129H	Open specification port number error
112DH	Specified IP address error
112EH	Connection establishment failed
1133H	Socket communications response send error
1134H	TCP connection timeout
1152H	IP address error
1155H	Connection number acquisition error
1157H	Receive buffer securement error
1165H	UDP/IP send failed
1166H	TCP/IP send failed
1167H	Unsend data send error
1180H	Redundant IP address error
1200H, 1210H	Module moderate error
1220H	Another CPU module moderate error
1240H, 1241H	Inter-module synchronization processing error
1260H, 1262H	Multiple CPU synchronization processing error
1800H	Annunciator ON
1810H	Operation continuation error
1830H	Receive queue full

Error code	Error name
1831H	Receive processing error
1832H	Transient data error
1900H	Constant scan time error
1B00H	Network configuration mismatch
1B20H	System consistency check error (operating status)
1B40H, 1B42H, 1B43H	Redundant system error
1B60H, 1B61H	Standby system CPU module error
1B70H	Tracking communications disabled
1B71H, 1B78H	Tracking communication error
1B80H, 1B81H, 1B82H	Tracking transfer error
1BA0H	Redundant function module error
1BD0H	System switching error
2120H, 2121H	Memory card error
2400H, 2401H	Module verification error
2420H	Fuse blown error
2441H, 2442H, 2450H	Major error in module
2461H, 2462H, 2470H	Another CPU module major error
2610H	Inter-module synchronization signal error
2630H	Multiple CPU synchronization signal error
2800H, 2801H, 2802H, 2803H, 2804H, 2805H, 2806H, 2807H, 2810H	I/O numbers/Network numbers specification error
2820H, 2821H, 2822H, 2823H, 2824H	Device/Label/Buffer memory specification error
2840H, 2841H, 2842H	File name specification error
3400H, 3401H, 3402H, 3403H, 3404H, 3405H, 3406H, 3420H, 3421H, 3422H, 3423H, 34A0	Operation error

Point

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

How to clear errors

Errors can be cleared in two ways:

■Using the engineering tool

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

Point

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

■Using SM/SD

Clear errors by operating SM/SD.

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

Precautions

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

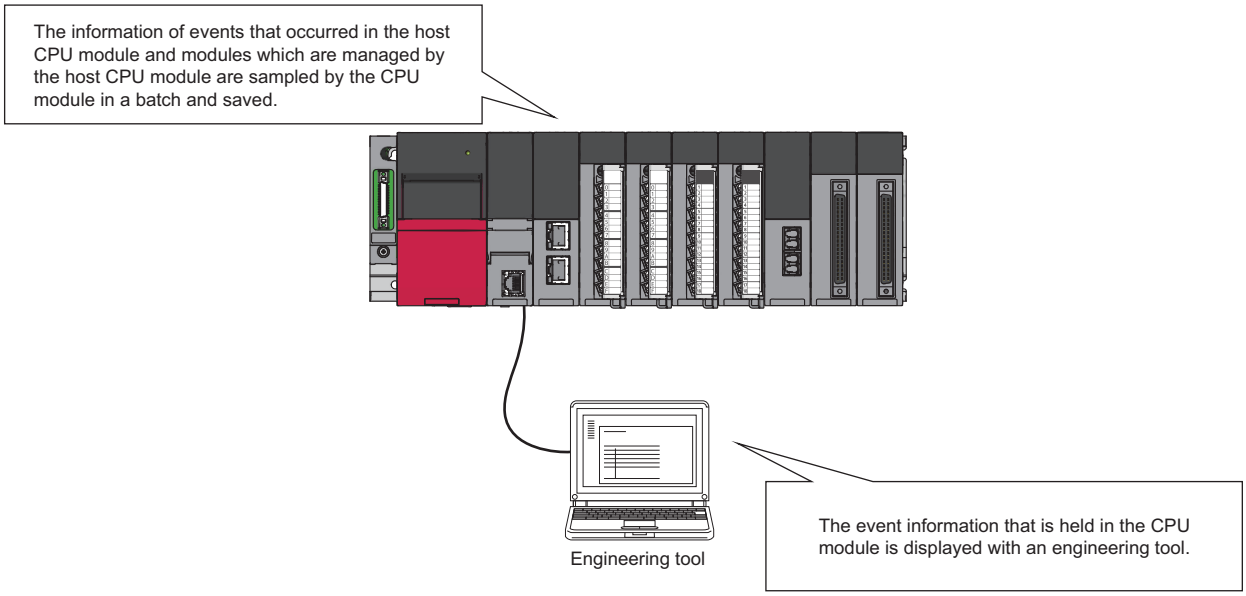
- Since the function clears all of the currently detected continuation errors at once, errors that should not yet be cleared may be cleared.
- Use the RST instruction to reset each annunciator individually.
- Running the error clear function does not remove the cleared errors from the event history.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

26.3 Event History Function

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

• When using the Safety CPU, refer to the following as well.
☞ Page 486 FUNCTIONS

The CPU module collects event information from each module, such as errors detected by the module, operations performed to the module, and network errors, and stores them.*1 Once errors and operations are stored, they can be checked chronologically. This function helps to determine the causes of problems that have occurred in the equipment/devices, check the update status of the programmable controller control data, and detect unauthorized access.



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

Point

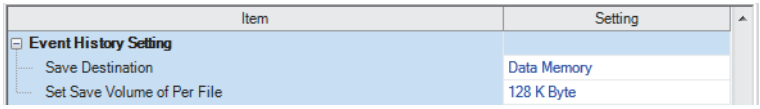
The event history information is constantly collected regardless of the operating state of the CPU module. There are occasions, however, when the event history information cannot be collected due to a major error in a module, a base unit error, a cable failure, or some other cause.

Event history settings

Under normal circumstances, the event history function can be used with its default settings and need not be manually configured. The storage memory and size settings for event history files can be changed as needed. (☞ Page 237 Event history file)

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

Window



Displayed items

Item	Description	Setting range	Default
Save Destination	Specify the storage location of event history files.	• Data Memory • Memory Card	Data Memory
Set Save Volume of Per File	Specify the storage capacity per event history file.	1 to 2048K bytes (in 1K bytes)	128K Byte

Logging of the event history

This section describes events saving for the event history.

Modules from which event history information is collected

Event history information is collected from the CPU module and other modules installed on the same base unit (i.e., the main base unit plus any additional extension base units). Event history information may or may not be collected from devices on the network depending on the specifications of the network modules used to connect to them. Refer to the manuals of the respective modules for more information including the coverage of event history collection regarding devices on the network. For a multi-CPU system, note that each CPU module logs only events detected on the modules under its control.

Events logged by the CPU module

Information logged in the event history includes operation initiator and other detailed information for troubleshooting purposes. For events that are logged in the event history on the CPU module, refer to the event history (☞ Page 594 Event List)

Event history file

The storage memory and file size for event history files can be changed in event history setting. (☞ Page 236 Event history settings)

■Storage memory

Choose either the data memory or SD memory card. If the storage memory is the SD memory card: when the write protect switch of the SD memory card is enabled, an event history will not be stored. (It is possible to read the event history file in SD memory card using the engineering tool.) Therefore, if the write protect switch of the SD memory card is changed into "Disabled→Enabled" during operation, the write error to the SD memory card occurs when the event for storing into event history is generated (SD memory card available event is registered) after the write protect switch is enabled. (At the time immediately after an error is generated, the module diagnostics of engineering tool can detect it, however, after Power is turned off→on or after resetting operation, the generated error will not be stored into the event history.)

Point

For a system where file writes occur frequently or the communication conditions are unstable and frequently change, the event history file size should be made large enough to accommodate a greater number of events. If this is the case, the SD memory card is highly recommended as the storage memory.

■File size

The size for event history files can be changed in event history setting (☞ Page 236 Event history settings). If the storage size exceeds the specified size, records are deleted in order from the oldest one and the latest one is stored. An event history file size is obtained from the following calculation formula.

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

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

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

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

[Calculation formula]

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

Ex.

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

Operating procedure

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

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

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

■When files are created

An event history file is created when:

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

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

Point

When a new event history file is created, that operation is logged into the event file as an event.

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

Operation	Operation for the event history
Memory initialization	When this event occurs, the event history is stored into the internal memory. If the internal memory reaches the maximum number of event history records it can store, all subsequent events are lost (☞ Page 239 Loss of event history information)
Event history creation	The event history, which has been stored in the internal memory during absence of the event history file, is stored into the data memory or the SD memory card (If any event was lost, it is logged as "*HST LOSS*").

Indicates the operation of the event history for the SD memory which was removed and mounted in case that the save destination memory is the SD memory.

Operation	Operation for the event history
Removal of the SD memory card	When this event occurs, the event history is stored into the internal memory. If the internal memory reaches the maximum number of event history records it can store, all subsequent events are lost (☞ Page 239 Loss of event history information)
Installation of the SD memory card	The event history, which have been stored in the internal memory during absence of the SD memory card, is stored to the SD memory card. If the re-inserted SD memory card contains an event history file of the same file size, the CPU module continues to store the event history information. If the file size is different, the CPU module removes the existing event history file and creates a new event history file.

■When parameters take effect

Any changed parameters take effect when:

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

Point

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

Loss of event history information

If events are detected frequently, or the power is turned OFF or reset immediately following event detection, some events may be lost without being collected. When event loss occurs, "*HST LOSS*" appears in the "Event Code" field of the engineering tool.

Viewing the event history

The event history can be viewed using the menus of the engineering tool. For operating procedures and how to interpret the displayed information, refer to the following:

📖 GX Works3 Operating Manual

Clearing the event history

The event history can be cleared using the event history window. Once the event history is cleared, the CPU module deletes all the event history information stored in the specified storage memory. For operating procedures and other details, refer to the following:

📖 GX Works3 Operating Manual

Precautions

Clearing the event history during execution of another function

No event history can be cleared during execution of the following functions. Check that the following functions are not being executed and then clear the event history.

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

Reading the event history during execution of another function

No event history can be read during execution of the following function. Check that the following function is not being executed and then read the event history.

- CPU module data restoration function

26.4 Program Cache Memory Auto Recovery Function

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

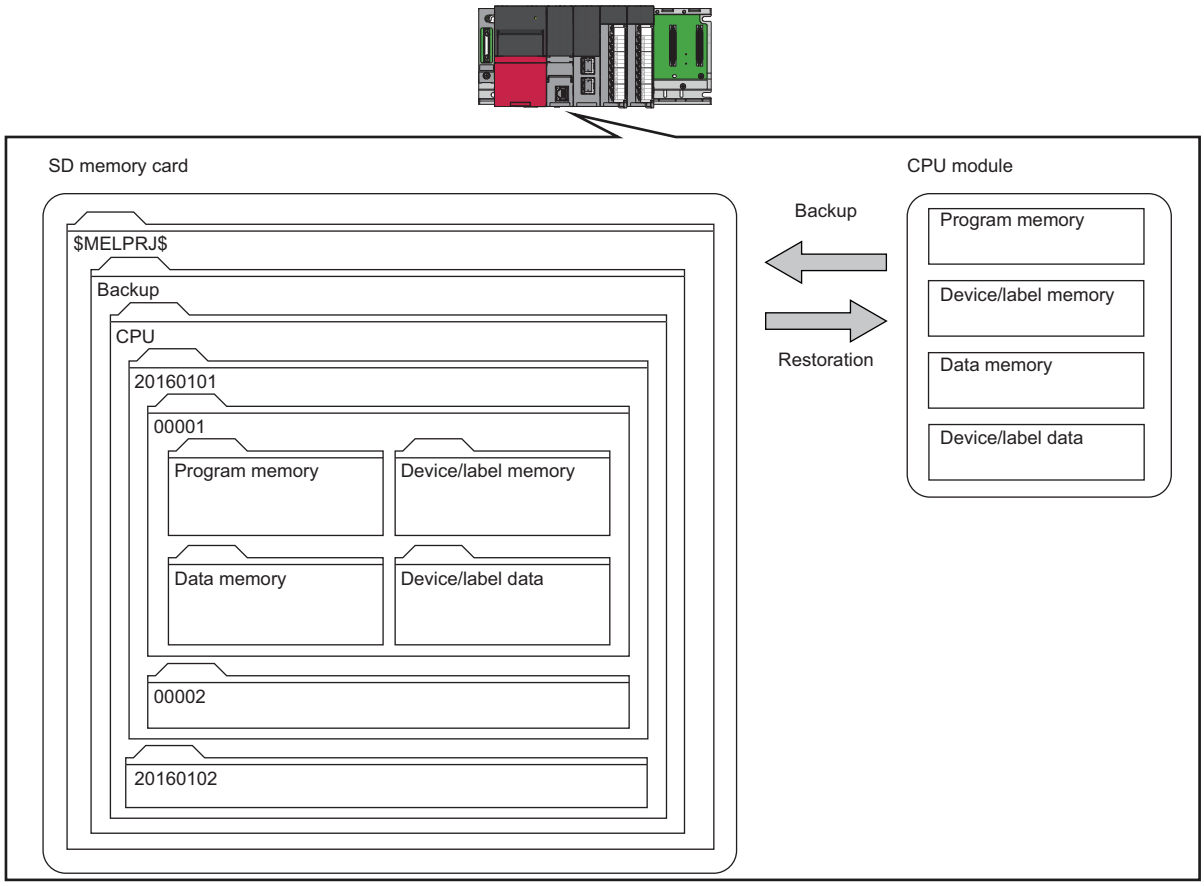
If the contents of memory of the CPU module have been rewritten by itself due to the factors such as excessive electrical noise, the program cache memory recovers the corresponding areas automatically during the run of the program. This function becomes active with RUN state of the CPU module and works during the run of the program. No special settings are required as the system performs this task automatically.

27 CPU MODULE DATA BACKUP/RESTORATION FUNCTION

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

This function backs up data such as program files, a parameter file, and device/label data*1 of a CPU module to an SD memory card. The data backed up in the SD memory card can be restored as required.

*1 Module access devices and buffer memory are excluded.



The following table lists the backup and restoration methods of the CPU module data backup/restoration function.

Function		Reference
Backup function	Backup processing triggered by turning on SM1351	Page 249 Backup processing triggered by turning on SM1351
	Automatic backup using SD944	Page 250 Automatic backup using SD944
Restoration function	Restoration processing triggered by turning on SM1354	Page 258 Restoration processing triggered by turning on SM1354
	Automatic restoration using SD955	Page 259 Automatic restoration using SD955

Point

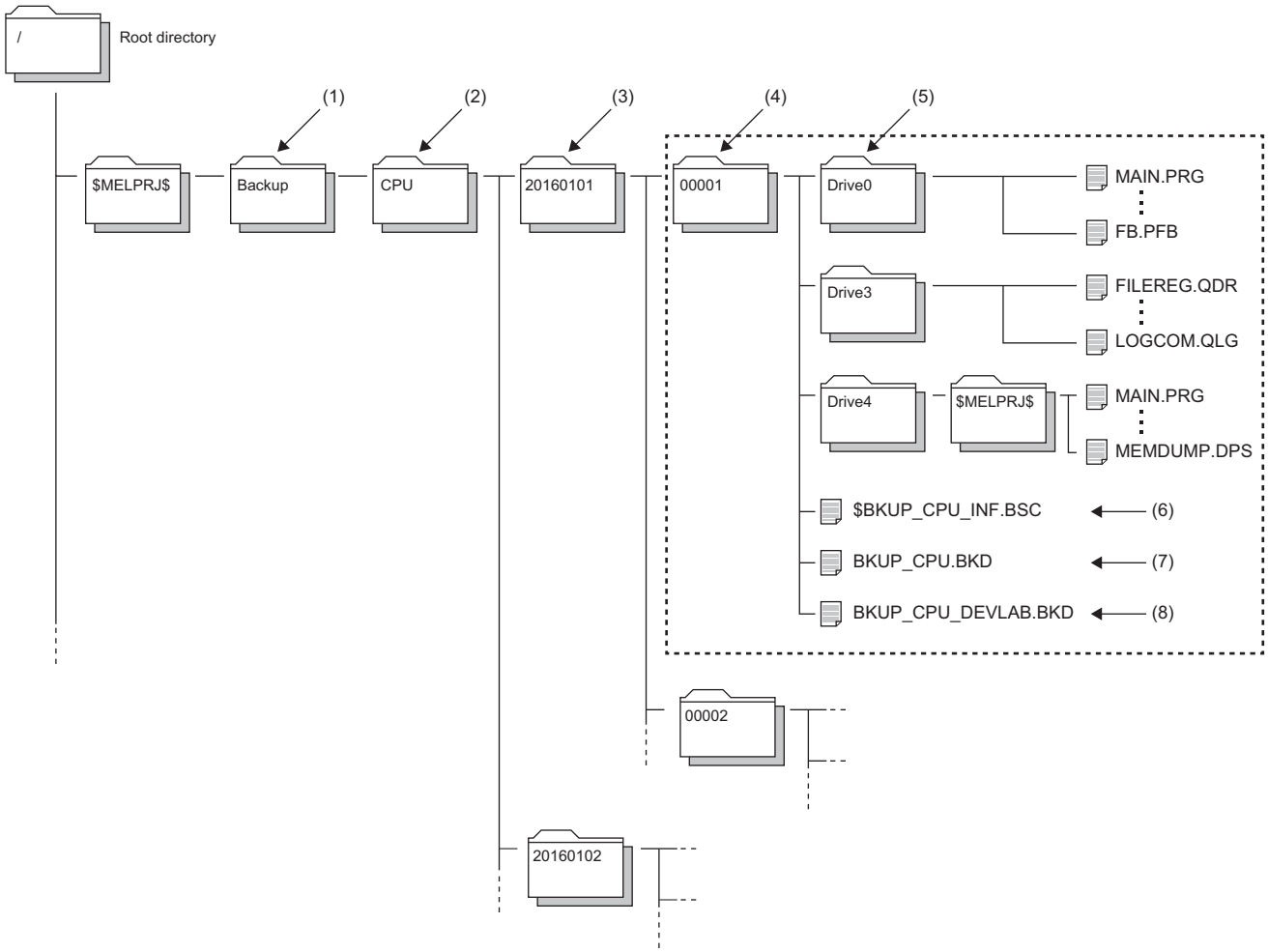
The restoration function modifies programs, parameters, or device/label data of the CPU module. After restoration, check the restored data carefully before an actual operation. (Check the data with an engineering tool.)

Restriction

When using the CPU module data backup/restoration function, check the versions of the CPU module and engineering tool used. (☞ Page 779 Added and Enhanced Functions)

Backup data

Backup data is saved in an SD memory card. The following shows the folder structure of backup data.



No.	Folder type	Folder name	Number of storable folders	Description
(1)	Backup data folder	Backup (Fixed)	1	A folder for storing all backup data
(2)	CPU data folder	CPU (Fixed)	1	A folder for storing backup data of the CPU module
(3)	Date folder	Automatically determined*1 Folder name format: YYYYMMDD • YYYY: Year when the data was backed up (four digits) • MM: Month when the data was backed up (two digits) • DD: Day when the data was backed up (two digits)	Depends on the capacity of the SD memory card used*2 or 1 to 100 folders	Folders for storing backup data by date For the setting of the upper limit value for the number of CPU module backup data, the number of backup data indicates the number of date folders. (☞ Page 247 Setting the upper limit value for the number of CPU module backup data)
(4)	Number folder	Automatically determined*1 Folder name: Sequentially numbered from 00001 to 32767 (five digits)	Depends on the capacity of the SD memory card used*2	Folders for storing information per backup data. Each backup data created on a date is stored in sequentially numbered folders.
(5)	Drive folder	Drive0 (Fixed), Drive3 (Fixed), and Drive4 (Fixed)	One set of the folders in a number folder	Folders for storing folders/files stored in each drive of the backup target CPU module by each drive
(6)	System file for backing up CPU module data	$BKUP_CPU_INF.BSC	One folder in a number folder	Files for storing the information required at restoration of data, such as a list of backup data and identification information of the CPU module
(7)	Backup data file for backing up CPU module data	BKUP_CPU.BKD	One folder in a number folder	The following data is stored. • Data on operations of the data logging setting • Data for restarting the SFC program from the block and step where the processing was stopped
(8)	Device/label data file for backing up CPU module data	BKUP_CPU_DEVLAB.BKD	One folder in a number folder	Device/label data is stored.

*1 Date folders and number folders are automatically named by the CPU module.
*2 The maximum number of storable folders is 32767.

Backup/restoration target data

Backup target data is all target data in the CPU module. (☞ Page 244 Backup/restoration target files)
Restoration target data is set with SD954 (Restoration target data setting). (☞ Page 256 Restoration target data)

■Backup/restoration target drives

Target drives are Drive0 (Program memory), Drive3 (Device/label memory), and Drive4 (Data memory).
Drive2 (SD memory card) is not a target drive.

■Backup/restoration target files

The following table lists backup/restoration target files.
○: Available, ×: Not available

File type		Backup/restoration
Program		○
FB file		○
CPU parameter		○
System parameter		○
Module parameter		○
Module extension parameter		○
Memory card parameter		×
Device comment		○
Device initial value		○
Global label setting file		○
Initial label value file	Initial global label value file	○
	Initial local label value file	○
File register		○
Event history		○
Device data storage file		○
General-purpose data		○
Data logging setting file	Common setting file	×
	Individual setting file	○
Memory dump setting file		○
Remote password		○
System file for the iQ Sensor Solution data backup/restoration function		×
Backup data file for the iQ Sensor Solution data backup/restoration function		×
ODBC server setting file		○

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

■The number of CPU module backup data that can be stored in an SD memory card

The number of CPU module backup data that can be stored in an SD memory card is 32767. This number is equal to the maximum number of storable folders.
The number of files that can be backed up and restored (the number of backup source data files) depends on the maximum number of files of each model or drive.

■Backup/restoration target device data

○: Available, ×: Not available

Classification	Device name	Symbol	Backup*5	Restoration*5
User device	Input	X	○	○
	Output	Y	○	○
	Internal relay	M	○*3	○*3
	Link relay	B	○	○
	Annunciator	F	○	○
	Link special relay	SB	○	○
	Edge relay	V	○*3	○*3
	Step relay	S	○	○
	Timer	T	○*3	○*3
	Retentive timer	ST	○*3	○*3
	Long timer	LT	○*3	○*3
	Long retentive timer	LST	○*3	○*3
	Counter	C	○*3	○*3
	Long counter	LC	○*3	○*3
	Data register	D	○*3	○*3
	Link register	W	○	○
	Link special register	SW	○	○
	Latch relay	L	○	○
System device	Function input	FX	○	×
	Function output	FY	○	×
	Function register	FD	○	×
	Special relay	SM	○	○*1*2
	Special register	SD	○	○*1*2
Link direct device	Link input	Jn\X	×	×
	Link output	Jn\Y	×	×
	Link relay	Jn\B	×	×
	Link special relay	Jn\SB	×	×
	Link register	Jn\W	×	×
	Link special register	Jn\SW	×	×
Module access device	Module access device	Un\G	×	×
CPU buffer memory access device	CPU buffer memory access device	U3En\G	○	○*2
		U3En\HG	○	○*2
Index register	Index register	Z	○*4	○*4
	Long index register	LZ	○*4	○*4
File register	File register	R/ZR	○	○
Refresh data register	Refresh data register	RD	○	○
Nesting	Nesting	N	×	×
Pointer	Pointer	P	×	×
	Interrupt pointer	I	×	×
Other devices	Network No. specification device	J	×	×
	I/O No. specification device	U	×	×
	SFC block device	BL	○	×
	SFC transition device	TR	×	×

*1 Whether to restore the device or not can be set with the bit 14 of SD955 (Restoration function setting).
*2 Values may be overwritten to the areas used by the system after the restoration processing.
*3 Includes local devices.
*4 Includes the local index register.
*5 Device data may be overwritten depending on the mounting status (I/O refresh) of each module or the refresh settings.

■Backup/restoration target label data

○: Available, ×: Not available

Classification	Backup*2	Restoration*2
Global label (including module labels)	○	○*1
Global label with latch specified	○	○
Local label	○	○
Local label with latch specified	○	○

*1 For module labels, data may be overwritten to the write areas from a module to the CPU module when the refresh settings have been made.

*2 Device data may be overwritten depending on the mounting status (I/O refresh) of each module or the refresh settings.

Progress of the backup/restoration processing

The progress of the backup/restoration processing can be checked in SD1350 (Number of uncompleted folders/files of CPU module data backup/restoration) or SD1351 (Progression status of CPU module data backup/restoration). However, the progress of the automatic restoration cannot be checked with the special register.

Special register	Description
SD1350	Displays the number of remaining backup/restoration target folders and files. • When the backup/restoration processing is started, the total number of backup/restoration folders and files is stored. • When the backup/restoration processing is completed, 0 is stored.
SD1351	Displays the progress of the backup/restoration processing in percentage (0 to 100%).*1

*1 When program files are restored, the progress in SD1351 stops while data is being written to the program memory in the restoration processing because the data is transferred from the program cache memory to the program memory. The progress of data transfer to the program memory can be checked in SD629 (Program memory write (transfer) status).

27.1 Backup Function

This function backs up data of a specified CPU module in an SD memory card.

Point

The backup function operates even when the CPU module is in the RUN state.
When executing the backup function with the CPU module in the RUN state, do not change device/label data during execution of the function. Doing so may cause data inconsistency of the device/label data and the contents of the backup data may unintentionally change.

Setting the upper limit value for the number of CPU module backup data

When the backup processing has not been executed (when no backup data folder (CPU data folder) exists in the SD memory card), the upper limit value for the number of CPU module backup data can be set.
The number of CPU module backup data in the upper limit value setting is the number of date folders.
The upper limit value for the number of CPU module backup data is enabled by turning on the bit 5 of SD944 (Enable the upper limit value for the number of CPU module backup data). The set value can be checked in SD960 (Upper limit status for the number of CPU module backup data).
When the bit 5 of SD944 is turned off, no upper limit value setting is applied.

Special relay/Special register	Description
SM960	This relay specifies an operation of backup when the number of CPU module backup data reaches the upper limit value. (This relay is valid only when the bit 5 of SD944 is on.) • Off: The oldest date folder is deleted, and the backup operation continues. • On: The backup operation does not continue if the upper limit value for the number of CPU module backup data is exceeded. (In this case, the backup is completed with an error.)
Bit 5 of SD944	Set to enable or disable the upper limit value for the number of CPU module backup data. • Off: Disable (No limit (Date folders are created within the maximum capacity of the SD memory card.) • On: Enable
SD960	This register displays the value (1 to 100) set in SD1353. If the bit 5 of SD944 is off, 0 is stored in this special register (SD).
SD1353	Set the upper limit value (1 to 100) for the number of CPU module backup data.

Restriction

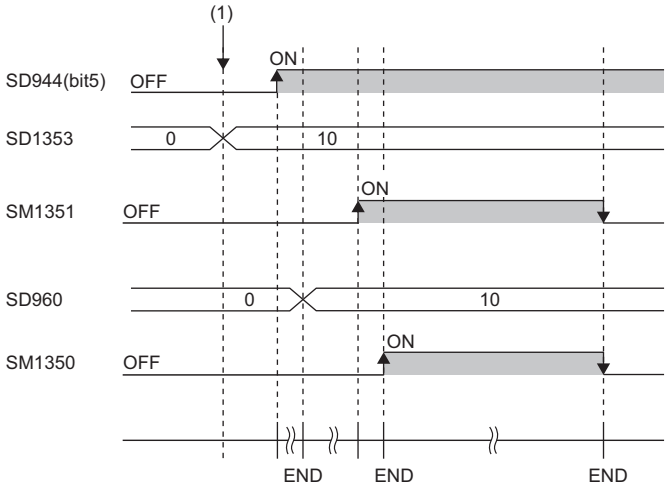
If data folders are created more than the capacity of the SD memory card before the number of them reaches the upper limit value for the number of CPU module backup data, the backup processing is completed with an error and the data cannot be backed up to reach the upper limit value.

■Operation of the special relay and special register

The following figure shows the operations of the special relay and special register of when the upper limit value for the number of CPU module backup data has been set.

Check the following at the timing on when the bit 5 of SD944 (Enable the upper limit value for the number of CPU module backup data) is turned on, and enable the upper limit value for the number of CPU module backup data.

- The backup processing has not been executed (when no backup data folder (CPU data folder) exists in the SD memory card).
- The value set in SD1353 (Upper limit value setting for the number of CPU module backup data) is within the range (1 to 100).



(1) The upper limit value for the number of CPU module backup data is set. (0 → 10)

Backup processing triggered by turning on SM1351

Data in the CPU module is backed up at a desired timing.

Operating procedure

Data in the CPU module is backed up by turning on SM1351.

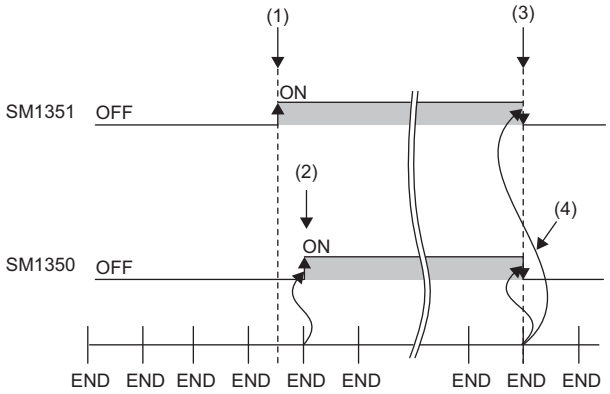
1. To set the upper limit value for the number of CPU module backup data, follow the steps below.

- Set SD1353 (Upper limit value for the number of CPU module backup data).
- Set SM960 (Upper limit operation setting flag for the number of CPU module backup data).
- Turn on the bit 5 of SD944 (Backup function setting).

For the upper limit value for the number of CPU module backup data, refer to the following.

☞ Page 247 Setting the upper limit value for the number of CPU module backup data

2. Turn on SM1351 (CPU module data backup execution request).



(1) A backup execution request is sent.
(2) The system turns on SM1350 (CPU module data backup status flag).
(3) The system turns off SM1351 after the backup processing is completed.
(4) The system turns off SM1350.

If the backup processing is completed with an error and SM953 (CPU module data backup error check flag) turns on, check SD953 (Backup error cause), take actions, and then back up the data again as required.

Point

The execution status of the backup processing can be checked in SD1350 (Number of uncompleted folders/files of CPU module data backup/restoration) and SD1351 (Progression status of CPU module data backup/restoration). (☞ Page 246 Progress of the backup/restoration processing)

Automatic backup using SD944

Data in the CPU module can be automatically backed up at a preset execution timing. Set the execution timing of the automatic backup with SD944 (Backup function setting). Multiple execution timing settings can be set.

Bit pattern of SD944	Execution timing
Bit 0: On	On the time set in SD948 and SD949 on the day set in SD947
Bit 1: On	On the time set in SD950 and SD951 on the day of the week set to SD952
Bit 15: On	When a stop error has occurred in the CPU module

Point

Since the special register set for the automatic backup is a latch area, setting data is held.

Retrying the automatic backup

Set whether to retry the automatic backup when the automatic backup is executed during execution of an operation or function that cannot be executed (☞ Page 255 Operations and functions that cannot be performed).
The retry interval is three minutes and the number of retries is 10.

Special relay/Special register	Description
SM961	This relay turns on when the retry of the automatic backup is completed with an error after the retry of the automatic backup for the number of retries was attempted. This relay turns off at the start of the automatic backup. (This relay does not turn off when SM1351 (CPU module data backup execution request) is on.) • Off: Retry not executed/Retry being executed • On: Retry failed
SM1356	This relay turns on during execution of the retry of the automatic backup. This relay turns on at the start of the retry of the automatic backup and turns off when the retry of the automatic backup is started or the retry of the automatic backup for the number of retries is attempted without execution of an operation or function that cannot be executed. • Off: Retry is not being executed. • On: Retry is being executed.
Bit 10 of SD944	Set whether to retry the automatic backup. • Off: Retry is not executed. • On: Retry is executed.

The setting of the bit 10 of SD944 (Backup function setting) cannot be changed during execution of the automatic backup.
The retry setting of the automatic backup is enabled at the following timing.

- When the bit 0, bit 1, or bit 15 of SD944 turns on

Operating procedure (when date and time are specified)

Data is automatically backed up on the specified date and time.

1. Set the upper limit value for the number of CPU module backup data. (The setting method and operating procedure for the upper limit value are the same as those for the upper limit value for the backup processing triggered by turning on SM1351. (☞ Page 249 Operating procedure))
2. Set the date and time with SD947 to SD949.

Special register	Description
SD947	Set the date when the data is to be automatically backed up.
SD948	Set the time (hour) when the data is to be automatically backed up.
SD949	Set the time (minute) when the data is to be automatically backed up.

3. To retry the automatic backup, turn on the bit 10 of SD944 (Backup function setting). For the retry of the automatic backup, refer to the following.

☞ Page 250 Retrying the automatic backup

4. Turn on the bit 0 of SD944.

If the backup processing is completed with an error and SM953 (CPU module data backup error check flag) turns on, check SD953 (Backup error cause), take actions, and then back up the data again as required.

Restriction

- In months that does not have the specified date, the automatic backup is not executed. For example, when SD947 has been set to "31", the months when the automatic backup is executed are January, March, May, July, August, October, and December.
- If the time less than 1 hour from the start time of daylight saving time has been set with the daylight saving time function, the automatic backup is not executed.

Operating procedure (when time and day of the week are specified)

Data is automatically backed up on the specified time on the specified day of the week.

1. Set the upper limit value for the number of CPU module backup data. (The setting method and operating procedure for the upper limit value are the same as those for the upper limit value for the backup processing triggered by turning on SM1351. (☞ Page 249 Operating procedure))
2. Set the time and day of the week with SD950 to SD952.

Special register	Description
SD950	Set the time (hour) when the data is to be automatically backed up.
SD951	Set the time (minute) when the data is to be automatically backed up.
SD952	Set the day of the week when the data is to be automatically backed up. b0: Sunday, b1: Monday, b2: Tuesday, b3: Wednesday, b4: Thursday, b5: Friday, b6: Saturday

3. To retry the automatic backup, turn on the bit 10 of SD944 (Backup function setting). For the retry of the automatic backup, refer to the following.

☞ Page 250 Retrying the automatic backup

4. Turn on the bit 1 of SD944.

If the backup processing is completed with an error and SM953 turns on, check SD953, take actions, and then back up the data again as required.

Restriction

If the time less than 1 hour from the start time of daylight saving time has been set with the daylight saving time function, the automatic backup is not executed.

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

Data is automatically backed up when a stop error occurs in the CPU module.

1. Set the upper limit value for the number of CPU module backup data. (The setting method and operating procedure for the upper limit value are the same as those for the upper limit value for the backup processing triggered by turning on SM1351. (☞ Page 249 Operating procedure))
2. To retry the automatic backup, turn on the bit 10 of SD944 (Backup function setting). For the retry of the automatic backup, refer to the following.

☞Page 250 Retrying the automatic backup

3. Turn on the bit 15 of SD944.

If the backup processing is completed with an error and SM953 turns on, check SD953, take actions, and then back up the data again as required.

Restriction

If a major error has occurred, the automatic backup may not be performed.

Checking backup errors

When an error has occurred, a diagnostic error is not detected and an error code is stored in SD953 (Backup error cause). (☞ Page 512 List of error codes)

Precautions

The following describes the precautions for the backup function.

Prohibited operation during execution of the backup processing

Do not perform the following operations during execution of the backup processing.

- Removing and inserting the SD memory card
- Powering off or resetting the CPU module

The above mentioned operations leave the backup data in the SD memory card in an incomplete state which is middle of the backup processing. Do not use these data for a restoration. If these data are used, the restoration is completed with an error.

Suspending backup processing

The following operation can suspend a backup processing.

- Setting the SD memory card forced disable

Suspending a backup processing leaves the backup data in the SD memory card in an incomplete state which is middle of the backup processing. Do not use these data for a restoration. If these data are used, the restoration is completed with an error.

Device/label data

To execute the backup processing, do not change device/label data during execution of the processing. Since device/label data is divided into multiple scans and backed up, changes in the device/label data may cause data inconsistency.

When labels accessible from external devices have been set in the CPU module

For the CPU module with firmware version "24" or earlier, the backup cannot be executed if labels accessible from external devices have been set in the CPU module by access label settings from external devices.
Check that no such labels have been set in the CPU module and then execute the backup processing.

When parameter settings were changed before execution of the backup processing

When programs or parameter settings were changed, check that operations are performed with the new programs and parameter settings and then execute the backup function. If the backup processing is executed without the check of the operations with the new programs and parameter settings, the restoration processing may not be executed.

Changing backup target data

Do not change backup target data in the CPU module during execution of the backup processing. If the target data was changed during execution of the backup processing, the changes will not applied.

Special relay and special register that function as flags to execute other functions

Before executing the backup processing, turn off the special relay and special register that function as flags to execute other functions. If the backup processing is executed when they are on, the corresponding function request may turn on and the function may be executed at the restoration of data in the special relay and special register.

Data protected by security functions

■File password function

Unlock the file passwords of the files in the backup target CPU module. If any files to which file passwords have been set exist in the CPU module, the files are not backed up.

■Security key authentication function

Locked programs are backed up in the locked state regardless of whether security keys have been written or not.

Timing for setting the upper limit value for the number of CPU module backup data

Set the upper limit value for the number of CPU module backup data before execution of the backup processing.
When the backup processing has been executed (a CPU data folder exists in the SD memory card) and the bit 5 of SD944 (Enable the upper limit value for the number of CPU module backup data) is turned on, an error will occur.
Even though the backup processing has been executed, turning on the bit 5 of SD944 can disable the upper limit value for the number of CPU module backup data. To set the upper limit value for the number of CPU module backup data again, replace the SD memory card with another one where no CPU data folder exist, or turn off and on the bit 5 of SD944 after deleting the backup data folder.

SFC program status

Do not change the status of the SFC program, such as step active status and transition conditions during execution of the backup processing. If the status of the SFC program was changed, the backup processing is completed with an error.

Time required for completing the backup processing

The backup processing takes more time depending on the size of data or the number of folders/files stored in the CPU module.

Operations and functions that cannot be performed

While the following operations or functions are being executed, the backup processing cannot be executed.

The following operations and functions cannot be executed during execution of the backup processing.

Operation or function		
Operation from the engineering tool	Initializing the CPU built-in memory/SD memory card	
	Clearing values (Devices, labels, file registers, latches)	
	Writing data to the programmable controller (including online change of files)	
	Deleting data in the programmable controller	
	User data operation	Writing user data
		Deleting user data
		Creating a folder
		Deleting a folder
		Changing a folder name
	Online change (online change (ladder block))	
	Event history function (Clearing event history)	
	File password function	
	Security key authentication function (Writing/deleting a security key to/in the CPU module)	
	Predefined protocol support function (Writing protocol setting data)	
	Memory dump function (Registering/clearing memory dump)	
Operation using the CPU module logging configuration tool	Data logging function (Writing/deleting a logging setting file, registering/clearing a logging setting, stopping a logging)	
	Deleting a logging file	
Others	• SLMP • MC protocol	Remote latch clear (Remote Latch Clear)
		Creating a new file (New File)
		Writing data to a file (Write File)
		Deleting a file (Delete File)
		Copying a file (Copy File)
		Changing a file attribute (Change File State)
		Changing file creation date (Change File Date)
		Opening a file (Open File)*1
	File transfer from an Ethernet-equipped module (FTP server)	Writing a file (put, mput, pm-write)
		Deleting a file (delete, mdelete)
		Changing a file name (rename)
		Changing a file attribute (change)
	File transfer function (FTP server) of the built-in Ethernet function	
	File transfer function (FTP client) of the built-in Ethernet function	
	Changing an IP address	
	iQ Sensor Solution data backup/restoration function	

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

Backup during execution of the backup processing

The backup processing triggered by turning on SM1351 or automatic backup cannot be executed during execution of the backup processing. (The latter backup processing is ignored.)

27.2 Restoration Function

This function restores backup data in the SD memory card to the CPU module.

Restoration target folder

Set restoration target data among backup data in the SD memory card with SD956 (Restoration target date folder setting) to SD958 (Restoration target number folder setting). The latest backup data can be restored with the bit 13 of SD955 (Restoration function setting).

Special register	Description
Bit 13 of SD955	Set the restoration function setting with bit patterns. Off: Data specified with the restoration target folders is restored. On: The latest data is restored.*1
SD957, SD956	Specify a date folder of the restoration target data in BCD. SD957 (upper): Year, SD956 (lower): Month and date
SD958	Specify the folder number (00001 to 32767) of restoration target data.

*1 The latest data is the backup data with the largest number in the latest date folder.

Restoration target data

Restoration target data is set with SD954 (Restoration target data setting).

Value of SD954	Restoration target data setting
0	All target data
1	Only device/label data
2	All target data excluding device/label data

Restoration of the special relay and special register

Set whether to restore the special relay and special register with the bit 14 of SD955.

Bit 14 of SD955	Restoration target data setting
Off	The special relay and special register are not restored.
On	The special relay and special register are restored.

The following special relay/register areas, which are used for the CPU module data backup/restoration function, will not be restored even though bit14 of SD955 is on.

- SM953, SM959, SM961, SM1350, SM1351, SM1353, SM1354, SM1356, SD953, SD959, SD1350, SD1351, SD1353

Operation setting after restoration

As an operation after the restoration processing, set whether to operate the CPU module in the status at the backup processing or to operate the CPU module in the initial status with the bit 15 of SD955. The following table lists the operations of each item to be performed according to the operation setting after restoration.

Item	Operation setting after restoration	
	Operating the CPU module in the status at data backup (bit 15 of SD955 = On)	**Operating the CPU module in the initial status (bit 15 of SD955 = Off)**
Device initial value	The device initial value is not set after the restoration processing.	The device initial value is set after the restoration processing. (The device data at data backup will be overwritten with the device initial value.)
Initial values of global/local labels	The initial values of global/local labels are not set after the restoration processing.	The initial values of global/local labels are set after the restoration processing. (The label data at data backup will be overwritten with the initial value of the initial global/local value.)
SFC program*1	When a continuous start has been specified before data backup, the SFC program is restarted from the block and step where the processing was stopped after the restoration processing.	Even though a continuous start has been specified before data backup, the SFC program is not restarted from the block and step where the processing was stopped after the restoration processing.
Event history	The event history at data backup is set.	The event history at data backup is not set, a new file is created and a restoration event is registered.

*1 When the firmware version of the CPU module at the backup processing is different from that of at the restoration processing, the SFC program will be started from the block 0 and step 0 regardless of the operation setting after the restoration.

Note that this setting is invalid since the device initial value file, initial global label value file, initial local label value file, event history file, and SFC program are not restored when the value in SD954 (Restoration target data setting) is 1 (restoration target data are only device/label data).

Point

For the operation setting after restoration, specify the operation at a completion of restoration.
When the CPU module is switched from STOP to RUN, values of devices are changed according to the operation of the device memory at an operating status change of the CPU module. (☞ Page 72 Operation Processing When Operating Status of the CPU Module Changed)

Restoration processing triggered by turning on SM1354

Backup data is restored at a desired timing.

Point

Use the restoration function by turning on SM1354 to check the backup data and to test before actual operation.

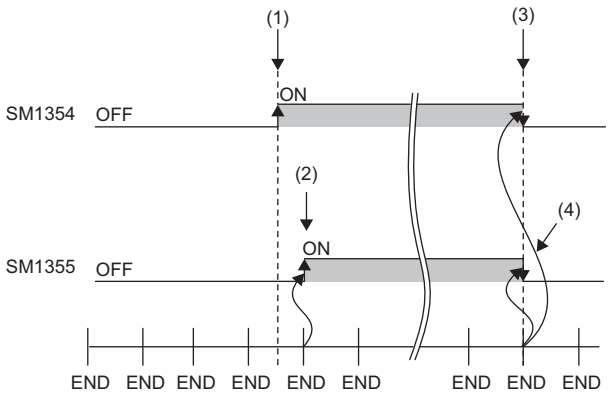
To operate the system using the backup data, use the automatic restoration with SD955. (☞ Page 259 Automatic restoration using SD955)

Restriction

The restoration processing triggered by turning on SM1354 (CPU module data restoration execution request) can be executed only when the CPU module is the STOP state.

Operating procedure

1. Set restoration target data with SD954 (Restoration target data setting).
2. Set restoration target folders with SD956 (Restoration target date folder setting) to SD958 (Restoration target number folder setting). (These settings are not required when the bit 13 of SD955 (Restoration function setting) is on.)
3. Set each setting with the bit 13 to 15 of SD955.
4. Set the CPU module to the STOP state.
5. Turn on SM1354.



(1) A restoration execution request is sent.
(2) The system turns on SM1353 (CPU module data restoration status flag).
(3) The system turns off SM1354 after the restoration processing is completed.
(4) The system turns off SM1353.

If the restoration processing is completed with an error and SM959 (CPU module data restoration error check flag) turns on, check SD959 (Restoration error cause), take actions, and then restore the data again as required.

Point

The execution status of the restoration processing can be checked in SD1350 (Number of uncompleted folders/files of CPU module data backup/restoration) and SD1351 (Progression status of CPU module data backup/restoration). (☞ Page 635 List of Special Register Areas)

Automatic restoration using SD955

Backup data is automatically restored when the CPU module is powered on or is reset.

Initialization at the automatic restoration

Set whether or not to initialize drives other than the SD memory card at execution of the automatic restoration with the bit 1 of SD955 (Restoration function setting).

This setting is enabled only when SD954 (Restoration target data setting) is set to 0 (All target data).

Bit 1 of SD955	Restoration target data setting
Off	Not initialized
On	Initialized

Operating procedure

1. Set restoration target data with SD954.
2. Set restoration target folders with SD956 (Restoration target date folder setting) to SD958 (Restoration target number folder setting). (These settings are not required when the bit 13 of SD955 (Restoration function setting) is on.)
3. Set each setting with the bit 1 and bit 13 to 15 of SD955.
4. Turn on the bit 0 of SD955.
5. Power on or reset the CPU module.

If the restoration processing is completed with an error and SM959 (CPU module data restoration error check flag) turns on, check SD959 (Restoration error cause), take actions, and then restore the data again as required.

Point

- Since the special register set for the automatic restoration is a latch area, setting data is held.
- SD955 holds its setting even after the CPU module is powered on or is reset. Thus, if the CPU module is powered on or is reset while the bit 0 of SD955 is on, the automatic restoration is executed again. Not to perform the automatic restoration when the CPU module is powered on or is reset next time, turn off the bit 0 of SD955 after a restoration is completed and then power on or reset the CPU module.

Checking restoration errors

- When an error occurs in the restoration processing triggered by turning on SM1354, a diagnostic error is not detected and an error code is stored in SD959 (Restoration error cause). (☞ Page 512 List of error codes)
- When an error occurs at the automatic restoration using SD955, a diagnostic error is detected and an error code is stored in SD959. (☞ Page 512 List of error codes)

Precautions

The following describes the precautions for the restoration function.

Prohibited operation during execution of the restoration processing

Do not perform the following operations during execution of the restoration processing.

- Removing and inserting the SD memory card
- Powering off or resetting the CPU module

The above mentioned operations leave the data in the CPU module in an incomplete state which is middle of the restoration processing. Do not run the CPU module with this incomplete state. Doing so may cause an unintended operation. Restore the data again or format each drive in the CPU module, and clear devices/labels before writing programs or parameters to the programmable controller.

Suspending the restoration processing

The following operation can suspend a restoration processing.

- Setting the SD memory card forced disable

Suspension during a restoration leaves the data in the CPU module in an incomplete state which is middle of the restoration processing. Do not run the CPU module with this incomplete state. Doing so may cause an unintended operation. Restore the data again or format each drive in the CPU module, and clear devices/labels before writing programs or parameters to the programmable controller.

Model name of the CPU module to which data is restored

Restore the data to the CPU module whose model name is the same as that of the CPU module where the backup data has been stored. If not, the data cannot be restored.

When labels accessible from external devices have been set in the CPU module

For the CPU module with firmware version "24" or earlier, the restoration cannot be executed if labels accessible from external devices have been set in the CPU module by access label settings from external devices.
Check that no such labels have been set in the CPU module and then execute the restoration processing.
For the CPU module with firmware version "25" or later, the restoration by turning on SM1354 cannot be executed if labels accessible from external devices have been set in the CPU module or in a backed up data from external devices. When the restoration is required, execute the automatic restoration using SD955.

Automatic restoration using SD955 and functions that cannot be executed

Do not set both of the automatic restoration using SD955 and the boot operation at a time.
If the CPU module is powered on or is reset with both of the automatic restoration using SD955 and the boot operation set, the automatic restoration using SD955 does not function.

Status of the restoration destination CPU module

If the status of the restoration destination CPU module differs from that of the CPU module at the backup processing (such as programs or parameters), the restoration may not be executed.
When the backup data to be restored is backed up in a different status from that of the restoration destination CPU module, store 0 (All the target data) to SD954 (Restoration target data setting) and execute the automatic restoration.

When the same name folder or file exists in the restoration target CPU module

If the name of a folder or file in the restoration target CPU module and the name of a folder or file in backup data are identical, the folder or file in the module will be overwritten by that in the backup data.

Changing the operating status during execution of restoration

During execution of the restoration processing, the CPU module remains in the STOP state even though the RUN/STOP/RESET switch is changed from the STOP to RUN position or the remote RUN or the remote PAUSE is executed. If the operating status of the CPU module is changed, the status will changes to the set status after the restoration processing is completed.

Time required for completing the restoration processing

The restoration processing takes more time depending on the number of backup data sets (folders), file size, and the number of files in the SD memory card.
In a multiple CPU system, if the automatic restoration has taken time and an error has occurred in another CPU module, another error may occur in the CPU module to which the automatic restoration was executed after the completion of the restoration.
In that case, shorten the time for restoration. Change the setting of SD954 (Restoration target data setting) to only the device data that is to be cleared at a system start-up, and execute the automatic restoration again.

Monitoring stop at restoration

Stop monitoring before executing the restoration processing.
When the restoration processing is executed, programs, parameters, and device/label values may not be properly monitored because they are changing.

Reflecting restored data

Some of the parameters are reflected only when the CPU module is powered on or is reset. If the restoration processing is executed while the CPU module is in the STOP state and then is switched to the RUN state, the CPU module may not operate with the backup data. In that case, power on or reset the CPU module. For device/label data, since device/label data except for latch-specified devices/labels is initialized when the CPU module is powered on or is reset, restore the device/label data again as required.

Abnormal completion of restoration

Since the restoration processing will be completed with an error, do not execute the restoration processing in the following cases.

- The name of a file in the restoration destination CPU module and the name of a file in backup data are identical, and a file password has been set.
- Data in a backup folder has been deleted. (Do not delete the data in backup folders that are likely to be used for restoration.)
- Backup data has problems. (Backup data has been changed or the CPU module was powered off during execution of the backup processing.)

Instruction executed at the rising/falling edge when data is restored

When the program file is restored by turning on SM1354, the execution status of instructions is undefined and the instructions executed at the rising/falling edge may not work properly.
To operate the system by restoring the backup data, use the automatic restoration with SD955.

When the special relay and special register are restored

When a restoration is operated with the setting to restore the special relay and special register, the system will be operated with the values (time, day of the week, day, time, and the upper limit value of the number of backup data for the automatic backup set to the backup function setting) before restoration. (The backup function setting will not be re-set although the special register areas are overwritten by the restoration.)
To operate the system with the restored backup function setting, set the backup function setting again.

Data protected by security functions

■File password function

Unlock the file passwords of the files in the backup target CPU module. If any files to which file passwords have been set exist in the CPU module, the files are not restored.

■Security key authentication function

Locked programs can be restored regardless of whether security keys have been written or not. However, when the security key has not been written to the CPU module after the restoration processing, the program cannot be executed. Restore unlocked backup data or set the same security key.

When the SFC program is restarted from where the program was stopped

Specify the continue start. When the continue start has not been specified, the SFC program will be started from the block 0 and step 0 even though the bit 15 of SD955 is on (the continue start is executed).

When the IP address change function is used

If the backup processing is executed with the IP address stored in the IP address storage area (system memory), the IP address will be changed at the following timing in the restoration processing.

- Restoration processing triggered by turning on SM1354: When the CPU module is powered on or is reset after the restoration processing
- Automatic restoration using SD955: When the restoration processing is executed

Operations and functions that cannot be performed

While the following operations or functions are being executed, the restoration processing cannot be executed.

The following operations and functions cannot be executed during execution of the restoration processing.

Operation or function		
Operation from the engineering tool	Initializing the CPU built-in memory/SD memory card	
	Clearing values (Devices, labels, file registers, latches)	
	Reading data from the programmable controller	
	Writing data to the programmable controller (including online change of files)	
	Verifying data with the programmable controller	
	Deleting data in the programmable controller	
	User data operation	Reading user data
		Writing user data
		Deleting user data
		Creating a folder
		Deleting a folder
		Changing a folder name
	Online change (online change (ladder block))	
	Event history function (Updating event history data, clearing event history)	
	File password function	
	Security key authentication function (Writing/deleting a security key to/in the CPU module)	
	Predefined protocol support function (writing/reading/verifying protocol setting data)	
	Memory dump function (Memory dump setting/reading results, registering/clearing memory dump)	
Operation using the CPU module logging configuration tool	Data logging function (Writing/reading/deleting a logging setting file, registering/clearing a logging setting, stopping a logging)	
	Deleting a logging file	
Others	• SLMP • MC protocol	Remote latch clear (Remote Latch Clear)
		Creating a new file (New File)
		Writing data to a file (Write File)
		Deleting a file (Delete File)
		Copying a file (Copy File)
		Changing a file attribute (Change File State)
		Changing file creation date (Change File Date)
		Opening a file (Open File)
		Reading a file (Read File)
	File transfer from an Ethernet-equipped module (FTP server)	Reading a file (get, mget)
		Writing a file (put, mput, pm-write)
		Deleting a file (delete, mdelete)
		Changing a file name (rename)
		Changing a file attribute (change)
	File transfer function (FTP server) of the built-in Ethernet function	
	File transfer function (FTP client) of the built-in Ethernet function	
	Changing an IP address	
	iQ Sensor Solution data backup/restoration function	

Operation of when the data logging function is used

If data is backed up during execution of the data logging function and the function has been set to be started automatically when the operating status of the CPU module is changed to RUN, the data logging function will be automatically executed when the status of the CPU module changes to RUN after the restoration processing. To restart the data logging function after the restoration processing without the above setting, use the CPU module logging configuration tool.

Restoration during execution of the restoration processing

The restoration processing triggered by turning on SM1354 cannot be executed during execution of the restoration processing. (The latter restoration processing is ignored.)

When using the CPU module database access function

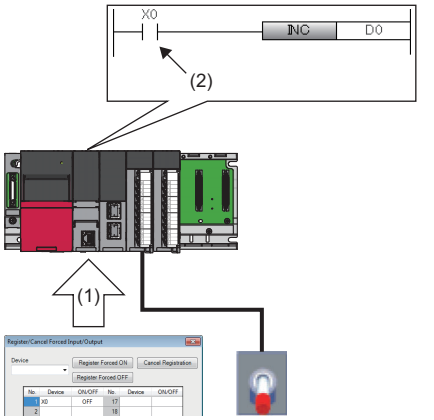
When the data is backed up with the CPU module data backup/restoration function and restored by turning on SM1354 while using the CPU module database access function, power off and on or reset the CPU module regardless of the completion status of the restoration. (For the automatic restoration by SD955, the CPU module is not required to be powered off and on or reset.)

28 EXTERNAL INPUT/OUTPUT FORCED ON/OFF FUNCTION

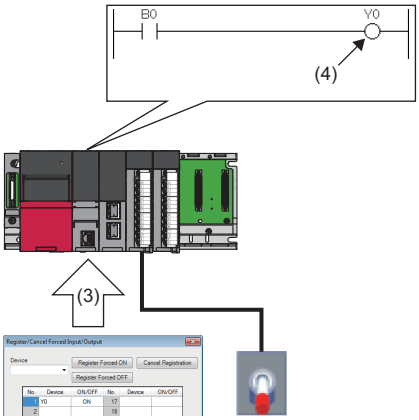
RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnSFCPU (Standard) RnSFCPU (Safety)

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

(External input)



(External output)



(1) Turn off X0 forcibly.
(2) The input device is turned off regardless of the on/off state of the external input.
(3) Turn on Y0 forcibly.
(4) The external output is turned on regardless of the operation result of the program.

Point

The external input/output forced on/off function is available for MELSEC iQ-R series modules and Q series modules mounted on an extension base unit.

Restriction

Before executing the external input/output forced on/off function, check the versions of the CPU module and engineering tool used. (☞ Page 779 Added and Enhanced Functions)

Devices that allow forced on/off registration

The following lists the devices that allow forced on/off registration.

Device	Range
Input	X0 to X2FFF (12288 points)
Output	Y0 to Y2FFF (12288 points)

Number of device points that allows forced on/off registration

A maximum of 32 points can be registered for input devices and output devices in total.

Inputs/outputs for which forced on/off can be set

The following describes the inputs/outputs for which forced on/off can be set.

■Input

After the refreshed data from the module is reflected, the input devices for which forced on/off registration is set are forcibly turned on or off.

■Output

The following external outputs are turned on or off by refreshing output devices that have been forcibly turned on or off.[*1]

- Refresh target output of modules mounted on the base unit
- Link refresh target output of network modules
- Link refresh target output of CC-Link IE Field Network Basic

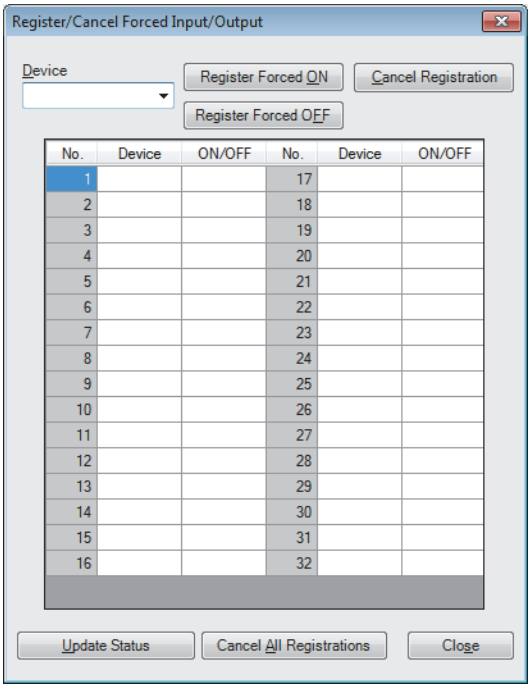
*1 When no modules are mounted on the base unit (no refresh targets), no data is output to the external outputs.

Operation method of forced on/off

Use the engineering tool for the forced on/off operation.

[Debug] ⇨ [Register/Cancel Forced Input/Output]

Window



Displayed items

Item	Description
Device	Enter target devices (X, Y).
[Register Forced ON] button	Registers forced on for the entered devices (X, Y).
[Register Forced OFF] button	Registers forced off for the entered devices (X, Y).
[Cancel Registration] button	Cancels forced on/off for the registered devices (X, Y).
[Update Status] button	Displays the latest on/off states.
[Cancel All Registrations] button	Batch-cancels forced on/off for the registered devices (X, Y).

Behavior in forced on/off registration

The following describes the behavior in forced on/off registration.

■Behavior of an input device

Registering forced on/off turns on or off the input device regardless of the status of the external input. When an input device for which the forced on/off has been registered is changed in the program, the input device is turned on and off in accordance with the operation result of the program.

Operation	Change in the input device in the program	Behavior of an input device
Forced on registration	Changed	• The input device is on when the program operation at every scan starts. • The input device is turned on or off in accordance with the operation result of the program after the program operation at every scan started.*1
	Not changed	The on state is held.
Forced off registration	Changed	• The input device is off when the program operation at every scan starts. • The input device is turned on or off in accordance with the operation result of the program after the program operation at every scan started.*1
	Not changed	The off state is held.

*1 If the forced on/off is registered for a device set to perform link refresh, the device value may be updated at the timing of refresh. Thus, a different value from the operation result may be displayed when the device value is monitored (performed in END processing).

■Behavior of an output device

Registering forced on/off turns on or off the external output regardless of the operation result of the program. When an output device has been changed in the program, the output device is turned on and off in accordance with the operation result of the program. Therefore, the on/off states between the output device and external output may differ.

Operation	Change in the output device in the program	Behavior of outputs	
		Behavior of an output device	Behavior of an external output
Forced on registration	Changed	• The output device is on when the program operation at every scan starts. • The output device is turned on or off in accordance with the operation result of the program after the program operation at every scan started.*1	The external output is turned on regardless of the operation result of the program.
	Not changed	The on state is held.	
Forced off registration	Changed	• The output device is off when the program operation at every scan starts. • The output device is turned on or off in accordance with the operation result of the program after the program operation at every scan started.*1	The external output is turned off regardless of the operation result of the program.
	Not changed	The off state is held.	

*1 If the forced on/off is registered for a device set to perform link refresh, the device value may be updated at the timing of refresh. Thus, a different value from the operation result may be displayed when the device value is monitored (performed in END processing).

CPU module operating status

Forced on/off registration is allowed regardless of the CPU module operating status. However, when a stop error has occurred, the external outputs are turned off regardless of the forced on/off registration setting. While the operating status of the CPU module is STOP due to a stop error, forced on/off is enabled only for the input devices. If the CPU module is powered off and on or is reset, all the forced on/off registration information will be canceled.

Forced on/off timing

The following table lists the timing to reflect forced on/off registration information to the input/output devices or external outputs.

Inputs/outputs for which forced on/off can be set	Reflection timing for the input devices	Reflection timing for the output devices or external outputs
Input/output of the modules mounted on the base unit	• END processing (at input refresh) • At COM instruction execution (at input refresh) • At instruction execution using the direct access input (DX) (LD, LDI, AND, ANI, OR, ORI, LDP, LDF, ANDP, ANDF, ORP, ORF, LDPI, LDFI, ANDPI, ANDFI, ORPI, ORFI)*1 • At execution of the RFS instruction and MTR instruction • At execution of instructions used in the interrupt by the system (UDCNT1, UDCNT2, SPD) • At program execution*2 • At execution of the inter-module synchronization cycle program (I44) • At execution of the multiple CPU synchronization program (I45) and non-execution of the multiple CPU synchronization program (I45) (at END processing)*3	• END processing (at output refresh) • At COM instruction execution (at output refresh) • At instruction execution using the direct access output (DY) (OUT, SET, DELTA(P), RST, PLS, PLF, FF, MC, SFT(P))*1 • At execution of the RFS instruction and MTR instruction • At execution of instructions used in the interrupt by the system (PLSY, PWM) • At program execution*2 • At execution of the inter-module synchronization cycle program (I44)
Input/output of the CPU module assigned to LX and LY of the CC-Link IE Controller Network module or MELSECNET/H module	• END processing (at link refresh of the CC-Link IE Controller Network module or MELSECNET/H module) • At COM instruction execution (at link refresh of the CC-Link IE Controller Network module or MELSECNET/H module) • At ZCOM instruction execution (at link refresh of the CC-Link IE Controller Network module or MELSECNET/H module)	
Input/output of the CPU module assigned to RX and RY of the CC-Link module	• END processing (at link refresh) • At COM instruction execution (at link refresh) • At ZCOM instruction execution (at link refresh)	
Input/output of the CPU module assigned to RX and RY of the CC-Link IE Field Network module	• END processing (at link refresh) • At COM instruction execution (at link refresh) • At ZCOM instruction execution (at link refresh) • At execution of the inter-module synchronization cycle program (I44)	
Input/output of the CPU module assigned to RX and RY of the CC-Link IE Field Network Basic	• END processing (at link refresh) • At COM instruction execution (at link refresh)	

*1 For the precautions for using the direct access device, refer to the following.
☞ Page 271 Precautions

*2 At input/output refresh execution where input/output refresh is registered for each program, and for interrupt programs.

*3 When an output device or external output is specified as a refresh device in multiple CPU refresh and the forced on/off function is executed, the device is not forcibly turned on or off.

Checking the forced on/off execution status

The execution status of the forced on/off can be checked in the following ways.

■Engineering tool

The execution status can be checked with the [Update Status] button of the engineering tool. (☞ Page 267 Operation method of forced on/off)

■FUNCTION LED

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

Point

The LED indicator follows the status of the external input/output forced on/off function even when the LED Display Setting has been set and other functions that use the FUNCTION LED are being executed. (☞ Page 232 LED display setting)

The display status of the FUNCTION LED can be checked with "Module Information List" in the module diagnostics. (GX Works3 Operating Manual)

■Special register

SD1488 (Debug function usage status) can be used to check whether the external input/output forced on/off function is used. (☞ Page 635 List of Special Register Areas)

Behavior in cancellation of forced on/off

Forced on/off registration can be canceled for each input/output device individually.

■Behavior of the device

Inputs/outputs for which forced on/off can be set		Change in input/output devices in the program	
		Changed	Not changed
Input	Input from the modules mounted on the base unit	The input device is turned on or off in accordance with the on/off state of the modules.	
	Input from LX of CC-Link IE Controller Network or MELSECNET/H	The input device is turned on or off in accordance with the on/off state refreshed from CC-Link IE Controller Network and MELSECNET/H.	
	Input from RX of CC-Link	The input device is turned on or off in accordance with the on/off state refreshed from CC-Link.	
	Input from RX of CC-Link IE Field Network	The input device is turned on or off in accordance with the on/off state refreshed from CC-Link IE Field Network.	
	Input from RX of CC-Link IE Field Network Basic	The input device is turned on or off in accordance with the on/off state refreshed from CC-Link IE Field Network Basic.	
	Input other than above (input without modules actually mounted)	The input device is turned on or off in accordance with operation result of the program.	The input device is turned on or off in accordance with the registered on/off state.
Output	Output to the modules mounted on the base unit	The operation result of the program is output.	Data is output in accordance with the registered on/off state.
	Output to LY of CC-Link IE Controller Network or MELSECNET/H		
	Output to RY of CC-Link		
	Output to RY of CC-Link IE Field Network		
	Output to RY of CC-Link IE Field Network Basic		
	Output other than above (output without modules actually mounted)	The output device is turned on or off in accordance with operation result of the program. (Refresh to external output is not executed.)	The output device is turned on or off in accordance with the registered on/off state. (Refresh to external output is not executed.)

■CPU module operating status

Forced on/off registration can be canceled regardless of the CPU module operating status.

■LED status

The following describes the LED status after forced on/off registration is canceled.

Forced on/off registration status of when the registration is canceled	FUNCTION LED status
Registered on/off information is remaining.	Flashing (every 200ms)
No registered on/off information is remaining.	Off*1

*1 When other functions that use the FUNCTION LED are being executed, the LED indicator changes its status in accordance with the status of the functions.

Behavior in batch-cancellation of forced on/off registrations

All the forced on/off registrations can be canceled in a batch.

■Behavior of the device

The behavior of the device is the same as that of cancellation of forced on/off (for each device). (☞ Page 270 Behavior of the device)

■CPU module operating status

The behavior of the device is the same as that of cancellation of forced on/off (for each device). (☞ Page 270 CPU module operating status)

■LED status

The following describes the LED status after forced on/off registration is canceled in a batch.

Forced on/off registration status of when the registration is canceled	FUNCTION LED status
No registered on/off information is remaining.	Off*1

*1 When other functions that use the FUNCTION LED are being executed, the LED indicator changes its status in accordance with the status of the functions.

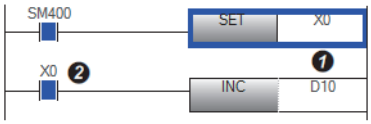
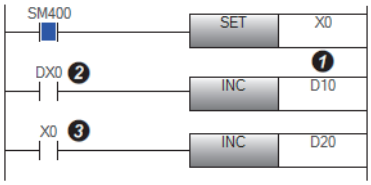
Precautions

The following describes precautions for using the external input/output forced on/off function.

- Multiple engineering tools connected to the network can be used to register forced on/off for the same CPU module. In this case, note the following.
  - The forced on/off state registered last is handled as the on/off state of input/output devices.
  - Since the on/off state displayed in engineering tools may differ from that of the CPU module, update the on/off state displayed in engineering tools.
- If a direct device is used in a program, forced on/off registration is reflected when an instruction is executed.

Ex.

When X0 is forcibly turned off

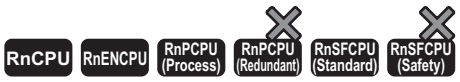
When the input device (X0) is used	When the direct input device (DX0) is used
 ❶ Set X0 by using the SET instruction. ❷ Turn on X0 because the SET instruction is being executed.	 ❶ Set X0 by using the SET instruction. ❷ Reflect the registered off information. ❸ Turn off X0 although the SET instruction is being executed.

- When the external input/output forced on/off function is executed while the program of high-speed internal timer interrupt (I48, I49) is executed, the interval of programs of high-speed internal timer interrupt is short, and a large number of forced on/off is registered, the scan time will increase and a WDT error may occur.
- Execution of interrupt programs which are executed at a fixed scan may delay depending on the number of forced on/off registrations and the number of refresh points of each refresh processing.
- When forced on/off is registered in the multiple CPU system configuration, registration is allowed for input/output devices[*2] and external outputs regardless of the out-of-group I/O fetch setting[*1]. Note, however, that the devices of other CPU modules and the external outputs controlled by other CPU modules are not turned on or off even if forced on/off is registered from the host CPU for the external outputs controlled by other CPU modules. Only the devices of the host CPU module are turned on or off.

*1 For the out-of-group I/O fetch setting, refer to the following.
☞ Page 274 Out-of-group I/O Fetch setting

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

29 MULTIPLE CPU SYSTEM FUNCTION



• When using the Safety CPU, refer to the following as well.
☞ Page 486 FUNCTIONS

With multiple CPU modules mounted on the base unit, each of the CPU modules controls their own assigned I/O modules and intelligent function modules. In addition, the CPU modules communicate with each other.

Point

- For details on the concept of the multiple CPU system configuration specification (System configuration specifications such as the mounting position of the CPU modules and assignment of CPU number/IO number), refer to the MELSEC iQ-R Module Configuration Manual.
- For the start-up (Setting, operating procedures, etc) of the Multiple CPU system, refer to the GX Works3 Operating Manual.

Restriction

- The startup time of the multiple CPU system may be slowed, depending on the configuration of installed modules, boot operation, functions performed before CPU modules enter into the RUN state (e.g. setting of initial device/label values), the configuration of system parameters and CPU parameters (e.g. the number of programs to run).
- When diagnostics of the SD memory card is performed due to operation such as power-off during access to the SD memory card, the startup time of the multiple CPU system may be slowed.
- Create a program so that only one CPU module accesses the Q series module, if possible. If multiple CPU modules access the Q series module simultaneously, the scan time (including the execution processing of interrupt programs) may be extended due to access waiting time.
- Before using the multiple CPU system function with the RnENCPU, check the versions of the CPU module and engineering tool used. (☞ Page 779 Added and Enhanced Functions)

29.1 Out-of-group I/O Fetch

The access range to the controlled module is different from that to the non-controlled module. To fetch the data which cannot be accessed, use the out-of-group I/O fetch function.

Accessing controlled module

The access range to the controlled module of the CPU module is the same as that to the single CPU system, and I/O refresh for the controlled module and/or reading/writing to buffer memory of the intelligent function module are enabled.

Accessing non-controlled module

Access to the non-controlled module of the CPU module is limited to reading input (DX) and buffer memory of the intelligent function module. Note that On/Off data of input (X)/output (Y) of the non-controlled module can be fetched using the out-of-group I/O Fetch function. (☞ Page 274 Out-of-group I/O Fetch setting)

○: Accessible, ×: Not accessible

Communication method	Access target		Access range	
			When "Import" is selected in the settings for out-of-group I/O fetch	When "Not Imported" is selected in the settings for out-of-group I/O fetch
Communication through refresh	Input (X)	Read to another CPU module	○*1	× (Non-processing)
	Output (Y)	Write to another CPU module	× (Non-processing)	
		Read to another CPU module	○*1	
	The buffer memory of the intelligent function module	Read	× (Cannot be specified from engineering tools)	× (Cannot be specified from engineering tools)
		Write		
	Link direct device	Read	× (Cannot be specified from engineering tools)	× (Cannot be specified from engineering tools)
		Write		
Communication through direct access	Input (DX)	Read to another CPU module	○	○
	Output (DY)	Write to another CPU module	× (Non-processing)	× (Non-processing)
		Read to another CPU module	× (Cannot be specified)	× (Cannot be specified)
	The buffer memory of the intelligent function module	Read	○	○
		Write	× (An error occurs in the CPU module.)	× (An error occurs in the CPU module.)
	Link direct device	Read	× (An error occurs.)	× (An error occurs.)
		Write		

*1 Cannot read from CPU modules which are synchronized through the inter-module synchronization function.

Precautions

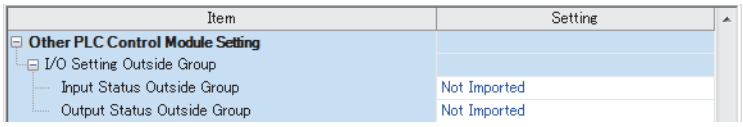
An error cannot is not detected in reading input (DX) for the non-controlled module and buffer memory of the intelligent function module.

Out-of-group I/O Fetch setting

In this menu item, whether or not out-of-group I/O status is fetched can be specified.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Other PLC Control Module Setting] ⇨ [I/O Setting Outside Group]

Window



Displayed items

Item	Description	Setting range	Default
Input Status Outside Group	Specify whether or not out-of-group input status is fetched.	• Not Imported • Import	Not Imported
Output Status Outside Group	Specify whether or not out-of-group output status is fetched.	• Not Imported • Import	Not Imported

Fetching input (X)

When "Import" is selected for "Input Status Outside Group" in "I/O Setting Outside Group", the input refresh before the start of the program operation fetches On/Off data from the input module and/or intelligent function module controlled by other CPU modules.

■Modules from which input (X) can be fetched

Input (X) can be fetched from the following modules mounted on the base/extension base unit.

- Input module
- I/O combined module*1
- Intelligent function module

*1 Data in the areas assigned to the output portion (area not used) are not fetched. On/Off state is maintained.

Restriction

Input (X) cannot be fetched from the Q series modules.

Fetching output (Y)

When "Import" is selected for "Output Status Outside Group" in "I/O Setting Outside Group", the output refresh before the start of the program operation fetches On/Off data output to the output module and/or intelligent function module controlled by other CPU modules into output (Y) for the host CPU module.

■Modules from which output (Y) can be fetched

Output (Y) can be fetched from the following modules mounted on the base/extension base unit.

- Output module
- I/O combined module
- Intelligent function module

Restriction

Input (X) cannot be fetched from the Q series modules.

Output to output/intelligent function module

On/off data cannot be output to non-controlled modules. When turning on or off output of the output module and/or intelligent function module controlled by other CPU modules by the program or others, the output is turned on or off within the CPU module. However, it is not output to output/intelligent function module.

Accessing buffer memory of intelligent function module

■Reading data on buffer memory

The following instructions can be used to read data stored in the buffer memory of the intelligent function module.

- FROM instruction
- Instruction using the CPU module access device (Un\Gn)

■Writing data to buffer memory

The following instructions cannot be used to write data to the buffer memory of the intelligent function module.

- TO instruction
- Instruction using the CPU module access device (Un\Gn)

Accessing a module by using link direct devices

The CPU module can access only to modules under its control by using link direct devices. Non-controlled modules cannot be accessed.

29.2 Operation Settings

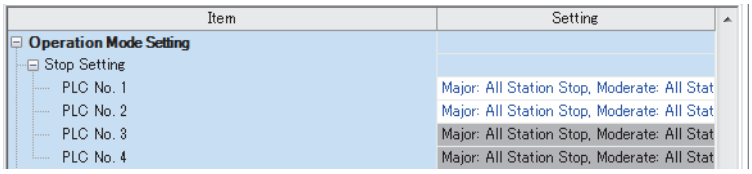
This section describes the operation settings of the multiple CPU system function.

Stop setting

An operating status, whether to stop the operation of all the CPU modules or not, if a major or moderate error occurs in any of the CPU modules is set.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Operation Mode Setting] ⇨ [Stop Setting]

Window



Displayed items

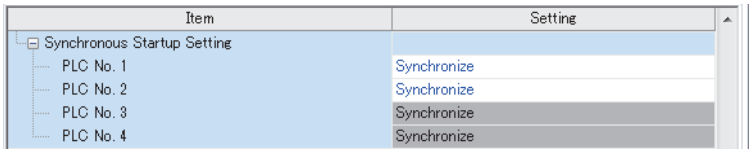
Item	Description	Setting range	Default
PLC No. 1	Set whether to stop the operation of all the CPU modules or not if a major or moderate error occurs in CPU No.1.	• Major: All Station Stop, Moderate: All Station Stop • Major: All Station Stop, Moderate: All Station Continue • Major: All Station Continue, Moderate: All Station Continue	Major: All Station Stop, Moderate: All Station Stop
PLC No. 2	Set whether to stop the operation of all the CPU modules or not if a major or moderate error occurs in CPU No.2.		
PLC No. 3	Set whether to stop the operation of all the CPU modules or not if a major or moderate error occurs in CPU No.3.		
PLC No. 4	Set whether to stop the operation of all the CPU modules or not if a major or moderate error occurs in CPU No.4.		

Synchronous startup setting

Startup time is synchronized among the CPU modules. This setting eliminates the need for an interlock program that monitors the startup time of another CPU module when accessing it. Note, however, that the startup of the entire system delays because the system starts up with the last CPU module.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Operation Mode Setting] ⇨ [Synchronous Startup Setting]

Window



Displayed items

Item	Description	Setting range	Default
PLC No. 1	Set the CPU modules whose startup time is synchronized in the multiple CPU system.	• Synchronize • Do not Synchronize	Synchronize
PLC No. 2			
PLC No. 3			
PLC No. 4			

Point

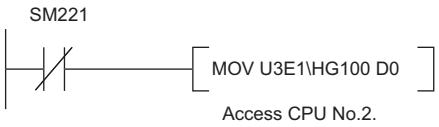
- Group setting for the synchronized start-up is available. For example, a setting in which only CPUs No.1 and No.2 start synchronously within a multiple CPU system with four CPU modules is possible.
- If a reserved (empty) CPU is specified to synchronize, it is skipped and the other CPUs in the group start synchronously.
- This setting is designed to access to each CPU module without interlock in the multiple CPU system. It is not intended to be used for starting operation processing at the same time across CPU modules after startup.

Program to check start-up of each CPU module

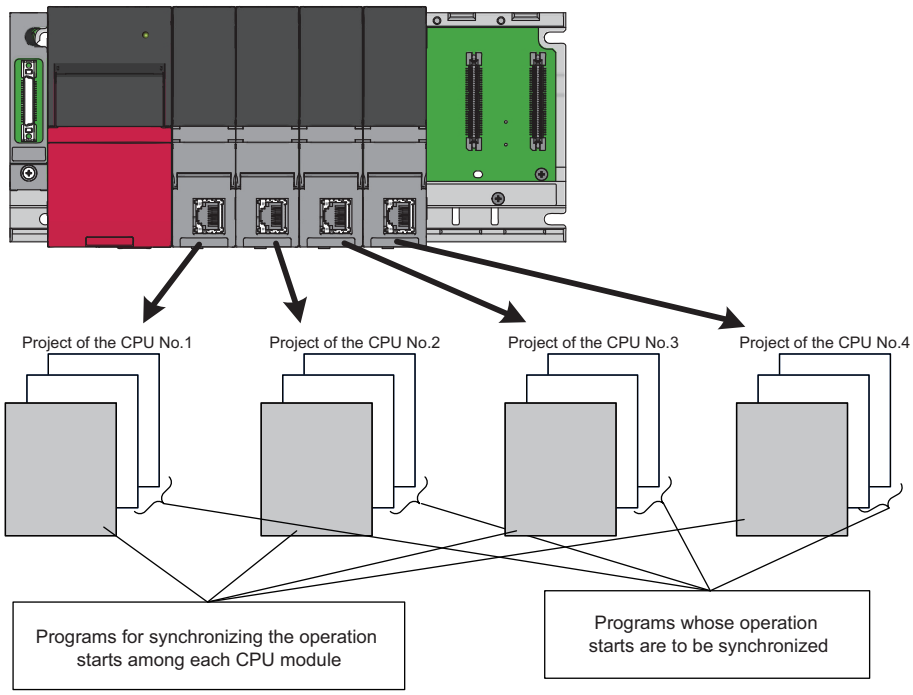
It is recommended to create a program that checks start-up of each CPU module using SM220 to SM223 (CPUs No.1 to No.4 preparation completed) when the multiple CPU synchronized startup is disabled. If a certain instruction is issued against a CPU module that has not started, the instruction executes no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

Ex.

Program to check start-up of CPU module of CPU No.2



To synchronize the start of operation processing across CPU modules, a program is required to check whether individual CPU modules are ready to start operation processing or not.



The following is an example of the configuration of a program which synchronizes the start of operation processing across CPU modules and programs which should start operation processing synchronously.

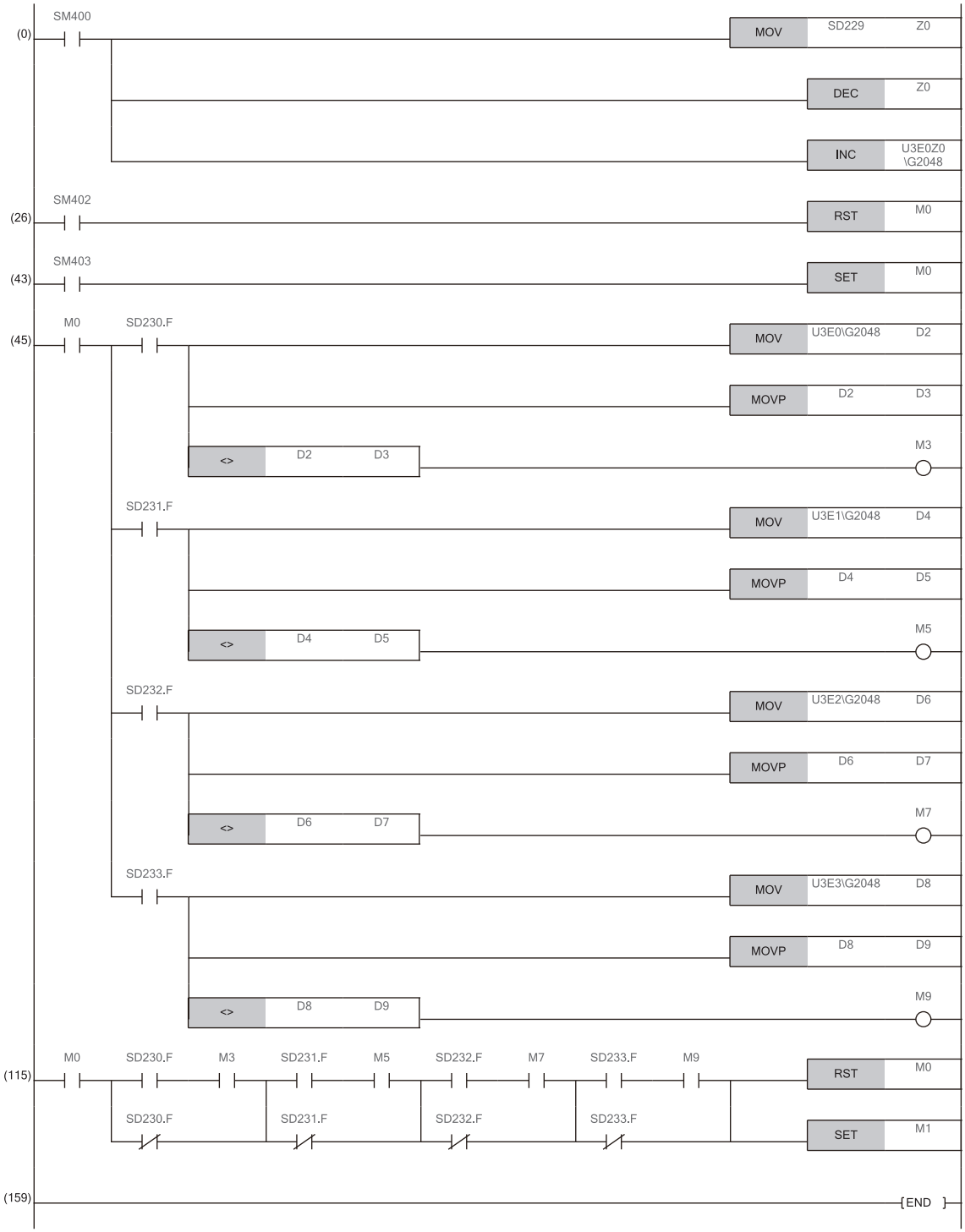
Program	Execution type	Description
Program to synchronize the start of operation processing	Scan	This is a program used to synchronize the start of operation processing across all CPU modules. This program must be specified as one for running at the beginning of the scan execution type of program. Also, when all the CPU modules are ready to start operation processing, the flag indicating an operation start turns on for only one scan.
Program to start operation processing synchronously	Event	This is a program which should be executed when all the CPU modules are ready to start operation processing. This program must be specified as an event execution type program which is triggered when the operation start flag is turned on.

■Program example

• Devices to be used for programs to start operation processing synchronously

Device to be used	Application
M0	Flag that indicates the operation processing is ready to be started (after a flag that indicates the operation start turns on, this flag turns off.)
M1	Flag that indicates an operation start (this flag turns on for only one scan.)
U3En\G2048	The synchronization counter of each CPU module (n=0: CPU No.1, n=1: CPU No.2, n=2: CPU No.3, n=3: CPU No.4)
D2 to D9	The storage location for values of the synchronization counter of each CPU module
M2 to M9	The operation processing start wait state of each CPU module

• Example of a program to synchronize the start of operation processing



Clock data

CPUs No.2 to No.4 automatically synchronize their clock data to the one set for CPU No.1 (even if setting up clock data individually for each CPU, they will be overwritten). Therefore, simply setting up the clock data for CPU No.1 allows to manipulate a unified clock data across the entire multiple CPU system (☞ Page 91 Time Setting)

Point

As with clock data, the same settings as those for CPU No.1 apply to CPU Nos. 2 to 4.

- Time zone setting (☞ Page 93 Setting Time Zone)
- Daylight saving time setting (☞ Page 94 Daylight Saving Time Function)

29.3 Multiple CPU Parameter Checking

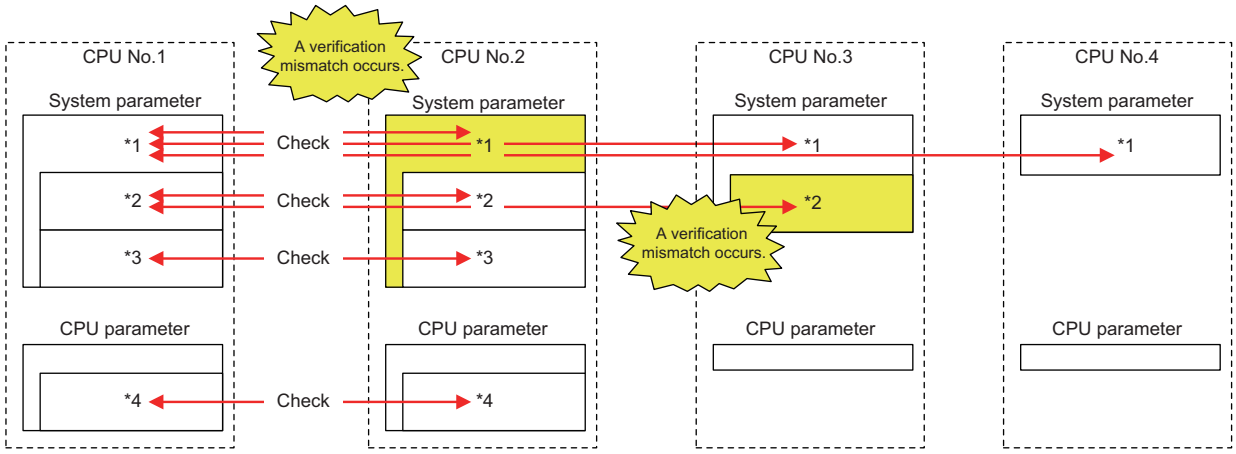
Whether the same setting is configured for between the system parameter of each CPU module and multiple CPU refresh number of points of CPU parameter is checked by the multiple CPU system at the timing shown below. However, as for the fixed scan communication setting and inter-module synchronization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

- Powered-on
- When the CPU No.1 is reset
- STOP→RUN state after parameter was changed

Timing	Parameters targeted for checking	Checking conditions for CPU No.1	Checking conditions for CPU No.2 and over
When power is on or the CPU No.1 is reset	System parameters (other than fixed scan communication setting and inter-module synchronous setting)	Checking is not conducted.	Compares with the parameters of the CPU of the lowest number.
	Settings of fixed scan communication	Checking is not conducted by the CPU module of the number for which the fixed scan communication setting is not configured. The CPU module of the number for which the fixed scan communication setting is configured will compare the parameters with those of the CPU of the lowest number.	
	Inter-module synchronization setting	Checking is not conducted by the CPU module of the number for which the Inter-module synchronization setting is not configured. The CPU module of the number for which the Inter-module synchronization setting has been configured will compare the parameters with those of the CPU of the lowest number.	
	CPU parameters (number of points of refresh settings)	Checking is not conducted by the CPU module of the number for which the fixed scan communication setting is not configured. The CPU module of the number for which the fixed scan communication setting is configured will compare the parameters with those of the CPU of the lowest number.	
STOP→RUN state after parameter was changed	—	Compares with the parameters of the host CPU module before parameters are changed.	

Ex.

An error occurs in the module of each CPU No 2 and 3, and the module of each CPU No 1 and 4 starts up normally when parameter checking. (Operation in error)



○: there is setting (if the number immediately next is the same number, it refers to the same parameter) , ×: there is no setting

Item		Setting conditions for the CPUs of each number			
		PLC No. 1	PLC No. 2	PLC No. 3	PLC No. 4
With or without the setting for each parameter	System parameters (other than fixed scan communication setting and inter-module synchronous setting) *1	○1	○5	○1	○1
	Settings of fixed scan communication*2	○2	○2	○6	×
	Inter-module synchronization setting*3	○3	○3	×	×
	CPU parameters (number of points of refresh settings)*4	○4	○4	×	×

29.4 Data Communication Between CPU Modules

CPU modules within a multiple CPU system can send and transfer data to each other. The refresh communication and direct access communication enable data writing or reading between CPU modules. Indicates the Data communication method.

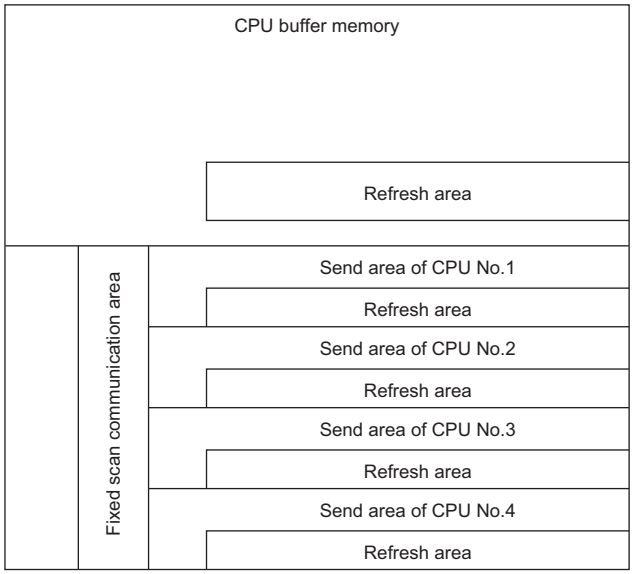
Communication method	Application	Description
Data communication with CPU buffer memory	Used when data is sent or received at the timing of each CPU module.	The CPU module for sending the data writes data into the CPU Buffer memory of the host CPU module. The CPU module for receiving data reads data from the CPU Buffer memory of the sender CPU module (other CPU modules).
Data communication with fixed scan communication area	Used when data is sent or received through adjusting the timing between CPU modules.	The CPU module for sending the data writes data into the fixed scan communication area (send area) of the host CPU module. The CPU module for receiving data reads data from the fixed scan communication area (receive area) of the host CPU module of the send source CPU module.

Memory to be used

CPU buffer memory is utilized for data communication between the CPU modules.

Memory configuration of CPU buffer memory

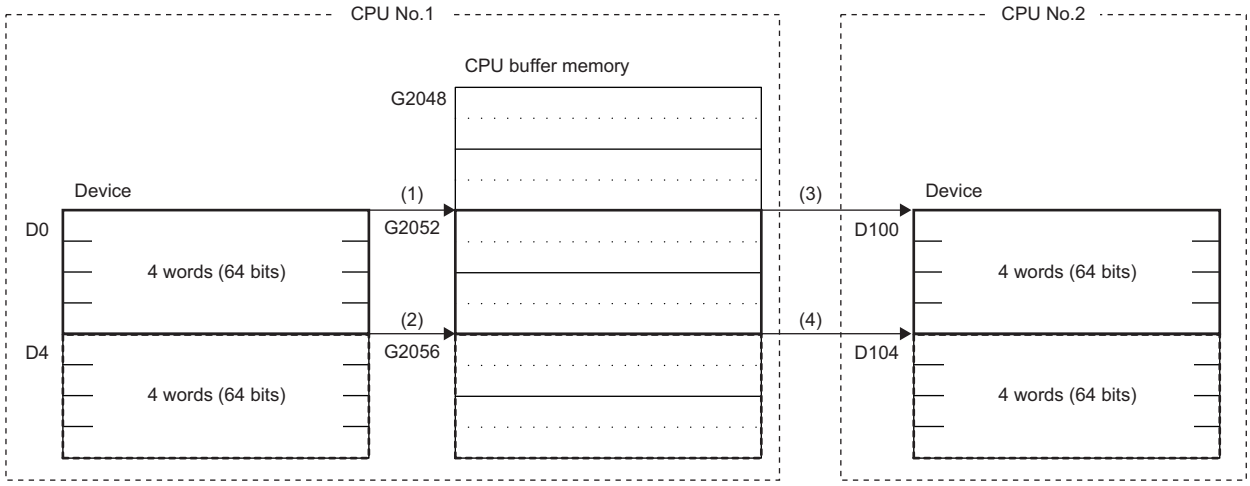
Indicates the memory configuration of CPU buffer memory.



Memory	Communication method	Description	Area size
CPU buffer memory	Communication through direct access	This area is used to read/write data to the areas of the host CPU module or other CPU modules.	512K words fixed
Refresh area	Communication through refresh	Data communication is conducted through refresh at END processing.	
Fixed scan communication area	Communication through direct access	This area communicates the data to the fixed scan communication area of the host CPU module, and is used when the host CPU module area and another CPU module area transfer the data at the fixed scan communication intervals.	It is possible to change within the range of 0 to 24K words in total. The send area as per unit can be set within the range of 0 to 12K words. (☞ Page 284 Setting fixed scan communication area)
The Refresh Area in the Fixed Scan Communication Area	Communication through refresh	The refresh operation is conducted at the fixed scan communication cycle.	

■Avoidance of 64-bit data inconsistency

To avoid 64-bit data inconsistency, access the CPU buffer memory by specifying the start address as a multiple of four similarly to the device to be specified.



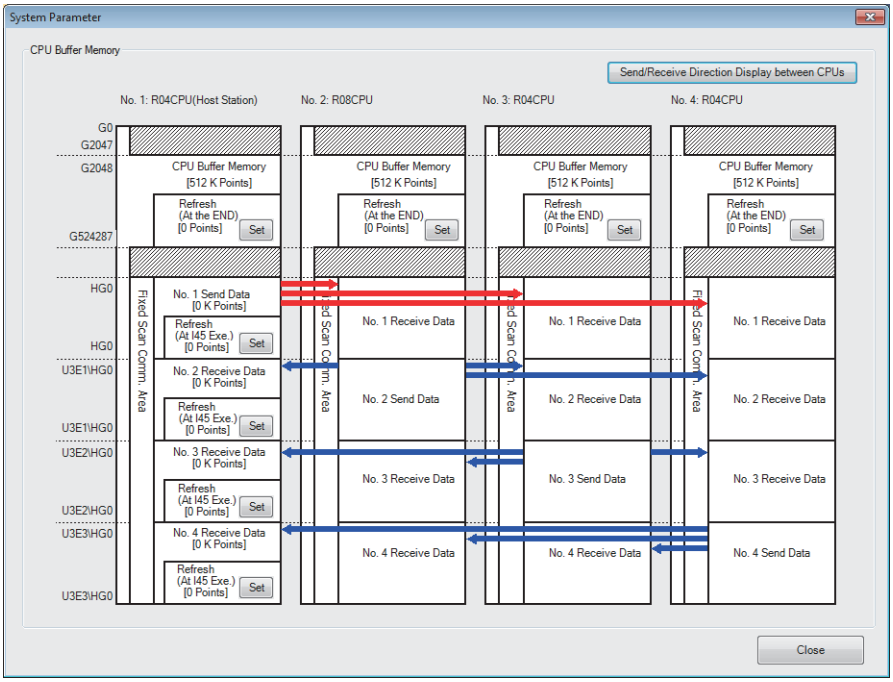
(1) The CPU module assures a 64-bit data integrity and write the data to the CPU buffer memory. (TO U3E0 K2052 D0 K4)
(2) The CPU module assures a 64-bit data integrity and write the data to the CPU buffer memory. (TO U3E0 K2056 D4 K4)
(3) The CPU module assures a 64-bit data integrity and read the data from the CPU buffer memory. (FROM U3E0 K2052 D100 K4)
(4) The CPU module assures a 64-bit data integrity and read the data from the CPU buffer memory. (FROM U3E0 K2056 D104 K4)

Checking for the memory configuration

Indicates the CPU buffer memory configuration of each CPU No. Besides, the refresh setting, which is set by the CPU parameter, can also be set through this section. (☞ Page 292 Refresh settings)

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Communication Setting between CPU] ⇨ [CPU Buffer Memory Setting] ⇨ [Detailed Setting]

Window



Displayed items

Item	Description	Setting range	Default
[Setting] button of each refresh area	This menu item sets up the refresh settings used for data communication between CPU modules. (Linked with the refresh settings of CPU parameter)	—	0 points
[Inter-CPU send/receive direction display] button	Click the button, and the arrow indicates the send/receive direction.	—	—

Setting the data communication with fixed scan communication area

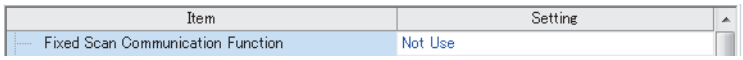
This section describes the setting for making the data communication with fixed scan communication area.

■Setting whether or not it should be used

To communicate data with the fixed scan communication area, "Enable" must be set to "Fixed Scan Communication Function".

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Communication Setting between CPU] ⇨ [Fixed Scan Communication Function]

Window



Displayed items

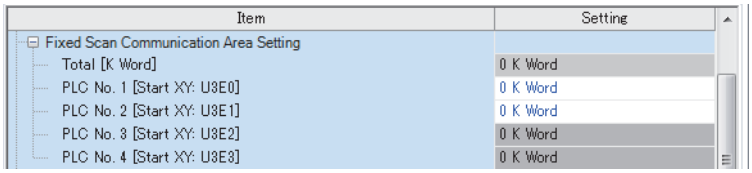
Item	Description	Setting range	Default
Fixed Scan Communication Function	Specify whether or not the fixed scan communication function should be used.	• Not Use • Use	Not Use

■Setting fixed scan communication area

This menu item sets up the range of send area for each CPU in the fixed scan communication area (the refresh area and the total of areas used for direct access communication). Only the fixed scan communication area can be changed with the parameter settings. The other areas are not configurable.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Communication Setting between CPU] ⇨ [Fixed Scan Communication Area Setting]

Window



Displayed items

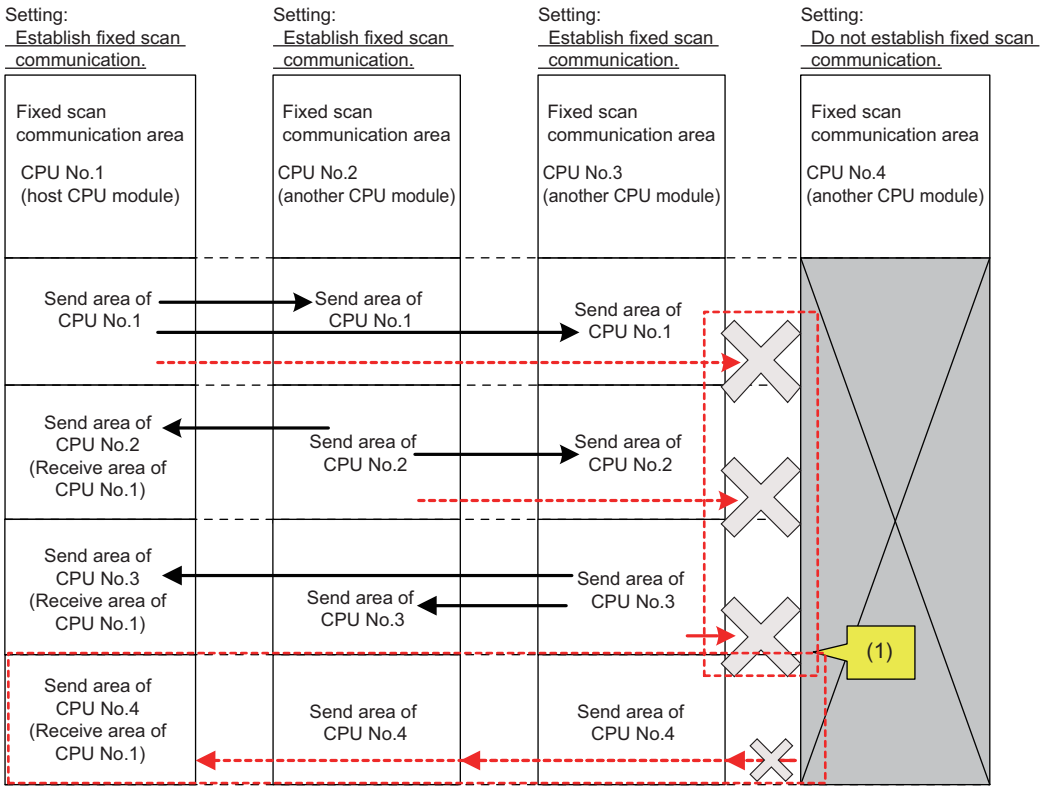
Item	Description	Setting range	Default
Total [K Word]	View the total value.	Entire system: 0 to 24K words*1	—
PLC No. 1 [Start XY: U3E0]	Set the send area size for CPU No.1.	0 to 12K words	0K word
PLC No. 2 [Start XY: U3E1]	Set the send area size for CPU No.2.	0 to 12K words	0K word
PLC No. 3 [Start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

*1 The max area size that can be set through setting of the fixed scan communication area may differ depending on the fixed scan communication cycle.

Fixed scan communication cycle	Max area size that can be set through setting of the fixed scan communication area
0.10ms	12K points
0.15ms	20K points
Except for shown above	24K points

Point

When there exists a CPU module for which "Disable" is set to "Fixed Scan Communication Function", if the send area of the fixed scan communication area is set to a CPU module for which "Disable" is set to "Fixed Scan Communication Function" (unspecified) in the parameter setting on the host CPU module, no error is generated because the unspecified CPU module is considered as a reserved one for future configuration.
Example: in four module configuration with the host CPU module is set to CPU No.1 and the fixed scan communication function of CPU No.4 set to "Disable"



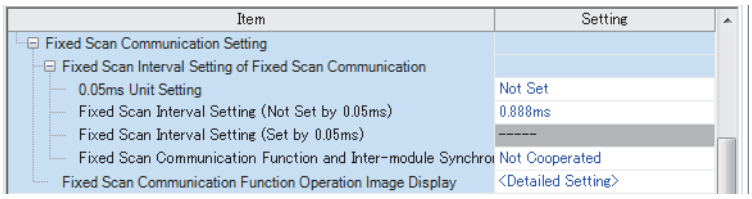
(1) Since the setting does not allow fixed scan communication, this CPU module does not send data to the other CPU modules although send areas have been reserved. This CPU module does not receive data from the other CPU modules.

Fixed scan communication setting

This menu item sets up the interval for data transfer between CPU modules. The data transfer interval can be synchronized with the timing for inter-module synchronization cycle (MELSEC iQ-R Inter-Module Synchronization Function Reference Manual)

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Fixed Scan Communication Setting] ⇨ [Fixed Scan Interval Setting of Fixed Scan Communication]

Window



Displayed items

Item	Description	Setting range	Default
0.05ms Unit Setting	Specify whether to set the fixed scan communication cycle in increments of 0.05ms.	• Not Set • Set	Not Set
Fixed Scan Interval Setting (Not Set by 0.05ms)	Select the fixed scan communication cycle to be set from the list items. The same option should be specified only for CPU modules which will use the fixed scan communication function.	• 0.222ms • 0.444ms • 0.888ms • 1.777ms • 3.555ms • 7.111ms	0.888ms
Fixed Scan Interval Setting (Set by 0.05ms)	Specify the fixed scan communication cycle freely in increments of 0.05ms.	0.10ms to 10.00ms	0.10ms
Fixed Scan Communication Function and Inter-module Synchronization Function	Set whether the fixed scan communication cycle cooperates with the inter-module synchronization cycle.	• Not Cooperated • Cooperate	Not Cooperated

Point

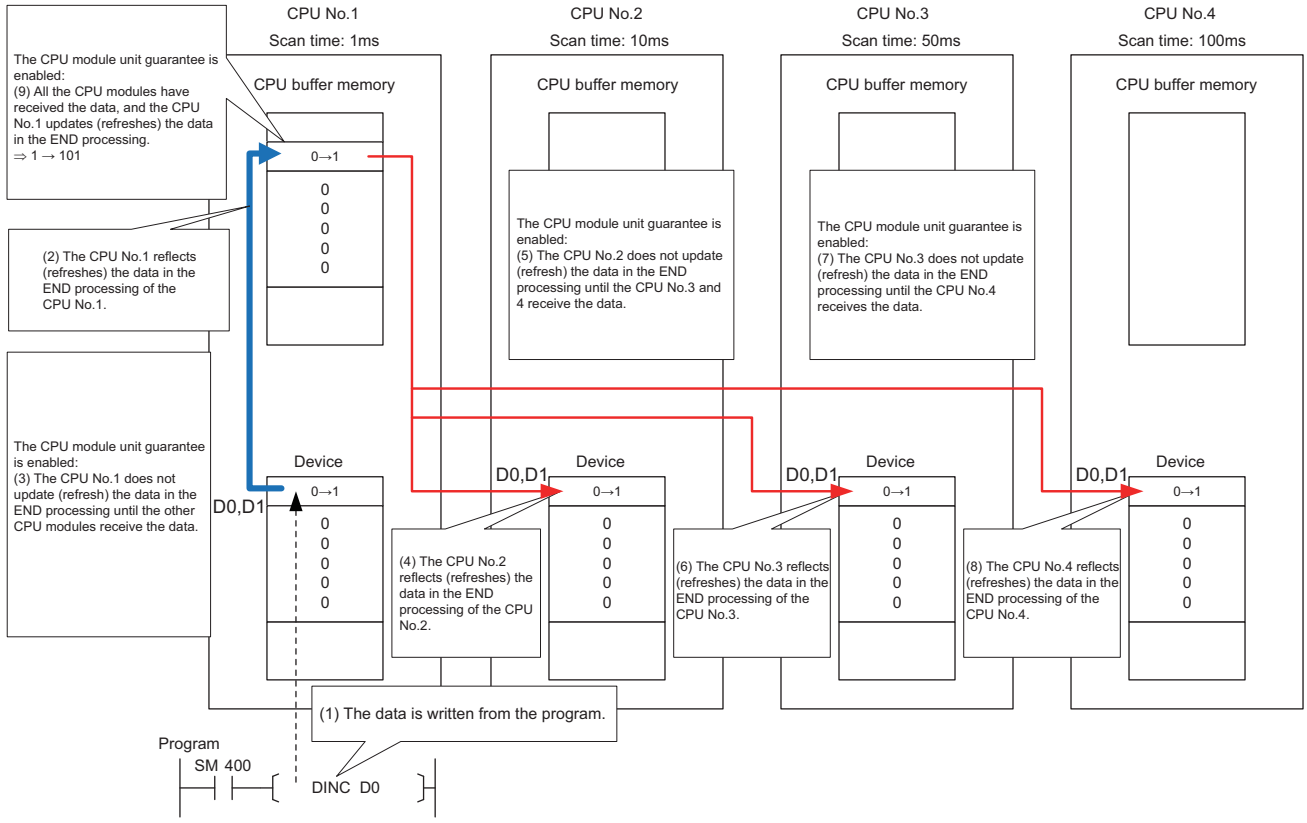
The send image for the fixed scan communication can be reviewed through the "Fixed Scan Communication Function Operation Image Display".

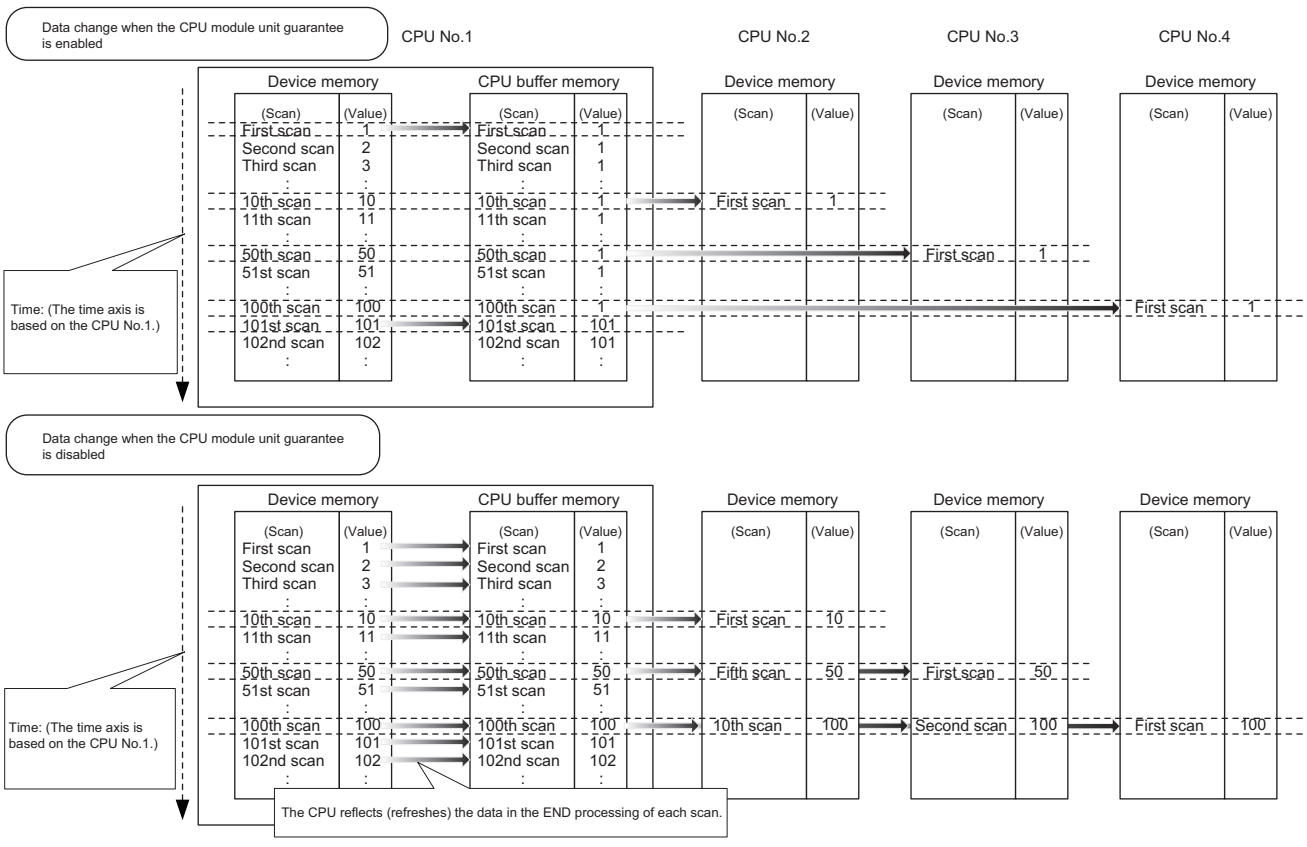
Error detection setting

When data is communicated among the CPU modules in a multiple CPU system, some data writing timing may lead to a failure of the complete data communication in the fixed scan communication cycle after the data writing and data inconsistency may occur. In this setting, set whether to detect a continuation error or not in this case. (☞ Page 230 Error Detection Setting)

Module-by-module data guarantee

In data communication, data is handled in units of 64 bits. Therefore, when data larger than 64 bits is handled, old and new data may be mixed for each CPU module depending on the timing between data reading by the host CPU module and data writing by other CPU modules/data receiving from other CPU modules. (Data inconsistency) To avoid this problem, the system conducts the refresh operation by exchanging handshake signals between the CPU modules for guaranteeing refresh data on a module-by-module basis.





Prevention of data inconsistency by module-by-module data guarantee

The following table shows the preventive control against data inconsistency according to the presence or absence of module-by-module data guarantee.

○: With the preventive control against data inconsistency by the system, ×: Without the preventive control against data inconsistency by the system*1

Communication method	CPU buffer memory		Fixed scan communication area	
	Module-by-module data guarantee enabled	Module-by-module data guarantee disabled	Module-by-module data guarantee enabled	Module-by-module data guarantee disabled
Communication through refresh	○	×	○	×
Communication through direct access	×	×	○*2	×

*1 The countermeasure by the program is required. (☞ Page 296 Data guarantee by program)

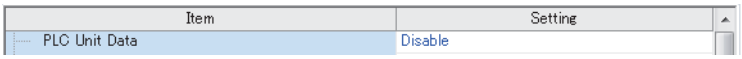
*2 Limited to the case of access within the multiple CPU synchronous interrupt program (I45). (☞ Page 298 When accessing fixed scan communication area)

Module-by-module data guarantee

Set up the module-by-module data guarantee.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Communication Setting between CPU] ⇨ [PLC Unit Data]

Window



Displayed items

Item	Description	Setting range	Default
PLC Unit Data	Specify whether data is guaranteed on a module-by-module basis.	• Disable • Enable	Disable

Point

- For communications by the direct access with the CPU buffer memory other than the fixed scan communication area, data is not guaranteed on a module-by-module basis even when this setting is enabled, because the data on the CPU buffer memory of other CPU modules is directly read after execution of the read instruction.
- Data send/receive timing coincides with an update interval of the CPU module with the slowest scan time, because the next send/receive starts only after all CPU modules have completed data reception.

Communication through refresh

The device data for each CPU module is written/read only by the parameter settings. Using refresh areas allows data communication between all or a part of the CPU modules in the multiple CPU system, thereby enabling devices of other CPU modules to be used by the host CPU module.

Types of refresh

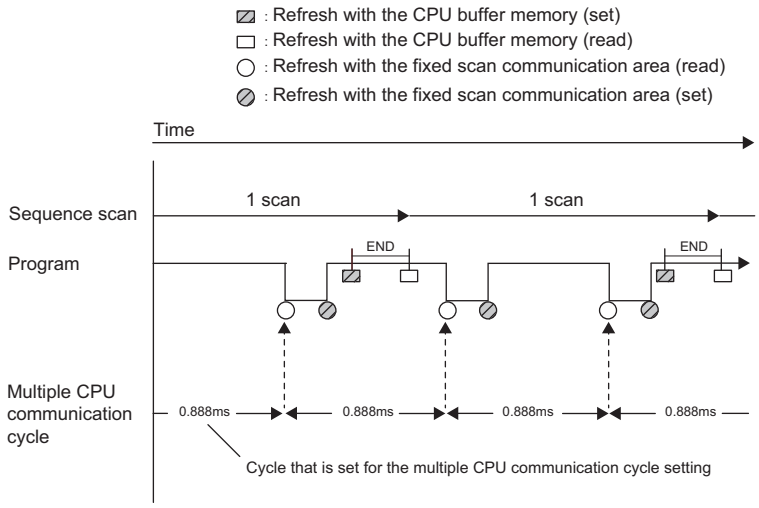
The following table shows the types of refresh.

Refresh timing	Application	Memory to be used
At the END processing	For data communication of devices in synchronization with the scan	CPU buffer memory
At the execution of multiple CPU synchronous interrupt program (I45)*1*2	For data communication of devices at the period specified in the fixed scan communication setting	Fixed scan communication area

*1 Executed at the period specified in the fixed scan communication setting.

*2 Refresh fails if the multiple CPU synchronous interrupt program (I45) does not exist on the program.

The following figure shows the timing of each refresh.

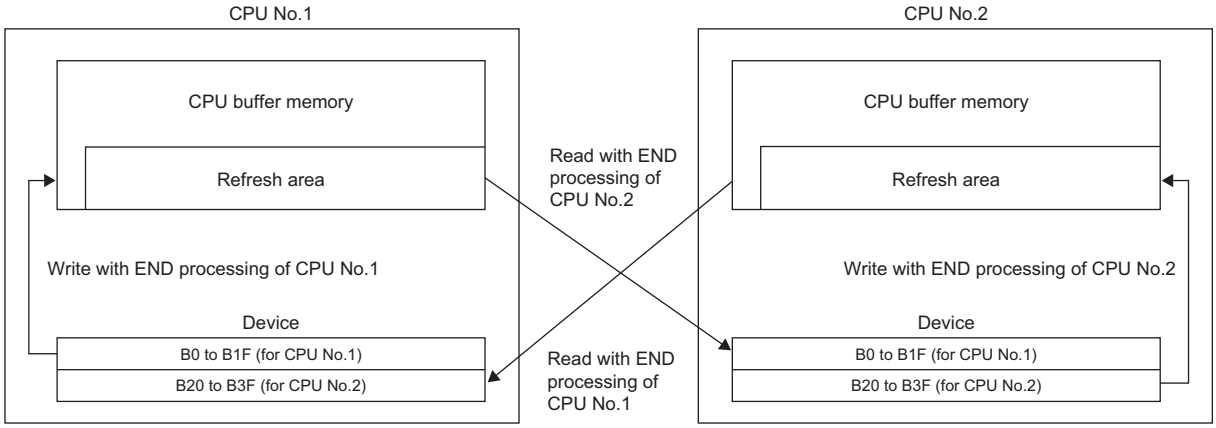


■Refresh using CPU buffer memory

At the END processing of the host CPU module, device data of the host CPU module is written to the refresh area within the CPU buffer memory on the host CPU module. The data written to the refresh area is transferred to the device of another CPU module at the END processing of another CPU module.

Ex.

When CPU No.1 refreshes 32 points (B0 to B1F) and CPU No.2 refreshes 32 points (B20 to B3F):

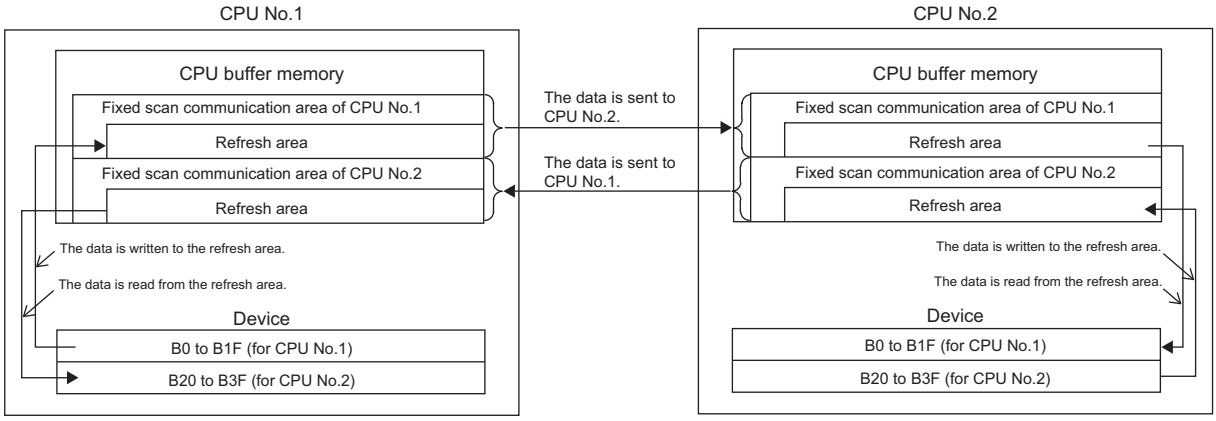


■Refresh using fixed scan communication areas

At the period specified in the fixed scan communication setting, the device data of the host CPU module is written to the refresh area within the fixed scan communication area of the host CPU module. The data written to the refresh area is sent to the refresh area within the fixed scan communication area of another CPU, which in turn reads the transferred data into the device.

Ex.

When CPU No.1 refreshes 32 points (B0 to B1F) and CPU No.2 refreshes 32 points (B20 to B3F):



Executing refresh

Refresh is executed when the CPU module is in RUN and/or STOP (PAUSE) state. For details on the behavior when the CPU module is in stop error state, refer to CPU module operation upon error detection setting. (☞ Page 231 CPU module operation upon error detection setting)

■Behavior during the multiple CPU synchronous interrupt program (I45) execution

If refresh is set to be performed during the multiple CPU synchronous interrupt program (I45) execution, the refresh behavior when the CPU module is in RUN state varies depending on either of the following conditions. (☞ Page 300 Multiple CPU Synchronous Interrupt)

- A program (I45 to IRET) which includes the multiple CPU synchronous interrupt program (I45) exists.
- Event execution type program exists.

The following table lists the refresh behaviors.

○: Execute refresh, ×: Not execute refresh

Execution type		Refresh behavior		Refresh behavior after the program control instruction is executed
		STOP state	RUN state	RUN state
Other than event execution type	The I45 interrupt pointer doesn't exist	○	×	×
	The I45 interrupt pointer exists	○	○	○
Event execution type (I45 interrupt specified)		○	○	× (stop after changing the execution type*1)

*1 The PSCAN/PSTOP instruction changes after the next scan, and the POFF instruction changes after the next two scans.

○: Execute refresh, ×: Not execute refresh, —: Execution disabled

Execution type		Refresh behavior triggered by operating status change of the CPU module						
		RUN state	Upon addition of the I45 interrupt pointer during online program change	Upon removal of the I45 interrupt pointer during online program change	RUN to STOP state	STOP state	After the refresh setting is changed; when parameters are written	STOP to RUN state; turning power off and on; resetting
Other than event execution type	The I45 interrupt pointer doesn't exist	×	○	×	○	○	○*3	×
	The I45 interrupt pointer exists	○	○	×	○*2	○	○*3	○*4
Event execution type (I45 interrupt specified)		○	—	—	○*2	○	○*3	○*4

*2 Not affect to the behavior.
*3 Operates with the pre-change parameters.
*4 Operates with the post-change parameters.

Configurable data

This section lists the configurable data with the refresh settings.

■Maximum number of settings

Maximum of 32 settings can be configured per CPU module for any refresh setting.

■Data number of points

Data number of points can be configured within the number of points assigned to the area (within the send range of the fixed scan communication area) in increments of two points. Device duplication between multiple setting numbers is not allowed.

■Data that can be specified

The device other than local device can be specified. However, when "Use File Register of Each Program" is enabled, file registers cannot be specified. Doing so may prevent the device from operating at file registers for each program depending on the CPU operating status.

Type	Devices that can be specified
Bit device[*1]	X, Y, M, L, B
Word Device	D, W, R, ZR, RD

*1 This device can be specified only in units of 16 points (one word).

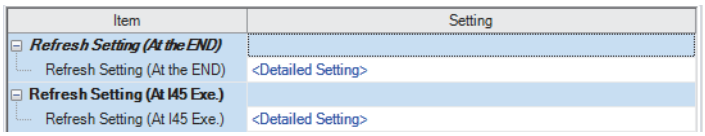
Refresh settings

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

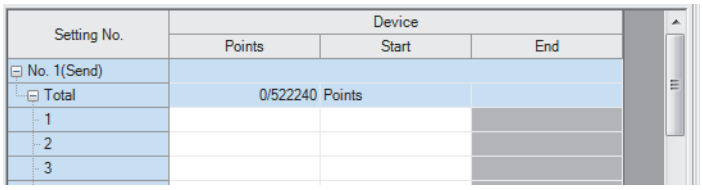
[CPU Parameter] ⇨ [Refresh Setting between Multiple CPU]

Operating procedure

"Refresh Setting between Multiple CPU" window



"Detailed Setting" window



1. Click "Detailed Setting" at the execution timing for each refresh.
2. Enter the start/end of the device.

Point

Set a blank column for "Head" and "End" and set only number of points so that the setting by which auto refresh is not conducted can be configured. (It is possible to perform setting which does not allow the refresh data of other CPU modules, which is unwanted for the host CPU module, to be obtained.)

■Enabling timing of refresh settings

The refresh settings are enabled at the following timing:

- When CPU module is powered off and on or is reset.
- When the operating mode is switched from STOP to RUN after the parameters are written to the CPU module (STOP → RUN → STOP → RUN).

CPU module operating status and refresh behavior

When the refresh timing coincides with the period specified in the fixed scan communication setting, the refresh behavior depends on the operating status of the CPU module.

CPU module operating status	Refresh enabled/disabled	Refresh timing	
		CPU buffer memory	Fixed scan communication area
RUN (including a continuation error period)	Execute refresh	During the END processing	Before and after the multiple CPU synchronous interrupt program (I45) execution[*1]
STOP (including a stop error period due to a moderate error)	Execute refresh	During the END processing	During the END processing
STOP (major error period)	Not execute refresh	—	—
PAUSE	Execute refresh	During the END processing	During the END processing

*1 In the disabling interrupt state by the DI instruction, the refresh is not operational, because the multiple CPU synchronous interrupt program (I45) does not work.

Communication through direct access

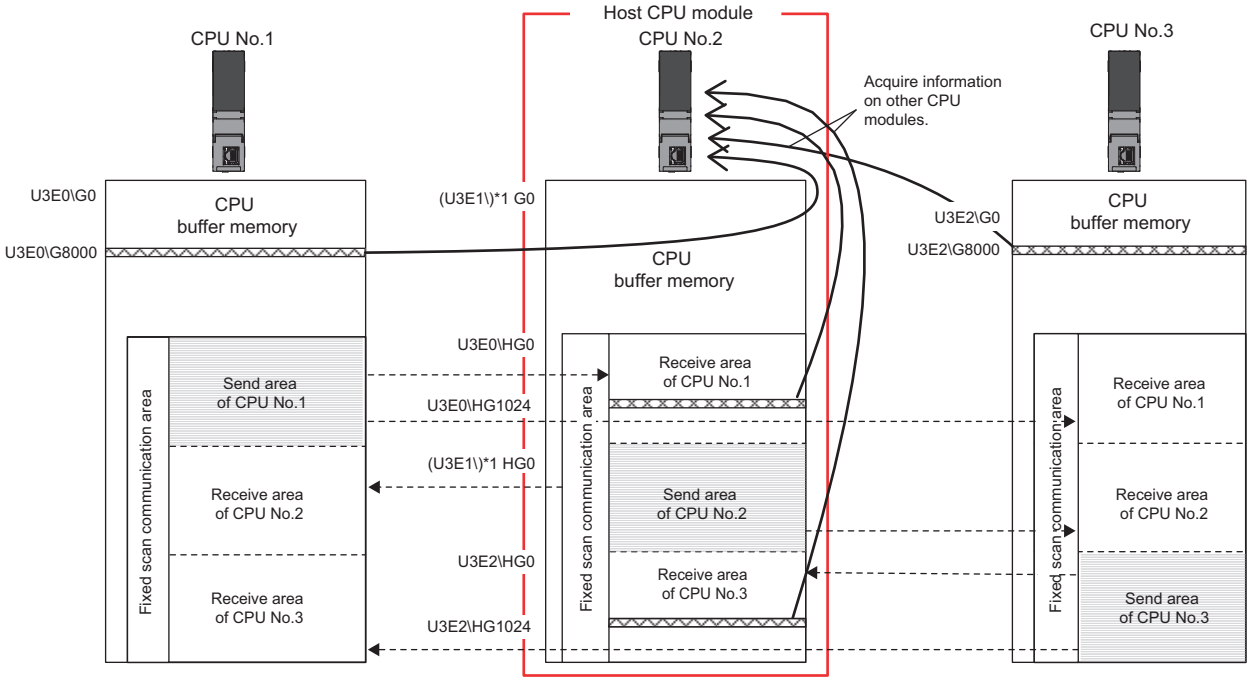
This method uses programs to communicate with other CPUs. The following table lists the communications using the direct access method.

Communication method	Description	Instruction to be used
Communication using CPU buffer memory	Transfer data between CPU modules using any area on the CPU buffer memory.	• FROM/TO instruction • DFROM/DTO instruction • Instruction using the CPU buffer memory access device (U3En\G□)
Communication using the fixed scan communication area	Transfer data between CPU modules using the fixed scan communication area.	Instruction using the CPU buffer memory access device (U3En\HG□)
Communication by a dedicated instruction	Transfer data between CPU modules in the multi CPU system and/or sends control command to a CPU module of other CPUs using a dedicated instruction.	DDWR/DDRD instruction and others

29

Specification method thorough CPU buffer memory access device

Specify the CPU buffer memory as "U3En\Gn" or the fixed scan communication area as "U3En\HGn" when accessing the memory or area. (☞ Page 334 CPU Buffer Memory Access Device)



■Obtaining information stored on the CPU buffer memory

Access the CPU buffer memory.

■Obtaining information stored on the fixed scan communication area for other CPUs

Access the fixed scan communication area on the host CPU module. Obtain data sent to the fixed scan communication area on the host CPU module at the fixed scan communication cycle.

Communication using CPU buffer memory and fixed scan communication area

This section describes the communication using CPU buffer memory and fixed scan communication area.

■Available area for communication

The following area can be used for communication.

Area	Description
CPU buffer memory	All the CPU buffer memory area except for the refresh area is available. The start address of the available area for each CPU module varies depending on the refresh settings. The end address of the area is fixed by CPU module models.
Fixed scan communication area	All the fixed scan communication area except for the refresh area is available. The start address of the available area for each CPU module is HG0 and the end address varies depending on the refresh settings.

■Instructions to be used for communication

Communication with each CPU module is enabled by issuing the following read/write instructions to each area.

- Write instruction: the instructions using the CPU buffer memory access device[*1] and TO/DTO instruction
- Read instruction: the instructions using the CPU buffer memory access device[*1] and FROM/DFROM instruction

*1 Specify "U3En\G□" when accessing the CPU buffer memory and "U3En\HG□" when accessing the fixed scan communication area.

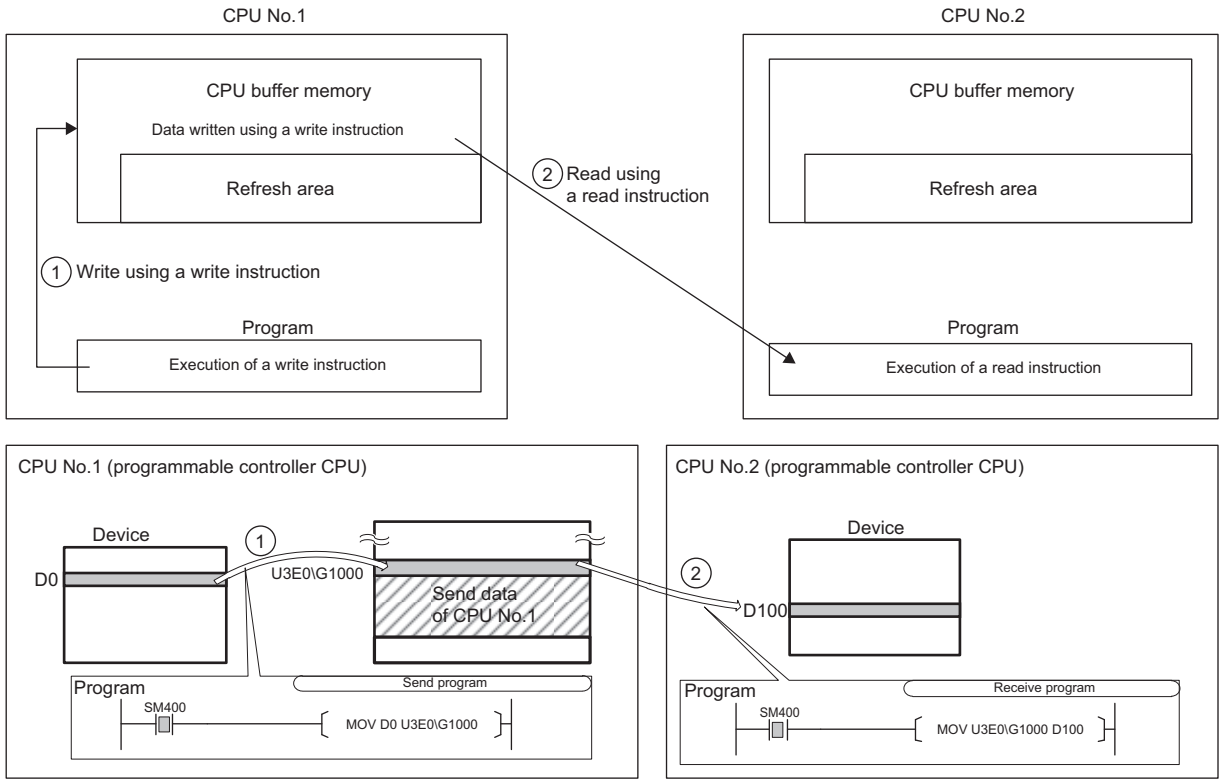
■Data communication behavior

- When using an area within the CPU buffer memory

Data written to the area within the CPU buffer memory on the host CPU module using the write instruction can be read by other CPU modules using the read instruction. Unlike the refresh, data registered during the instruction execution can be directly read.

Ex.

When data written to the CPU buffer memory on the CPU No.1 using the write instruction is read by CPU No.2 using the read instruction:

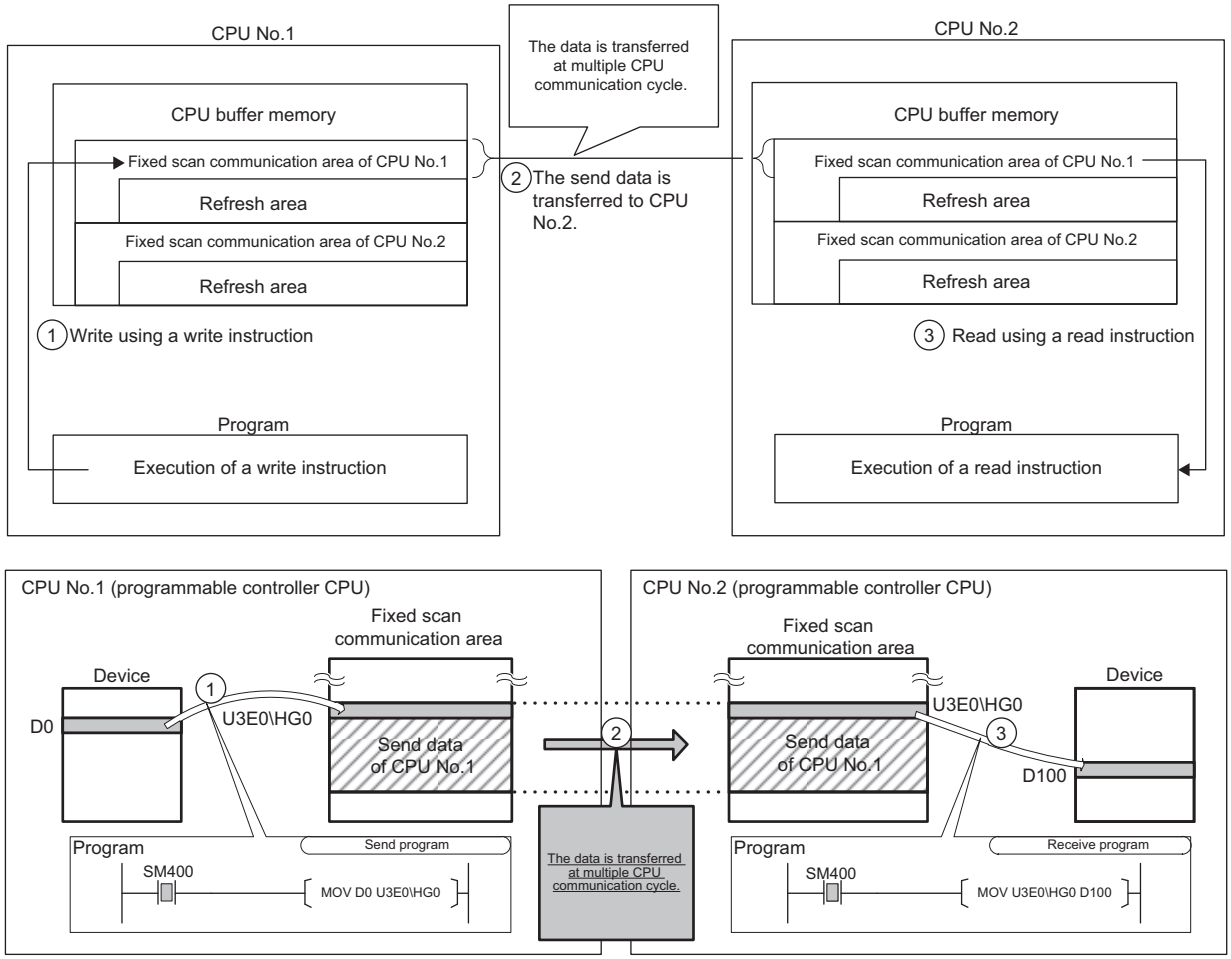


• When using an area within the fixed scan communication area

Data written to the area within the fixed scan communication area on the host CPU module using the write instruction is sent to other CPU modules at the period specified in the fixed scan communication setting. Other CPU modules read the received data using the read instruction. Unlike the refresh, data registered during the instruction execution can be directly read.

Ex.

When data written to the fixed scan communication area on the CPU No.1 using the write instruction is read by CPU No.2 using the read instruction:



Data guarantee by program

This section describes how to avoid the inconsistency of data larger than 64 bits using the program. To set up the module-by-module data guarantee using the parameters, use the multiple CPU setting. (☞ Page 288 Module-by-module data guarantee)

Data guarantee in communication through the refresh

Inconsistency of transferred data can be avoided by setting the interlock device to a transfer number lower than the one for the transferred data, because data is transferred in descending order from the highest setting number in the refresh settings.

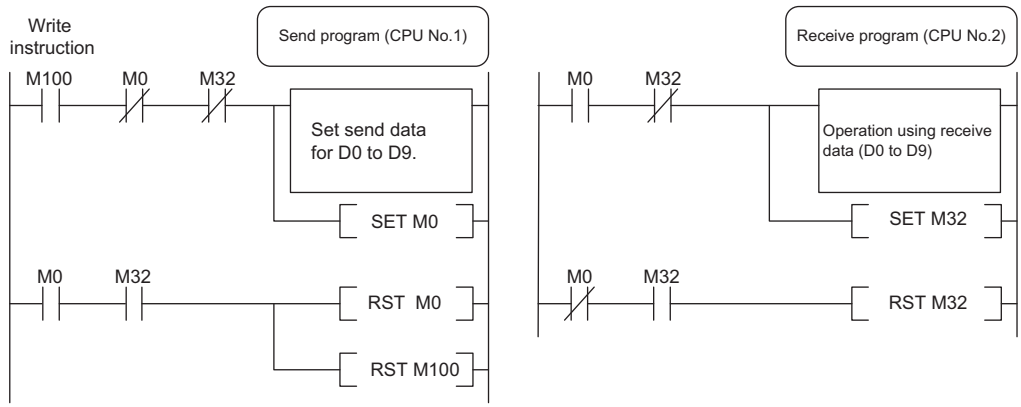
Ex.

Interlock program in communication by refresh

- Parameter settings

CPU No.1 refresh setting							Direction	CPU No.2 refresh setting						
CPU No.	Transfer No.	Send/receive range for each CPU module			Send/receive device setting			CPU No.	Transfer No.	Send/receive range for each CPU module			Send/receive device setting	
		Number of points	start	end	start	end				Number of points	start	end	start	end
CPU No.1	Transfer No.1	2	0	1	M0	M31	→	CPU No.1	Transfer No.1	2	0	1	M0	M31
	Transfer No.2	10	2	11	D0	D9			Transfer No.2	10	2	11	D100	D109
CPU No.2	Transfer No.1	2	0	1	M32	M63	←	CPU No.2	Transfer No.1	2	0	1	M32	M63

- Program example



Data guarantee for communication through direct access

The behavior varies depending on the area to be accessed.

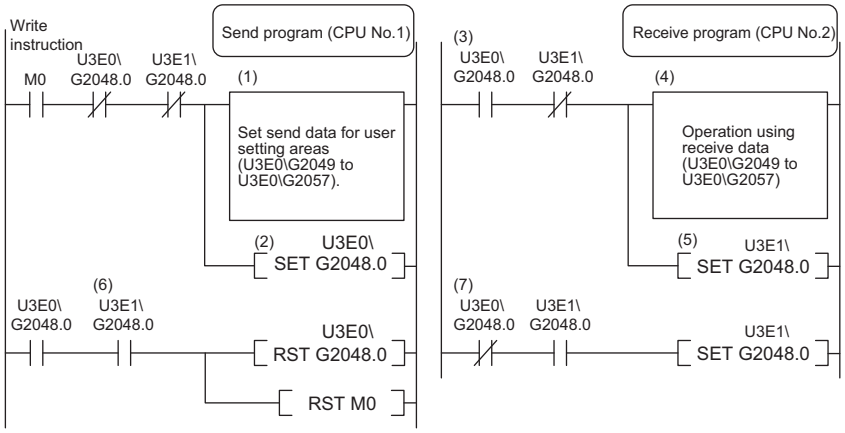
■When accessing CPU buffer memory:

The program reads data in ascending order from the start address of the CPU buffer memory other than the refresh area, and the write instruction writes send data in descending order from the end address of the CPU buffer memory other than the refresh area. Therefore data inconsistency can be avoided by setting an interlock device at the start position of data to be communicated.

Ex.

Interlock program in communication by direct access (when accessing CPU buffer memory)

- Program example



(1) CPU No.1 creates send data.
(2) CPU No.1 turns on the data setting complete bit.
[Data transfer with CPU No.2 END processing]
(3) CPU No.2 detects send data setting complete.
(4) CPU No.2 performs receive data processing.
(5) CPU No.2 turns on receive data processing complete.
[Data transfer with CPU No.1 END processing]
(6) CPU No.1 detects receive data processing complete, and turns off the data setting complete bit.
[Data transfer with CPU No.2 END processing]
(7) CPU No.2 detects that send data setting complete is turned off, and turns off receive data processing complete.

■When accessing fixed scan communication area

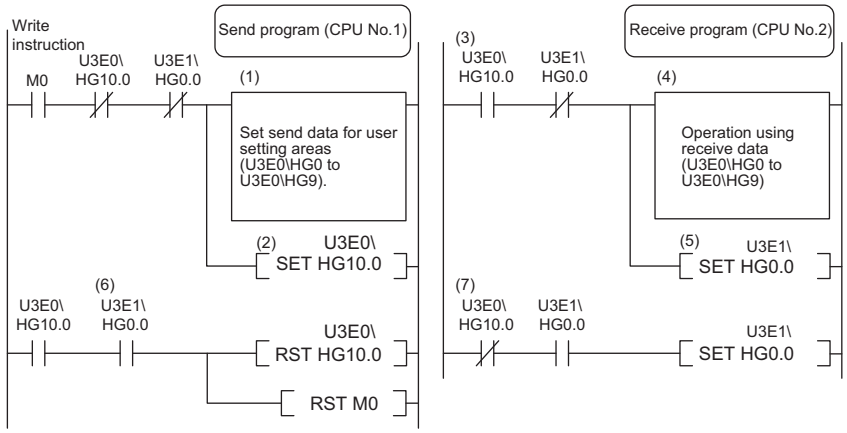
When accessing within the multiple CPU synchronous interrupt program (I45), enabling the setting of module-by-module data guarantee eliminates the need of an interlock circuit. When this refresh area is accessed within a program other than the above, or when the setting of module-by-module data guarantee is disabled, an interlock circuit is required, as with the access to the CPU buffer memory. (☞ Page 287 Module-by-module data guarantee)

The program reads data by transferring it in the order that it is written to the CPU buffer memory (fixed scan communication area). Data inconsistency can be prevented by using devices written after the transfer data for interlocks, regardless of the device type or address.

Ex.

Interlock program in communication by direct access (when accessing fixed scan communication area)

• Program example



(1) CPU No.1 creates send data.
(2) CPU No.1 turns on the data setting complete bit.
[Data transfer with multiple CPU communication cycle]
(3) CPU No.2 detects send data setting complete.
(4) CPU No.2 performs receive data processing.
(5) CPU No.2 turns on receive data processing complete.
[Data transfer with multiple CPU communication cycle]
(6) CPU No.1 detects receive data processing complete, and turns off the data setting complete bit.
[Data transfer with multiple CPU communication cycle]
(7) CPU No.2 detects that send data setting complete is turned off, and turns off receive data processing complete.

Also, with instructions such as BMOV instructions that involve writing data with two or more words to the CPU buffer memory, data is written from the end address to the start address. If combining and writing send data with interlock signals with a single instruction, data inconsistency can be prevented with an interlock signal at the start of the data.

Communication between CPU modules in error state

The following section describes communication between CPU modules in an error state.

Behavior in receive data error state

A CPU module receiving illegal data due to noise and/or failure discards the received data. If a received data is discarded, the receive-side CPU module keeps the last data received before discarding. When the CPU module receives the next correct data, it updates the received data.

Refresh execution in an error state

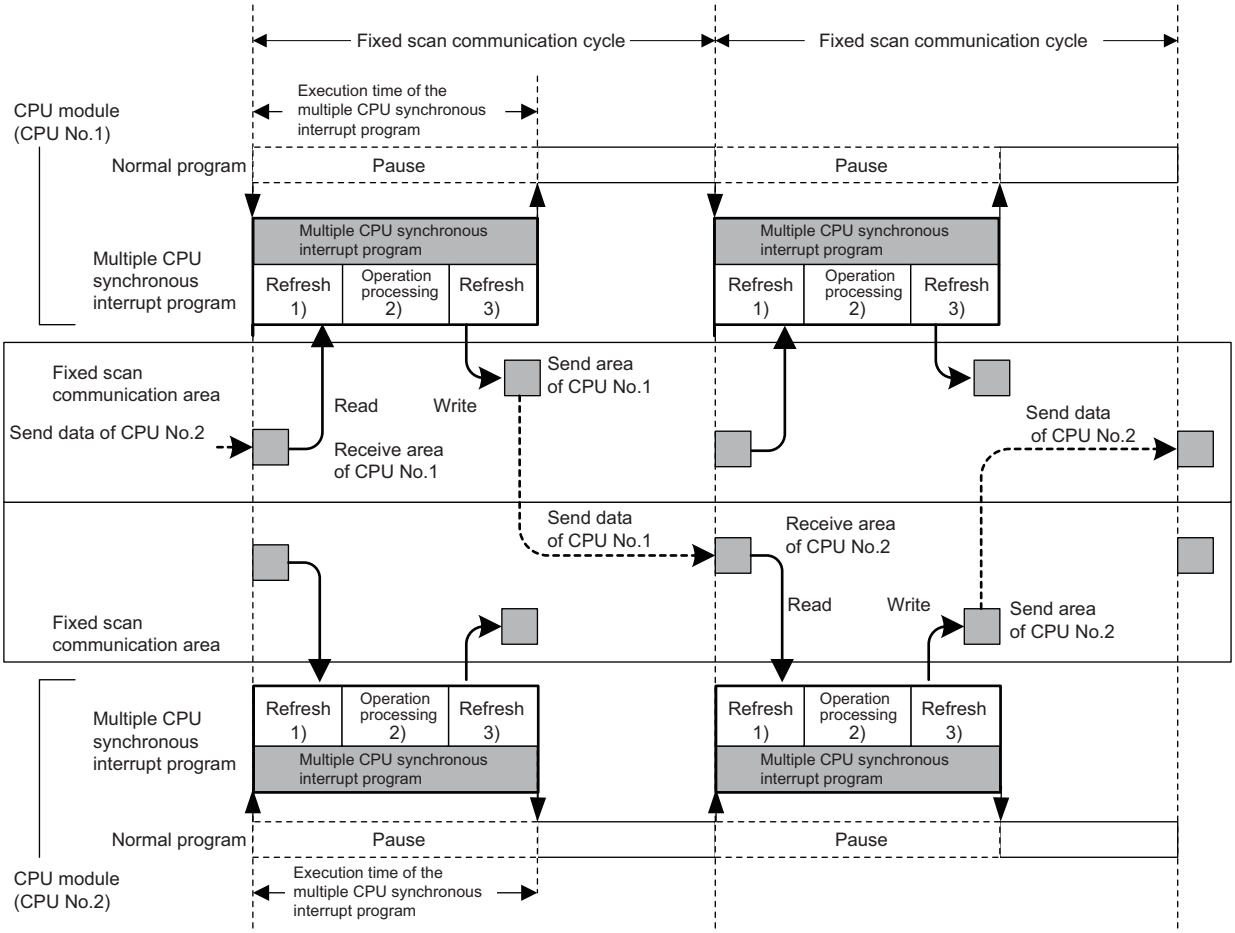
The following table lists the refresh and send/receive operation between CPU modules when the host CPU module detects a self diagnostic error. If one of the CPU modules enters into stop error state, the other CPU modules which are not in stop error state keep data stored before the stop error occurs.

Error		Refresh*1	Data communication between CPU modules*2
Minor error		○	○
Moderate error	Causes other than the following item	○	○
	Parameter error for fixed scan communication function (including the consistency check during start-up)	×*3*4	×*3*4
Major error		×*3	×*3

*1 Indicates data transfer between user devices and the fixed scan communication area on the host CPU module.
*2 Indicates data communication between the fixed scan communication areas on the host CPU module and other CPU modules.
*3 When an error occurs during normal operation, normal data generated immediately before the error occurs is continued to be sent between the fixed scan communication areas on the host CPU module and other CPU modules.
*4 If the consistency check fails due to a parameter change in normal operation, refresh and data send/receive between the CPU modules are continued.

29.5 Multiple CPU Synchronous Interrupt

Triggers an interrupt program at the fixed communication timing set by a parameter. The interrupt program executed at the fixed scan communication cycle is called "Multiple CPU synchronous interrupt program". Using the multiple CPU synchronous interrupt enables synchronizing with the fixed scan communication cycle so that data communication between CPU modules can be made. (It allows synchronizing the data communication timing between CPU modules.)



❶ Data reading of other CPU modules (refresh): Data sent from other CPU modules is read to a device or a label. (Data is read from the receive area of the host CPU module.)

❷ Operation processing: The multiple CPU synchronous interrupt program is executed.

❸ Data sending of other CPU modules (refresh): Data to be sent to other CPU modules is written from a device or a label. (Data is written to the send area of the host CPU module.)

Point

The operation methods required when an interrupt factor occurs and the program creating methods are the same as those for normal interrupt program. (☞ Page 55 Interrupt program)

Execution timing

The multiple CPU synchronous interrupt program (I45) is executed at the timing for the fixed scan communication cycle. The fixed scan communication cycle can be changed through the fixed scan communication setting. (☞ Page 286 Fixed scan communication setting)

Point

It is also possible to perform refresh during the multiple CPU synchronous interrupt program (I45) in execution. (☞ Page 289 Communication through refresh)

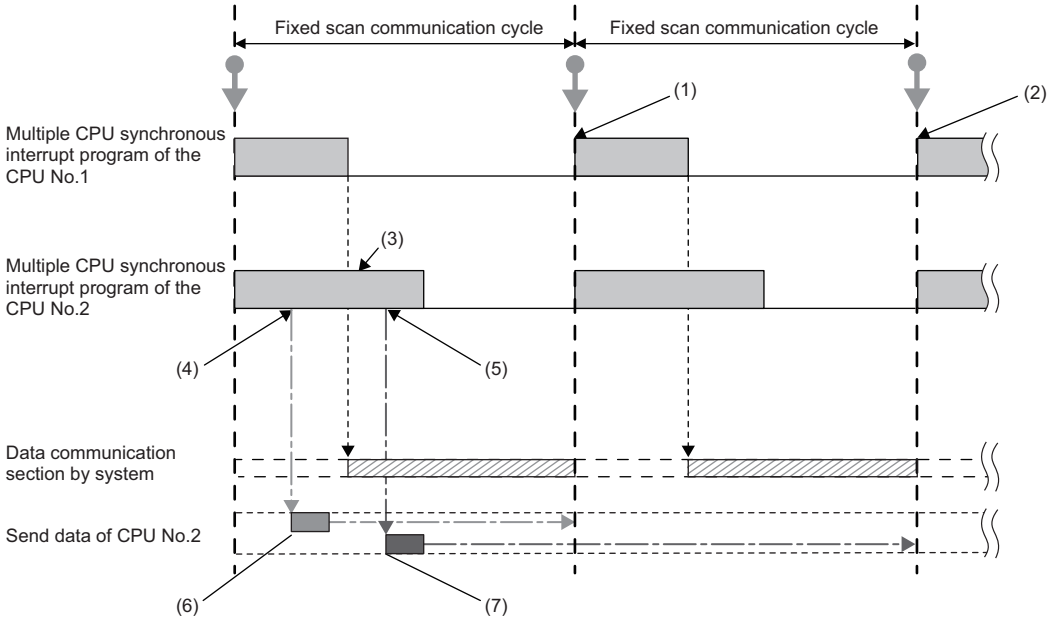
Multiple interrupt

For the multiple interrupt of the multiple CPU synchronous interrupt program, refer to the multiple interrupt function. (☞ Page 106 Multiple Interrupt Function)

Precautions

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

- Create a multiple CPU synchronous interrupt program so that it has the execution processing time shorter than the fixed scan communication cycle. If the interrupt program has the execution processing time equal to or longer than the cycle, the multiple CPU synchronous interrupt interval cannot be guaranteed. (☞ Page 56 Operation upon occurrence of an interrupt factor) The execution time of the multiple CPU synchronous interrupt program (I45) can be monitored using the RAS setting of the CPU parameter. (☞ Page 230 Error Detection Setting)
- To send data successfully in the next fixed scan communication cycle, select "Detect" for "Program Execution Section Exceed (I45)" in "RAS Setting" of [CPU Parameter]. (☞ Page 230 Error Detection Setting) With this setting, when data is written after the host CPU module starts data transfer in the multiple CPU synchronous interrupt program, SM484 (Execution section excess error flag for multiple CPU synchronization interrupt program) is turned on and the number of data sending errors in the next cycle is stored in SD484 (Number of execution section excess errors for multiple CPU synchronization interrupt program). In addition, when the data is written while the data cannot be sent in the fixed cycle communication cycle set by the parameter, an error can be detected. (The CPU module continues its operation.)



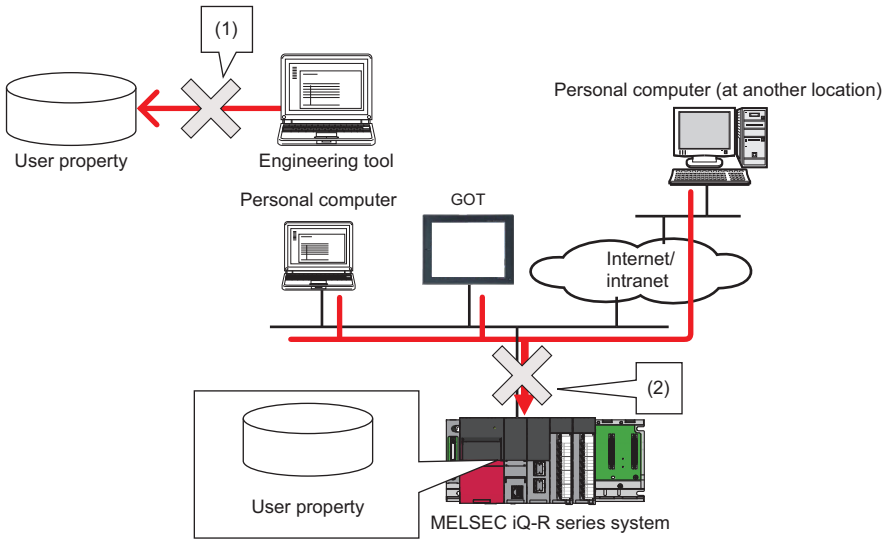
(1) CPU No.1 receives send data (A) of CPU No.2 in the next scan.
(2) CPU No.1 receives send data (B) of CPU No.2 after two scans.
(3) SM484 turns on and SD484 counts up because CPU No.2 continuously executes its program even after the start of data communication.
(4) Send data (A) of CPU No.2 is written.
(5) Send data (B) of CPU No.2 is written.
(6) This data is sent in the next scan because it has been written before data update.
(7) This data is sent after two scans because it has been written after data update.

30 SECURITY FUNCTION

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

• Only the Safety CPU supports the user authentication function.

This function serves to protect the user property stored in a personal computer and the user property inside modules in the MELSEC iQ-R series system against threats such as theft, tampering, faulty operation, and unauthorized execution due to the unauthorized access by an outsider. Use an appropriate security function according to the purpose as shown in the following table:



(1) Protection is provided against theft, tampering, and faulty operation resulting from unauthorized access by an outsider. (Protection for user property on a personal computer)

(2) Protection is provided against theft, tampering, faulty operation, and unauthorized execution resulting from unauthorized access by an outsider. (Protection for user property in CPU modules)

Data to be protected	Purpose	Function	Reference
Project	Prevents unauthorized access to programs (in units of POU). (A password is used.)	Block password function	GX Works3 Operating Manual
	Prevents unauthorized access to programs (in units of program file). (A security key is used.) Project data locked with a security key can only be viewed with an engineering tool for which the same security key has been registered.	Security key authentication function	
	Restricts access to projects on the personal computer with a combination of user name and password. The security level can be set individually for each user.	User authentication function	
CPU module	Prevents unauthorized execution of programs. (A security key is used.) Programs locked with a security key can only be executed at modules for which the same security key has been set.	Security key authentication function	
	Restricts access to projects in the CPU module with a combination of user name and password. The security level can be set individually for each user.	User authentication function	
	Prevents unauthorized read and write of files. (A password is used.)	File password function	• GX Works3 Operating Manual • MELSEC iQ-R Ethernet User's Manual (Application) • SLMP Reference Manual • MELSEC- iQ-R Serial Communication Module User's Manual (Application)
	Blocks access from an invalid IP address by identifying the IP address of an external device via Ethernet.	IP filter function	MELSEC iQ-R Ethernet User's Manual (Application)
	Limits access to just specific communication routes via Ethernet. (A password is used.)	Remote password function	

Point

If a personal computer with a security key registered is abused by an outsider, there is no way to prevent the outflow of the program property, and thus the user needs to take adequate measures as shown below:

- Preventive measures against the theft of a personal computer (for example, wire locking)
- Management of users of the personal computer (for example, deletion of unnecessary user accounts, strict control of login information, and implementation of fingerprint authentication)

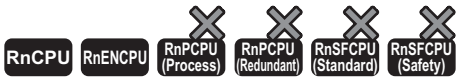
Furthermore, if a personal computer with a security key registered has failed, the locked project data cannot be viewed and edited. We assume no responsibility whatsoever for any damage or loss to the user and any other individual or organization, resulting from such a situation. Therefore, the user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

- Import the registered security key to another personal computer.
- Export the registered security key to a file and store the file in a safe place.

The following information in the CPU module can be deleted by selecting [Online] ⇨ [User Authentication] ⇨ [Initialization of all PLC Data].

- User management information set with user authentication function (user authentication function returns to disabled status)
- Security key
- All program memory, data memory files
- All files in file storage areas in device/label memory
- Safety CPU safety operation mode status (returns to TEST MODE)

31 SEQUENCE SCAN SYNCHRONIZATION SAMPLING FUNCTION



Data of the CPU module are collected when each module synchronizes data with the sequence scan of the CPU module.
The modules which can synchronize and collect with sequence scan are as follows:

- MES interface module (high-speed access)
- High speed data logger module (high-speed collection)
- C intelligent function module (data sampling in sequence scan)

For details on the function using the sequence scan synchronization sampling function and collectable data for each module, refer to the manual for the module used.
(📖 Manual for the module used)

Precautions

For one CPU module, the total number of points that can be collected by multiple modules is 32K points.
However, use points in 8K-point increments to use all of 32K points.
The points are assigned in 8K-point increments as follows:

- 6K points used → 8K points to be assigned
- 9K points used → 16K points to be assigned

MEMO

32 LABEL INITIALIZATION FUNCTION

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

In the Process CPU, the labels assigned to label areas will be initialized (Initial values are set if the values have been set, or the labels are cleared to zero if not) when the CPU module is powered on or the operating status of the CPU module is switched from STOP to RUN after data is rebuilt (reassigned) and then written to the programmable controller.

32.1 Initialization of Labels After Rebuilt All (Reassignment)

In the Process CPU, the labels assigned to label areas will be initialized (Initial values are set if the values have been set, or the labels are cleared to zero if not) when the CPU module is powered on or the operating status of the CPU module is switched from STOP to RUN after data is rebuilt (reassigned) with the engineering tool and then written to the programmable controller.

Restriction

This function is supported by only Process CPUs. Before using this function, check the versions of the CPU module and engineering tool used. (☞ Page 779 Added and Enhanced Functions)

Label initialization operation

The following table lists the label initialization operations.

Target label	Label initial value setting	Label initialization operation*3	
		When the CPU module is powered on	When the operating status of the CPU module is switched from STOP to RUN*2
Labels outside the latch range	Set	Initial values are set.	Initial values are set.
	Not set	Zero clear*1	Zero clear*1
Latch type labels	Set	Initial values are set.	Initial values are set.
	Not set	Zero clear*1	Zero clear*1

*1 The labels (whose attributes are other than CONSTANT) to be assigned to label areas are cleared to zero.
*2 This operation is also performed when the operating status of the CPU module is switched from PAUSE to RUN.
*3 Since the labels assigned to devices are assigned to device areas, the labels are not initialized by this function.

Operating procedure

The following describes the operating procedure for this function.

1. Rebuild (reassign) all data.

[Convert] ⇨ [Rebuild All]

2. Set the CPU module to the STOP state.
3. Write the new program files. When initial values have been set in the labels used in the program, write the label initial value file together with the program files.

[Online] ⇨ [Write to PLC]

4. Set the CPU module to the RUN state. (Labels are initialized only when the operating state is changed from STOP to RUN for the first time.)
5. Labels are not initialized at the second state change or later. Label initial values can be set at the second state change or later depending on the parameter settings. (☞ Page 308 Label Initial Value Reflection Setting)

Point

Since this function automatically initializes labels (zero clear), the reset operation after writing is unnecessary.

Restriction

Labels are initialized only when the data is written to the programmable controller for the first time after data is rebuilt (reassigned). To write the same project to other programmable controller, rebuild (reassign) the data again before writing.

Precautions

The following describes precautions on this function.

- Even though SM326 (SFC device/label clear mode) is on (device/label values are kept), only labels are initialized when data is rebuilt and then written to the programmable controller. (Devices are kept.)
- This function is applied to the writing of all rebuilt (reassigned) data to the programmable controller with the engineering tool. This function is not applied to the writing of programs using the SLMP or FTP command or the boot function.
- Once all data is rebuilt (reassigned) and written, the label initialization cannot be canceled. For example, when all data is written to the programmable controller without being rebuilt (reassigned) again after the data is rebuilt (reassigned) and written to the programmable controller, labels will be initialized after the CPU module is powered on or the operating status of the CPU module is switched from STOP to RUN.

32.2 Label Initial Value Reflection Setting

With the default settings of the Process CPU, label initial values are not set in labels when the operating status of CPU module is switched from STOP to RUN even though the label initial values have been set for the labels.
This function can set whether or not to set label initial values when the operating status of the CPU module is switched from STOP to RUN.

Restriction

This function is supported by only Process CPUs. Before using this function, check the versions of the CPU module and engineering tool used. (☞ Page 779 Added and Enhanced Functions)

Label initialization operation

The following table lists the label initialization operations.

Target label	Label initial value setting	Label initialization operation*4			
		When "Label Initial Value Reflection Setting" has been set to "Disable"		When "Label Initial Value Reflection Setting" has been set to "Enable"	
		When the CPU module is powered on	When the operating status of the CPU module is switched from STOP to RUN*3	When the CPU module is powered on	When the operating status of the CPU module is switched from STOP to RUN*3
Labels outside the latch range	Set	Initial values are set.*1	Initial values are not set. (Values do not change.)	Initial values are set.	Initial values are set.
	Not set	Zero clear*2	Values do not change.	Zero clear*2	Values do not change.
Latch type labels	Set	Initial values are not set. (Values do not change.)	Initial values are not set. (Values do not change.)	Initial values are set.	Initial values are set.
	Not set	Values do not change.	Values do not change.	Values do not change.	Values do not change.

*1 Since values in labels outside the latch range are cleared when the CPU module is powered off, initial values are set in the labels when the CPU module is powered on even though "Label Initial Value Reflection Setting" has been set to "Disable".
*2 The labels (whose attributes are other than CONSTANT) to be assigned to label areas are cleared to zero.
*3 This operation is also performed when the operating status of the CPU module is switched from PAUSE to RUN.
*4 Since the labels assigned to devices are assigned to device areas, the labels are not initialized by this function.

Setting procedure

The following describes the procedure of disabling label initial values to be set when the operating status of the CPU module is switched from STOP to RUN.

1. Check that "Label Initial Value Reflection Setting" has been set to "Disable". (☞ Page 309 Setting method)
2. Write data to the programmable controller.
3. Set the CPU module to the RUN state. (Label initial values are not set.*1)

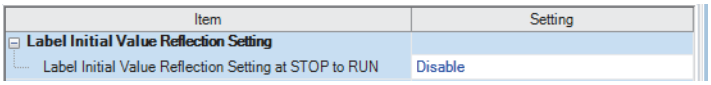
*1 Initial values are set when the CPU module is powered on or the operating status of the CPU module is switched from STOP to RUN for the first time after all data is rebuilt (reassigned) and written to the programmable controller.

Setting method

The following describes how to configure the label initial value reflection setting.

[CPU Parameter] ⇨ [File Setting] ⇨ [Label Initial Value Reflection Setting]

Window



Displayed items

Item	Description	Setting range	Default
Label Initial Value Reflection Setting at STOP to RUN	Set whether or not to set label initial values when the operating status of the CPU module is switched from STOP to RUN.	• Disable • Enable	Disable

Point

When this setting is set to "Enable" (Label initial values are set when the operating status of the CPU module is switched from STOP to RUN), the Process CPU will perform the same operation as the one performed by the programmable controller CPU. In the default setting, this setting has been set to "Disable". Thus, set this setting to "Enable" to perform the same operation as the one performed by the programmable controller CPU.

Precautions

The following describes precautions on the label initial value reflection setting.

- If the battery runs out and latch type labels cannot be kept at a power failure, values in the labels are cleared when the CPU module is powered off. Thus, initial values are set when the CPU module is powered on even though "Label Initial Value Reflection Setting" has been set to "Disable".

MEMO

PART 3 DEVICES AND CONSTANTS

This part consists of the following chapters.

33 DEVICES

34 CONSTANTS

35 HOW TO USE DEVICES CONVENIENTLY

33 DEVICES

This chapter describes the devices.

33.1 Device List

This section lists the devices.

Classification	Type	Device name	Symbol	Number of points of Default	Parameter-set range	Notation
User device	Bit	Input	X	12K points	Unchangeable	Hexadecimal
	Bit	Output	Y	12K points		Hexadecimal
	Bit	Internal relay	M	12K points	Changeable (☞ Page 154 Device Setting)	Decimal
	Bit	Link relay	B	8K points		Hexadecimal
	Bit	Annunciator	F	2K points		Decimal
	Bit	Link special relay	SB	2K points		Hexadecimal
	Bit	Edge relay	V	2K points		Decimal
	Bit	Step relay*3	S	0 points		Decimal
	Bit/word	Timer	T	1K points		Decimal
	Bit/word	Retentive timer	ST	0 points		Decimal
	Bit/double word	Long timer	LT	1K points		Decimal
	Bit/double word	Long retentive timer	LST	0 points		Decimal
	Bit/word	Counter	C	512 points		Decimal
	Bit/double word	Long counter	LC	512 points		Decimal
	Word	Data register	D	18K points		Decimal
	Word	Link register	W	8K points		Hexadecimal
	Word	Link special register	SW	2K points		Hexadecimal
	Bit	Latch relay	L	8K points		Decimal
System device	Bit	Function input	FX	16 points	Unchangeable	Hexadecimal
	Bit	Function output	FY	16 points		Hexadecimal
	Word	Function register	FD	5 points × 4 words		Decimal
	Bit	Special relay	SM	4K points		Decimal
	Word	Special register	SD	4K points		Decimal
Link direct device	Bit	Link input	Jn\X	16384 points (Max.)*1	Unchangeable	Hexadecimal
	Bit	Link output	Jn\Y	16384 points (Max.)*1		Hexadecimal
	Bit	Link relay	Jn\B	32768 points maximum*1		Hexadecimal
	Bit	Link special relay	Jn\SB	512 points (Max.)*1		Hexadecimal
	Word	Link register	Jn\W	131072 points maximum*1		Hexadecimal
	Word	Link special register	Jn\SW	512 points (Max.)*1		Hexadecimal
Module access device	Word	Module access device	Un\G	268435456 points (Max.)*1	Unchangeable	Decimal
CPU buffer memory access device	Word	CPU buffer memory access device	U3En\G	268435456 points (Max.)*1	Unchangeable	Decimal
			U3En\HG	12288 points maximum	Changeable	Decimal
Index register	Word	Index register	Z	20 points	Changeable (☞ Page 336 Index register setting)	Decimal
	Double word	Long index register	LZ	2 points		Decimal
File register	Word	File register	R/ZR	0 points	Changeable	Decimal
Refresh data register	Word	Refresh data register	RD	512K points	Changeable	Decimal
Nesting	—	Nesting	N	15 points	Unchangeable	Decimal

Classification	Type	Device name	Symbol	Number of points of Default	Parameter-set range	Notation
Pointer	—	Pointer	P	8192 points*2	Changeable (☞ Page 343 Pointer setting)	Decimal
	—	Interrupt pointer	I	1024 points	Unchangeable	Decimal
Other devices	—	Network No. specification device	J	—	Unchangeable	Decimal
	—	I/O No. specification device	U	—		Hexadecimal
	—	SFC block device*3	BL	320 points		Decimal
	—	SFC transition device*3*4	TR	0 points		Decimal

*1 These are the maximum points that can be handled in the CPU module. The number of points actually used differs depending on the module used.

*2 The default number of points is 16384 for the R120CPU, R120ENCPU, R120PCPU, and R120SFCPU.

*3 If using the items with the programmable controller CPU or Process CPU, check the versions of the CPU module and engineering tool. (☞ Page 779 Added and Enhanced Functions)
Note that the items cannot be used with the Safety CPU.

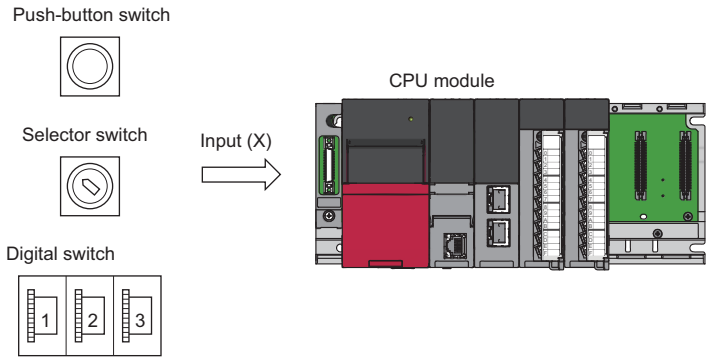
*4 Can be used as a device comment in the SFC program.

33.2 User Device

This chapter describes the user device.

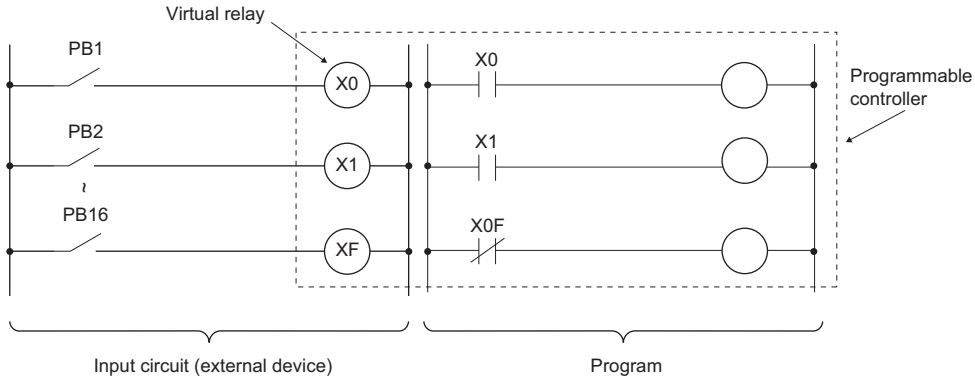
Input (X)

This device provides the CPU module with commands and/or data using an external device, such as pushbutton, transfer switch, limit switch, and digital switch.



Concept of input

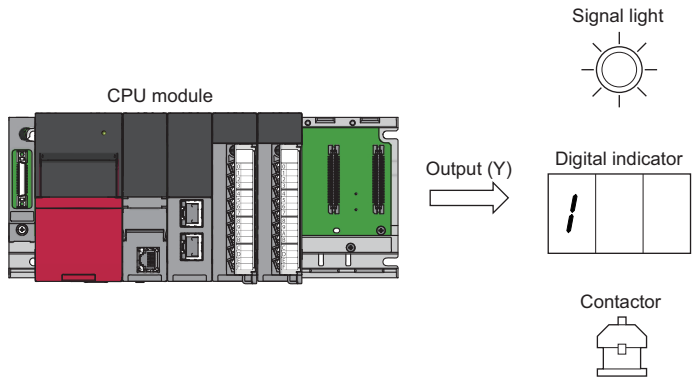
Assume that one virtual relay Xn is incorporated into the CPU module for each input point. In the program, a normally open contact and normally closed contact for the Xn are used.



Also, the input can be used as a target re-flesh (CPU module side) device of the remote input (RX), such as the CC-Link IE Field Network.

Output (Y)

This device outputs the control results of the program to various devices, such as external signal light/digital HMI/ electromagnetic switch (contactor)/solenoid.



Internal relay (M)

This device is used as an auxiliary relay within the CPU module. The following operations turn off all the internal relays.

- Turning power of the CPU module off and on
- Reset
- Latch clear

Latch relay (L)

This device is an auxiliary relay which enables latching (data retention during power failure). This device is used within the CPU module. This device latches operation results (ON/OFF information) even after the following operations.

- Turning power of the CPU module off and on
- Reset

Link relay (B)

This device is used as a CPU side device when refreshing bit data between the network module, such as the CC-Link IE Controller Network module and the CPU module.

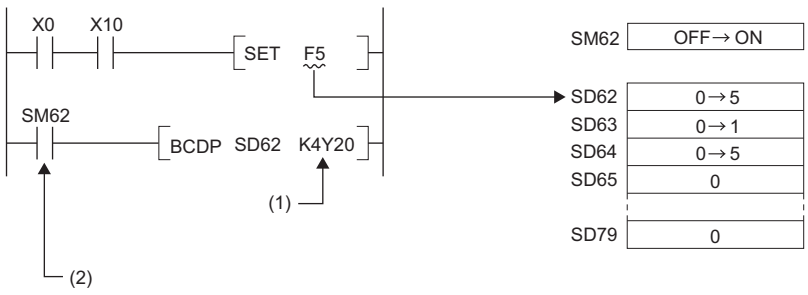
Refreshing network modules using link relay

Data are transferred/received between the link relay (B) within the CPU module and the link relay (LB) of the network module, such as the CC-Link IE Controller Network module. The refresh range is specified using parameters on the network module. The part which is not used for refreshing can be used for other applications.

Annunciator (F)

This device is an internal relay used for the program which detects malfunction/failure of customer-created equipment. When the annunciators are turned on, SM62 (Annunciator) is turned on, the number of activated annunciators and their device numbers are stored in SD62 (Annunciator number) to SD79 (Table of detected annunciator numbers).

Ex.
Failure detection program



(1) Outputs the annunciator number of the annunciator that turned on.
(2) ON detection of the annunciator

Also the number of the annunciator turned on first (the number stored in SD62) is registered in the event history.

Point

Only one annunciator number is registered in the event history while power is turned on.

On/off method for annunciator

Annunciators are turned on by either the SET F□instruction or the OUT F□ instruction. Annunciators are turned off by the RST F□instruction or the LEDR instruction or the BKRST instruction.

Point

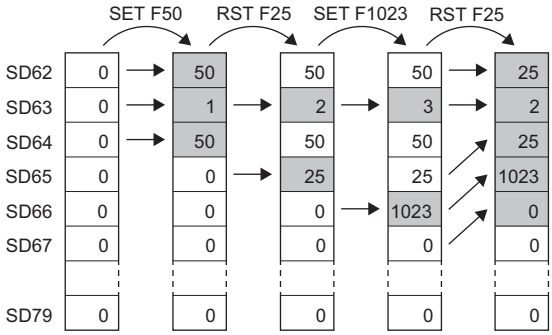
When the annunciators are turned on/off using any methods (e.g. the MOV instruction) other than shown above, the operation is the same as that of internal relays. As a result, SM62 is not turned on and annunciator numbers are not stored into SD62 and SD64 (Table of detected annunciator numbers) to SD79.

■Operations when annunciators are turned on

1. The annunciator numbers turned on are stored sequentially into SD64 to SD79.
2. The annunciator number stored into SD64 is stored into SD62.
3. SD63 value (Number of annunciators) is incremented by one.

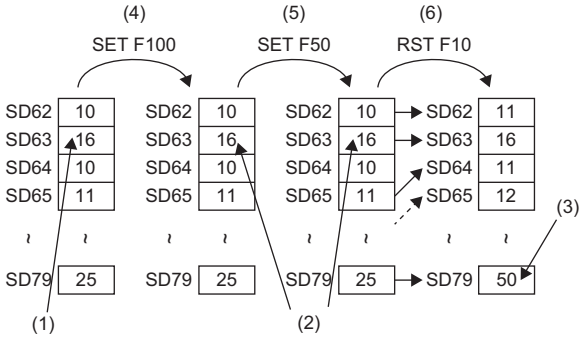
■Operations when annunciators are turned off

1. The number of the annunciator deactivated is removed, and the numbers of annunciators, which were lined up behind the removed one, move forward one by one.
2. When the annunciator number stored into SD64 is turned off, the annunciator number newly stored into SD64 is stored into SD62.
3. The value of SD63 is decremented by one. When the SD63 value gets decremented to 0, SM62 is turned off.



Point

If more than 16 annunciators are turned on, the 17th annunciator onwards are not stored into SD64 to SD79. However, if the numbers of annunciators registered in SD64 to SD79 are turned off, the lowest numbers, which are not registered in SD62 to SD79, of the numbers of annunciators which were turned on for the 17th on and after, are stored into SD64 to SD79.



(1) Annunciators have been stored for the maximum number (16).

(2) Because annunciators have been stored for the maximum number, the value does not change.

(3) The smallest number is stored.

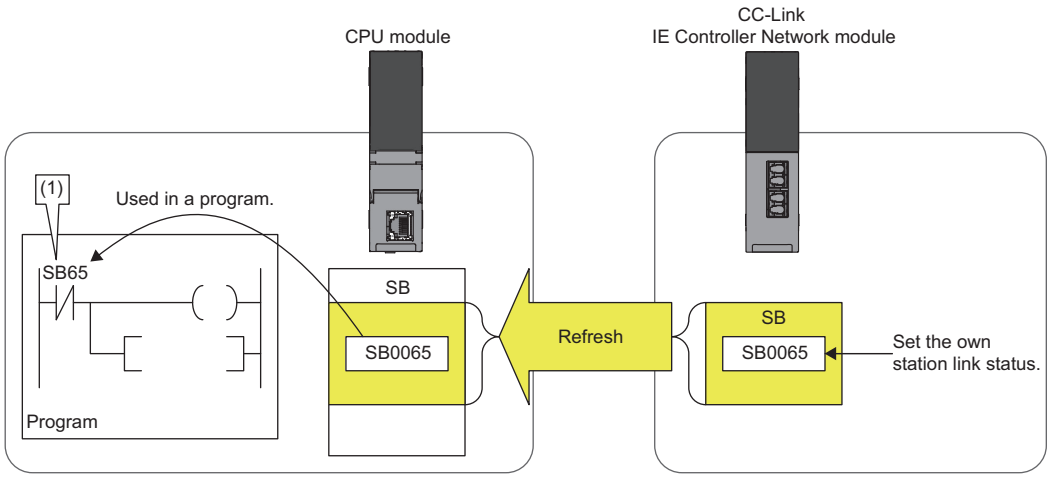
(4) Turn on the 17th device.

(5) Turn on the 18th device.

(6) Turn off the first device.

Link special relay (SB)

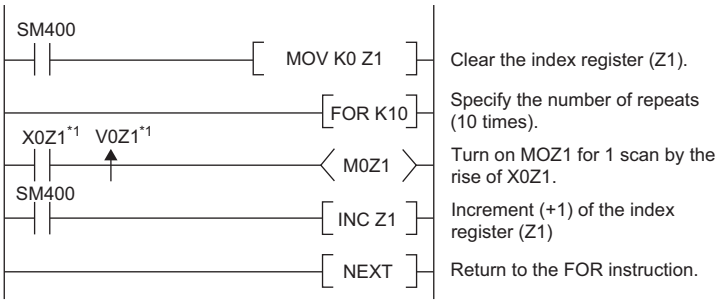
The communication status and error detection status of network modules, such as the CC-Link IE Controller Network module, are output to the link special relay (J□\SB□) on the network. The link special relay (SB) is a device for using as a refreshing target of link special relays in the network. The part which is not used for refreshing can be used for other applications.



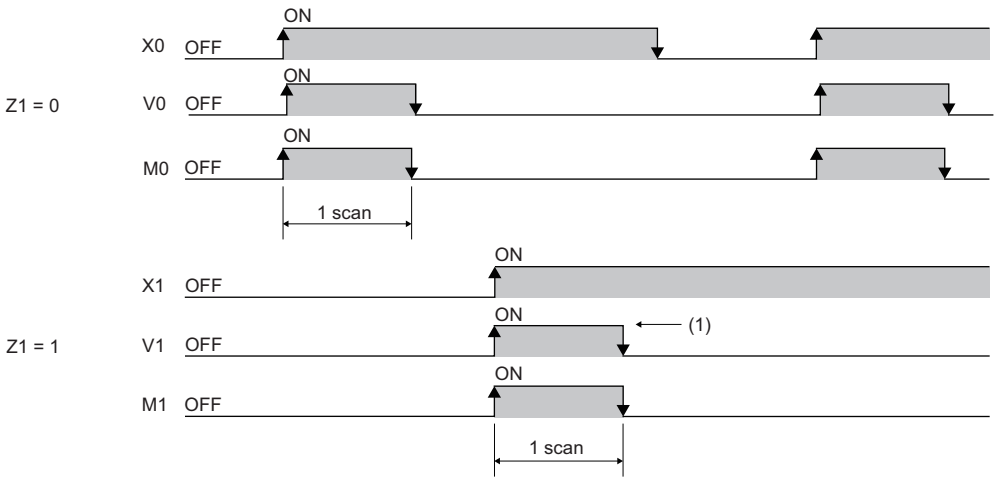
(1) The network status is checked.

Edge relay (V)

The edge relay is a device that memorizes operation results (on/off information) from the head of the ladder block, allowing its use only by the EGP/EGF instruction. This device is executed for various objectives such as the rising (off and on) detection in the structured programs by the index modification.



*1 Edge relay V0Z1 memorizes on/off information of X0Z1.



(1) When X1 is rising, this device is turned on for one scan duration.

Step relay (S)

This device is used when specifying SFC program steps. This device is also used when specifying step No. through such methods as verifying (monitor, current value changes) SFC programs with SFC control instructions or the engineering tool. (MELSEC iQ-R Programming Manual (Program Design))

Timer

This device starts measurement when the timer coil is turned on. When the current value reaches a setting value, time is up and the contact is turned on. This timer is an up-timing type device and therefore the current value matches a setting value when the timer time is up.

Types of timers

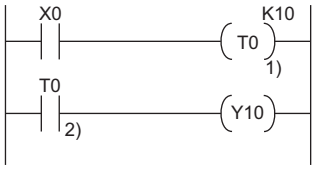
There are two types of timers: timer (T) which retains the current value in 16-bit units and long timer (LT) which retains it in 32-bit units. The timer (T) and the long timer (LT) are different devices and the number of device points can be set for each of them. In addition, there are the retentive timer (ST) and the long retentive timer (LST), both of which retain the current value even if the coil is turned off.*1

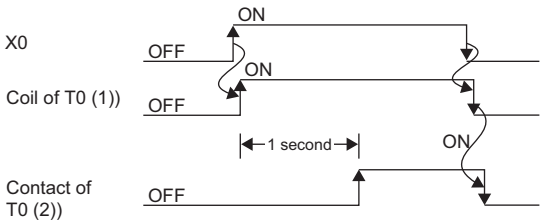
Types of timers			
Timer	Current value = 16 bits	Timer (T)	Low-speed timer
			High-speed timer
		Retentive timer (ST)	Low-speed retentive timer
			High-speed retentive timer
	Current value = 32 bits	Long timer (LT)	
		Long retentive timer (LST)	

*1 For the timer (T)/long timer (LT), the current value returns to 0 when the coil is turned off.

■Timer (T)

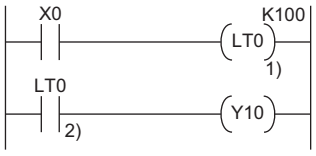
This device starts measurement when the coil of the timer is turned on. When the timer current value matches a setting value, time is up and the timer contact is turned on. When the timer coil is turned off, the current value returns to 0 and the timer contact is turned off.

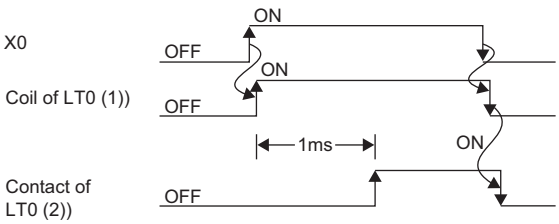




■Long timer (LT)

This device can count from 0 to 4294967295 to measure the time. If the measuring unit is set to 0.01ms, the measurable time range of this device is from 0 to about 11.9 hours. The current value of the long timers is updated by adding a difference of the counter which is used in the system when the OUT LT□ instruction is executed. Proper time can be measured even when the coil instruction of the long timer cannot be executed once per scanning because the counters used in the system count asynchronously with scanning. When the long timer coil is turned on, measurement starts and when the long timer current value matches a setting value, time is up and the long timer contact is turned on. When the long timer coil is turned off, the current value returns to 0 and the long timer contact is turned off.





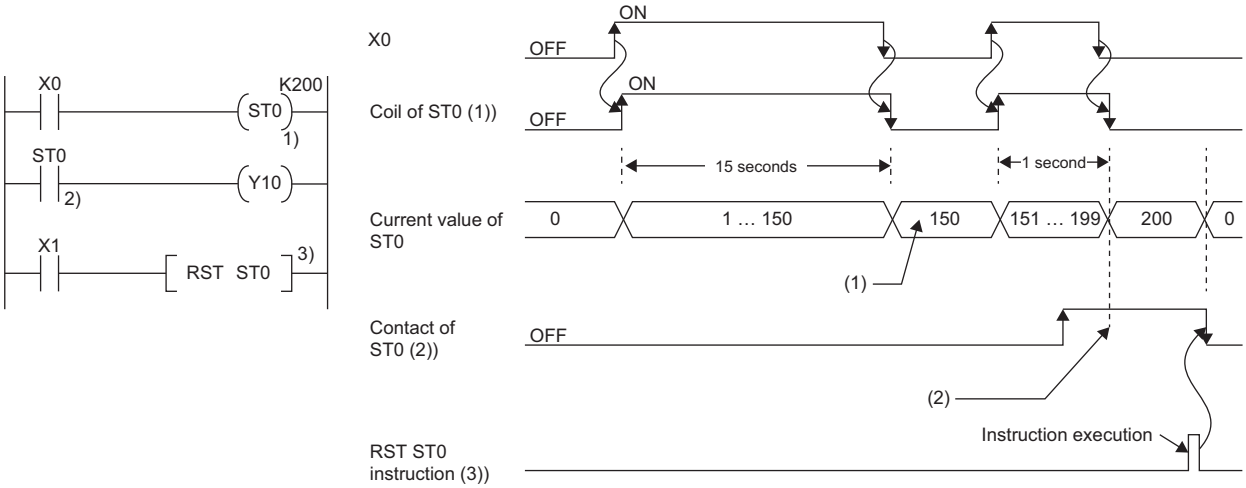
*1 This figure shows the example when the long timer time limit value setting is 0.01ms.

Point

- For the long timer contact to be turned on after the long timer coil is turned on, the next coil must be executed after the long timer coil turned off and on. The long timer contact is not turned on simultaneously when the long timer coil is off and on.
- The long timer (LT) can be used in interrupt programs. (☞ Page 55 Interrupt program)

■Retentive timer (ST)

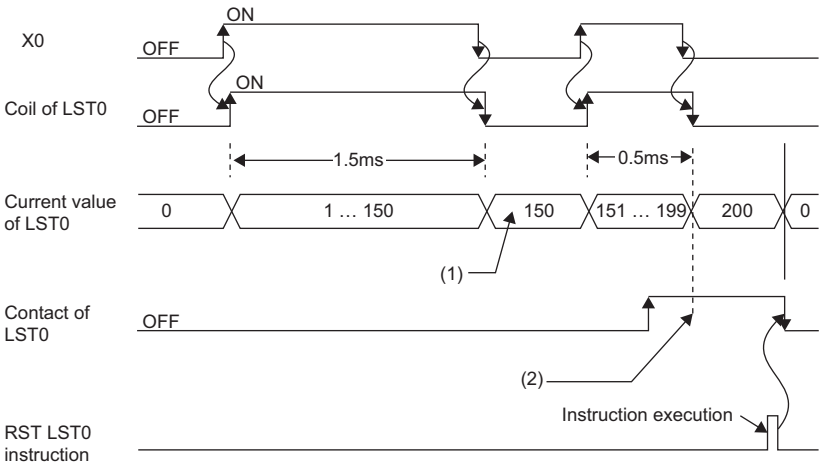
This device counts the sum of time duration in which the coil is turned on. When the retentive timer coil is turned on, measurement starts and when the timer current value matches a setting value (when time is up), the retentive timer contact is turned on. The current value and the contact on/off state is retained even when the retentive timer coil is turned off. When the coil is turned on again, measurement starts with the retained current value. To clear the retentive timer current value and turn off the contact, issue the RST ST□ instruction.



(1) Even though the coil (1)) turns off, the current value is held.
(2) Even though the coil (1)) turns off, the contact remains on.

■Long retentive timer (LST)

This device counts the sum of time duration in which the coil is turned on. When the long retentive timer coil is turned on, measurement starts and when the timer current value matches a setting value (when time is up), the contact is turned on. The current value and the contact on/off state is retained even when the long retentive timer coil is turned off. When the coil is turned on again, measurement starts with the retained current value. To clear the long retentive timer current value and turn off the contact, issue the RST LST□ instruction.



(1) Even though the coil turns off, the current value is held.
(2) Even though the coil turns off, the contact remains on.

Point

The long retentive timer (LST) can be used in interrupt programs. (☞ Page 55 Interrupt program)

■Low-speed/high-speed timer (T/ST)

The low-speed timer and high-speed timer are the same device which is set to a low speed or high speed timer by writing the instruction accordingly to specify it on the timer. For example, specifying OUT T0 generates a low-speed timer and specifying OUTH T0 produces a high-speed timer even when using the same T0 device. This also applies to the retentive timer.

Timer time limit value

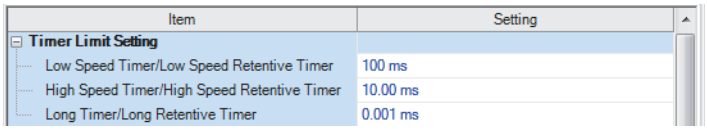
Although the low-speed timer and high-speed timer are the same device, timer limit value are different depending on how to specify the timer device (how to write the instruction). For example, specifying OUT T0 generates a low-speed timer and specifying OUT H T0 produces a high-speed timer even when using the same T0 device. This also applies to the retentive timer. The long timer cannot be set to a low-speed or high speed device. The time limit value for each timer is set in "Timer Limit Setting".

■Timer limit setting

The following window is to set the timer time limit values.

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Timer Limit Setting]

Window



Displayed items

Item	Description	Setting range	Default
Low Speed Timer/Low Speed Retentive Timer	Set the timer time limit value of T and ST used for the low-speed timer and low-speed retentive timer.	1 to 1000ms (unit: 1ms)	100ms
High Speed Timer/High Speed Retentive Timer	Set the timer time limit value of T and ST used for the high-speed timer and high-speed retentive timer.	0.01 to 100ms (unit: 0.01ms)	10.00ms
Long timer/Long retentive timer	Set the timer time limit value of LT and LST used for the long timer and long retentive timer.	0.001 to 1000ms (unit: 0.001ms)	0.001ms

Timer current value and the measurable range

Indicates the timer current value and the measurable range.

■Timer (T/ST)

The current value range is 0 to 32767. The measurable time range is from 0 to (timer time limit value × 32767).

■Long timer (LT/LST)

The setting range of the current value is 0 to 4294967295, which is the same as the range of unsigned 32-bit integers. The measurable time range is from long timer time limit value to (timer time limit value × 4294967295).

Handling timers

When executing the timer coil (the OUT T☐ instruction), the timer coil is turned on/off, the current value is updated, and the contact is turned on/off.

Accuracy of timers

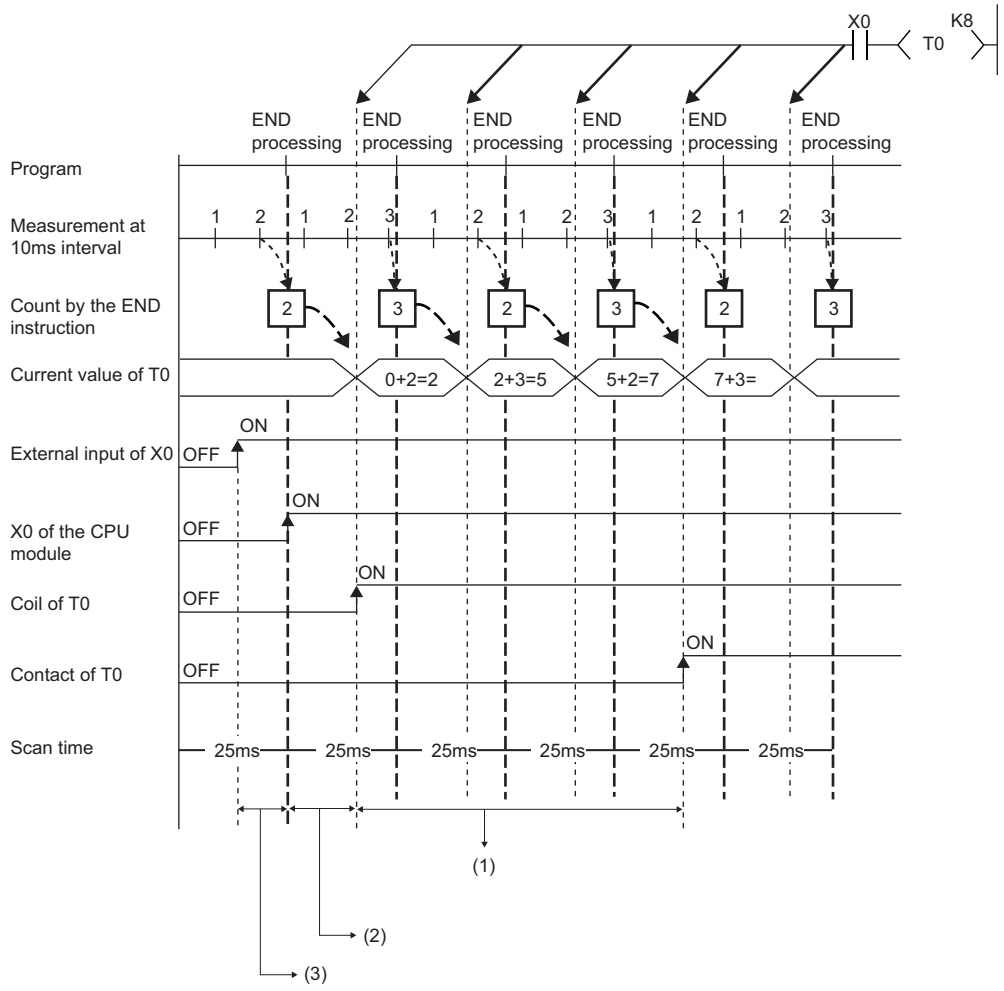
Indicates the accuracy of timers.

■Timer (T/ST)

The scan time value measured by the END instruction is added to the current value when the OUT T☐ instruction is executed. If the timer coil is turned off when the OUT T☐ instruction is executed, the current value is not updated. The maximum response accuracy of the timer (the time duration from capture of an input (X) to output of it) is "2 scan time + timer time limit setting."

Ex.

Timer limit setting = 10ms, setting value of T0 = 8 (10ms × 8 = 80ms), scan time = 25ms



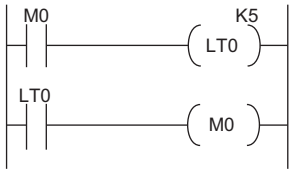
(1) Accuracy from when the coil of the timer turns on until when the contact of the timer turns on
- (1 scan time + timer limit setting) to (1 scan time)

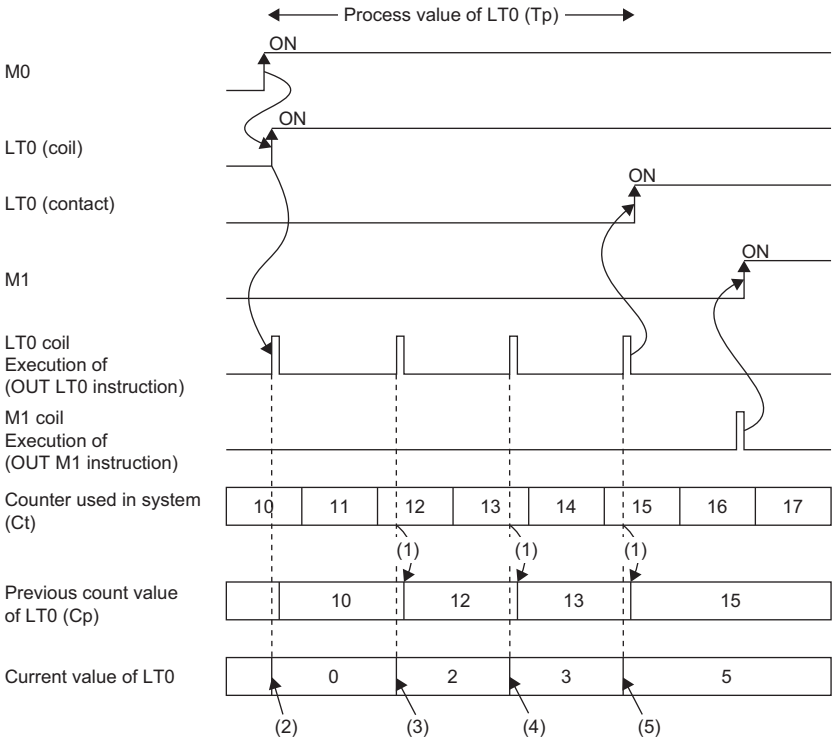
(2) Timing on when the coil of the timer turns on

(3) Input fetching timing

■Long timer (LT/LST)

In the following program, the accuracy of Tp (the time duration from the long timer coil activation to long timer contact activation) is (Ts-Tu) ≤ Tp < (Ts + Tu).

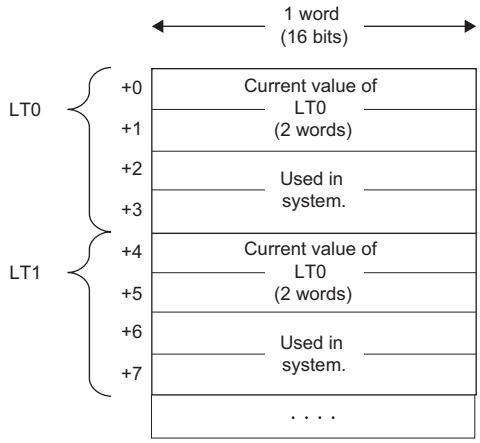




(1) Cp is updated.
(2) LT0 is initialized to 0.
(3) Add the result of (Cp - Ct): 0 + (12 - 10) = 2
(4) Add the result of (Cp - Ct): 2 + (13 - 12) = 3
(5) Add the result of (Cp - Ct): 3 + (15 - 13) = 5
Tp: time duration from the long timer coil activation to long timer contact activation
Ts: setting value of the long timer
Tu: time limit value of the long timer

Data configuration of Long timer (LT/LST)

The long timer (LT) and the long retentive timer (LST) use four words (64 bits) for each point. If the most significant two words are changed in a program, it is impossible to measure the time properly, because they are used by the system.



Point

The current value of the long timer (LT) and the long retentive timer (LST) is 32-bit data. It can be specified by the instructions which can specify signed or unsigned 32 bit data. (It cannot be specified with the BK + Instruction.)

Precautions

This section describes the precautions when using the timer and long timer.

■Precautions about timer usage

- Don't describe more than one coil (the OUT T☐ instruction) on the same timer during a single scanning. Doing so results in improper measurement because the timer current value is updated when the coil for each timer is executed.
- When timer is not used for data collection for each scan: While the coil on a timer (e.g. T1) is turned on, the timer coil (the OUT T☐ instruction) cannot be skipped by the instructions such as the CJ. When the timer coil is skipped, proper measurement is impossible because the timer current value is not updated. In addition, when the timer exists in a subroutine program, be sure to execute a subroutine call including T1 coil only once for each scanning operation while the coil of the timer (e.g. T1) is turned on. Otherwise proper measurement is impossible.
- The timer cannot be used in the initial execution type program, the fixed scan execution type program, or the event execution type program where the occurrence of an interrupt is set to be a trigger. The timer can be used in standby type programs if the coil of timer (OUT T☐ instruction) is executed one time for one scan using a subroutine program.
- The timer cannot be used in interrupt programs. The timer can be used in subroutine programs or FB programs if the coil of timer (OUT T☐ instruction) is executed one time for one scan.
- When setting value is 0: The contact is turned on when the OUT T☐ instruction is executed.
- Even when the setting value is increased after the timer time is up, the timer status does not change (time is continues to be up) and the timer does not operate.
- Do not set the timer setting value to 32768 or above. If used when set to 32768 or above, the timer contact may not turn on.

■Precautions about long timer usage

This section describes the precautions when using long timers (LT/LST).

- The long timer cannot be used in initial execution type programs.
- Even when the setting value is increased after the long timer time is up, the long timer status does not change (time continues to be up) and the long timer does not operate.

Counter

This device counts the number of rising operation of the input condition in the program. The counter is an up-timing type device and therefore when the count value matches a setting value, the count reaches its upper limit and the contact is turned on.

Types of counters

There are two types of counters: counter (C) which retains the counter values in 16-bit units and long counter (LC) which retains them in 32-bit units. The counter (C) and the long counter (LC) are different devices and the number of device points can be set for each of them.

■Counter (C)

This device uses one word for each point. The measurable range is 0 to 65535.

■Long counter (LC)

This device uses two words for each point. The measurable range is 0 to 429467295.

Point

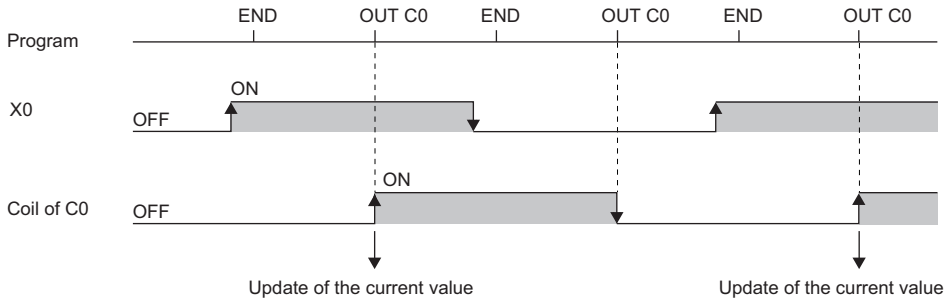
The long counter (LC) can be used in interrupt programs. (☞ Page 55 Interrupt program)

Counting process

When executing the counter coil (OUT C☐instruction / OUT LC☐instruction), the counter coil is turned on/off, the current value is updated (count value +1), and the contact is turned on/off. The current value is updated (count value +1), when the counter coil input is rising (off and on). The current vale is not updated when the coil input is turned off, on and on, and on and off.

[Ladder example]

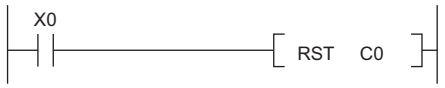
[Update timing of the current value]



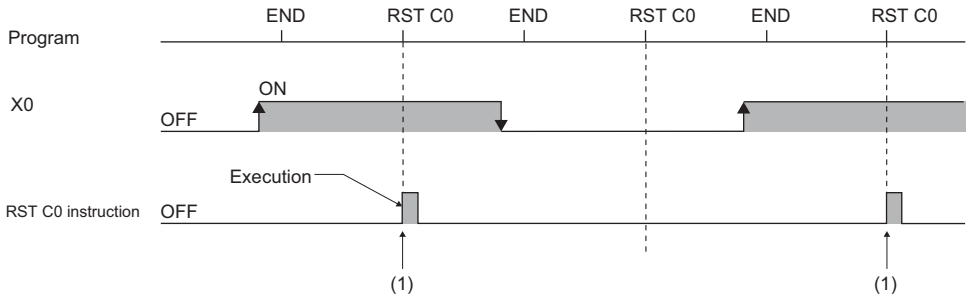
Resetting counters

The counter current value is not cleared even when the counter coil input is turned off. To clear the counter current value (resetting) and turn off the contact, issue the RST C☐/RST LC☐ instruction. When executing the RST C☐ instruction, the counter value is cleared and the contact is turned off.

[Ladder example]



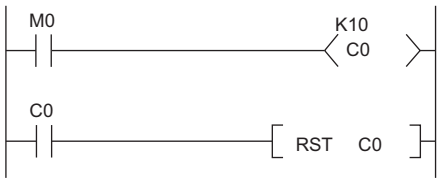
[Reset timing of the counter]



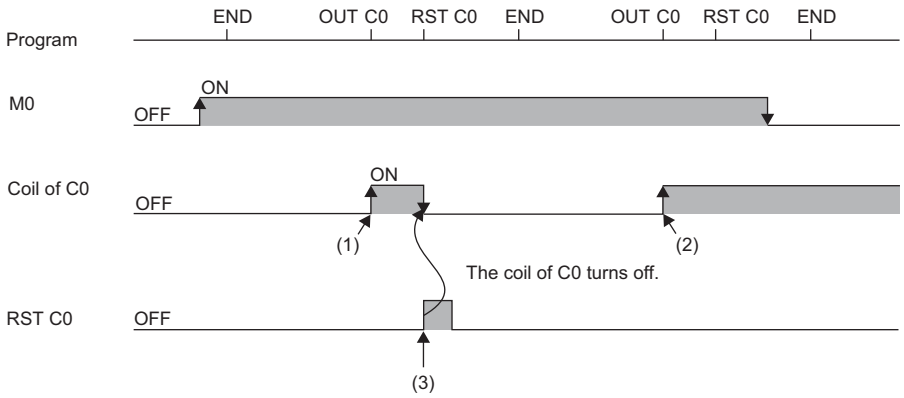
(1) The count value is cleared, and the contact turns off.

■Precautions about counter reset

When executing the RST C☐ instruction, the coil for C☐ is also turned off. If the execution condition for the OUT C☐ instruction is turned on after the RST C☐ instruction is executed, the coil of C☐ is turned on and the current value is updated (count value +1) when the OUT C☐ instruction is executed.

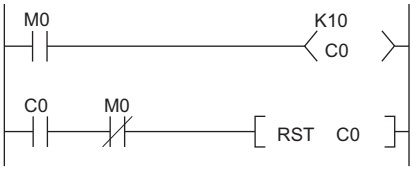


In the above ladder example, the coil of C0 is turned on by turning off and on M0 and as a result the current value is updated. When C0 count reaches its upper limit, C0 contact is turned on and C0 current value is cleared by execution of the RST C0 instruction. At this time C0 coil is also turned off. When M0 is turned on at the next scanning, the current value is updated because C0 coil is turned off and on during the OUT C0 instruction execution (the current value is changed to 1).



(1) The current value update contact turns on.
(2) The current value is updated because the coil of C0 turns on.
(3) The count value is cleared, and the contact turns off.

To address the above problem, insert a normally closed contact of the execution condition for the OUT C0 instruction into the execution condition for the RST C0 instruction to prevent C0 coil from being turned off while the execution condition (M0) of the OUT C0 instruction is turned on as shown in the following ladder example.



Maximum counting speed for counters

Counting is possible only when on/off time of the input condition is longer than the execution interval of the same OUT C☐ instruction. Maximum counting speed for counters can be obtained by the following equation:

$$\text{Maximum counting speed Cmax} = \frac{n}{100} \times \frac{1}{T} \text{ [times/s]}$$

*1 Duty (n) is a value which expresses the ratio of on/off time of the count input signal as a percent (%) value.

- When $T1 \geq T2$, $n = \frac{T2}{T1+T2} \times 100\%$
- When $T1 < T2$, $n = \frac{T1}{T1+T2} \times 100\%$

Data register (D)

This device can store numerical values.

Link register (W)

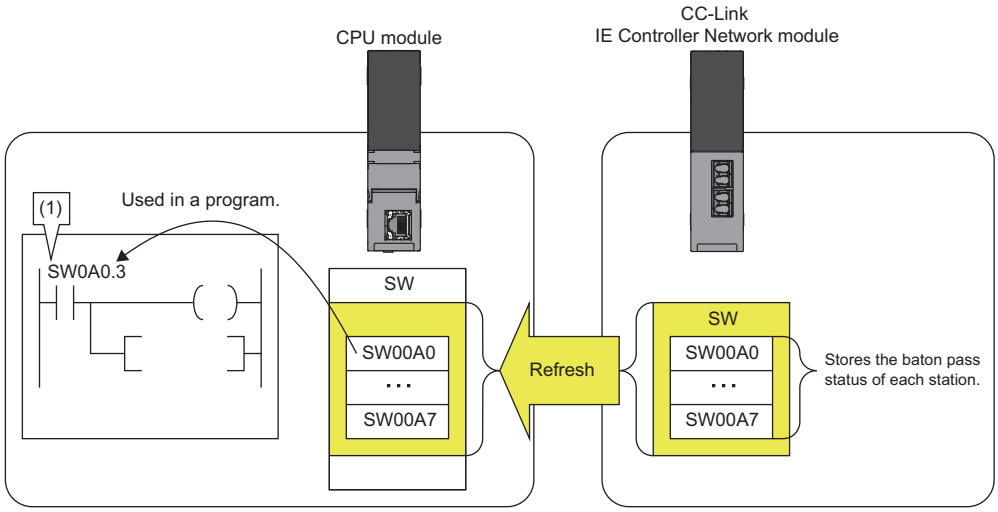
This device is used as a CPU module side device when refreshing word data between the network module, such as the CC-Link IE Controller Network module and the CPU module.

Refreshing network modules using link register

Data are transferred/received between the link register (W) within the CPU module and the link register (LW) of the network module, such as the CC-Link IE Controller Network module. The refresh range is specified using parameters on the network module. The part which is not used for refreshing can be used for other applications.

Link special register (SW)

Word data information on the communication status and error detection status of a network, such as CC-Link IE Controller Network, are output into the link special register (J□\SW□) on the network. The link special register (SW) is a device for using as a refreshing target of link special registers in the network. The part which is not used for refreshing can be used for other applications.



(1) The network status is checked.

33.3 System Device

The system device is used by the system. Assignment/capacity is fixed and cannot optionally be altered.

Function device (FX/FY/FD)

This device is used for the subroutine programs with argument passing. Data is written/read between the subroutine call sources with argument passing and the subroutine programs with argument passing. When using the function device in a subroutine program, the device used in each subroutine program call source can be determined. As a result, when using the same subroutine program, it can be used without being aware of the call source of other subroutine programs.

Function input (FX)

This device is used when passing on/off data to a subroutine program. In the subroutine program, bit data specified in a subroutine call instruction with argument passing are captured and used for operation. All the bit data specification devices of the CPU module are available.

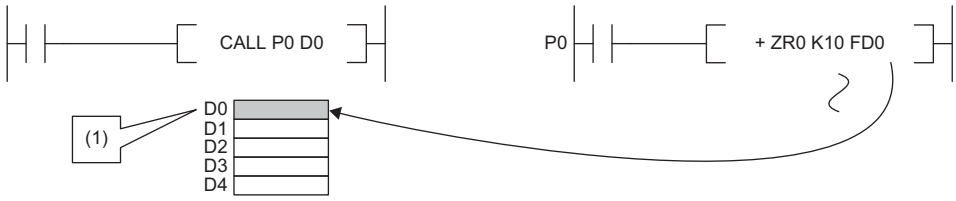
Function output (FY)

This device is used when passing operation results (on/off data) in a subroutine program to a subroutine program call source. Operation results are stored into the device specified in the subroutine program with argument passing. The bit data specification device other than the CPU module input (X) are available.

Function register (FD)

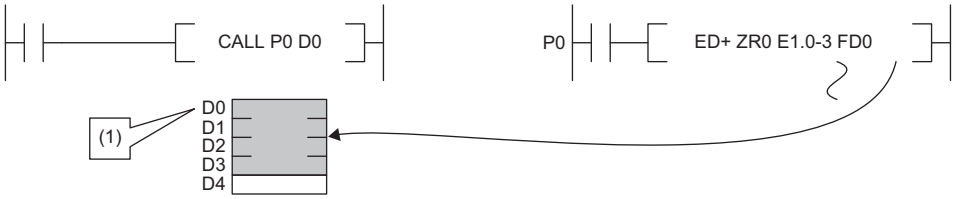
This device is used for writing/reading between the subroutine call source and the subroutine program. The input/output condition of the function register is automatically identified by the CPU module. "Source data" in a subroutine program means data inputted into the subroutine program. "Destination data" in a subroutine program means data outputted data from the subroutine program. One point of a function register occupies maximum of four words and can store 16-bit data, 32-bit data, 64-bit data, single-precision real number, and double-precision real number. However the number of words to be used depends on the instruction in the subroutine program.

For example, for the destination of addition instruction (+instruction) of 16-bit signed integer, one word is used.



(1) The data is stored in one point of D0.

Besides, for the destination of addition instruction (ED+instruction) of double-precision real number, four words are used.



(1) The data is stored in four points of D0 to D3.

Special relay (SM)

This is the internal relay for which the specification is defined in the CPU module, where the status of the CPU module is stored. (☞ Page 606 List of Special Relay Areas)

Special register (SD)

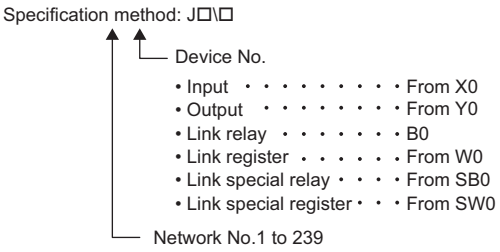
This is the internal register for which the specification is defined in the CPU module, where the status (diagnostics information, system information, etc) of the CPU module is stored. (☞ Page 635 List of Special Register Areas)

33.4 Link Direct Device

This device directly access link relays and/or link registers of the network module in the CC-Link IE Controller Network and/or CC-Link IE Field Network.

Specification method

Specify as shown below: The link register 10 (W10) of the network number 2 can be specified as "J2\W10".



Point

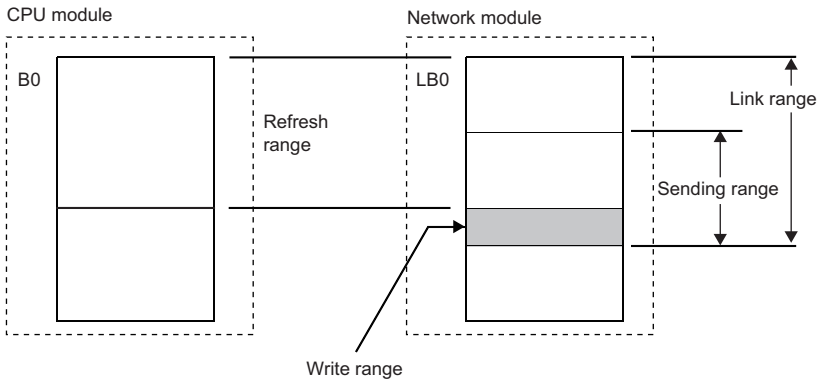
For bit devices, digit specification is allowed. (Example: J1\K1X0, J10\K4B0)

Specification range

All the link devices of the network module can be specified. The link devices which fall outside the range specified with the network refresh parameter can also be specified.

Specification range for writing

Writing should be done for the range which is within the link device range specified as a sending range of network parameters and outside the range specified as a refresh range for the network refresh parameter. Writing for a range specified as the refresh range overwrites link device data of the link module during refresh process. When writing data to a writing range of another station using the link direct device, data is overwritten with received data during data reception from another station.



Specification range for reading

Data can be read from all the range of link devices of the network module. There is only one network module for which writing/reading is allowed with the link direct device for each network number. When more than one network module with the same network number is mounted, the network module with the lowest slot number is the target for writing/reading by the link direct device.

Difference from link refresh

The following table shows the difference between the link direct device and link refresh.

Item		Link direct device	Link refresh
Description method in program	Input	Jn\K4X0...	X0...
	Output	Jn\K4Y0...	Y0...
	Link relay	Jn\K4B0...	B0...
	Link register	Jn\W0...	W0...
	Link special relay	Jn\K4SB0...	SB0...
	Link special register	Jn\SW0...	SW0...
Access range in relation to network module		All the link devices for each network module	The range specified in the refresh parameter
Guarantee range of access data		In word (16-bit) units	In word (16-bit) units

33.5 Module Access Device

This device directly accesses from the CPU module to the buffer memory of the intelligent function module mounted on the main base unit and extension base unit.

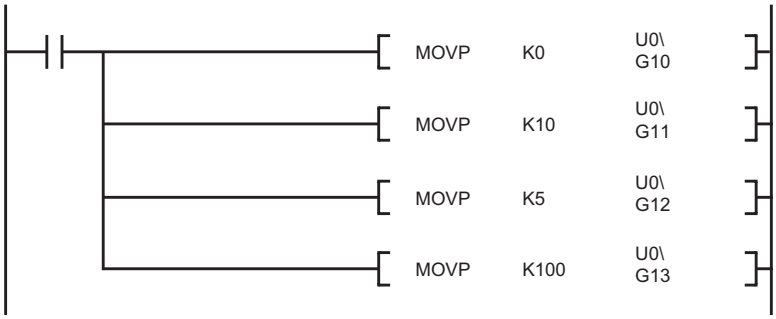
Specify this device with 'Un\Gn'. (Example: U5\G11)

Specified item		Value to be specified
Un	Start I/O number of intelligent function modules	Upper two digits when a start I/O number is described in three digits (00H to FFH) Example: 1F stands for X/Y1F0
Gn	Buffer memory address	0 to 268435455 (decimal)

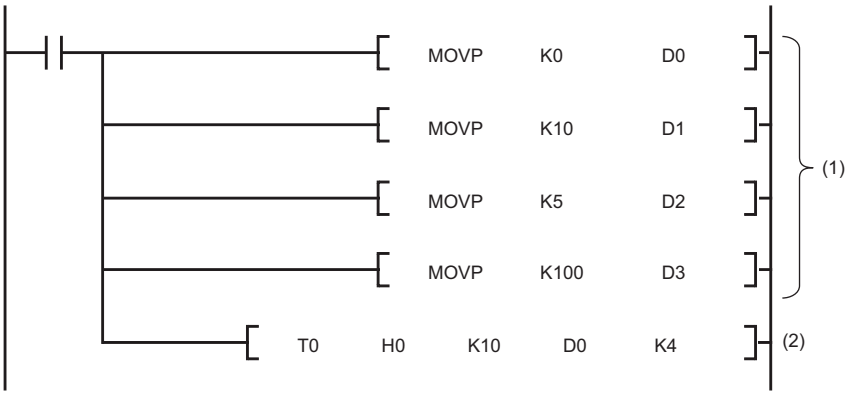
Point

When reading/writing the buffer memory data using the module access device more than twice within a program, the processing speed can be increased by conducting read/write operations at a single point of the program using the FROM/TO instruction.

• When data is written using more than one module access device:



• When data is written at a single point of the program using the TO instruction:



(1) Store data to devices such as the data register (D).

(2) Write data to only one point in the program.

Precautions

The following describes the precautions for when the module access device is used.

- If data is written to the refresh-target memory using a program while the refresh function is being used, the CPU module overwrites the data in the target memory at the execution of the refresh function. Thus, the expected operation may not be acquired. When the refresh function is used, do not write the data directly to the refresh-target memory but write it to the refresh-source memory.

*1 When the data is transferred from the CPU module to the module, the target memory is the buffer memory or link device. When the data is transferred from the module to the CPU module, the target memory is the specified device of the CPU module.

33.6 CPU Buffer Memory Access Device

This device accesses memory used by the built-in function of the CPU module, such as data writing/reading between CPU modules on the multiple CPU system and Ethernet function (☞ Page 293 Specification method thorough CPU buffer memory access device)

Specification method

Specify this device with 'Un\Gn'. (Example: U3E1\G4095, U3E2\HG1024)

Specified item		Value to be specified
Un (I/O number of the CPU module)	CPU No.1	3E0
	CPU No.2	3E1
	CPU No.3	3E2
	CPU No.4	3E3
G (CPU buffer memory area)	CPU buffer memory	G
	Fixed scan communication area	HG
n (CPU buffer memory address)		0 to 268435455 (decimal)

33.7 Index Register (Z/LZ)

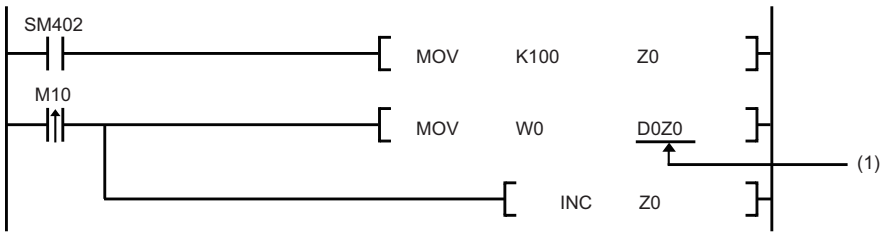
This device is used for the index modification of the device. The index modification is the indirect specification using the index register.

Types of index registers

There are two types of index registers: index register (Z) and long index register (LZ).

Index register (Z)

Used for 16-bit index modification

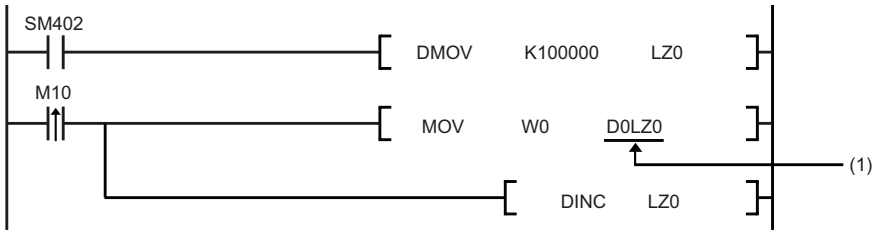


(1) Access D0Z0 = D100.

In addition, 32-bit index modification with ZZ expression using two index registers is also available.

Long index register (LZ)

Used for 32-bit index modification



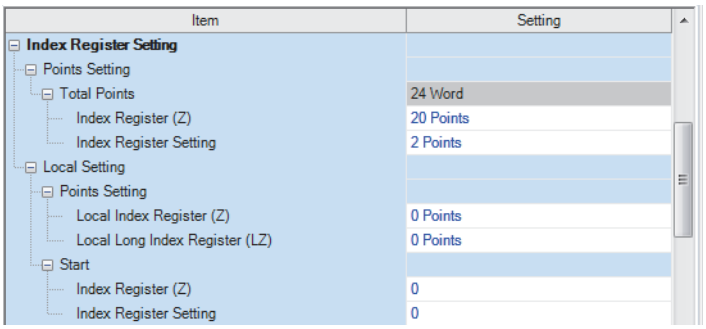
(1) Access D0LZ0 = D100000.

Index register setting

The following window allows to specify the number of points for the index register (Z) and long index register (LZ) and the range where they are used as a local device. The total number of points of the index register (Z) and the long index register (LZ) must be set to 24 words.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Index Register Setting]

Window



Displayed items

Item			Description	Setting range	Default
Number of points setting	Total Points		Show the total number of points for index register and long index register.	—	—
	Index register (Z)		Set the number of points for the index registers.	0 to 24 points (in two-point increments)	20 points
	Long index register (LZ)		Set the number of points for the long index registers.	0 to 12 points (in one-point increments)	2 points
Local setting	Number of points setting	Local Index register (Z)	Set the number of points for index registers used as a local device. Set within the range of the index register.	0 to 24 points (in one-point increments)	0 points
		Local long index register (LZ)	Set the number of points for long index registers used as a local device. Set within the range of the long index register.	0 to 12 points (in one-point increments)	0 points
	Head	Index register (Z)	Set the head for the local index registers. Set within the range of the index register.	0 to 23	0
		Long index register (LZ)	Set the head for the local long index registers. Set within the range of the long index register.	0 to 11	0

33.8 File Register (R/ZR)

This device is a word device for extension. This device is specifically a file register file which exists in the file storage area on the device/label memory.

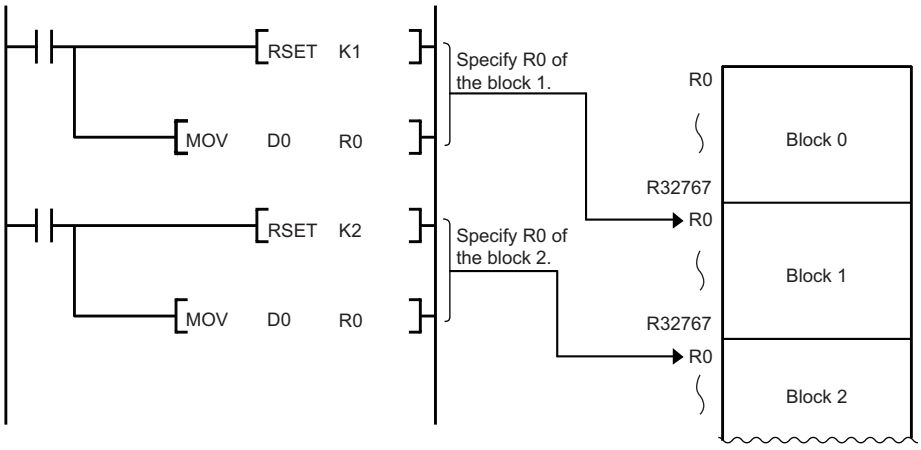
Specification method

There are two types of the specification methods for the file register: block switching and serial number methods.

Block switching method

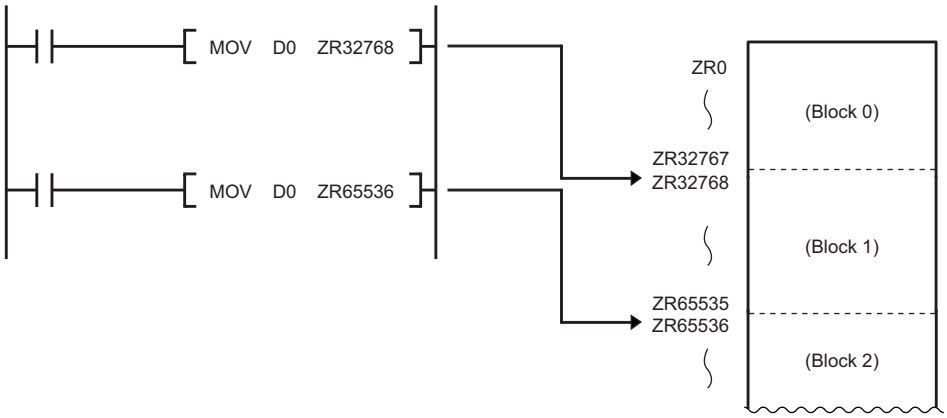
In this method the number of points of file register being used is specified by being divided in increments of 32K point (R0 to R32767). When using more than one block, specification is conducted by switching to the block number used by the RSET instruction. "R" is used as the device symbol. The range of "R" is from R0 to R32767. However in the following cases the upper limit of the device number is "block size (unit: word) -1."

- The file register file size is smaller than 64K bytes.
- The file register file size is not a multiple of 64K bytes and the end block is specified in the RSET instruction.



Serial number method

In this method file registers having more than 32K points are specified using serial device numbers. File registers of the multiple blocks can be used as consecutive file registers. "ZR" is used as the device symbol. The range of ZR is from ZR0 to (file register file size (unit: word) -1).



Setting file registers

This section describes the settings required to use the file registers.

Configuration procedure

This section describes the procedure to use the file registers.

1. Set the file register usage with [CPU Parameter].
2. To use the file register for each program, previously create the device memory which will become the file register file. (GX Works3 Operating Manual)
3. When using the file registers, which are common for all programs, the file registers of the file name and capacity, which have been set through setting of the file register, are created.*1

*1 If the capacity is not set, it must be set when creating a file register file and writing it to the programmable controller in the same manner as the procedure 2.

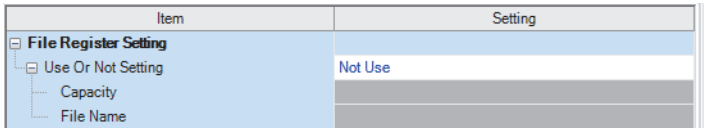
4. Write parameters and file register files into the CPU module.

File register setting

This setting must be completed before using the file registers.

[CPU Parameter] ⇨ [File Setting] ⇨ [File Register Setting]

Window



Displayed items

Item	Description	Setting range	Default
Use Or Not Setting	Specify whether or not file registers should be used.	• Not Use • Use File Register of Each Program • Use Common File Register in All Programs	Not Use
Capacity	Specify the capacity of the file register in units of 1K words when "Use Common File Register" in All Programs is selected.	This value depends on whether or not the extended SRAM cassette is mounted and its capacity. (☞ Page 151 The Setting Range of the Capacity of Each Area)	—
File name	Assign a file name to the file register when "Use Common File Register in All Programs" is selected.	1 to 60 characters	—

■Precautions

To access the CPU module from either of the following modules or engineering tools, the size of the file register needs to be 5696K words or less with an extended SRAM cassette (NZ2MC-16MBS) of 16MB.

- C Controller module
- MES interface module
- GOT2000 (when the programmable controller program monitor (R ladder) is used)
- Engineering tool version 1.015R or earlier
- CPU Module Logging Configuration Tool version 1.49B or earlier
- GX LogViewer version 1.49B or earlier
- MX Component version 4.10L or earlier

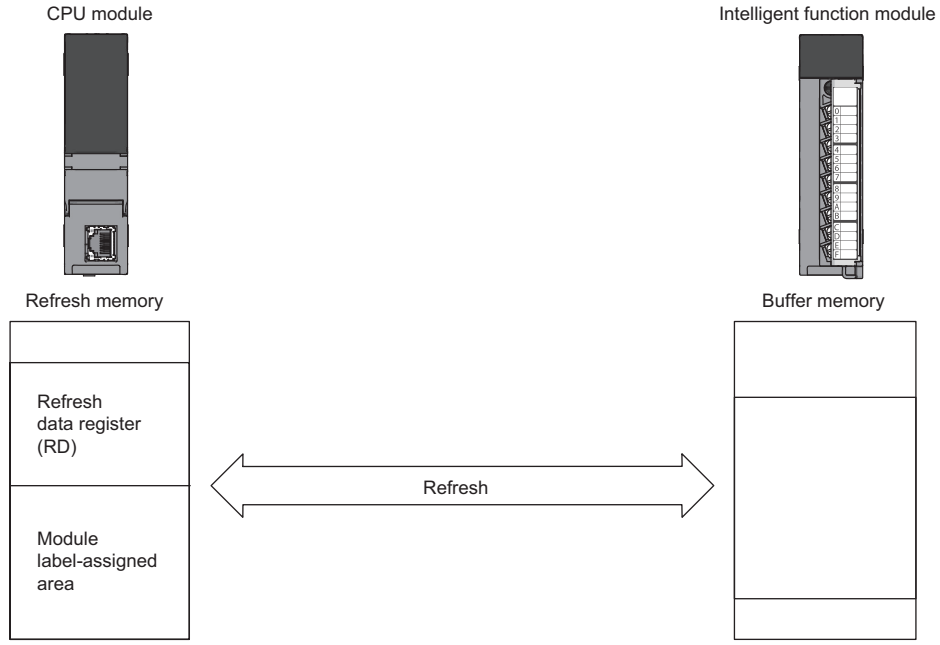
Clearing file registers

To clear the file registers, use the following methods (☞ Page 80 Memory Operation)

- Clearing in the program: write 0 into the file register range to be cleared.
- Clearing with engineering tool: clear them using engineering tool (📖 GX Works3 Operating Manual)

33.9 Refresh Data Register (RD)

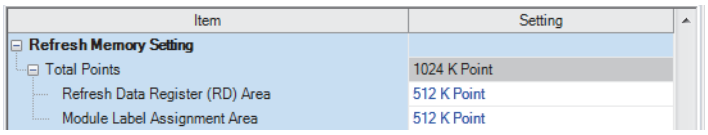
This device is provided for using as a refreshing target of buffer memory on the various devices, such as an intelligent function module. Refresh Data Register (RD) is assigned into the refresh memory area. (☞ Page 78 Refresh memory)



Refresh memory setting

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Refresh Memory Setting]

Window

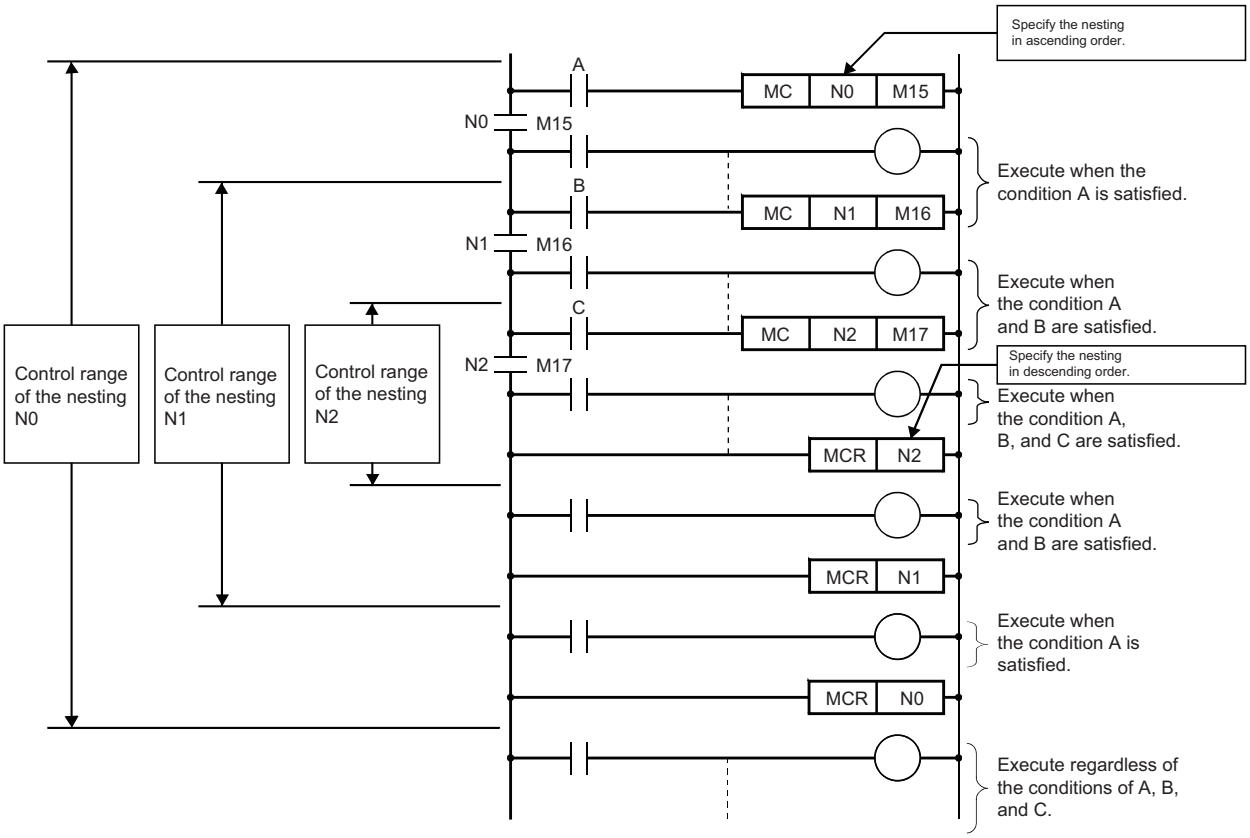


Displayed items

Item	Description	Setting range	Default
Total Points	Show the total number of points for the refresh data register and the assigned area.	—	1024K Point
Refresh Data Register (RD) area	Set the number of points of refresh data register.	0 to 1024K points (in one-point increments)	512K Point
Module Label Assignment Area	Set the number of points for the module label assigned area.	0 to 1024K points (in one-point increments)	512K Point

33.10 Nesting (N)

This device is used in the master control instructions (the MC/MCR instruction)*1 and enables the programming of operation conditions in a nesting structure. Specify this device from outside the nesting structure starting with the lowest number (in ascending order from N0 to N14).



*1 This instruction creates an effective ladder-switching program using opening/closing the common rail of the ladders.

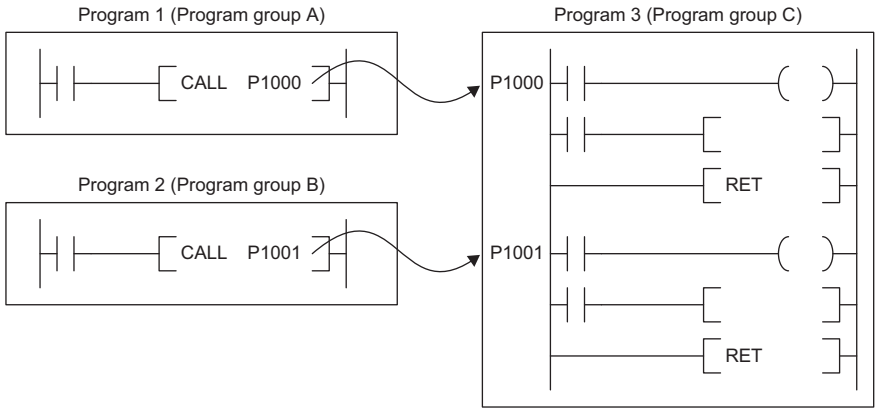
33.11 Pointer (P)

This device is used in the jump instructions (the CJ/SCJ/JMP instruction) and/or subroutine program call instructions (e.g. the CALL instruction). The type of pointer type includes the Global pointer and Local pointer. This pointer is used for the following objectives:

- Specifies the jumping target specification and label of the jump instructions (the CJ/SCJ/JMP instruction).
- Specifies the call target and label (the head of subroutine program) of the subroutine call instructions (e.g. the CALL instruction).

Global pointer

This is the pointer which enables calling by the subroutine call instruction from all the program being executed.



Effective use of number of points

The number of points for global pointers can be obtained with the following formula: "pointer number of points specified by the parameter" - "the total point number of local pointer being used for each program."

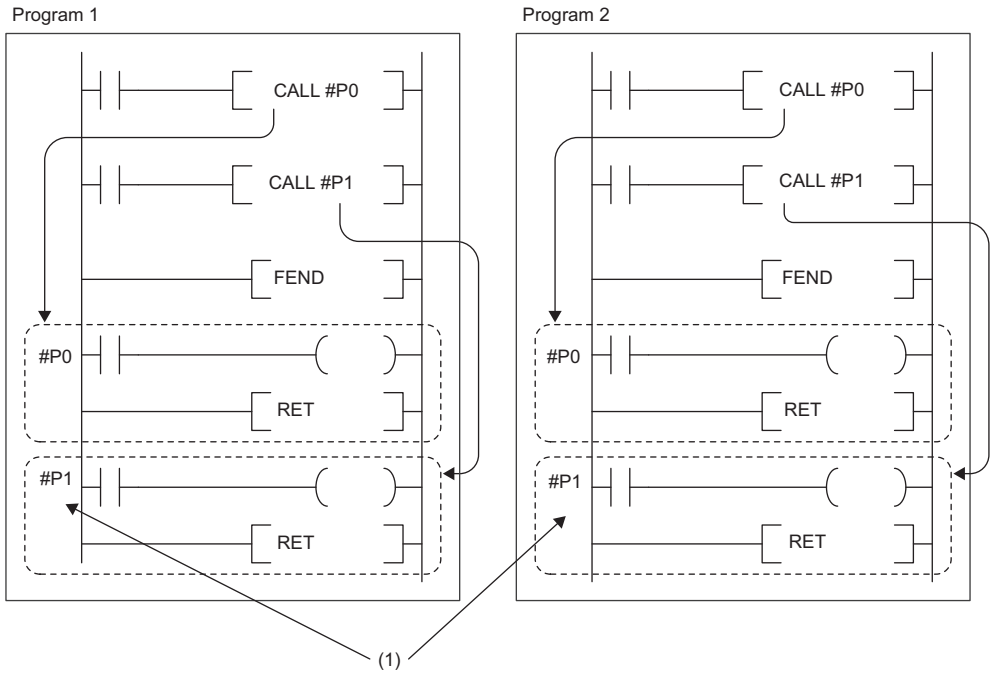
Precautions

This section describes the precautions when the global pointer is used.

- A global pointer with the same pointer number cannot be set as a label in multiple points.

Local pointer

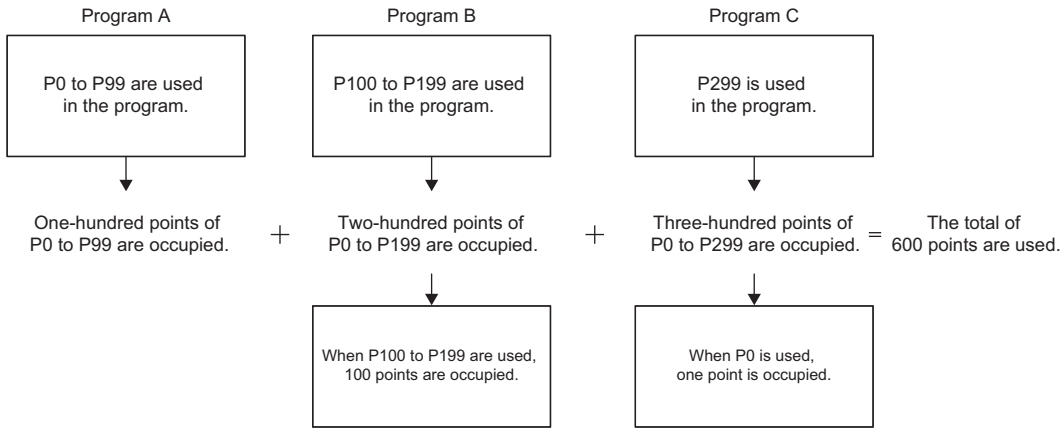
This is the pointer to be independently used in each program where the same pointer number can be used. This pointer is specified in the following format: # (pointer number) (Example: #P0) (☞ Page 354 Specification method for the local devices).



(1) The local pointers with the same number can be used between different programs.

Effective use of number of points

Local pointer number of points are shared among all the programs. The range of the local pointer number of points used by each program is from P0 to the maximum value of the local pointer being used in that program. For example, even when a program actually uses only P99, 100 points (P0 to P99) are considered to be used. When using local pointers in multiple programs, they can be effectively used by using them in ascending order from P0 in each program group.

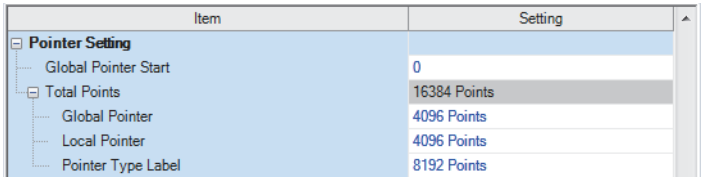


Pointer setting

The following menu item is to set pointers.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Pointer Setting]

Window



Displayed items

Item	Description	Setting range	Default
Global Pointer Start	Set the head of the global pointer.	P0 and over*1	0
Total Points	Show the total number of points for the pointer.	—	16384 points*2
Global pointer	Set the number of points of the global pointer.	• R04CPU, R04ENCPU, R08CPU, R08ENCPU, R08PCPU, R08SFCPU, R16CPU, R16ENCPU, R16PCPU, R16SFCPU, R32CPU, R32ENCPU, R32PCPU, R32SFCPU: 0 to 16384 points (in increments of 1 point) • R120CPU, R120ENCPU, R120PCPU, R120SFCPU: 0 to 32768 points (in increments of 1 point)	4096 points*3
Local pointer	Set the number of points of the local pointer.		4096 points*3
Pointer-type label	Set the number of points for the Pointer-type label assignment area.		8192 points*4

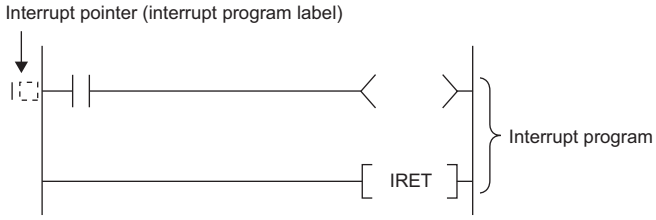
*1 Up to the number of "Total points of pointer device area" - "Number of points of global pointer".
*2 The default number of points is 32768 for the R120CPU, R120ENCPU, R120PCPU, and R120SFCPU.
*3 The default number of points is 8192 for the R120CPU, R120ENCPU, R120PCPU, and R120SFCPU.
*4 The default number of points is 16384 for the R120CPU, R120ENCPU, R120PCPU, and R120SFCPU.

Point

Specify a pointer number which is equal to or lower than "(end number of the pointer range specified in the parameter) - (number of points of the global pointers)."

33.12 Interrupt Pointer (I)

This device is used as a label located at the head of the interrupt program. This pointer can be used in all the programs being executed.



Point

Setting the execution type of program to the event execution type eliminates the need to write (I□) the interrupt pointer. (☞ Page 47 Interrupt occurrence by the interrupt pointer (I))

Interrupt factors of the interrupt pointer numbers

The interrupt factors of the interrupt pointer numbers are indicated.

Interrupt factor	Interrupt pointer number	Description
Interrupt from module	I0 to I15	This is a pointer used for modules which have the interrupt function.
Interrupt by the internal timer	I28 to I31	This interrupt pointer is used in fixed scan interrupts by the internal timer.
Inter-module synchronous interrupt	I44	This fixed scan interrupt pointer is used in the inter-module synchronization function.
Multiple CPU synchronous interrupt	I45	This fixed scan interrupt pointer is used in the multiple CPU synchronization function.
High-speed internal timer interrupt 2	I48	This interrupt pointer is used in fixed scan interrupts by the internal timer and can be specified in a shorter interval than interrupt pointer numbers I28 to I31.
High-speed internal timer interrupt 1	I49	
Interrupt from module	I50 to I1023	This is a pointer used for modules which have the interrupt function.

The priority for the interrupt pointer numbers and interrupt factors

The priority for the interrupt pointer numbers and interrupt factors are indicated.

Interrupt pointer number	Interrupt factor		Interrupt priority	Interrupt priority order
I0	Interrupt from module	1st point	5 to 8	9
I1		2nd point		10
I2		3rd point		11
I3		4th point		12
I4		5th point		13
I5		6th point		14
I6		7th point		15
I7		8th point		16
I8		9th point		17
I9		10th point		18
I10		11th point		19
I11		12th point		20
I12		13th point		21
I13		14th point		22
I14		15th point		23
I15		16th point		24
I28	Interrupt by the internal timer		4	8
I29				7
I30				6
I31				5
I44	Inter-module synchronous interrupt		3	4
I45	Multiple CPU synchronous interrupt			3
I48	High-speed internal timer interrupt 2		2	2
I49	High-speed internal timer interrupt 1		1	1
I50 to I1023	Interrupt from module		5 to 8	25 to 998

Point

- The interrupt priority is the order which is executed at the time of the multiple interrupt. (☞ Page 106 Interrupt priority)
- The interrupt priority order is the order which is executed when the interrupt factor with the same interrupt priority is generated. (☞ Page 109 Multiple interrupt execution sequence)

33.13 Other Devices

This section describes other devices.

Network No. specification device (J)

This device is used when specifying a network number with the Link dedicated instruction. (📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks))

I/O No. specification device (U)

This device is used when specifying an I/O number with the intelligent function module dedicated instruction. (📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks))

SFC block device (BL)

This device is used when specifying SFC program blocks. This device is also used when specifying step No. through such methods as verifying (monitor, current value changes) SFC programs with SFC control instructions or the engineering tool. (📖 MELSEC iQ-R Programming Manual (Program Design))

SFC transition device (TR)

This device is used when specifying SFC program transition conditions. This device can only be used for device comments for transition conditions. (📖 MELSEC iQ-R Programming Manual (Program Design))

34 CONSTANTS

This section describes the constant.

34.1 Decimal Constant (K)

Used when decimal data is specified in the program. Specify this device using K☐ character (e.g. K1234). The specification range depends on the argument data type of the instruction using the decimal constant as shown in the following table:

Argument data type of the instruction		Specification range for decimal constant
Data size	Data type name	
16 bit	Word (signed)	K-32768 to K32767
	Word (unsigned)/bit string (16 bit)	K0 to K65535
	16-bit data	K-32768 to K65535
32 bit	Double word (signed)	K-2147483648 to K2147483647
	Double word (unsigned)/bit string (32 bit)	K0 to K4294967295
	32-bit data	K-2147483648 to K4294967295

34.2 Hexadecimal Constant (H)

Used when hexadecimal data is specified in the program. Specify this device using H☐ character (e.g. H1234). To specify data in BCD, specify each digit of hexadecimal value with numerical characters 0 to 9. The specification range depends on the argument data type of the instruction using the hexadecimal constant. The range is H0 to HFFFF for 16-bit data size, and H0 to HFFFFFFFF for 32-bit data size.

34.3 Real Constant (E)

Used when real number is specified in the program. There are two types of real numbers: single-precision real number and double-precision real number. Specify it using E□character. (e.g. E1.234).

Setting range for real numbers

The setting ranges are different between the single-precision real number and double-precision real number.

Specification range for single-precision real numbers

Indicates the specification range for single-precision real numbers.

$-2^{128} < \text{device} \leq -2^{-126}$, 0, $2^{-126} \leq \text{device} < 2^{128}$

(E-3.40282347+38 to E-1.17549435-38, 0, E1.17549435-38 to E3.40282347+38)

Specification range for double-precision real number

Indicates the specification range for single-precision real numbers.

$-2^{1024} < \text{device} \leq -2^{-1022}$, 0, $2^{-1022} \leq \text{device} < 2^{1024}$

(E-1.7976931348623157+308 to E-2.2250738585072014-308, 0, E2.2250738585072014-308 to E1.7976931348623157+308)

Point

For the number of significant digits of real number data input by the engineering tool, refer to the following.

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

Activity when operation is in progress

Indicates the activity when operation is in progress.

When overflow and/or underflow occurs:

If an overflow and/or underflow occurs when the operation is in progress, the following process is performed:

- Activity when overflow occurs: an error is returned.
- Activity when underflow occurs: 0, rather than an error, is returned.

When a special value is input:

When a special value*1 is used for the input data operation, an error is returned. Note that when "-0" is generated during the operation is in progress, it is handled as "+0," and therefore "-0" is not returned as an operation result.

*1 The special values include -0, denormalized number, nonnumeric number, and ±∞.

Expression on program

The following expressions are used:

- Normal expression: specifies the numerical value to be set with no modification (e.g.: 10.2345 is expressed as "E10.2345").
- Exponent expression: specifies the numerical value to be set as (numerical value) × 10^n (e.g. 1234 is expressed as E1.234+3*1).

*1 +3 means 10^3.

34.4 Character String Constant

The character string can be specified by enclosing it with double-quotation marks (" "). (Example: "ABCDE") Note that the NULL character (00H) becomes the termination character.

Rules for special character

The special characters shown below cannot be used after they are inputted into the character string as they are. They become available by adding the Escape Sequence "$". (Example: ABC$'D$$E)

Special character	Character string to be entered when editing the circuit diagram
$	$$
'	$'
"	$"

Point

For details on character string data, refer to the following.

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

35 HOW TO USE DEVICES CONVENIENTLY

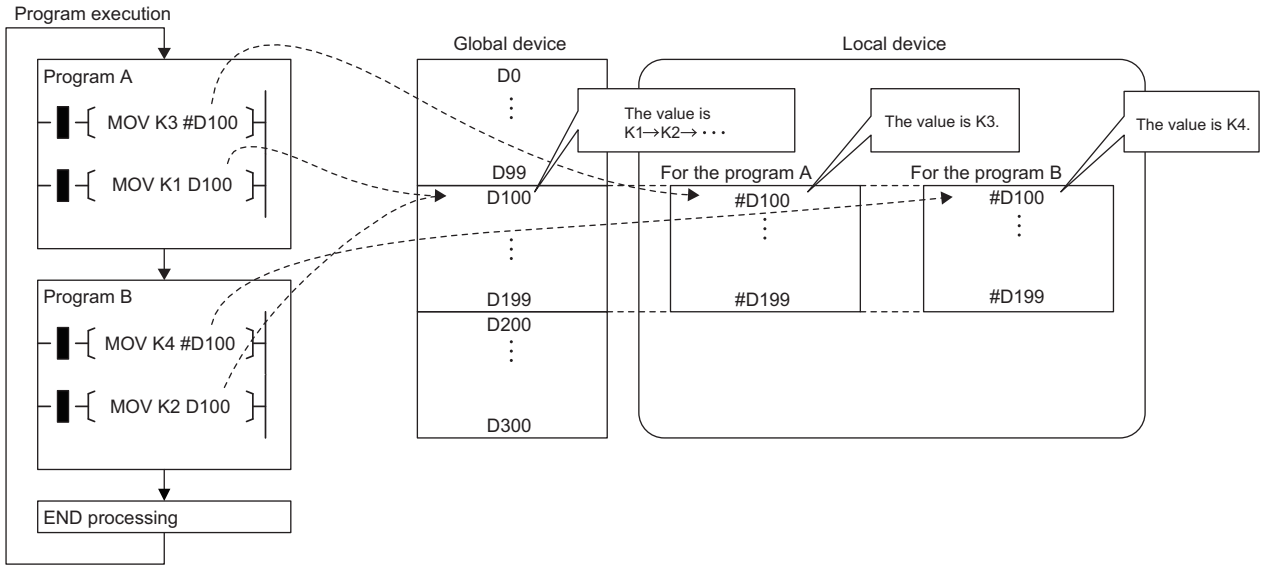
The device includes the global device and local device. As the convenient usage, there are index modification and indirect specification.

35.1 Global Device

This device can be shared by all the programs. All the devices that do not set as local device are handled as global device.

35.2 Local Device

This device can be used independently in each program. When creating multiple programs, programming can be completed without being aware of devices used in other programs.



Point

The local device area must be reserved independently from the global device area. Therefore a local device and global device with the same device number can exist.

Devices available as local device

This section lists devices available as local device.

- Internal relay (M)
- Edge relay (V)
- Timer (T, LT, ST, LST)
- Counter (C, LC)
- Data register (D)
- Pointer (P)

Point

Because the index register (Z, LZ) saves/returns during program execution, it should be regarded separately as the local index register unlike other local devices. (☞ Page 335 Index Register (Z/LZ))

Local device area

The CPU module reserves the local device area on the device/label memory based on the number of points setting of the local device when:

- CPU module is powered off and on or is reset.
- CPU module status is changed from STOP to RUN.

When local device is used in subroutine program:

Local devices to be used vary depending on whether SM776 (Local device setting at CALL) is turned on or off. Local index register to be used is also determined according to the SM776 setting.

SM776	Local device to be used
Off	Uses local devices of the program file from which subroutine program is called.
On	Uses local devices of the program file into which the subroutine program is stored.

Restriction

- In terms of on/off setting for SM776, the value (on or off) used when the subroutine call occurs is considered to be effective. Therefore when on/off setting for SM776 is switched in the subroutine program, the modified value (on or off) is not effective until the next subroutine call occurs.
- On/off setting for SM776 cannot be specified for each program file because it is enabled for each CPU module.

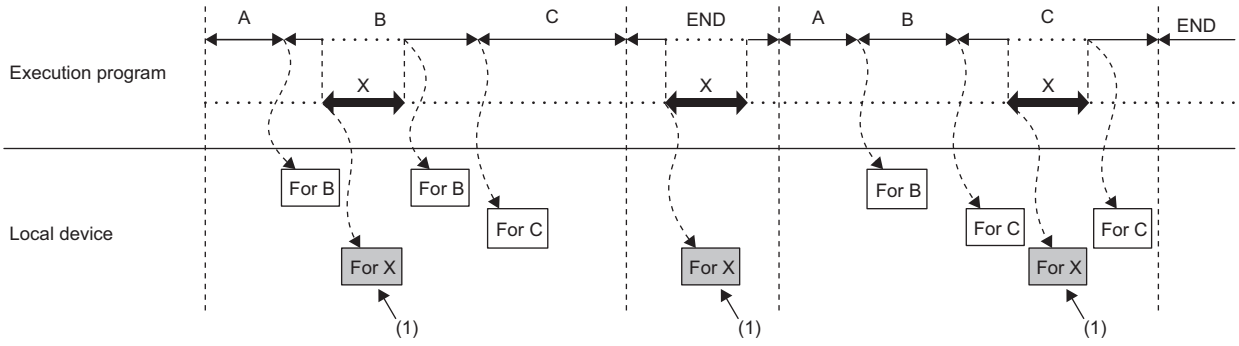
When local device is used in interrupt and other programs:

If using a local device for the interrupt program/fixed scan execution type program/event execution type program which uses the occurrence of an interrupt as a trigger, turn SM777 (Local device setting in interrupt programs) on. The programs will not function properly if SM777 is turned off.

Ex.

Operation if SM777 turned on in following setting

Program name	Execution type	Local device use/not use
A	Scan	Not Use
B	Scan	Use
C	Scan	Use
X	Fixed scan	Use



(1) Uses the program X local device.

For local index register, the register of the program file which has been executed before these programs is used regardless of the SM777 setting.

Restriction

- In terms of on/off setting for SM777, the value (on/off) used when executing the event execution type program which uses the interrupt program/fixed scan execution type program/occurrence of an interrupt as a trigger is considered to be effective. Therefore when on/off setting for SM777 is switched in the program to be executed, the modified value (on or off) is not effective until the event execution type program which uses the next interrupt program/fixed scan execution type program/occurrence of an interrupt as a trigger is executed.
- On/off setting for SM777 cannot be specified for each program file because it is enabled for each CPU module.
- When the local device monitor is executed, the monitor switches to the applicable local device. Consequently, if SM777 is off, when an interrupt occurs immediately after switching, and a local device is accessed, the local device being monitored by the local device monitor is used. (The local device for the program being run prior to the interrupt (program immediately before END) is not accessed.)

Clearing local device

Local device can be cleared to 0 by the following operations:

- CPU module is powered off and on or is reset.
- CPU module status is changed from STOP to RUN.
- CPU module status is changed from PAUSE to RUN.

Setting method for the local devices

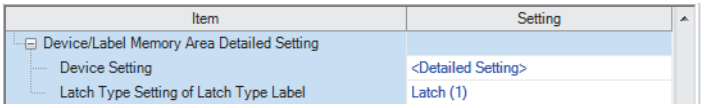
Set the range where each device will be used as a local device and also set whether or not it should be used.

Range setting

The range setting for local devices is common to all the programs. Therefore the range for local devices cannot be set for each program.

Operating procedure

"Device/Label Memory Area Detailed Setting" window



"Device Setting" window

Item	Symbol	Local Device		
		Start	End	Points
Input	X			
Output	Y			
Internal Relay	M			
Link Relay	B			
Link Special Relay	SB			
Annunciator	F			
Edge Relay	V			
Step Relay	S			
Timer	T			
Long Timer	LT			
Retentive Timer	ST			
Long Retentive Time	LST			
Counter	C			
Long Counter	LC			
Data Register	D			
Link Register	W			
Link Special Registe	SW			
Latch Relay	L			
Total Device				0.0K Word
Total Word Device				0.0K Word
Total Bit Device				0.0K Bit

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

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

2. Set the range where each device will be used as a local device.

Point

Configure the setting range of the local device within the range which has been set for the number of device points. The number of local devices used is calculated by the following calculation formula. Set the number of local devices used so that the number is equal to or less than the capacity of the local device area.

Total number of local devices used = ((A ÷ 16) + B + (C × 2) + (D × 4) + ((E × 2) ÷ 16)) × F

- A: Number of points of the local devices M and V
- B: Number of points of the local devices D, T (current value), ST (current value), and C (current value)
- C: Number of points of the local device LC (current value)
- D: Number of points of the local devices LT and LST
- E: Number of points of the local devices T (contact/coil), ST (contact/coil), C (contact/coil), and LC (contact/coil)
- F: Number of programs using the local device

Setting unit

The increment of setting of a local device is the same as that of a global device. (☞ Page 155 Range of use of device points)

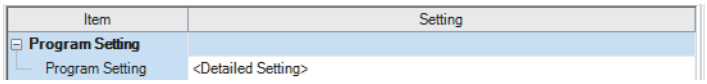
Setting whether or not it should be used

Whether or not local devices should be used can be set for each program. Since the local device area of program for which "Do not use" has been set is not assured, it can suppress unnecessary consumption of device/label memory.

[CPU Parameter] ⇨ [Program Setting]

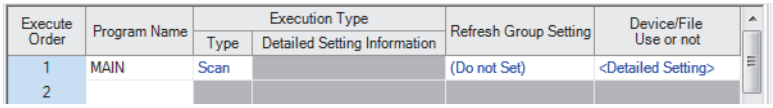
Operating procedure

"Program Setting" window



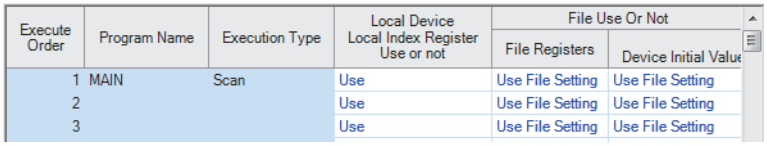
Item	Setting
Program Setting	
Program Setting	<Detailed Setting>

"Detailed Setting" window



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

"Setting of Device/File Use Or Not" window



Execute Order	Program Name	Execution Type	Local Device Local Index Register Use or not	File Use Or Not	
				File Registers	Device Initial Value
1	MAIN	Scan	Use	Use File Setting	Use File Setting
2			Use	Use File Setting	Use File Setting
3			Use	Use File Setting	Use File Setting

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

2. Click "Detailed Setting" of "Device/File Use or not".

3. Set whether or not it should be used for each program in the setting of "Local Device Local Index Register Use or not".

Restriction

Do not use local devices in a program which is configured not to use local devices.

Specification method for the local devices

To specify the local device in the program, add "#".

Ex.

For example, #D100, K4#M0, and @#D0 can be used.

Point

Local device is listed with a preceding # symbol in the program. This symbol is helpful to differentiate local devices from global devices.

Precautions

This section describes the precautions when using local devices.

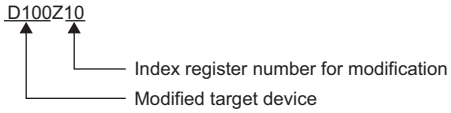
- Like global devices, the timer (T, LT, ST, LST) and counter (C, LC) specified as a local device cannot be checked for the device range. Therefore when operates the address in the index modification or indirect specification, be careful not to exceed the specified device range.
- Accessing the range including both global and local devices by the index modification is not allowed.
- When the range of the 32-bit index modification is across the setting ranges of local devices of the index register, proper index modification is impossible.
- Local devices are not latched.

35.3 Index Modification

Specify the device number using the index register. The device number to be used is "Device number of device targeted for modification" + "Contents of index register".

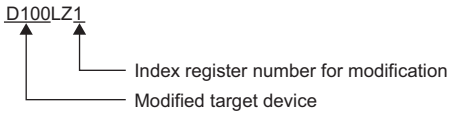
16-bit index modification

The device number is modified using the index register (Z). The modification range for the device in the case of the 16-bit index modification is -32768 to 32767.



32-bit index modification

The device number is modified using the long index register (LZ). The modification range for the device in the case of the 32-bit index modification is -2147483648 to 2147483647.



Device for which Index modification can be performed

The following table lists the devices that can be targeted for index modification.

Item	Description
16-bit index modification	X, Y, M, L, B, F, SB, V, S*4, T*1, LT*1, ST*1, LST*1, C*1, LC*1, D, W, SW, SM, SD, Jn\X, Jn\Y, Jn\B, Jn\SB, Jn\W, Jn\SW, Un\G, U3En\G, U3En\HG, R, ZR, RD, P*3, I*3, BLn\S*4, J, U, K, H
32-bit index modification	M, B, SB, T*1, LT*1, ST*1, LST*1, C*1, LC*1, D, W, SW, Jn\B*2, Jn\W*2, Un\G*2, U3En\G*2, U3En\HG*2, R, ZR, RD, K, H

*1 The devices can be used for the contact, coil and current value.
*2 For network numbers and the specification source of I/O numbers, 32-bit index modification cannot be performed.
*3 When the devices are used as an interrupt pointer, index modification cannot be performed.
*4 If specifying the devices with the programmable controller CPU or Process CPU, check the versions of the CPU module and engineering tool used. (☞ Page 779 Added and Enhanced Functions)
For details on the modification range, refer to the following.
📖 MELSEC iQ-R Programming Manual (Program Design)
Note that the devices cannot be specified with the Safety CPU.

Combination of index modification

This section describes the combination of index modification.

Modification order for the device specification and index modification

According to the priority order shown below, the device specification (digit specification, bit specification, indirect specification) and index modification can be applied. However, some word devices may not follow the priority order shown below.

Order of priority	When the device targeted for the device specification and index modification is the bit device	When the device targeted for the device specification and index modification is the word device
High ↑ ↓ Low	1: Index modification 2: Digit specification	1: Index modification 2: Indirect specification 3: Bit specification

Specification method combined with device specification

The device targeted for specification is modified in order of: 1st modification, 2nd modification and then 3rd modification. Besides, the following contents can be used only for the device for which the 1st modification can be applied. (For example, index modification + digit specification is impossible for the function input (FX).)

Device targeted for specification	1st modification	2nd modification	3rd modification	Example
Bit device	Index modification	Digit specification	—	K4M100Z2
Word device	Index modification	Bit specification	—	D10Z2.0
	Index modification	Indirect Specification	—	@D10Z2
	Bit specification	Index modification	—	D10.8Z2
	Indirect Specification	Bit specification	—	@D10.8
	Index modification	Indirect Specification	Bit specification	@D10Z2.8
	Indirect Specification	Bit specification	Index modification	@D10.8Z2

Precautions

This section describes the precautions on using index modification.

Index modification between the FOR and NEXT instructions

Between the FOR instruction and the NEXT instruction, pulse output is provided through the edge relay (V). However, pulse output by the PLS, PLF, or pulse conversion (□P) instruction is not available (☞ Page 318 Edge relay (V))

Index modification by the CALL instruction

In the CALL instruction, pulse output is provided through the edge relay (V). However, pulse output by the PLS, PLF, or pulse conversion (□P) instruction is not available (☞ Page 318 Edge relay (V))

Device range check for index modification

For details on the device range check when index modification is performed, refer to the following.

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

of the index modification range due to switching from 16-bit to 32-bit

To change the index modification range for switching from 16-bit to 32-bit, the user must:

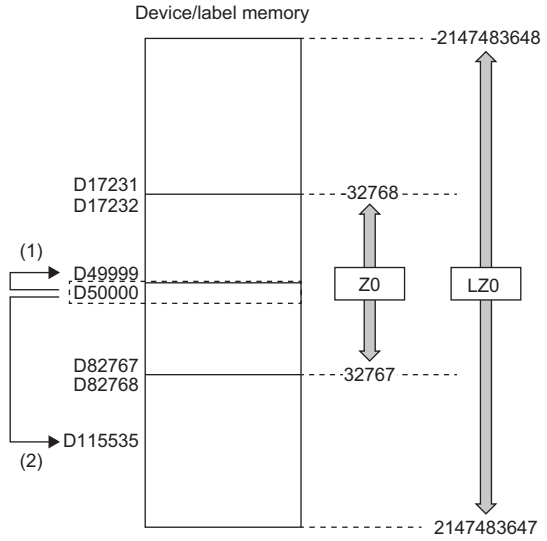
- Review the index modification block(s) within the program.
- For 32-bit index modification with ZZ expression, because the specified index register (Zn) and the immediately following index register (Zn+1) are used, caution must be taken to prevent duplicated index registers from being used.
- Review the number of points of the index register (Z) and that of the long index register (LZ), which are specified in "Index Register Setting" (☞ Page 336 Index register setting)

When values are stored in the index registers

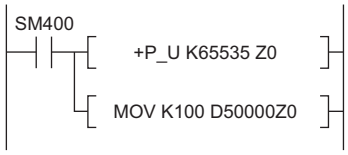
For 16-bit index modification using the index register (Z), the range is -32768 to 32767. Therefore, when values within the range from 32768 to 65535 are stored in the index register (Z) for an instruction which processes unsigned data, the instruction doesn't work in design because the range of the index modification will be -32768 to 32767. For the range of values larger than or equal to 32768, the long index register (LZ) must be used so that 32-bit-based index modification can be applied.

Ex.

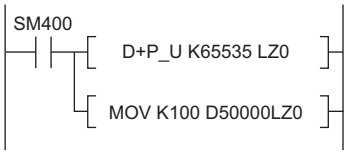
Operation for Index modification



(1) When unexpected operation is executed



(2) When proper operation is executed

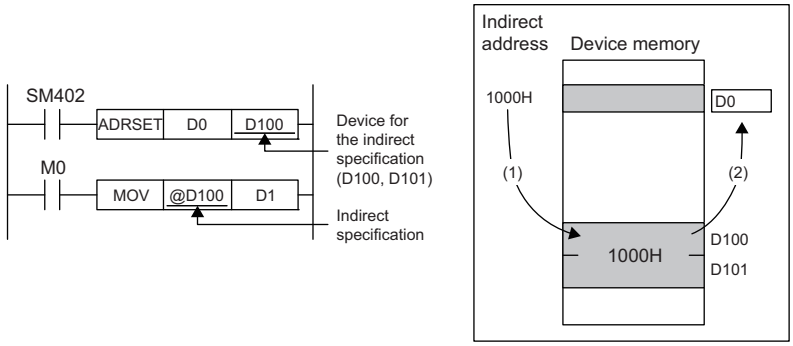


(1) When the value 65535 is stored in the index register (Z), D50000(-1) to D49999 are accessed because the value is turned into -1 when an index modification is performed.

(2) When a value larger than or equal to 32768 is used for an index modification, the value must be stored in the long index register (LZ). In doing so, the value 65535 is used as such for an index modification using the long index register (LZ) and D50000 (65535) to D115535 become accessible.

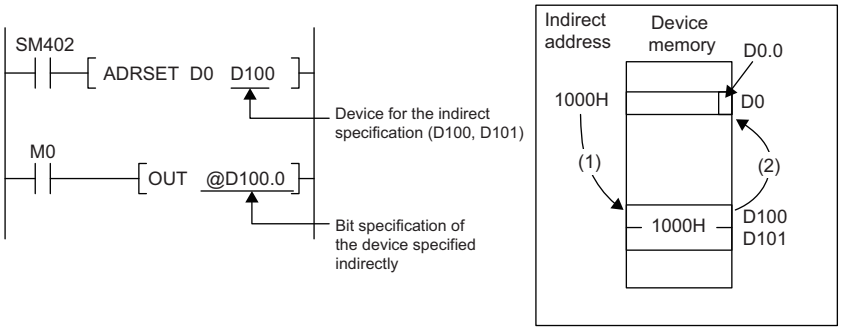
35.4 Indirect Specification

Specify the device using the indirect address of device. Store the indirect address of device to be specified into the device for indirect specification, and write as "@ + Device for indirect specification".



(1) The indirect address of D0 is read into D100, D101.
(2) The indirect address is used to indirectly specify D0.

Besides, specifying a bit of a word device allows the indirect specification for the instruction that specifies bits.



(1) The indirect address of D0 is read into D100, D101.
(2) The indirect address is used to output at the 0th bit of D0 which was indirectly specified using the indirect address.

The indirect specification can be used in the device/label memory or refresh memory.

Indirect address of device

To specify, use the 32-bit data, and to hold the value, use the device of two words. The indirect address of the device can be obtained with the ADRSET instruction. The ADRSET instruction specifies the indirect address of the device using instructions that handle 32-bit data. For ADRSET instruction, refer to the following.

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

Point

When the block or the file of the file register is switched through the RSET or QDRSET instruction, the indirect address refers to the one of the block or the file before they are switched. To allow the indirect address in the device for indirect specification to specify the block or file after the file register is switched, specify the ADRSET instruction to obtain the indirect address again after block or file are switched.

Devices that can allow indirect specification

This section lists devices that can allow indirect specification.

Type	Device*3
Devices that can allow indirect specification where @ is added*1	T, ST, C, D, W, SW, FD, SD, Un\G, Jn\W, Jn\SW, U3En\G, U3En\HG, R, ZR, RD
Device that can acquire the indirect address through the ADRSET instruction*2	X, Y, M, L, B, F, SB, T, ST, C, D, W, SW, FX, FY, FD, SM, SD, R, ZR, RD

*1 Also can be used for the local device. (e.g.: @#D0)
*2 The indirect address of device can be obtained for the local device as well. (e.g.: ADRSET #D0 D100)
*3 Devices that cannot be used as operands of instructions cannot be used even when they are indirectly specified.

PART 4 WHEN BUILDING A REDUNDANT SYSTEM

This part consists of the following chapters. Please read these chapters when building the redundant system. Since information same as that of the Process CPU is not described in these chapters, refer to Part 1 to Part 3.

36 BASIC CONCEPT

37 PROCEDURE FOR STARTING UP A REDUNDANT SYSTEM

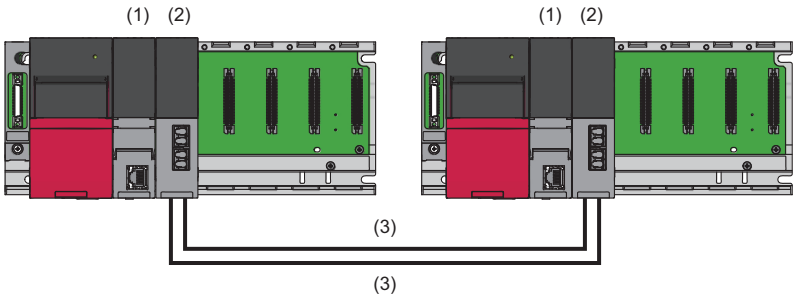
38 FUNCTIONS

39 PRECAUTIONS ON PROGRAMMING

40 MAINTENANCE AND INSPECTION FOR A REDUNDANT SYSTEM

36 BASIC CONCEPT

This system consists of two basic systems that have a CPU module, a power supply module, a network module, or other modules for each of them. Even if an error occurs in one system, control is continued with the other system.
A redundant configuration of the systems of main base units is available when redundant function modules are used and Process CPUs are operated in the redundant mode. To build a redundant system, build the same system with modules on two main base units, and connect the redundant function modules of each system by using tracking cables. Connect the redundant function modules with two tracking cables to configure a redundant system of tracking cables.



(1) Process CPU
(2) Redundant function module
(3) Tracking cable

36.1 System

System A and system B

In a redundant system, one system is specified as a system A and the other is specified as a system B to distinguish between two systems connected with tracking cables. Set the system A or B with the engineering tool. (☞ Page 383 System A/B Settings)

Point

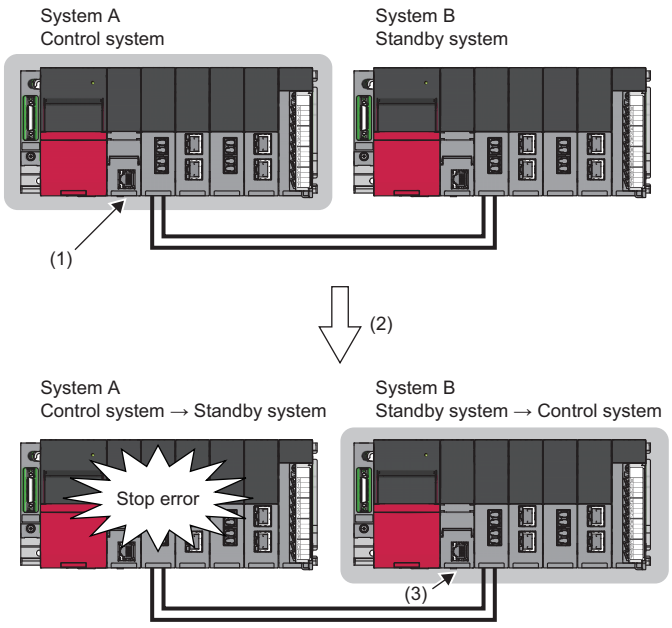
When one system is set to system A, the other system is automatically set to system B, and vice versa.

Control system and standby system

In a redundant system, the CPU module in one system executes programs to perform controls. The other system is in the standby state and does not perform controls. The system that performs controls is called a control system and the system in the standby state is called a standby system.
A control system or standby system is determined when both the systems are started and ready for tracking communications. (☞ Page 364 Determination of Control System/Standby System)

36.2 System Switching Between the Control System and Standby System

In a redundant system, data link is performed between the redundant function modules connected with tracking cables and data required for operation is transferred (tracking transfer) at every scan from the control system to the standby system. If an error occurs in the control system, the standby system will function as the new control system and continue the control using the data that the system has received.



(1) The control system is operating without errors. (Data is transferred to the standby system from the control system at every scan.)
(2) A stop error occurs on the control system.
(3) The standby system now functions as the new control system and continues the control using the data that the system has received.

For details on the system switching, refer to the following.
☞ Page 392 System Switching

36.3 System Consistency Check

In a redundant system, whether both systems have the same configuration is checked to switch the system and continue the operation without causing a system failure. (☞ Page 428 System Consistency Check)

36.4 Operation Mode of the Process CPU

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

Process mode

Use this mode when a redundant system is not built. When "Mode" has been set to "Process" at project creation with the engineering tool, the Process CPU is started up in the process mode.

Redundant mode

Use this mode when a redundant system is built. When "Mode" has been set to "Redundant" at project creation with the engineering tool, the Process CPU is started up in the redundant mode. (☞ Page 382 Creating a Project)

Point

Set the redundant mode for both systems in a redundant system.

36.5 Operation Modes of the Redundant System

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

Operation mode	Description
Backup mode	A mode used to normally operate the redundant system. When an error or failure has occurred in the control system, the standby system is switched to the control system to continue the operation. Even if a system failure has occurred in the control system, the standby system of the redundant system continues the operation because data is transferred from the control system to the standby system at every scan.
Separate mode	A mode used to maintain the redundant system without stopping controls. In separate mode, the control system CPU module and the standby system CPU module can execute different programs. The program of the standby system can be modified and the operation of the program can be checked with the data tracked from the control system while the system is being controlled in the control system.

A redundant system is started in the backup mode. To switch the operation mode, use the engineering tool. (☞ Page 389 Operation Mode Change)

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

Item	Backup mode	Separate mode
Program operation	The control system executes programs. The standby system CPU module executes programs according to "Both Systems Program Executions Setting" in "Program Setting" of "CPU Parameter". With the default setting, the standby system does not execute programs. To switch the mode to the backup mode, refer to the following. ☞ Page 390 Mode switching to the backup mode	Both of the control system and the standby system execute programs. To switch the mode to the separate mode, refer to the following. ☞ Page 389 Mode switching to the separate mode
System switching	Automatic system switching and manual system switching are supported.	Only manual system switching is supported.
Tracking transfer	Data is transferred according to "Tracking Setting" of "CPU Parameter".	Data is transferred according to "Tracking Setting" of "CPU Parameter". However, the following data causes tracking transfer to stop. • Local devices, local labels • Special relay, special register • PID control instruction information • Signal flow memory
Memory copy from control system to standby system	The following copy operations can be performed. • Automatic memory copy • Memory copy with the engineering tool • Memory copy with the special relay and special register	The following copy operations can be performed. • Memory copy with the engineering tool • Memory copy with the special relay and special register
System consistency check	The system consistency check is performed.	The system consistency check is not performed.
Program execution type	■When the operation mode is changed to the backup mode The control system and the standby system take over their execution types before the operation mode change.	■When the operation mode is changed to the separate mode The control system and the standby system operate differently. • The control system takes over its execution type before the operation mode change. • The standby system operates in the execution type specified by the CPU parameter.
Interrupt program execution enable/disable status	■When the operation mode is changed to the backup mode The control system and the standby system take over their execution enable status before the operation mode change.	■When the operation mode is changed to the separate mode The control system and the standby system operate differently. • The control system takes over its execution enable status before the operation mode change. • The standby system disables execution of interrupt programs.
I/O refresh	Whether or not to refresh the output (Y) of the standby system depends on "Standby System Output Setting" of "CPU Parameter". With the default setting, the output (Y) of the standby system is not refreshed.	Even though "Standby System Output Setting" has been set to "Disable" in "CPU Parameter", the output (Y) of the standby system is enabled.
Link refresh	Data is not transferred from devices of the CPU module in the standby system to link devices.	Data is transferred from devices of the CPU module in the standby system to only the link special relay (SB) and link special register (SW). (Except for CC-Link)
Writing data to the programmable controller	Data can be written to both systems. Or, data can be written to only a system specified in the connection destination setting.	Data is written to only a system specified in the connection destination setting.
Remote operation function	Remote operations can be performed on both systems.	Remote operation with the engineering tool can be performed only on a system specified in the connection destination setting.
Time synchronization	Time of the standby system is synchronized with that of the control system.	Time of both systems is not synchronized.

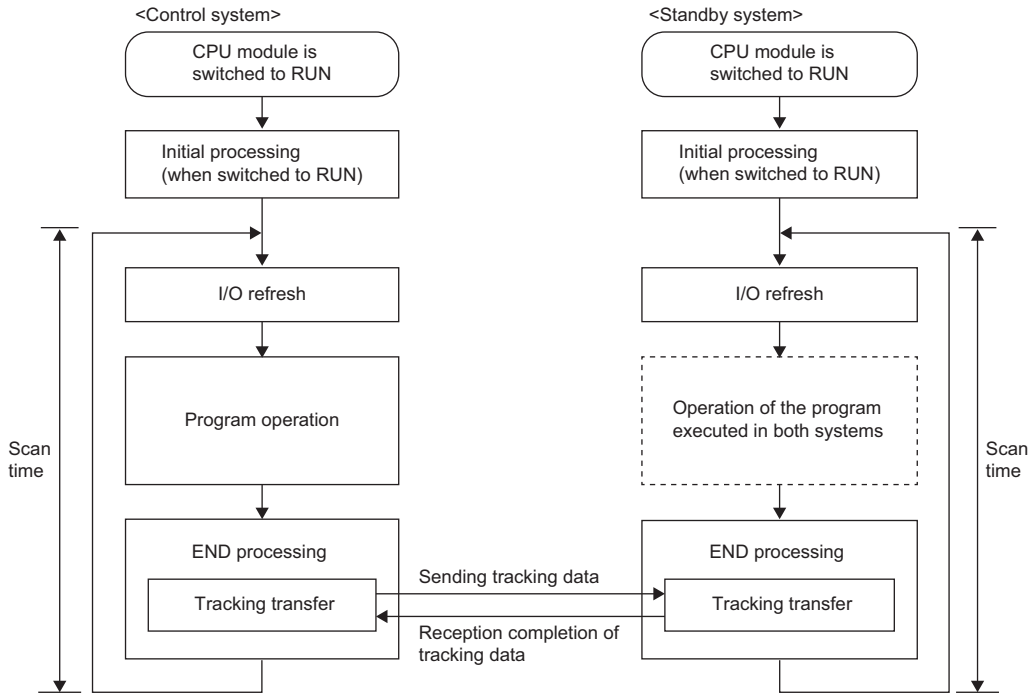
36.6 Scan Configuration

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

In a redundant system, tracking is performed during the END processing. (☞ Page 405 Tracking Transfer)

For details on other than tracking transfer, refer to Chapter 1. (☞ Page 32 Scan Configuration)

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



Both systems perform different processing depending on the set operation mode.

○: Performed, ×: Not performed

Processing	Backup mode		Separate mode	
	Control system	Standby system	Control system	Standby system
Initial processing/initial processing (when switched to RUN)	○	○	○*3	○*3
I/O refresh	○	○*1	○	○
Program operation	○	×*2	○	○
Tracking transfer	○	○	○	○
END processing	○	○	○	○

*1 Only the input refresh is performed. Whether or not to perform the output refresh depends on "Standby System Output Setting" of "CPU Parameter". (☞ Page 439 Redundant System Operation Setting)

*2 The program operation is not performed with the default setting. Programs are executed according to "Both Systems Program Executions Setting" in "Program Setting" of "CPU Parameter". (☞ Page 433 Program Execution in Both Systems)

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

36.7 Determination of Control System/Standby System

This section describes how to determine which system is the control system and the other is the standby system.

When starting up both systems simultaneously

The following describes the method of determining the system types of when both systems are started up simultaneously.

How to determine the system types

A control system or standby system is determined when both the systems are powered off and on or the CPU module is reset and then ready for tracking communications.

■When both systems are started up simultaneously

When both systems are started up simultaneously, the system A operates as the control system and the system B as the standby system.

"Simultaneously" here means that one CPU module is started up within three seconds after the other CPU module is started.

■When one of the system A and system B is started up first

When one of the systems is started up first, the system will wait for the start-up of the other system in three seconds. (☞ Page 366 The system waiting for the start-up of the other system)

When communications for both systems are established, system A operates as the control system and system B as the system B.

Check method

Check the LED of each redundant function module to check the status of the control system/standby system.

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

Point

Users can check the status of the control system/standby system with the engineering tool. (GX Works3 Operating Manual)

- System monitor
- Monitor status bar

Precautions

■If the system A/B settings have not been set for both systems or the same system setting has been set for both systems

If the system A/B settings have not been set for both systems or the same system setting has been set for both systems, a stop error occurs. To determine the control system and standby system, properly set the system A/B settings for both systems.

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

If both systems are restarted simultaneously in a system where the system A operates as the standby system due to a stop error and the system B normally operates as the control system, a stop error may occur on both systems.

If the stop error cause of the system A is a program error, a stop error occurs on the system A again after the systems are restarted. Even if the system B normally operates as the control system before the restart, a stop error occurs on the system B if a mismatch between files is detected in the system consistency check after the restart. As a result, a stop error occurs on both systems.

In this case, eliminate the cause of the error that has occurred on the system A and reset and restore both systems.

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

Do not power off the CPU module in the other system.

The system may start up without checking the system consistency even when the conditions between the both systems do not match. (☞ Page 429 Execution timing)

■When the configuration between the both systems do not match

When the configuration (such as insertion of an SD memory card) between the both systems do not match, it may take time to start communications between the both systems. Check that the both system configurations match in the process of starting up both systems at a time, and then start up a redundant system.

■Using an SD memory card

Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card. A stop error may occur on both systems and the systems may not be started as a redundant system.

If the CPU module is powered off or reset or an SD memory card is removed during an access to the SD memory card, the data in the SD memory card may be corrupted. In this case, diagnostics of the SD memory card (such as a file system check or restore processing) is performed when the CPU module is powered off and on or is reset.

During diagnostics of the SD memory card, tracking communications are disabled. If diagnostics of the SD memory card on the other system are not completed in the timeout period set in "Other system Start-up Timeout Setting", a stop error occurs on the own system. If diagnostics of the SD memory card takes time on the other system, a stop error occurs on the other system as well. In this case, a stop error occurs on both systems and the systems cannot be started as a redundant system. Reset and restore both systems.

Point

Even though diagnostics of the SD memory card takes time on the other system, only the own system can be started. (☞ Page 367 When one system is started automatically even though a tracking communication error has occurred)

When starting up one system first

One of the system A and system B can be started up first.

How to start one system first

Perform one of the following operations to the CPU module that is waiting for the start-up of the other system. The system of the CPU module will be started as the control system.

■Online operation

Perform the following operation on the engineering tool.

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

Select "Forced Start of Control System while Waiting for Other System to Start" and click the [Execute] button.

■Switch operation

Set the RUN/STOP/RESET switch of the CPU module to the RUN, STOP, and then the RUN position again.

Set "Control System Start-up Setting (Switch Operation)" to "Enable" in the CPU parameter in advance. (☞ Page 439 Redundant System Operation Setting)

■Operation with the input (X)

Turn on the input (X) set with the parameters.

Set "Control System Start-up Setting (Input (X))" to "Enable" in the CPU parameter in advance. (☞ Page 439 Redundant System Operation Setting)

The system waiting for the start-up of the other system

When the Process CPU (redundant mode) is started up and tracking communications cannot be established with the other system, the CPU module will start waiting for the start-up of the other system in three seconds.
Both of the CTRL LED and SBY LED of the redundant function module turn off because the system of the CPU module is not the control system or standby system yet. The BACKUP LED flashes because a system switching disable cause has been generated.
After tracking communications are established with the other system, the control and standby systems are determined and the system switching disable cause is eliminated.

■Operation while the system is waiting for the start-up of the other system

Even though the RUN/STOP/RESET switch of a CPU module is set to RUN while the system is waiting for the start-up of the other system, the CPU module is in the STOP state because the system has not yet been set to control system or standby system. Thus, the CPU module does not execute programs.
The following table lists the refresh operations to be performed while the system is waiting for the start-up of the other system.

Type	Operation
I/O refresh	Only the input refresh is performed. The output refresh is not performed.
Network module link refresh	Data is transferred from the network module to the CPU module and from the CPU module to the network module (SB/SW). Data is not transferred from the CPU module to the network module (other than SB/SW).
Intelligent function module refresh	Data is transferred between the intelligent function module and the CPU module.

■Timeout of waiting for the start-up of the other system

Measuring the length of the waiting time for the start-up of the other system is started upon completion of the initial processing.
Set the waiting time for the start-up of the other system in "Other system Start-up Timeout Setting" of "CPU Parameter". (☞ Page 439 Redundant System Operation Setting)
When a timeout occurs, a stop error will occur. To prevent occurrence of a stop error, set "Other system Start-up Timeout Setting" of "CPU Parameter" to "Not Set".

Precautions

- To start up the system that is waiting for the start-up of the other system as the control system, check that the other system is not operating as the control system.
- When tracking communications are established by connecting the other system with a tracking cable or powering on or resetting the CPU module in the other system, the CPU module that has been waiting for the start-up of the other system enters the RUN state and executes programs. Thus, always check that program execution by the CPU module does not cause any problems before powering on the other system or connecting the other system with a tracking cable.
- When the CPU module that has been waiting for the start-up of the other system becomes ready for tracking communications and a stop error has occurred in the other system, a stop error occurs in that CPU module as well. In this case, take a corrective action against the error of the other system, and turn off and on or reset the CPU module in both systems.
- When the system waits for the start-up of the other system, tracking communication cannot be made because the other system is off or either of tracking cables has problems. Check that the other system is on or tracking cables have no problems. (☞ Page 603 When the L ERR LED turns on)
- Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card. (☞ Page 365 Using an SD memory card)

When one system is started automatically even though a tracking communication error has occurred

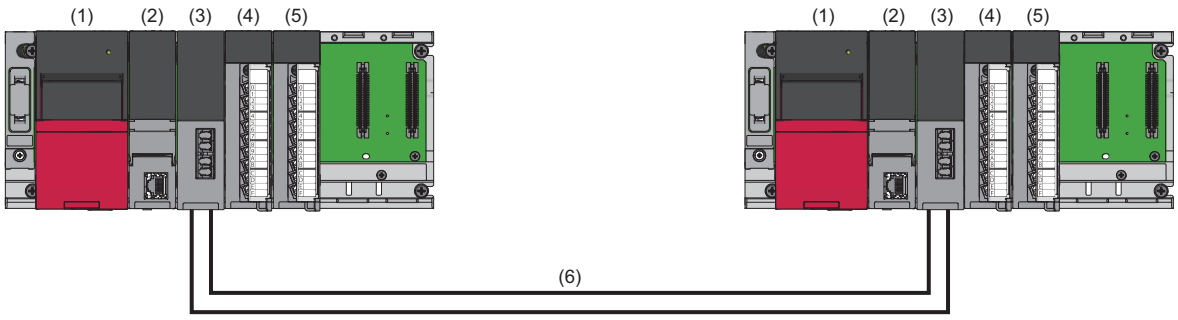
When the other system is powered off or a tracking cable has an error at a system start, the CPU module waits for the start-up of the other system. The following shows a program example to start up only one of the systems using external signals without waiting for the start-up of the other system. In the following example, the program prevents both systems from operating as control systems.

Restriction

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

System configuration

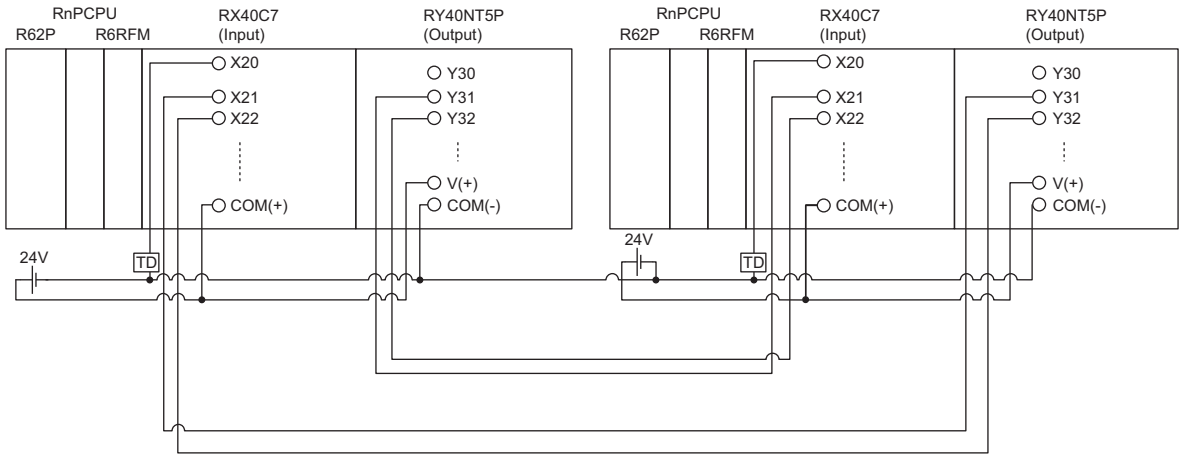
The following figure shows a configuration example of the system.



(1) Power supply module (R62P)
(2) CPU module (RnPCPU)
(3) Redundant function module (R6RFM)
(4) Input module (RX40C7)
(5) Output module (RY40NT5P)
(6) Tracking cable

Wiring example

The following figure shows a wiring example.



- TD is an on delay timer wired externally. Input the contact output of the on delay timer to X20.
- The 24V power supply is always input.

■I/O signals

The following table lists the details on the I/O signals.

Device No.	Signal name
X20	Control System Start-up Setting (Input (X)) With the on delay timer wired externally, X20 turns on and the CPU module starts as the control system after a certain time.
X21	Starting up (other system)
X22	Control system (other system)
Y31	Starting up (own system)
Y32	Control system (own system)

■Setting time of the external on delay timer

For the on delay timer for starting up the control system, set the setting time longer than the time duration of the start-up of both systems.*1

The following shows a reference setting time.

• Setting time of the external on delay timer = Start-up time of the CPU module + Time lag of power-on + α

Check the start-up time with the actual system and adjust the setting time (α) because the start-up time of the CPU module depends on the system configuration.

*1 If the setting time of the external on delay timer is shorter than the above, a stop error may occur on one system even though tracking cables are connected properly because turning on X20 (Control System Start-up Setting (Input (X))) starts this program example.

Parameter settings

The following shows parameter settings.

■System parameter

Set the system parameter according to the system configuration in "I/O Assignment Setting".

Slot	Module Name	Module Status Setting	Points	Start XY
Main				
CPU	R32PCPU(Host Station)			3E00
0(0-0)	R6RFM	No Setting	32 Points	0000
1(0-1)	RX40C7	No Setting	16 Points	0020
2(0-2)	RY40NT5P	No Setting	16 Points	0030

■CPU parameter (program settings)

Set this program example (INITIAL in this example) in "Program Setting" as follows.

Execute Order	Program Name	Execution Type		Refresh Group Setting	Device/File Use or not	Both Systems Program Executions Setting
		Type	Detailed Setting Information			
1	INITIAL	Initial		(Do not Set)	<Detailed Setting>	Control System Execution
2	MAIN	Scan		(Do not Set)	<Detailed Setting>	Control System Execution

• Set "INITIAL" for "Execution Type".
• Set "Control System Execution" in "Both Systems Program Executions Setting".

■CPU parameter (redundant settings)

Set the CPU parameter in "Redundant System Settings" as follows.

Item	Setting	
Redundant Behavior Setting		
Watching Standby System Setting	Enable	
Standby System Output Setting	Disable	
Backup Mode Setting	Check Operating Status Consistency	
Auto Memory Copy Setting	Disable	
Control/Standby System Start-up Setting		
Other system Start-up Timeout Setting	Not Set	(1)
Time-out Period	60 Second	
Control System Start-up Setting(Switch Operation)	Disable	
Control System Start-up Setting(Input(X))	Enable	(2)
Input(X)	X20	(3)
Tracking Setting		
Signal Flow Memory Tracking Setting	Transfer	
Tracking Device/Label Setting	Detailed setting	(4)
Tracking Block No.1 Auto Transfer Setting	Transfer Automatically	
Device/Label Detailed Setting	<Detailed Setting>	

(1) Set "Not Set" in "Other system Start-up Timeout Setting".
(2) Set "Enable" in "Control System Start-up Setting (Input (X))".
(3) Set "X20" in "Input (X)".
(4) Set "Detailed setting" in "Tracking Device/Label Setting".

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

Tracking Block No. 1

Total Global Device 36.2K Word

File Register Setting

File Name

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

(5)

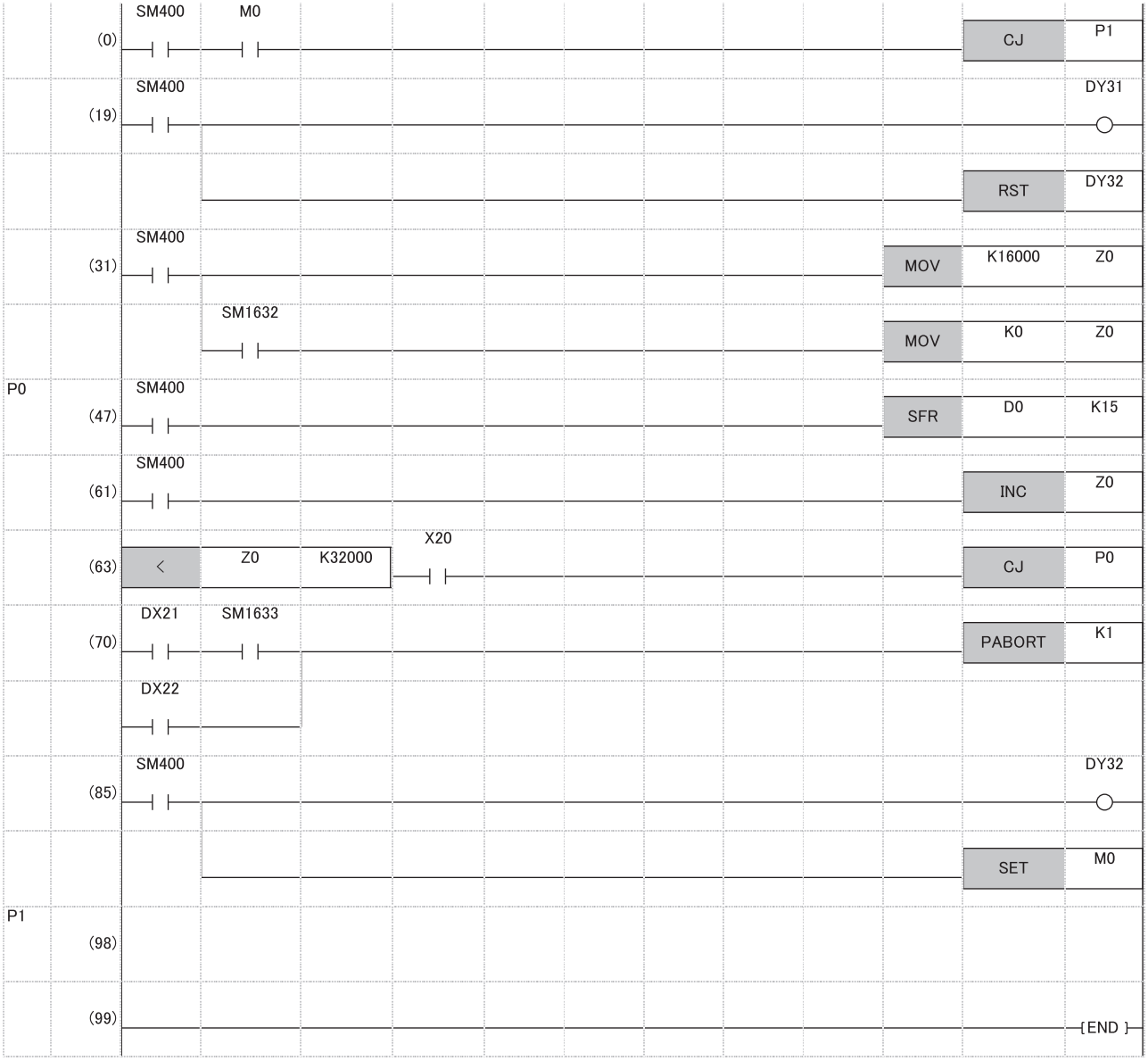
(5) Do not include X20 to X2F and Y30 to Y3F, which are used in the program example, in the tracking transfer range.

Point

To switch enabling and disabling "Control System Start-up Setting (Input (X))", set a switch for the input to X20 so that "Control System Start-up Setting (Input (X))" is enabled only when the switch is on.

Program example

The following shows a program example and the overview of the operation.



■Execution determination of start-up processing

(0) Set an interlock signal (M0) to prevent the above program from being executed if the operating status of the CPU module is changed from STOP to RUN during operation. Do not set the latch setting for the device and label used for the interlock signal.

■Signal initialization

(19)With the direct access output, the starting up (own system) (Y31) is turned on and the control system (own system) (Y32) is turned off.

■Standby time setting

(31)Set the standby time. For the system A (SM1632), set the standby time longer.

■Standby processing

(47) to (63)If the control system is started up with the on delay timer (X20), the system stands by temporarily. If both systems start up successfully, the system does not stand by.

■Condition judgment of the other system

(70)When the other system is operating and the own system is the system B (DX21 AND SM1633) or when the own system checks that the other system has already been started up as the control system (DX22), a stop error occurs on the own system.

■Output of the control system (own system)

(85)With the direct access output, the control system (own system) (Y32) is turned on and the other system is notified of the start-up of the own system as the control system. The interlock signal (M0) turns on as well.

Point

- If both systems start up successfully, a stop error does not occur because an initial execution type program is not executed on the other system.
- If a stop error has occurred in this program, the possible cause is a tracking cables failure. Check the connection of tracking cables and power off and on or reset the CPU module on the system in which the stop error has occurred.

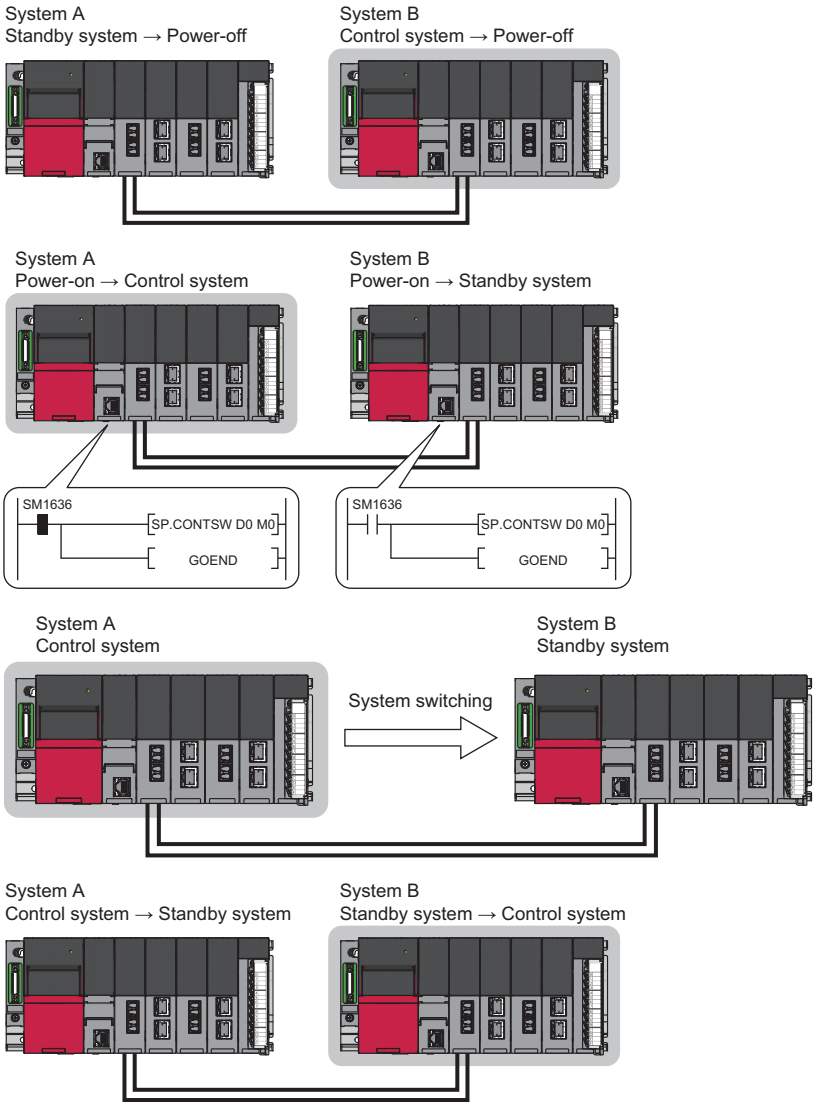
When starting up the previous control system as the control system

In a redundant system, the system A is always specified as the control system when both systems are started up simultaneously.
Even though both systems are temporarily powered off due to a power failure or other causes while the system B is operating as the control system, the system A is started up as the control system when both systems are powered on again.
To start up the system B that was the previous control system as the control system again, create a program that uses SM1636 (Previous control system identification flag).
For a system where a network module has been mounted, execute the SP.CONTSW instruction after a network module on the other system is started up.

Program example

Executing a system switching instruction when the CPU module in the system A is set to the RUN state for the first time switches the system B to the control system.
Turn on SM1646 (System switching by a user) in advance.

Operation example



1. Both systems are temporarily powered off due to a power failure or other causes while the system B is operating as the control system.

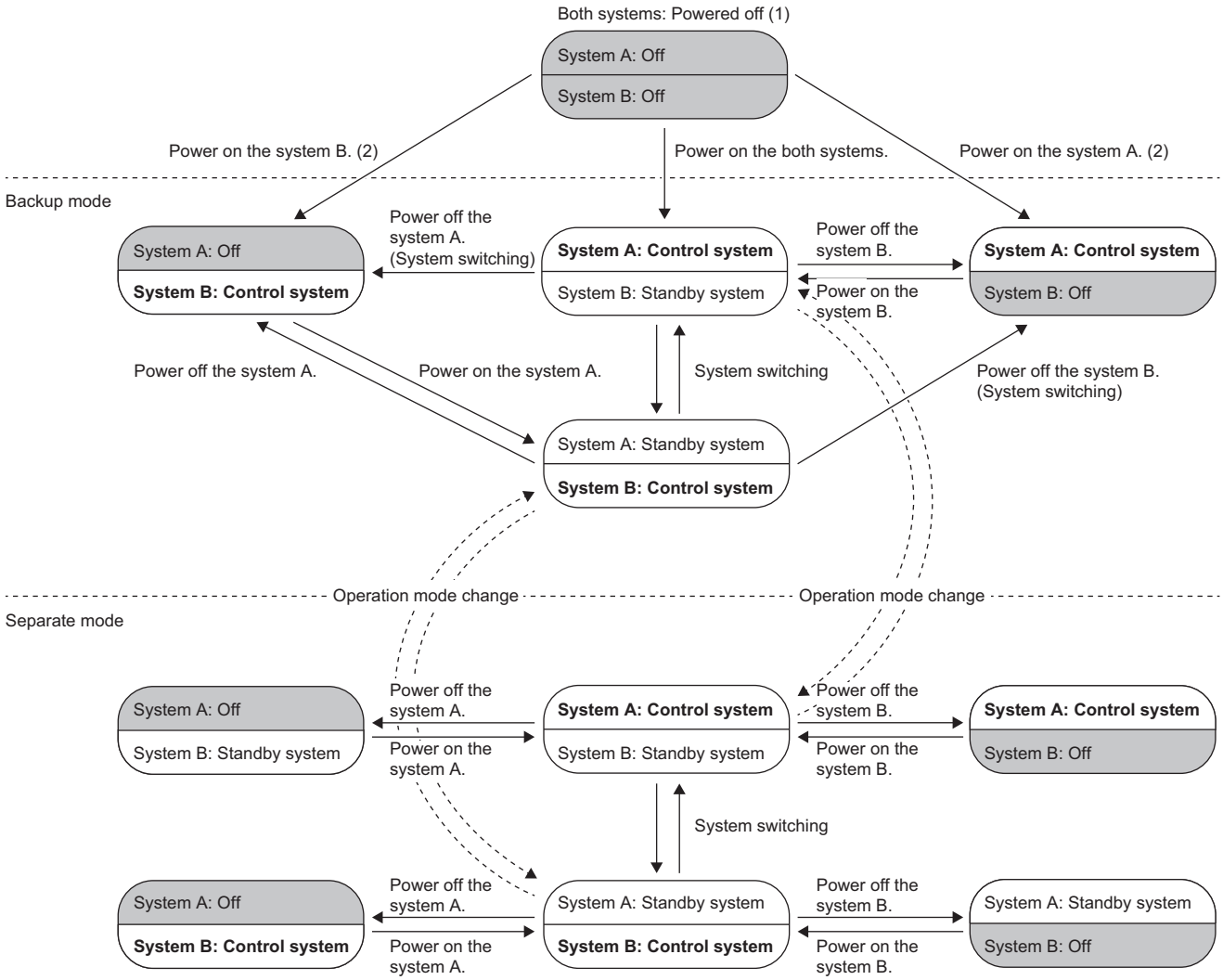
2. The system A is started as the control system when both systems are powered on. SM1636 turns on for only one scan after the CPU module in the system A is set to the RUN state.

3. The system switching is performed by the SP.CONTSW instruction.

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

36.8 State Transition of the Redundant System

The following figure shows the state transition due to the operation mode change and system switching for the redundant system after start-up.



(1) When both systems are powered off, both systems enter the power-off state regardless of the operation mode or system type.
(2) When a system that has been powered is started as the control system. (☞ Page 365 When starting up one system first)

37 PROCEDURE FOR STARTING UP A REDUNDANT SYSTEM

This chapter describes the procedures for starting up a redundant system starting from the start-up procedure of CPU modules to execution of programs.

37.1 Overview

The following table lists two procedures for starting up a redundant system.

Start-up procedure	Description
Starting up both systems at a time	Prepare both systems and follow this procedure to start up both systems as a redundant system when the systems are powered on.
Starting up the systems one by one	Follow this procedure to start up a control system first to perform controls, and then start up a standby system to build a redundant system. • Debugging before operation can be performed with only one system. • If one system has failed and the system is powered off before module replacement, powering on this system again starts up only this system.

Starting up both systems at a time

1. Installing batteries

Install a battery to the CPU module of each system. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

2. Mounting extended SRAM cassettes and inserting SD memory cards

Mount an extended SRAM cassette and insert an SD memory card to the CPU module of each system as needed. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card. (☞ Page 365 Using an SD memory card)

3. Mounting modules and wiring each device

Mount modules on the base unit of each system and wire each device. Use modules of the same model, and mount them on the same slots on each base unit. (📖 MELSEC iQ-R Module Configuration Manual, ☞ Page 380 Wiring)

4. Powering on the systems

Check each system for the following and power on the systems.

- Wiring to the power supply module is correct.
- The power supply voltage is within the range of specifications.
- The CPU module is in the STOP state.

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

- Power supply module: POWER LED
- CPU module: READY LED
- Redundant function module: RUN LED

In step 4, the ERROR LED of each CPU module flashes and the ERR LED of each redundant function module turns on. Proceed to the next step.

5. Connecting a personal computer and a CPU module

Start an engineering tool installed on a personal computer. (☞ Page 382 Creating a Project)

Connect the personal computer on which the engineering tool has been installed and a CPU module. (☞ Page 382 Connecting a Personal Computer and a CPU Module)

6. Initializing the CPU module

Initialize the CPU module with the engineering tool. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

After initialing the CPU module, connect the other CPU module and the personal computer. (☞ Page 382 Connecting a Personal Computer and a CPU Module)

Initialize the CPU module in the same way.

7. Setting parameters

Set system parameters, CPU parameters, and module parameters. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

- To use the functions that require an SD memory card, set memory card parameters.
- To mount an intelligent function module, set intelligent function module parameters.

Point

Users can set system parameters by reading the actual system configuration to the module configuration of the engineering tool.

8. Creating a program

Create a program with the engineering tool. After creating the program, convert the program and save the project. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

9. Writing the system A/B setting

Set the system A or B with the engineering tool. (☞ Page 383 System A/B Settings)

10. Writing data to the programmable controller

Write the set parameters and created programs to both systems with the engineering tool. (☞ Page 385 Writing Data to the Programmable Controller)

11. Resetting the CPU modules

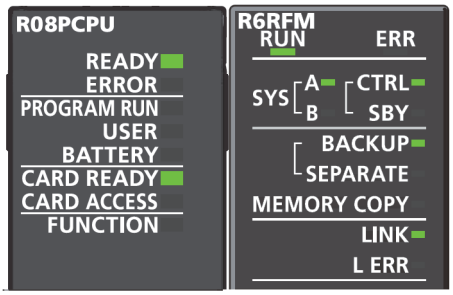
Restart both systems with either of the following methods.

- Powering on the systems
- Resetting the CPU modules

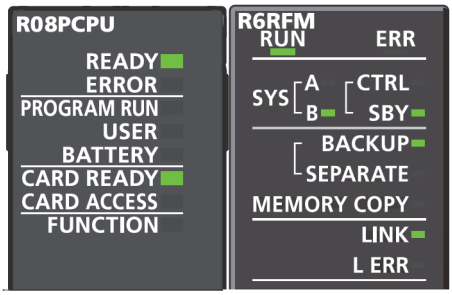
12. Checking the LEDs

Check that the LEDs of each CPU module and redundant function module are in the following states. The CARD READY LED status depends on whether an SD memory card has been installed to each CPU module or not.

• LEDs of System A



• LEDs of System B



When an error has occurred, the following LEDs are on. Check details of the error with the engineering tool and eliminate the error cause.

- CPU module: ERROR LED (📖 MELSEC iQ-R CPU Module User's Manual (Startup))
- Redundant function module: ERR LED, L ERR LED (☞ Page 603 When an error has occurred in a redundant function module)

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

☞ Page 379 Precautions on starting up the system when the data logging function is used

13. Running a program

Power off both systems. Set the RUN/STOP/RESET switch of the CPU module of each system to the RUN position and power on both systems.

Check that the PROGRAM RUN LED of the CPU module of the control system turns on.

Point

Individually setting the RUN/STOP/RESET switch of each CPU module to the RUN position with the CPU module powered on causes a continuation error due to the operating status mismatch and the error is detected in the standby system. Therefore, to start up both systems simultaneously, performing step 13 is recommended.

14. Monitoring the program

Check that the program is normally running on the engineering tool. (☞ Page 386 Monitoring the Program)

Starting up the systems one by one

To debug a program with only one system before operation, start up the control system.

To start up the control system for a purpose other than debugging a program, start up the standby system to build a redundant system.

Starting up the control system

Start up only the control system.

1. Installing a battery

Install a battery to the CPU module. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

2. Mounting an extended SRAM cassette and inserting an SD memory card

Mount an extended SRAM cassette and insert an SD memory card to the CPU module as needed. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card. (☞ Page 365 Using an SD memory card)

3. Mounting modules and wiring each device

Mount modules on the base unit and wire each device. (📖 MELSEC iQ-R Module Configuration Manual, ☞ Page 381 Wiring to the power supply modules in a redundant system)

4. Powering on the system

Check the system for the following and power on the system.

- Wiring to the power supply module is correct.
- The power supply voltage is within the range of specifications.
- The CPU module is in the STOP state.

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

- Power supply module: POWER LED
- CPU module: READY LED
- Redundant function module: RUN LED

In step 4, the ERROR LED of the CPU module flashes and the ERR LED of the redundant function module turns on. Proceed to the next step.

5. Connecting a personal computer and the CPU module

Start an engineering tool installed on a personal computer. (☞ Page 382 Creating a Project)

Connect the personal computer on which the engineering tool has been installed and the CPU module. (☞ Page 382 Connecting a Personal Computer and a CPU Module)

6. Initializing the CPU module

Initialize the CPU module with the engineering tool. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

7. Setting parameters

Set system parameters, CPU parameters, and module parameters. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

- To use the functions that require an SD memory card, set memory card parameters.
- To mount an intelligent function module, set intelligent function module parameters.

Point

Users can set system parameters by reading the actual system configuration to the module configuration of the engineering tool.

8. Creating a program

Create a program with the engineering tool. After creating the program, convert the program and save the project. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

9. Writing the system A/B setting

Set the system A or B with the engineering tool. (☞ Page 383 System A/B Settings)

10. Writing data to the programmable controller

Write the set parameters and created programs to the programmable controller with the engineering tool. (☞ Page 385 Writing Data to the Programmable Controller)

11. Resetting the CPU module

Restart the control system with either of the following methods.

- Powering on the system
- Resetting the CPU module

12. Starting up the system

Perform the following operation on the engineering tool to start up the system as a control system within the time set in "Other system Start-up Timeout Setting".

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

Select "Forced Start of Control System while Waiting for Other System to Start" and click the [Execute] button. (When "Other system Start-up Timeout Setting" of "CPU Parameter" has not been changed, a stop error occurs in 60 seconds.)

Check that the CTRL LED of the redundant function module turns on.

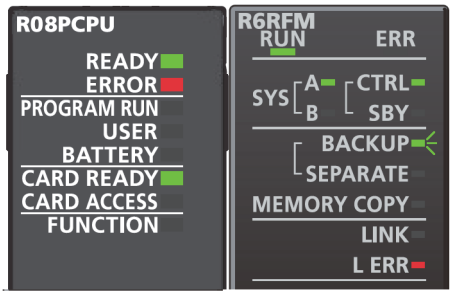
Point

When the CPU parameters have been set, users can start up only one system with the following operations. (☞ Page 439 Redundant System Operation Setting)

- Switch operation: Set the RUN/STOP/RESET switch of the CPU module to the RUN, STOP, and then the RUN position again.
- Input (X) operation: Turn on the input (X) set with the parameters.

13. Checking the LEDs

Check that the LEDs of the CPU module and redundant function module are in the following states. The following figure shows the LED status when own system is set as the system A in the system settings. The CARD READY LED status depends on whether an SD memory card has been installed to the CPU module or not.



Point

- Since only the control system has been started up, a continuation error occurs. Thus, the ERROR LED of the CPU module and the L ERR LED of the redundant function module turn on. The BACKUP LED flashes because a system switching failure cause has been generated.
- When "Watching Standby System Setting" of "CPU Parameter" is set to "Disable", the ERROR LED of the CPU module will turn off.

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

☞ Page 379 Precautions on starting up the system when the data logging function is used

14. Running a program

Set the CPU module to the RUN state and check that the PROGRAM RUN LED of the CPU module turns on.

15. Monitoring the program

Check that the program is normally running on the engineering tool. (☞ Page 386 Monitoring the Program)

Starting up the standby system

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

Follow step 1 to 6 in the start-up procedure for the control system to start up the standby system. (☞ Page 376 Starting up the control system)

Point

Check if the model of the module which is mounted on the base unit and insertion of the extended SRAM cassette or SD memory card are the same between the control system and standby system before the starting up.

1. Powering off the system

Power off the standby system.

2. Wiring the redundant function modules

Connect the redundant function modules of both systems with tracking cables. (☞ Page 380 Wiring the redundant function modules)

3. Starting up the system

Power on the standby system.

Connect the personal computer to the control system and copy the memory of the control system to the standby system with the engineering tool by following the procedure described below. (☞ Page 420 Memory Copy from Control System to Standby System)

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

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

When copying the memory is completed, the MEMORY COPY LED of the redundant function module of the standby system turns on.

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

Point

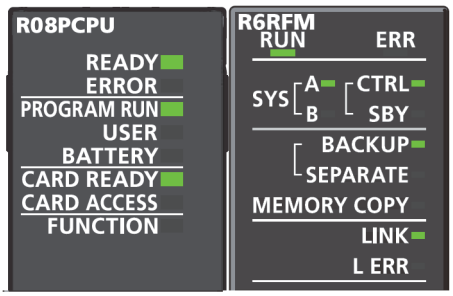
If "Auto Memory Copy Setting" has been set to "Enable" in the CPU parameter settings in advance, the system can be started up only with the following operation. (☞ Page 439 Redundant System Operation Setting)

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

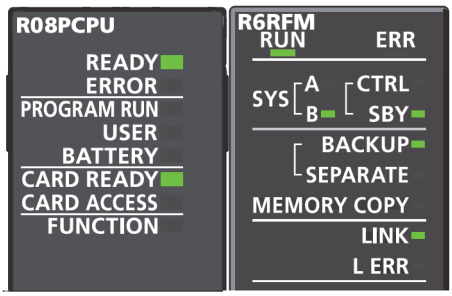
4. Checking the LEDs

Check that the LEDs of the CPU module and redundant function module are in the following states. The following figure shows the LED status when the control system has been started up as the system A. The CARD READY LED status depends on whether an SD memory card has been installed to the CPU module or not.

- LEDs of System A



- LEDs of System B



5. Clearing errors

When the ERROR LED of the CPU module of the control system is on, clear the error with the engineering tool. (☞ Page 233 Error Clear)

Point

When "Watching Standby System Setting" of the CPU parameter has been set to "Disable" in the procedure for starting up the control system, the ERROR LED may be off. (☞ Page 439 Redundant System Operation Setting)

6. Monitoring the program

Check that the program is normally running on the engineering tool. (☞ Page 386 Monitoring the Program)

Precautions on starting up the system when the data logging function is used

If the systems are switched after data logging has failed (processing overflow) due to frequent and continuous data logging collection, data logging continues frequently and continuously in the new control system, and thus the scan time of the new control system increases considerably. Therefore, a WDT error may occur in the new control system and both systems may stop. To avoid the above problem, check the following at system start-up.

- After the redundant system is started, the control system does not fail in data logging collection (processing overflow).*1
- The new control system does not fail in data logging collection (processing overflow) and the CPU module operates without errors after system switching.*1

*1 Whether a processing overflow has occurred or not can be checked with the special register (Number of processing overflow occurrences) corresponding to the data logging No. (☞ Page 635 List of Special Register Areas)

To prevent both systems from stopping, review the following settings.

- Collection interval and data to be collected in the data logging setting (☞ Page 746 Collection, ☞ Page 747 Data)
- "Scan Time Monitoring Time (WDT) Setting" of the CPU parameter (☞ Page 89 Scan Time Monitoring Time Setting)

Point

When using the data logging function, note that the number of writings to the SD memory card is limited.

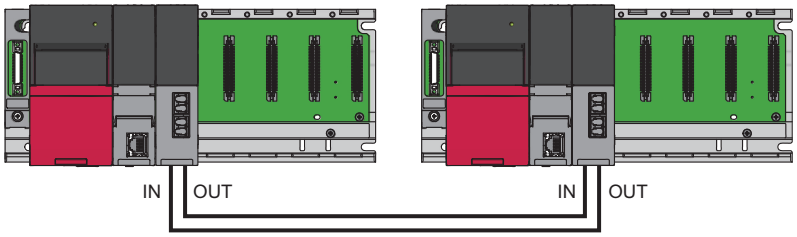
37.2 Wiring

Wiring the redundant function modules

This section describes how to wire the redundant function modules.

Wiring method

Connect the tracking cables from the OUT connector of a redundant function module to the IN connector of the other redundant function module.



Point

For the specifications of the tracking cables connected to redundant function modules, refer to the following.

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

Connecting/disconnecting tracking cables

■Connection procedure

1. Pay attention to the direction of a tracking cable connector and insert the cable to a redundant function module until it clicks. A redundant function module has one IN connector and one OUT connector. Connect the IN connector of the system A and the OUT connector of the system B with a tracking cable and connect the OUT connector of the system A and the IN connector of the system B.
2. Pull each cable lightly and check that it has been connected securely.

■Disconnection procedure

1. Disconnect the tracking cables while pressing the connector hook.

Precautions

- There are restrictions on the cable bending radius. For details, refer to the specifications of the tracking cables used.
- Place the cables in a duct or clamp them. If not, a dangling cable may swing or inadvertently be pulled, resulting in damage to the module or cables or malfunction due to poor contact.
- When connecting tracking cables, pay attention not to touch optical fiber cores of the connectors on the cables and module sides and prevent dirt and dust from adhering to them. If oil on hands, dirt, and dust adhere to the optical fiber cores, the transmission loss increases and tracking may not work properly.
- Hold the connector of the tracking cables to connect or disconnect the cable. Pulling the cable connected to the module may result in malfunction or damage to the module or cable or malfunction due to poor contact.

Wiring to the power supply modules in a redundant system

This section describes the wiring to the power supply modules.

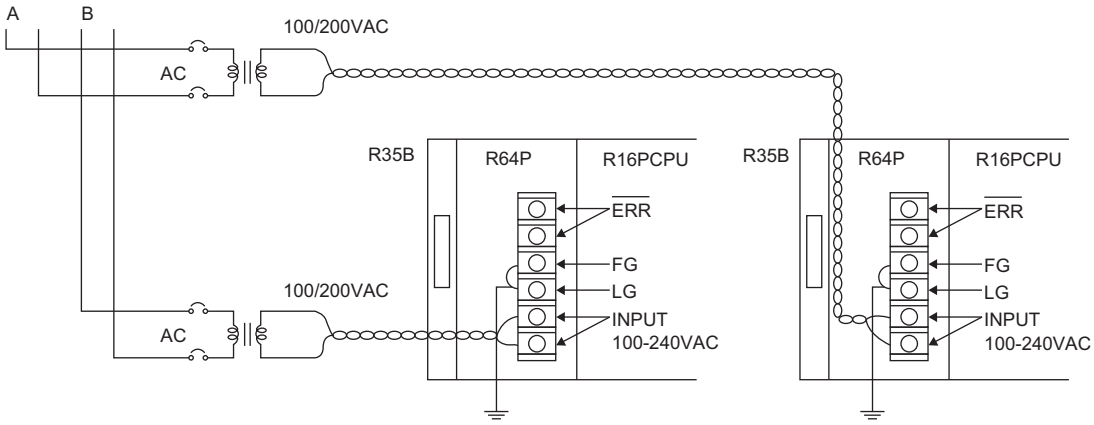
The terminal block of each power supply module has a screw size of M4. Wire cables to the terminal block with the applicable solderless terminal RAV1.25-4 or RAV2-4.

Separately supply power to the system A and B.

Wiring example

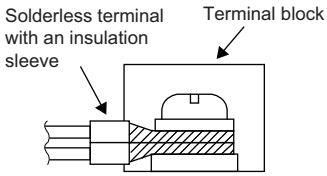
The following figure shows a wiring example of power cables to each main base unit and ground cables.

For wiring examples for each power supply module, refer to the manual included with the power supply module. (Before Using the Product)



Point

- 100VAC, 200VAC and 24VDC wires must be twisted starting from the terminal connected, and connect modules at the shortest distance. Also, use the thickest wire (maximum 2mm²) to reduce the voltage drop.
- For the wiring to a terminal block, use a solderless terminal.
- To prevent the short-circuit because of loosening screws, use the solderless terminal with an insulation sleeve of 0.8mm or less. Note that up to two solderless terminals can be connected per terminal block.

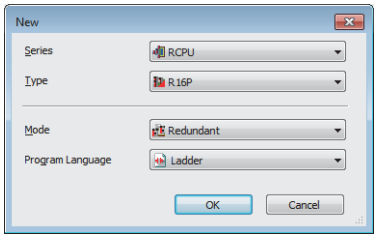


- Ground the LG and FG terminals after short-circuiting them. Failure to do so may be susceptible to the noise. The LG terminal has a half potential of the input voltage.
- When two redundant power supply modules operate in parallel as a redundant power supply system, it is recommended to connect the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

37.3 Creating a Project

Start the engineering tool and create a project.

[Project] ⇨ [New]



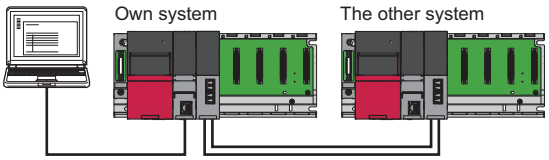
1. Select the Process CPU to be used for "Type". Select "Redundant" for "Mode". Select a programming language to be used for "Programming Language" and click the [OK] button.

37.4 Connecting a Personal Computer and a CPU Module

Connect the personal computer on which the engineering tool has been installed and a CPU module.

Connection procedure

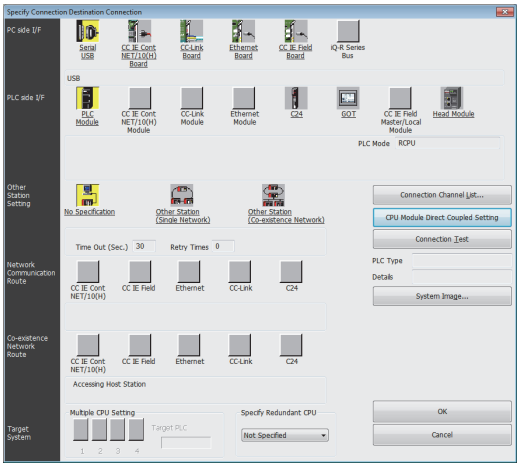
The following describes the procedure for directly connecting a CPU module and the personal computer.



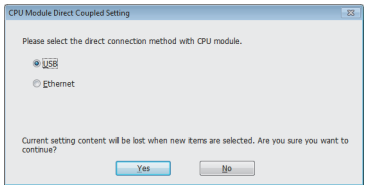
1. Connect a CPU module to the personal computer with a USB cable or an Ethernet cable. The CPU module connected with a USB cable or an Ethernet cable becomes own system.

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

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



4. Select a method of connection with the CPU module and click the [Yes] button.



5. Set "Specify Redundant CPU" to "Not Specified" on the "Specify Connection Destination Connection" window and click the [Connection Test] button to check that the personal computer has been connected with the CPU module.

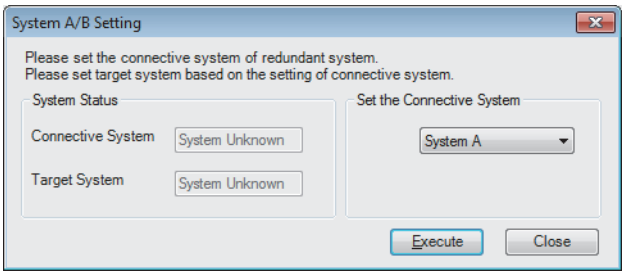
When connecting the personal computer and the CPU module with a USB cable for the first time, install a USB driver. (GX Works3 Operating Manual)

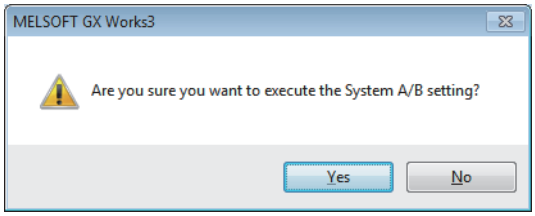
37.5 System A/B Settings

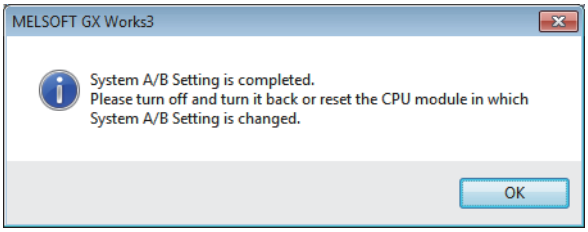
Set the system A or B with the engineering tool and write the system settings to the programmable controller.

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

Configuration procedure







1. Set the CPU module to the STOP state.
2. Select the setting to be written to own on the "System A/B Setting" window.
3. Click the [Execute] button.
4. Click the [Yes] button on the window on the left. Check that the SYS A LED or SYS B LED of the redundant function module flashes in accordance with the setting.
5. To change the system A/B settings, power off and on the system or reset the CPU module, and then click the [OK] button. Check that the SYS A LED or SYS B LED of the redundant function module turns on in accordance with the new setting.*1

*1 If the system A/B settings are set to the Process CPU that is not running in the redundant mode, the SYS A LED or SYS B LED of the redundant function module will turn off. Follow the procedure below (writing data to the programmable controller).

Point

Besides the method that uses the engineering tool for setting system A and B, system A and B can be automatically set by using the systems.

When one system is set to system A and gets ready for tracking communications, the other system is automatically set to system B, and vice versa.

For the system to which system A or system B has been assigned automatically, the system setting will be overwritten automatically. In that case, the system setting change can be checked in the event history. (Event code: 00700)

- If a system that has no system A/B setting yet is connected to the system specified as system A, the system that has no system A/B setting will be automatically set to system B. (This event is not recorded in the event history.)
- If a system specified as system A is connected to another system specified as system A, the system will be automatically set to system B. (This event is recorded in the event history.)

Checking method

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

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

Point

Users can check the system A/B setting with the engineering tool. (GX Works3 Operating Manual)

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

Precautions

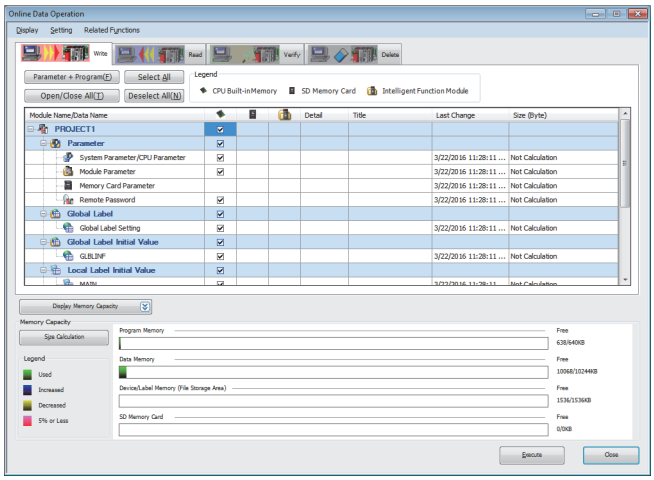
- Do not power off or reset the CPU module during system A/B setting. Doing so may not properly reflect the system A/B setting. If the system A/B setting has not been properly reflected, set the system again.
- For the system A/B setting, use the CPU module that has been connected with a USB cable or an Ethernet cable. Users cannot set a system if they have changed the connection destination on the engineering tool.
- The set system A/B information cannot be deleted. Only changing the set system A/B information is supported.
- When starting up both systems, if both systems have no system A/B setting yet or the same system A/B setting, a stop error will occur when tracking communications are established. Set a different system A/B setting for each system.
- If the system that has no system A/B setting yet is started first, a stop error will occur when tracking communications are established. Set a different system A/B setting for each system.
- Be careful not to connect between running system As or between system Bs with tracking cables. Otherwise, a continuation error will occur.

37.6 Writing Data to the Programmable Controller

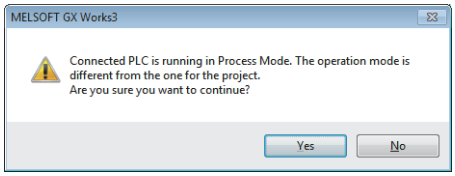
Write the set parameters and created programs to the programmable controller.

[Online] ⇨ [Write to PLC]

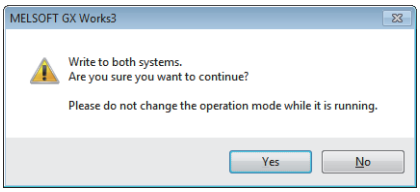
Operating procedure



1. Select system parameters, CPU parameters, module parameters, and program files on the "Online Data Operation" window. When FBs are used, select the corresponding FB/FUN files.
2. Click the [Execute] button.



3. When the Process CPU is not in the redundant mode, the window shown on the left appears. Click the [Yes] button.



4. Click the [Yes] button.*1
5. When writing the data to the programmable controller is completed, click the [Close] button.

*1 When one of both systems is started up first, the window asking for writing the data to only the connected system appears.

Point

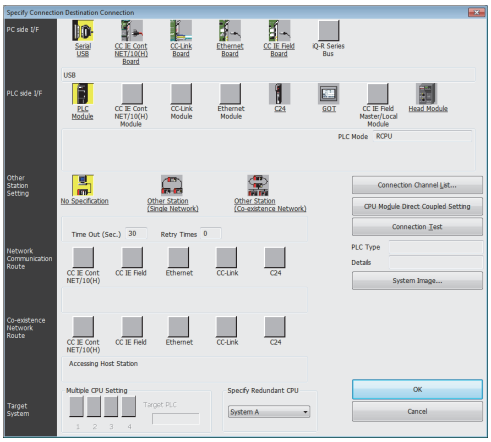
- The same data can be written to both systems when the operation mode is the backup mode. Always write the same data to both systems to prevent the occurrence of a file mismatch in the system consistency check. When writing programs, FB files, or global label setting after performing [Convert] or [Rebuild All], always write them to both systems. Otherwise, a file mismatch will be detected.
- To operate CPU modules, write system parameters, CPU parameters, and program files. To operate I/O modules and intelligent function modules, write module parameters or module extension parameters.
- When new parameters have been set or the set parameters have been changed, reset the CPU module. (MELSEC iQ-R CPU Module User's Manual (Startup))
- Use the [Select Favorites] button to register frequently used items such as system parameters, CPU parameters and programs. Select [Setting] ⇨ [Register Favorites Selection] from the menu on the "Online Data Operation" window and set the items for the [Select Favorites] button.

37.7 Monitoring the Program

Check the operation of a program on the engineering tool.

Change the connection destination with the engineering tool and check the operating status of the system A or B.

[Online] ⇨ [Current Connection Destination]



1. Select a system in "Specify Redundant CPU" on the "Specify Connection Destination Connection" window.

2. Click the [Connection Test] button to check whether the CPU module of the selected system has been connected.

For how to check the operation, refer to the following.

GX Works3 Operating Manual

38 FUNCTIONS

This chapter describes the redundant functions and the functions that are different (modified or restricted) from those of the process mode. Functions not described in this chapter are the same as those of the process mode. (☞ Functions in Part 2)
For the availability of the functions in the redundant system, refer to the following.
📖 MELSEC iQ-R CPU Module User's Manual (Startup)

Function			Description	Reference
Redundant function	Operation mode change		Switches the operation mode of the redundant system between the backup mode for normal operation and the separate mode for system maintenance while it is running.	Page 389 Operation Mode Change
	System switching		Switches the systems between the control system and the standby system to continue operation of the redundant system when a failure or an error occurs in the control system. For debugging and maintenance purpose, users can switch the systems at any desired timing.	Page 392 System Switching
	Tracking transfer		Transfers the control data from the control system to the standby system and maintains the consistency of the data in the two systems to continue operation of the redundant system when a failure or an error occurs in the control system.	Page 405 Tracking Transfer
	Memory copy from control system to standby system		Transfers data such as parameters and programs in the CPU module of the control system to the CPU module of the standby system to maintain the consistency of the memory in the two CPU modules.	Page 420 Memory Copy from Control System to Standby System
	System consistency check		Checks whether the system configurations and files in the CPU modules are the same between the control system and the standby system when the redundant system is in backup mode.	Page 428 System Consistency Check
	Program execution in both systems		Detects an error in the external device or network of the systems (control system and standby system) by executing a program that diagnoses external devices or networks of both systems.	Page 433 Program Execution in Both Systems
	Redundant function module communication test		Checks a redundant function module for error when the communications of the redundant function module are unstable.	Page 441 Redundant Function Module Communication Test
Functions different (modified or restricted) from those of the process mode	Scan monitoring function		• During system switching, scan time monitoring with the watchdog timer is interrupted. Thus, no error is detected even if the scan time monitoring time has elapsed. Therefore, a time taken for system switching does not need to be considered in the scan time monitoring time setting. • Scan time monitoring with the watchdog timer is interrupted while the standby system is waiting for tracking data reception. Thus, no error is detected even if the scan time monitoring time has elapsed while the standby system is waiting for tracking data reception. Therefore, a time taken for tracking data reception does not need to be considered in the scan time monitoring time setting.	—
	Clock function	Time synchronization	The time of the standby system is synchronized with the time of the control system in the backup mode.	Page 442 Time synchronization
		System clock	After system switching, SM420 (User timing clock No.0) to SM424 (User timing clock No.4) remain off on the new control system CPU module. To use SM420 to SM424 on the new control system CPU module, execute the DUTY instruction again.	—
	Online change	Online change (ladder block)	When the online change (ladder block) is performed on the CPU module in one system in the backup mode, the change is also reflected on the CPU module in the other system.	Page 442 Online Change
	Interrupt function		When the systems are switched during time measurement of I28 to I31, I48, or I49, the time measurement interrupts and the new control system starts the time measurement from 0.	—
	SFC function		This function cannot be used in redundant mode.	—
	Constant scan		• In the standby system, when the system is powered off, a hardware failure has occurred, a tracking cable has a failure, or the system is switched to the separate mode, a continuation error may occur due to the excess of constant scan time. • The constant scan function is invalid for the standby system in backup mode.	Page 443 Constant scan

Function			Description	Reference
Functions different (modified or restricted) from those of the process mode	Remote operation	Using a contact	When the backup mode setting is enabled in the CPU parameter, both systems may be recognized as being mismatched in the system consistency check even if the remote operation is simultaneously performed on the systems because their operating statuses are mismatched depending on the timing.	—
		Using an engineering tool	In a redundant system, the operation target of remote operations depends on the operation mode and type.	Page 443 Remote Operation
	Label access setting from external device		In the transfer setup of external devices that communicate with the CPU module through GOT, SLMP, or other methods, select "Control System/Standby System", "System A/System B", or "Not Specified" to start a communication with a global label name specified.	—
	Latch function		When the systems are switched, the new control system and the new standby system start the value in "Interval Setting at Time Setting" in "Device Latch Interval Setting" from 0.	—
	Memory card function	Boot operation	Use the boot operation only to simultaneously start up both systems. When one system is restarted and the boot operation is used, an error occurs.	Page 446 Boot Operation
	Data logging function		In a redundant system, the data logging function collects data only in the control system regardless of the operation mode.	Page 448 Data Logging Function
	RAS function	Self-diagnostic function	The following items related to the redundant system are added to the contents that can be checked with the system monitor. • Tracking cable status • Display of the main base units of systems A and B • Module configuration on the base unit of the selected system • Event history of own system • System to which a CPU module is belonged (control system/ standby system, system A/B) and operation mode of the CPU module (backup mode/separate mode)	GX Works3 Operating Manual
		Error clear	A continuation error occurred in own system can be cleared by using the special relay (SM50) or the connected engineering tool. An error on the standby system CPU module can be cleared by using the special relay (SM1679) of the control system CPU module.	Page 233 Error Clear,Page 451 Clearing an Error on the Standby System CPU Module from the Control System CPU Module
	Security function		The security key setting is written or deleted on both systems individually, regardless of the operation mode.	—
	Monitor function	Test function	Changes in values of devices and labels are transferred (tracking) from the control system to the standby system before the END processing. If the systems are switched between the change and the tracking transfer, the change is not reflected on the new control system.	—
	Multiple CPU system function		This function cannot be used in redundant mode.	—
	SLMP communication		• There are notes for system switching when the system IP address matching function is not used. • If a remote operation command is executed, the CPU modules enter different operating statuses and thus systems cannot be switched. • For SLMP communications via the built-in Ethernet port of the CPU module, when communications are performed to the other system that cannot respond (power-off, reset, or tracking cable disconnection), a timeout error may occur.	Page 447 SLMP communication
	Inter-module synchronization function		This function cannot be used in redundant mode.	—

38.1 Operation Mode Change

This function switches the operation mode of the redundant system between the backup mode for normal operation and the separate mode for system maintenance while it is running.

Switching procedure

Switch the operation mode of the control system CPU module in the "Redundant Operation" window of the engineering tool.

Mode switching to the separate mode

The following describes the switching procedure for the separate mode.

To prevent tracking transfer in separate mode, turn off the tracking trigger in advance. (☞ Page 410 Tracking trigger)

1. Connect the engineering tool to the control system CPU module.
2. Open the "Redundant Operation" window of the engineering tool.

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

3. Select "Separate Mode" in "Change the Operation", and click the [Execute] button. The SEPARATE LED of the redundant function module turns on.

4. The standby system CPU module is set to the RUN-transition instruction waiting state (same as the STOP state), and the PROGRAM RUN LED flashes.
5. Set the standby system as the connection destination with the engineering tool, and execute the remote STOP and then the remote RUN with the remote operation function.
6. The CPU module is set to the RUN state and executes a program, and the PROGRAM RUN LED turns on.

Point

For the step 5, the following methods are also available for setting the CPU module to the RUN state.

- Setting the RUN/STOP/RESET switch in the STOP position to the RUN position
- Setting the module in the STOP state to the RUN state by using the RUN-PAUSE contact

■Powering off and on or resetting the CPU module in separate mode

In separate mode, the systems are not switched even if the control system CPU module is turned off and on or reset.

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

Before turning off or resetting one system in the separate mode, check if the other system is in the following status.

- The other system has been started and the READY LED is on.
- When the other system is set to the RUN state, the PROGRAM RUN LED is on.

If one system is turned off or reset while the other system is in the initial processing or initial processing (when switched to RUN), the operation modes of both systems become different and a stop error occurs.

Mode switching to the backup mode

The operation mode can be switched to the backup mode only in the communication path of when the operation mode was switched to the separate mode.

The following describes the switching procedure for the backup mode.

1. Set the CPU modules of the control system and standby system to have the same file configuration and operating status. (To prevent an error detected by a restarted system consistency check after switching the separate mode to the backup mode)
2. Connect the engineering tool to the control system CPU module.
3. Open the "Redundant Operation" window of the engineering tool.

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

4. Select "Backup Mode" in "Change the Operation", and click the [Execute] button. The BACKUP LED of the redundant function module turns on.

5. When the tracking trigger has been turned off at the mode switching to the separate mode, turn on the tracking trigger. (☞ Page 410 Tracking trigger)

Point

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

Precautions

The following describes the precautions for switching the operation mode.

Switching of the operation mode in the RUN-transition instruction waiting state

When the operation mode is switched to the separate mode, the standby system CPU module is set to the RUN-transition instruction waiting state (same as the STOP state). If the operation mode is switched to the backup mode in this state, the standby system CPU module goes to the RUN state.

The following describes the precautions for this state.

- If the SP.CONTSW instruction is executed before the standby system CPU module goes to the RUN state (during initial processing (when switched to RUN)), the operating status of each of both systems becomes different and a continuation error will occur. (☞ Page 404 System switching during initial processing/initial processing (when switched to RUN))
- If the standby system CPU module takes time to switch to the RUN state due to causes such as access to a file in the standby system CPU module, the operating status of each of both systems becomes different and a continuation error occurs.

Switching the mode during initial/initial processing (when switched to RUN)

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

Consecutive operation mode switching

To consecutively switch the operation mode, switch the operation mode in certain intervals. If the operation mode is changed again before the operating status is switched by the previous operation mode switching, the operating status of each of both systems becomes different and a continuation error will occur.

38.2 System Switching

This function switches the systems between the control system and the standby system to continue operation of the redundant system when a failure or an error occurs in the control system. The systems can also be switched manually by a user for debug or maintenance.

System switching method

The following two methods are available for system switching: system switching (automatically performed by a redundant system) and manual system switching (performed by a user).
The following table lists the system switching types, causes, execution availability by operation mode, and priority of when multiple causes are simultaneously generated.
○: Operation possible, ×: Operation not possible

System switching type	System switching cause	Operation mode		Priority	
		Backup mode	Separate mode		
Automatic system switching	Power-off, reset, hardware failure of the CPU module	○	×	High ↑ ↓ Low	1
	Stop error of the CPU module				2
	System switching request from a network module				3
Manual system switching	System switching request by the SP.CONTSW instruction	○	○		4
	System switching request from the engineering tool				5

- When multiple system switching causes are simultaneously generated, the systems are switched according to the cause with higher priority. The switching cause determined according to the priority is stored in the event history and SD1643 (System switching cause).
- The manual system switching is requested for the control system CPU module.

Point

The systems are switched even when the CPU module is set to the STOP state. They are switched when any of the above switching causes is generated.

Automatic system switching

A redundant system judges whether system switching is required or not in backup mode, and automatically switches the systems between the control system and the standby system as required.

■System switching due to power-off, reset, or hardware failure of the CPU module

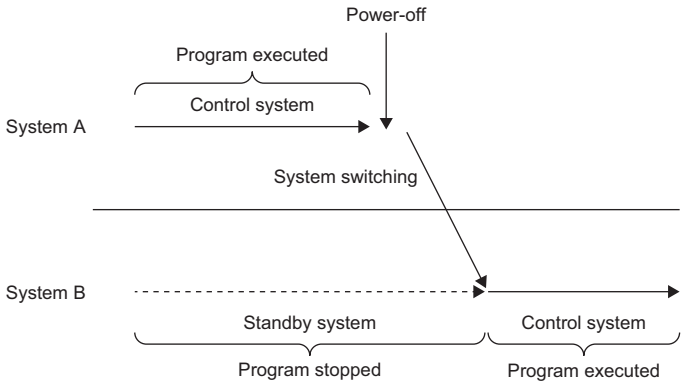
In a redundant system, the standby system CPU module monitors the control system status. If the control system is unable to control the redundant system in the following cases, the standby system CPU module is switched to the control system CPU module and the new control system CPU continues the control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

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

*1 The control system CPU module is switched to the standby system CPU module even when the existing standby system CPU module is not ready to switch. (☞ Page 398 Execution availability of system switching)

Ex.

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



Point

When a network is established using the CC-Link IE Field Network module, the systems can be switched even when a redundant function module is removed from the base unit or a failure has occurred in the base unit.

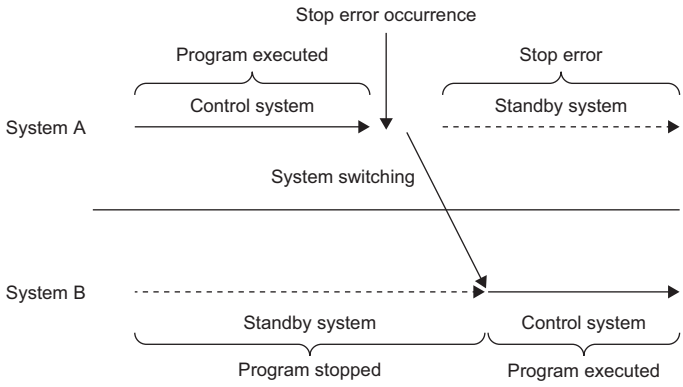
■Stop error of the CPU module

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

*1 If a WDT error has occurred, the control system CPU module is switched to the standby system CPU module even when the existing standby system CPU module is not ready to switch. (☞ Page 398 Execution availability of system switching)

Ex.

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

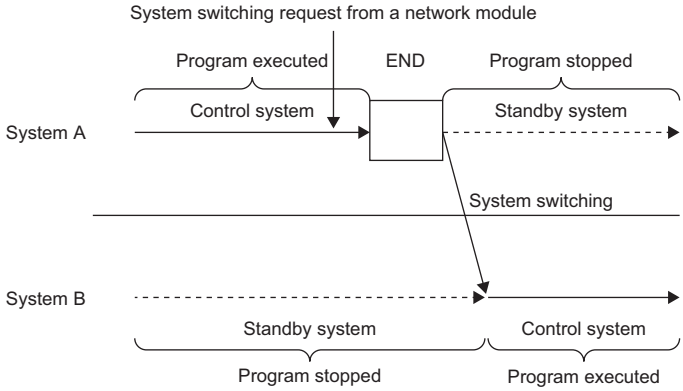


■System switching request from a network module

A control system network module requests the CPU module for system switching when a communication error or disconnection is detected. When the control system CPU module receives the system switching request from the network module, the systems are switched after the END processing.

Ex.

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



The following network modules send a system switching request.

- CC-Link IE Controller Network module
- CC-Link IE Field Network module
- Ethernet interface module with built-in CC-Link IE

If a network module cable is disconnected, the systems may not be switched depending on the timing of error detection on the control system and the timing on the standby system. (☞ Page 404 When the cable for the network module is disconnected)

Manual system switching

The user can manually switch the systems between the control system and the standby system.

Point

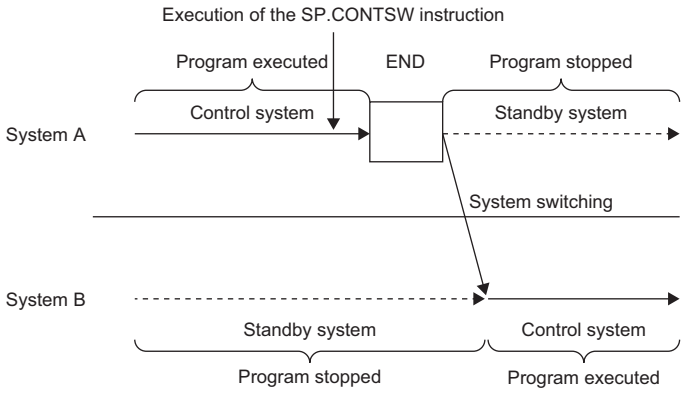
- After turning on SM1646 (System switching by a user), perform the manual system switching in the control system.
- When the manual system switching is disabled by the DCONTSW instruction, execute the ECONTSW instruction. The system switching is enabled in the initial status.

■System switching by the SP.CONTSW instruction

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

Ex.

Operation of system switching by the SP.CONTSW instruction



Restriction

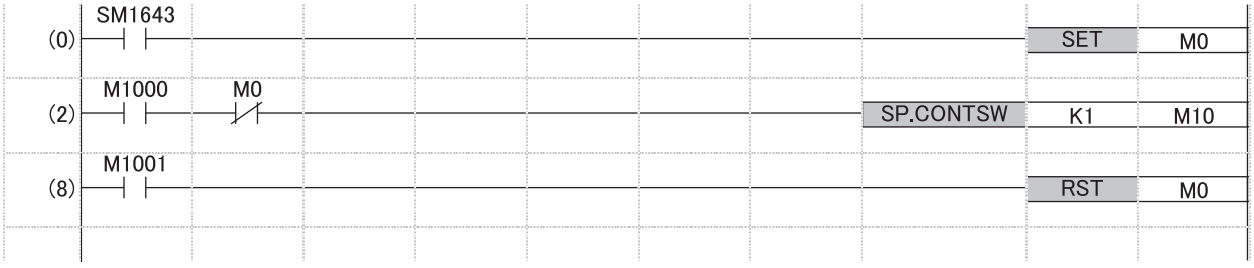
When the SP.CONTSW instruction is executed in separate mode, a several scans may be required for the system switching.

Point

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

When the SP.CONTSW instruction is used, create a program that does not execute the SP.CONTSW instruction again on the new control system CPU module by using SM1643 (ON for only one scan after system switching (standby system to control system)), as shown below.

M1000: System switching command, M1001: Clear signal



For details on the SP.CONTSW instruction, refer to the following.

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

■System switching request from the engineering tool

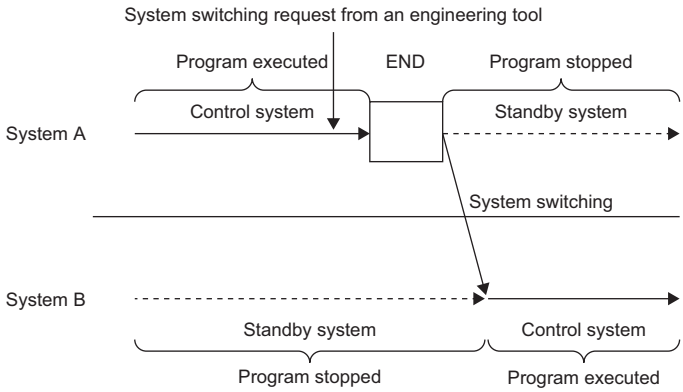
When the engineering tool sends a system switching request to the control system CPU module, the systems are switched after the END processing.

Switch the systems from the "Redundant Operation" window of the engineering tool.

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

Ex.

Operation of system switching from the engineering tool



Operation at system switching

The following table shows the operations of the CPU modules of when the control system and the standby system are switched.

These operations are for when both systems are operating and the operating statuses of the CPU modules are the same.

Item		New control system CPU module	New standby system CPU module
Program execution	Initial execution type program	This type of program is not executed. However, when an initial execution type program has not been completed on the old control system at system switching, it is executed again from its head.	The program execution stops.*1
	Scan execution type program	This type of program is executed from the step 0.	
	Fixed scan execution type program	The fixed scan execution interval is measured from 0.	The program execution stops.
	Standby type program	This type of program is not executed.	
	Event execution type program	• Interrupt program: The program is executed when an interrupt factor is generated. • ON of bit data (TRUE): The program is executed when a specified factor is generated. • Passing time: A specified time is measured from 0.	
Tracking transfer		Tracking data is transferred to the new standby system. However, when a stop error occurs on the new standby system, the new control system does not transfer the tracking data to the new standby system.	Tracking data is received.
Online change		Write operation at system switching continues.	
Device/label memory		The state before system switching is held.	
Signal flow memory (☞ Page 412 Tracking transfer setting for the signal flow memory)		• In backup mode, the signal flow of the old control system is reflected.*2 • In separate mode, the signal flow of the old standby system is reflected.	The state before system switching is held.
Special relay (SM), special register (SD)		The state before system switching is held. However, when the CPU module is in the RUN state, SD520 to SD531 are cleared.	
Output (Y)		The state before system switching is held and output refresh is performed.	After the output (Y) is turned off, output refresh stops.*3
Direct access input (DX)		In the program execution after system switching, data is fetched when an instruction using the direct access input (DX) is executed.	No operation is performed because the program does not operate.*1
Direct access output (DY)		In the program execution after system switching, data is output when an instruction using the direct access output (DY) is executed.	
FROM/TO instructions		In the program execution after system switching, this instruction is executed when the instruction execution condition is satisfied.	
Instruction that requires several scans		In the program execution after system switching, this instruction is executed when the instruction execution condition is satisfied.	• No operation is performed because the program does not operate.*1 • When the systems are switched while the instruction is being executed, the instruction execution continues. However, the completion device does not turn on in the new standby system, and it turns on in the control system after the next system switching.*1

*1 The operation varies in a program executed in both systems. (☞ Page 436 Operation at system switching)

*2 The operation varies when the signal flow memory is not transferred. (☞ Page 455 Instructions whose operations vary depending on tracking of the signal flow memory)

*3 The operation varies when the standby system output setting is enabled. (☞ Page 439 Redundant System Operation Setting)

Execution availability of system switching

The following tables show the execution availability of system switching in each operation mode.

In backup mode

○: Switching possible, ×: Switching not possible

Redundant system status		Execution availability of system switching						
		Automatic system switching					Manual system switching	
		Power-off or reset of the CPU module	Hardware failure of the CPU module	Stop error of the CPU module		System switching request from a network module*3	System switching request by the SP.CONTSW instruction*3	System switching request from the engineering tool*4
				WDT error	Error other than WDT error			
Normal operation or continuation error in the standby system		○	○	○	○	○	○	○
Tracking communications disabled (cable disconnection)		×	○*6	○*6	×	×	×	×
Power-off, reset, hardware failure of the standby system CPU module		×	○*6	○*6	×	×	×	×
Stop error in the standby system	WDT error	×	○*6	○*6	×	×	×	×
	Error other than WDT error	○	○	○*6	×	×	×	×
Network error detected in the standby system*1		○	○	○	○	×	×	×
During memory copy from control system to standby system		○	○	○*6	×	×	×	×
During online change		○	○	○	○	○*2*5	×	×
Mismatch between the CPU module operating status of both systems		○	○	○	○	×	×	×
During system switching		○	○	○	○	○	×	×
During online module change	Redundant module	×	○*6	○*6	×	×	×	×
	Other modules	○	○	○	○	○	○	○
System switching disabled by the DCONTSW instruction		○	○	○	○	○	×	×

*1 When the group specification has been set in the standby system, a network error is not detected if communication is available with the line of an Ethernet-equipped module after a communication error has occurred on the other Ethernet-equipped module. (📖 MELSEC iQ-R Ethernet User's Manual (Application))

*2 After the online change has been completed, the switching cause is detected and the systems are switched. However, when an online change is being executed only in the standby system, the systems cannot be switched.

*3 If system switching is disabled when a system switching request is sent, a continuation error occurs and a cause of a system switching failure is stored in SD1644 (Cause of system switching failure).

*4 If system switching is disabled when a system switching request is sent, the error code corresponding to a cause of the system switching failure is returned.

*5 When the systems are switched during an online change, a mismatch between the files is detected in the system consistency check and a stop error may occur on the new standby system. (☞ Page 430 File)

*6 Although the control system CPU module is switched to the standby system CPU module, the existing standby system is not switched.

In separate mode

○: Switching possible, ×: Switching not possible

Redundant system status		Execution availability of system switching				
		Automatic system switching			Manual system switching	
		Power-off, reset, hardware failure of the CPU module	Stop error of the CPU module	System switching request from a network module	System switching request by the SP.CONTSW instruction*2	System switching request from the engineering tool*3
Normal operation or continuation error in the standby system		×	×	×	○	○
Tracking communications disabled (cable disconnection)		×	×	×	×	×
Power-off, reset, hardware failure of the standby system CPU module		×	×	×	×	×
Stop error in the standby system		×	×	×	×	×
Network error detected in the standby system*1		×	×	×	○	○
During memory copy from control system to standby system		×	×	×	×	×
During online change		×	×	×	○	○
Mismatch between the CPU module operating status of both systems		×	×	×	○	○
During system switching		×	×	×	×	×
During online module change	Redundant module	×	×	×	×	×
	Other modules	×	×	×	○	○
System switching disabled by the DCONTSW instruction		×	×	×	×	×

*1 When the group specification has been set in the standby system, a network error is not detected if communication is available with the line of an Ethernet-equipped module after a communication error has occurred on the other Ethernet-equipped module. (MELSEC iQ-R Ethernet User's Manual (Application))

*2 If system switching is failed when a system switching request is sent, a continuation error occurs and a cause of a switching failure is stored in SD1644 (Cause of system switching failure).

*3 If system switching is disabled when a system switching request is sent, the error code corresponding to a cause of the system switching failureis returned.

Occurrence of a cause of the system switching failure

When a cause of the system switching failure occurs, the BACKUP LED flashes in backup mode and the SEPARATE LED flashes in separate mode.

Point

The cause to flash the BACKUP LED or SEPARATE LED can be checked in SD1642 (BACKUP/SEPARATE LED flashing cause). Check SD1642 and eliminate the cause to flash the LED.

Check method of system switching information

The following table lists the check methods of system switching information at system switching (automatic system switching and manual system switching).

Check method	Information	Reference
Event history	System switching result, system switching cause, and control system/standby system transition	Page 400 Event history
Special relay (SM)/Special register (SD)	System switching result and detailed information	Page 400 Special relay/Special register
CTRL LED and SBY LED of the redundant function module	System switching result	Page 401 CTRL LED and SBY LED of the redundant function module

When the systems have been switched, check the switching cause or detailed information and take corrective action to restore the system to a normal state as required.

Event history

The information related to system switching results, system switching cause, and control system/standby system transitions can be checked with the event history of the engineering tool.

When the systems are switched, the following items are stored in the event history of both systems.

- Automatic system switching: "System switching (system)" (event code: 00F00)
- Manual system switching: "System switching (user)" (event code: 2B000)

When the systems are switched by the SP.CONTSW instruction, the system switching instruction ID number specified with the SP.CONTSW instruction is also stored.

Special relay/Special register

System switching results and detailed information can be checked with the special relay and special register.

- Whether the systems have been switched or not can be checked by checking SD1649 (System switching Cause (when the systems are successfully switched)). When the systems have been switched, the switching cause is stored in SD1649 of the control system and standby system.
- If the systems have not been switched even after a switching cause is generated, the cause why the systems have not been switched can be checked by checking SD1644 (Cause of system switching failure). The switching cause is stored in SD1643 (System switching cause).

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

- Special relay (☞ Page 631 Redundant function)
- Special register (☞ Page 673 Redundant function)

■Special relay

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

○: Stored, ×: Not stored

SM number	Name	Storage status at system switching	
		New control system CPU module	New standby system CPU module
SM1637	System switching detection (standby system to control system)	○	○
SM1643	System switching check flag (standby system to control system)	○	×
SM1644	System switching check flag (control system to standby system)	×	○
SM1645	System switching request from a network module	○	○
SM1646	System switching by a user	×	×

■Special register

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

○: Stored, ×: Not stored

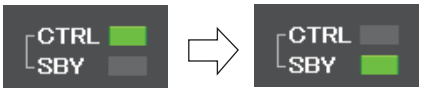
SD number	Name	Storage status at system switching	
		New control system CPU module	New standby system CPU module
SD1642	BACKUP/SEPARATE LED flashing cause	×	×
SD1643	System switching cause	×	○
SD1644	Cause of the system switching failure*1	×	×
SD1645	System switching request status from a network module of own system	○	○
SD1646	System switching request status from a network module of the other system	○	○
SD1648	Cause of the other system monitoring error	○	○
SD1649	System switching Cause (when the systems are successfully switched)	○	○
SD1650	System switching instruction ID number	○	○

*1 When the system switching has not been normally completed, a value is stored in this special register area of the control system.

CTRL LED and SBY LED of the redundant function module

System switching results can be checked with the CTRL LED and SBY LED on the front of the redundant function module.

Switching from the standby system to the control system



Switching from the control system to the standby system

Precautions

The following describes the precautions on system switching.

Item	Description	Reference
Error in the redundant function module	• When an error has been detected on the redundant function module, the control system and standby system continue operating without being switched. • When a communication error has been detected in the communication between the redundant function module and a CPU module, the systems are switched.	Page 403 Error in the redundant function module
When both systems operate as standby systems	• If a communication error has occurred due to a tracking cable error during system switching, both systems may operate as standby systems. • If a WDT error or a hardware failure of the CPU module has occurred in the control system while a stop error exists in the standby system, both systems operate as standby systems.	Page 403 When both systems operate as standby systems
When both systems operate as control systems	If the tracking cable and the network cable that is connected to the control system network module are simultaneously pulled out or disconnected, both systems may operate as control systems. In this case, connect the tracking cable again. After a stop error has occurred in the system B, power off and on or reset the system B.	—
Scan time monitoring function	During system switching, scan time monitoring with the watchdog timer is interrupted. Thus, in a scan in which the systems are switched, no error is detected even if the scan time monitoring time has elapsed. A time taken for system switching does not need to be considered in the scan time monitoring time (WDT) setting.	—
System switching disabled by online change	In a redundant system, the manual system switching is disabled during an online change.	Page 403 System switching disabled by online change
Operation of an event execution type program	When "ON of bit data (TRUE)" has been specified in the trigger type and the systems are switched before tracking transfer of device data, an event execution type program does not operate in the new control system.	—
When the cable for the network module is disconnected	If a network module cable is disconnected, the systems may not be switched depending on the timing of error detection on the control system and the timing on the standby system.	Page 404 When the cable for the network module is disconnected
System switching during initial processing/ initial processing (when switched to RUN)	• When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the control system, the systems are switched after the initial processing or initial processing (when switched to RUN). • When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the standby system, whether the systems are switched or not depends on the system switching cause.	Page 404 System switching during initial processing/initial processing (when switched to RUN)
CC-Link control at system switching	When the master operating station is switched with a program by using the CC-Link standby master function, the CC-Link control can be continued even after system switching.	MELSEC iQ-R CC-Link System Master/Local Module User's Manual (Application)

Error in the redundant function module

- When an error has been detected on a redundant function module, a continuation error occurs on the CPU module, and the control system and standby system continue operating without being switched. Check the error code, and perform online module change if the redundant function module has failed. (📖 MELSEC iQ-R Online Module Change Manual)
- When the redundant function module was removed from the base unit or a failure has occurred in the base unit and a communication error has been detected in the communication between the redundant function module and a CPU module, a stop error occurs on the CPU module and the systems are switched. In this case, check the error code and eliminate the error cause.

Restriction

When the CC-Link IE Field Network module has not been mounted, the systems may not be switched when a communication error has been detected in the communication between the redundant function module and the CPU module.
To switch the systems even though the redundant function module has been removed from the base unit or a failure has occurred in the base unit, mount the CC-Link IE Field Network module.

When both systems operate as standby systems

- If a communication error is generated due to a tracking cable error during system switching, both systems may operate as standby systems. When an error occurs with the tracking cable, each L ERR LED of the redundant function modules turns on. In this case, replace the tracking cable with a new one as soon as possible. If both systems operate standby systems, connect a new tracking cable properly and turn off and on or reset the CPU module of one system so that the other system will operate as the control system.
- If a WDT error or a hardware failure of the CPU module has occurred in the control system while a stop error exists in the standby system, the control system is switched to the standby system and the both systems operate as standby systems. In this case, eliminate error causes, and then power off and on or reset the CPU modules in the both systems.

The following table lists the operations of when both systems operate as standby systems.

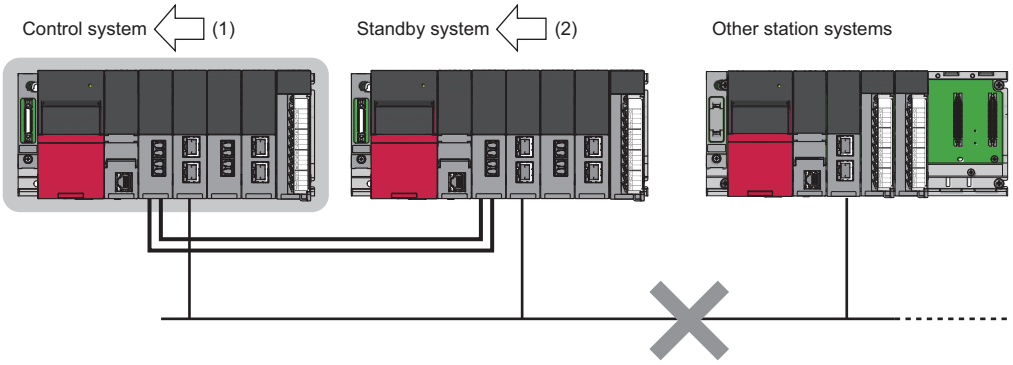
Item		Operation of both systems
LED of the redundant function module	BACKUP LED	Flashing (in backup mode)
	SEPARATE LED	Flashing (in separate mode)
	CTRL LED	Off
	SBY LED	On
Special relay	SM1635 (Standby system judgment flag)	On
System switching request	System switching request from a network module	System switching disabled
	System switching request by the SP.CONTSW instruction	
	System switching request from the engineering tool	

System switching disabled by online change

In a redundant system, the manual system switching is disabled during an online change. To disable the manual system switching during an online change, the CPU module is set to the manual system switching disabled state before an online change starts. After the online change is completed, it is set to the manual system switching enabled state.
If a communication with the engineering tool is interrupted due to communication cable disconnection or other causes during an online change, the online change cannot be completed, and the CPU module remains in the manual system switching disabled state. When the CPU module is in the manual system switching disabled state, the systems cannot be switched by manual system switching or a system switching request from a network module.
If an online change has failed, refer to the following and take actions.
☞ Page 442 Action for when an online change has failed

When the cable for the network module is disconnected

If a network module cable is disconnected, the systems may not be switched depending on the timing of error detection on the control system and the timing on the standby system.



(1) When the control system detects cable disconnection first, the systems are switched.
(2) When the standby system detects cable disconnection first, the systems are not switched. At this time, a continuation error occurs on the control system CPU module and the BACKUP LED flashes since a cause of the system switching failure has been generated.

In both case, replace the network module cable with a new one and clear the network error.

System switching during initial processing/initial processing (when switched to RUN)

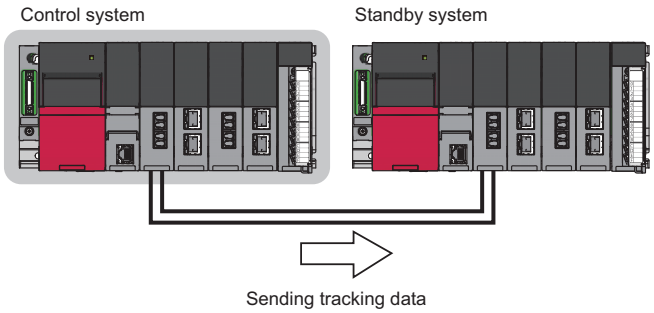
When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the control system, the systems are switched after the initial processing or initial processing (when switched to RUN). Even if the initial processing or initial processing (when switched to RUN) takes time and a communication error occurs, the systems are switched after the initial processing or initial processing (when switched to RUN).

When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the standby system, whether the systems are switched or not depends on the system switching cause.

System switching cause	Whether to switch the systems
Power-off, reset, hardware failure of the CPU module	The systems are switched after the initial processing or initial processing (when switched to RUN).
Stop error of the CPU module	
System switching request from a network module	The systems are not switched.
System switching request by the SP.CONTSW instruction	• In backup mode, the systems are not switched. • In separate mode, the systems will be switched after the initial processing or initial processing (when switched to RUN).
System switching request from the engineering tool	

38.3 Tracking Transfer

This function transfers the control data from the control system to the standby system and maintains the consistency of the data in the two systems to continue operation of the redundant system when a failure or an error occurs in the control system.



- The control system CPU module sends tracking data to the standby system CPU module before the END processing. The tracking data includes device/label data, special relay (SM), special register (SD), PID control instruction information, and system data.
- The standby system CPU module receives the tracing data, and then reflects the received data on the device/label memory.
- Tracking transefer is performed in both of backup mode and separate mode.
- Device/label data can be divided and transferred as required. To transfer device/label data within a desired range, set a range of global devices to be transferred or whether to transfer local devices/global devices for each tracking block and turn on the tracking triggers (SD1667 to SD1670) assigned for each tracking block.
- Up to 1M-word device/label data can be transferred in one scan.
- During an online change or when a stop error has occurred, tracking transfer is performed only for system data. (☞ Page 408 System data)

Point

- In the CPU parameter, "Tracking Device/Label Setting" is set to "Transfer collectively" by default. Thus, tracking transfer is performed without setting the parameter. (☞ Page 412 Batch transfer)
- When "Process Control Extension Setting" is enabled, the tracking transfer setting is automatically configured in detailed settings. The range of the file register specified to the system resource of the optional process control extension setting is registered to the tracking block No.64. (☞ Page 411 When the "Process Control Extension Setting" is enabled)

Restriction

When a tracking transfer is not performed, the possible causes are the following:

- The standby system CPU module has been powered off or reset.
- Hardware failure of the CPU module has occurred.*1
- An error has occurred in the redundant function module.*1
- A WDT error has occurred.*1
- Tracking cables has been pulled out or disconnected.

*1 When the tracking communication line properly works, "Link-up" (event code: 00100) may be stored in the event history.

Tracking data

The following table lists the tracking data that can be transferred from the control system to the standby system.

Item		Operation mode		Reference
		Backup mode	Separate mode	
Device data	User device	○	○	Page 407 Devices that can be specified
	Special relay	○*1	×	Page 408 Special relay
	Special register	○*1	×	Page 408 Special register
Global label*3*4		○	○	Page 413 Tracking device/ label setting
Local device		○*2	×	
Local label		○*2	×	
Signal flow memory		○*2	×	Page 412 Tracking transfer setting for the signal flow memory
PID control instruction information		○*1	×	Page 408 PID control instruction information
CPU buffer memory		×	×	—
System data		○*1	○*1	Page 408 System data

*1 The data is automatically transferred regardless of parameter settings. (☞ Page 408 Auto tracking data)

*2 The data that is used in a program executed in both systems is not transferred. (☞ Page 433 Program Execution in Both Systems)

*3 Module labels are not transferred.

*4 Global labels with devices assigned are not transferred as global labels. They are transferred according to the tracking transfer settings of the assigned devices.

Devices that can be specified

The following table lists the data that can be specified for tracking transfer.

○: Specifiable, ×: Not specifiable, —: Not settable as a local device

Classification	Device name	Transfer	
		Global device	Local device
User device	Input (X)	○	—
	Output (Y)	○	—
	Internal relay (M)	○	○
	Link relay (B)	○	—
	Annunciator (F)	○	—
	Link special relay (SB)	○	—
	Edge relay (V)	○	○
	Step relay (S)	×	—
	Timer (T)	○	○
	Retentive timer (ST)	○	○
	Long timer (LT)	○	○
	Long retentive timer (LST)	○	○
	Counter (C)	○	○
	Long counter (LC)	○	○
	Data register (D)	○	○
	Link register (W)	○	—
	Link special register (SW)	○	—
	Latch relay (L)	○	—
System device	Function input (FX)	×	—
	Function output (FY)	×	—
	Function register (FD)	×	—
	Special relay (SM)	○*1	—
	Special register (SD)	○*1	—
Index register	Index register (Z)	○	○
	Long index register (LZ)	○	○
File register	File register (R)	×	—
	File register (ZR)	○	—
Refresh data register	Refresh data register (RD)	○	—

*1 The data is automatically transferred regardless of parameter settings. (☞ Page 408 Auto tracking data)

Auto tracking data

The following tables list the data that is automatically transferred by the system regardless of parameter settings of tracking transfer.

■Special relay

The following table lists the special relay areas that are automatically transferred by the system.

SM number	Name
SM752	Dedicated instruction End bit control flag
SM754	BIN/DBIN instruction error control flag
SM755	Scaling data check settings
SM756	Module access completion wait control flag
SM775	Selection of link refresh processing during the COM instruction execution
SM776	Local device setting at CALL
SM777	Local device setting in interrupt programs
SM792	PID bumpless processing (for the complete differentiation PIDCONT instruction)
SM794	PID bumpless processing (for the inexact differential S.PIDCONT instruction)
SM816	Hold mode (S.IN instruction)
SM817	Hold mode (S.OUT instruction)
SM1646	System switching by a user

For details on the special relay, refer to the following.

☞ Page 606 List of Special Relay Areas

■Special register

The following table lists the special register areas that are automatically transferred by the system.

SD number	Name
SD250	Loaded maximum I/O
SD414	2n second clock setting
SD415	2n ms clock setting
SD771	Specification of the number of write instruction executions to data memory
SD775	Selection of refresh processing during the COM instruction execution
SD792, SD793	PID limit setting (for complete derivative)
SD794, SD795	PID limit setting (for incomplete derivative)
SD816, SD817	Basic period
SD818	Bumpless function availability setting for the S.PIDP instruction
SD1662	Tracking transfer data receive completion wait time
SD1667 to SD1670	Tracking transfer trigger

For details on the special register, refer to the following.

☞ Page 635 List of Special Register Areas

■PID control instruction information

The PID control data that is specified with the PIDINIT or S.PIDINIT instruction is transferred. (📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks))

■System data

The data related to the redundant system, such as system switching and operation mode change, is transferred.

Tracking block and tracking trigger

The devices or labels of a specified range is transferred by setting a range of devices or labels to be transferred for each tracking block and turning on the tracking trigger which is assigned for each tracking block.

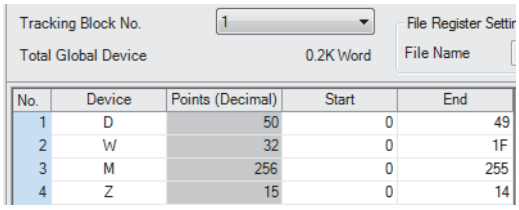
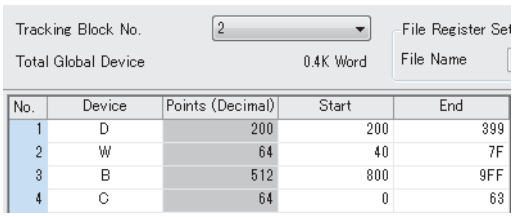
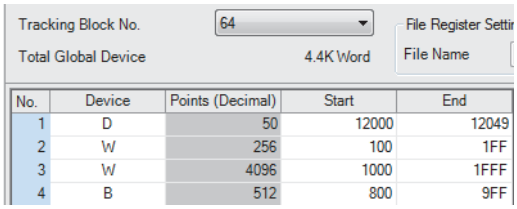
Tracking block

A tracking block is used for setting a tracking transfer range of global devices and whether or not to transfer local devices/global labels/local labels.

- Up to 64 blocks (No.1 to 64) are available.
- Up to 2048 devices can be set in a block. Up to 2048 devices can be transferred in a single tracking.
- The available capacity of tracking devices or labels is 1M words per block.

Ex.

When only the tracking block No.3 has no setting

Tracking block No.	Setting example
1	 No. / Device / Points (Decimal) / Start / End 1 D 50 0 49 2 W 32 0 1F 3 M 256 0 255 4 Z 15 0 14
2	 No. / Device / Points (Decimal) / Start / End 1 D 200 200 399 2 W 64 40 7F 3 B 512 800 9FF 4 C 64 0 63
3	Not set
⋮	⋮
64	 No. / Device / Points (Decimal) / Start / End 1 D 50 12000 12049 2 W 256 100 1FF 3 W 4096 1000 1FFF 4 B 512 800 9FF

Point

Global labels, local labels, and local devices are transferred only once in one scan. Even when the same global labels, local labels, and local devices are specified in multiple tracking blocks, the overlap does not affect the data capacity to be transferred.

Tracking trigger

By turning on a tracking trigger, the devices or labels of a range specified in the corresponding tracking block are transferred. Bits used as tracking triggers change depending on the CPU parameter setting, as shown below.

■When "Tracking Device/Label Setting" is set to "Transfer collectively"

The bit 0 of SD1667 is used as a tracking trigger. The bit is automatically turned on by the system at an initial processing or operating status change (STOP to RUN), and the tracking transfer is started. To stop the tracking transfer, turn off the bit. To restart, turn on the bit.

■When "Tracking Device/Label Setting" is set to "Detailed setting"

SD1667 to SD1670 (64 bits) are used as tracking triggers. The bit 0 of SD1667 to the bit 15 of SD1670 correspond to the tracking blocks No.1 to 64. To start the tracking transfer of a tracking block, turn on the corresponding bit. To stop the tracking, turn off the bit. To restart the tracking transfer, turn on the bit again.

When "Tracking Block No.1 Auto Transfer Setting" is set to "Transfer Automatically" in the CPU parameter, the bit 0 of SD1667 is automatically turned on by the system at initial processing or operating status change from STOP to RUN, and the tracking transfer is started.

Ex.

Program example for changing the status of tracking transfer triggers according to conditions

- When only M0 (Trigger switching condition 1) is on, the tracking transfer trigger of the tracking block No.1 is turned on. (Tracking blocks No.2 to No.64 are not transferred.)
- When only M1 (Trigger switching condition 2) is on, the tracking transfer trigger of the tracking block No.2 is turned on. (Tracking blocks No.1 and No.3 to No.64 are not transferred.)

Setting procedure for tracking transfer

The following describes the setting procedure for tracking transfer.

1. Set "Tracking Setting" in the CPU parameter. (☞ Page 411 Tracking transfer setting)
2. Add a program that controls tracking triggers. (☞ Page 410 Tracking trigger)
3. Write the set parameters and program to the CPU module with the engineering tool.
4. Start up the system again.
5. Turn on the tracking trigger with the program to start the tracking transfer.

When "Tracking Device/Label Setting" is set to "Transfer collectively", the above steps 2 and 5 are not required.

Point

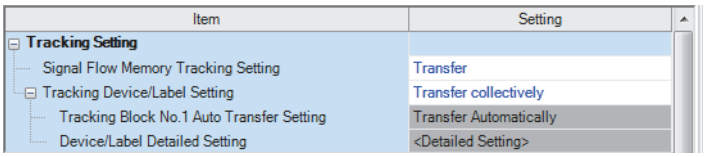
To continue the same control as before even after system switching, configure the settings in "Tracking Setting" so that all data necessary for the program is transferred.

Tracking transfer setting

The following describes the CPU parameters related to tracking transfer.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Tracking Setting]

Window



Displayed items

Item	Description	Setting range	Default
Signal Flow Memory Tracking Setting	Set whether to transfer the signal flow memory or not. (☞ Page 412 Tracking transfer setting for the signal flow memory)	• Do Not Transfer • Transfer	Transfer
Tracking Device/Label Setting	Set "Transfer collectively" or "Detailed setting" for devices or labels to be transferred. • When "Transfer collectively" is set, global devices, local devices, global labels, and local labels are assigned to the tracking block No. 1 and automatically transferred. (☞ Page 412 Batch transfer) • When "Detailed setting" is set, detailed settings can be configured in "Tracking Block No.1 Auto Transfer Setting" and "Device/Label Detailed Setting".	• Transfer collectively • Detailed setting	Transfer collectively
Tracking Block No.1 Auto Transfer Setting	Set whether to automatically transfer the tracking block No.1 or not. (☞ Page 409 Tracking block and tracking trigger)	• Do Not Transfer Automatically • Transfer Automatically	Transfer Automatically
Device/Label Detailed Setting	Set devices and labels to be transferred. (☞ Page 413 Detailed setting)	—	—

Point

Set "Detailed setting" of "Tracking Device/Label Setting" for the following purposes.

- To shorten a tracking transfer time
- To add tracking data, such as the file register (ZR)
- To transfer each data set individually

When the "Process Control Extension Setting" is enabled

The tracking transfer setting is automatically configured in detailed settings. The range of the file register specified to the system resource of the optional process control extension setting is registered to the tracking block No.64.

Since the other devices are not automatically registered, specify the other tracking block numbers for those devices. The devices and their ranges set in "Device/Label Memory Area Setting" can be input in a batch by clicking the [Device Setting Reflection] button in the "Global Device Setting" window. (☞ Page 414 Global device setting)

Tracking transfer setting for the signal flow memory

By transferring the signal flow memory, operations of rising/falling instructions in the old control system are taken over to the new control system even after system switching.

Point

"Signal Flow Memory Tracking Setting" is set to "Transfer" by default. Transferring the signal flow memory is recommended. For the operation not to transfer it, refer to the following.

☞ Page 455 Instructions whose operations vary depending on tracking of the signal flow memory

■Tracking of individual POUs

The following table shows whether the signal flow memory is transferred or not for each POU.

○: Transferred, ×: Not transferred, —: No signal flow memory

POU			Both systems program executions setting	
			Control system execution	Both systems executions
Program block			○	×
Function block	Macro type		○	×
	Subroutine type	Global FB	○	○*1
		Local FB	○	×
Function			—	

*1 To prevent the signal flow memory in the standby system from being overwritten in a both systems execution program, use a local FB. When a global FB is used, the signal flow memory in the standby system is overwritten with the signal flow memory in the control system.

Tracking device/label setting

The following two methods are available for transferring devices and labels: automatically transferring all the devices and labels in a batch or transferring specified devices and labels of a specified tracking block.

■Batch transfer

When "Tracking Device/Label Setting" is set to "Transfer collectively", the following devices and labels are assigned to the tracking block No. 1 and automatically transferred.

Type	Description
Global device	■Bit device • Input (X) • Output (Y) • Internal relay (M) • Link relay (B) • Edge relay (V) • Latch relay (L) ■Word device • Timer (T) • Long timer (LT) • Retentive timer (ST) • Long retentive timer (LST) • Counter (C) • Long counter (LC) • Data register (D) • Link register (W) • Index register (Z) • Long index register (LZ)
Local device*1	All local devices
Global label*2	All the global labels assigned to the device/label memory
Local label*1	All local labels

*1 The data used in both systems execution programs is not transferred.

*2 Global labels with devices assigned are not transferred as global labels. They are transferred according to the tracking setting of the assigned devices. To transfer global labels with devices assigned, specify the assigned global devices in the global device setting. (☞ Page 414 Global device setting)

Point

To transfer the annunciator (F), link special relay (SB), link special register (SW), file register (ZR), or refresh data register (RD), specify the corresponding data in "Device/Label Detailed Setting" of "Tracking Device/Label Setting". (☞ Page 413 Detailed setting)

Restriction

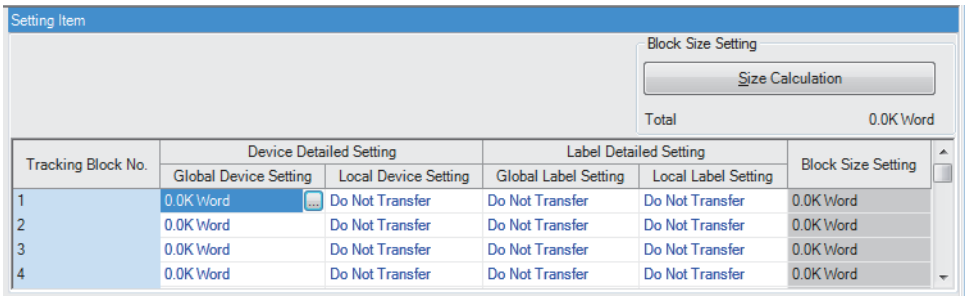
After setting "Transfer collectively", perform a test operation in the system design phase and check if the size of tracking data is 1M words or smaller. If the size of the tracking data is larger than 1M words, a stop error occurs when the CPU module is powered off and on or reset.

■Detailed setting

Set devices and labels to be transferred for each tracking block (No.1 to 64) to be used.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Tracking Setting] ⇨ [Device/Label Detailed Setting]

Window



Displayed items

Item	Item	Description	Setting range	Default
Device Detailed Setting	Global Device Setting	Set global devices to be transferred. (☞ Page 414 Global device setting)	—	0.0K Word
	Local Device Setting	Set whether to transfer local devices or not. When "Transfer" is set, all the local devices are transferred. For the devices that can be set as local devices, refer to the following. ☞ Page 407 Devices that can be specified	• Do Not Transfer • Transfer	Do Not Transfer
Label Detailed Setting	Global Label Setting*1	Set whether to transfer global labels or not. When "Transfer" is set, all the global labels assigned to the device/label memory are transferred.	• Do Not Transfer • Transfer	Do Not Transfer
	Local Label Setting	Set whether to transfer local labels or not. When "Transfer" is set, all the local labels are transferred.	• Do Not Transfer • Transfer	Do Not Transfer
Block Size Setting		Calculates "Block Size Setting" of each tracking block and "Total" of the setting capacity.	—	—

*1 Global labels with devices assigned are not transferred as global labels. They are transferred according to the tracking settings of the assigned devices.

Point

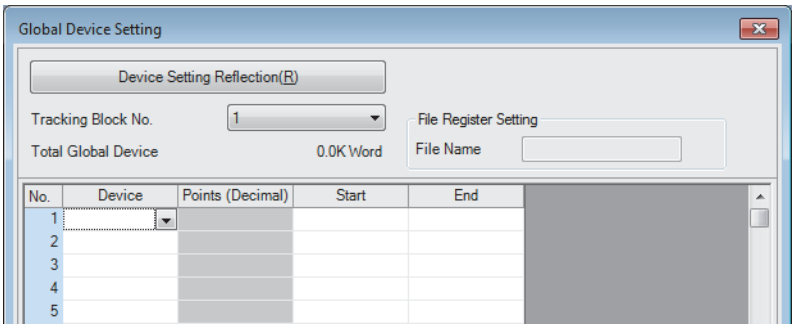
When setting devices and labels for tracking transfer, click the [Size Calculation] button to check if the tracking data capacity in one scan is equal to or less than 1M words.

■Global device setting

Set devices and their ranges for each tracking block No.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Tracking Setting] ⇨ [Device/Label Detailed Setting] ⇨ [Global Device Setting]

Window



Displayed items

Item	Description
Device Setting Reflection	Reflects the device setting of "Device/Label Memory Area Setting" in the CPU parameter. (Except for the annunciator (F), link special relay (SB), and link special register (SW))
Tracking Block No.	Select the number of a tracking block to be set.
File Register Setting	Enter a file name of the file register file. This item is valid only when the file register (ZR) is selected in the device name field.
Device	Select a device to be transferred. (☞ Page 407 Devices that can be specified)
Start/End	Specify a range of devices to be transferred.

Point

To input devices and their ranges set in "Device/Label Memory Area Setting" in a batch, click the [Device Setting Reflection] button.

Tracking mode

The following two modes are available for tracking.

Item	Description
Synchronous tracking mode	Tracking data is always transferred to the standby system once every scan of the control system. During a tracking transfer from the control system to the standby system, the next scan does not start in the control system.
Asynchronous tracking mode	When a tracking tansfer from the control system is to be performed and the previous tracking is still in progress, the tracking transfer from the control system is canceled and the previous tracking continues. The control system starts the next scan without waiting for notifications of data reception/reflection completion from the standby system.

The tracking mode depends on the operation mode and the CPU module operating status.

Operating status of standby system and control system		Operation mode	
Control system	Standby system	Backup mode	Separate mode
RUN	RUN	Synchronous tracking mode*1	Asynchronous tracking mode
	STOP, PAUSE	Asynchronous tracking mode	
STOP, PAUSE	RUN		
	STOP, PAUSE		

*1 When the RUN/STOP/RESET switch of the CPU module of each system is set to the RUN position and both systems are powered on, data is transferred in the asynchronous tracking mode at first. After the tracking data is reflected to the standby system, the mode is switched to the synchronous tracking mode.

In the following conditions, the tracking mode is the asynchronous tracking mode.

- During an online change
- When the operation mode is switched
- When the systems are switched
- When tracking communications disabled is detected

Effect on the scan time

The following describes the effect on the scan time depending on the tracking mode.

Point

For the calculation method for an increase in the scan time due to tracking transfer, refer to the following.

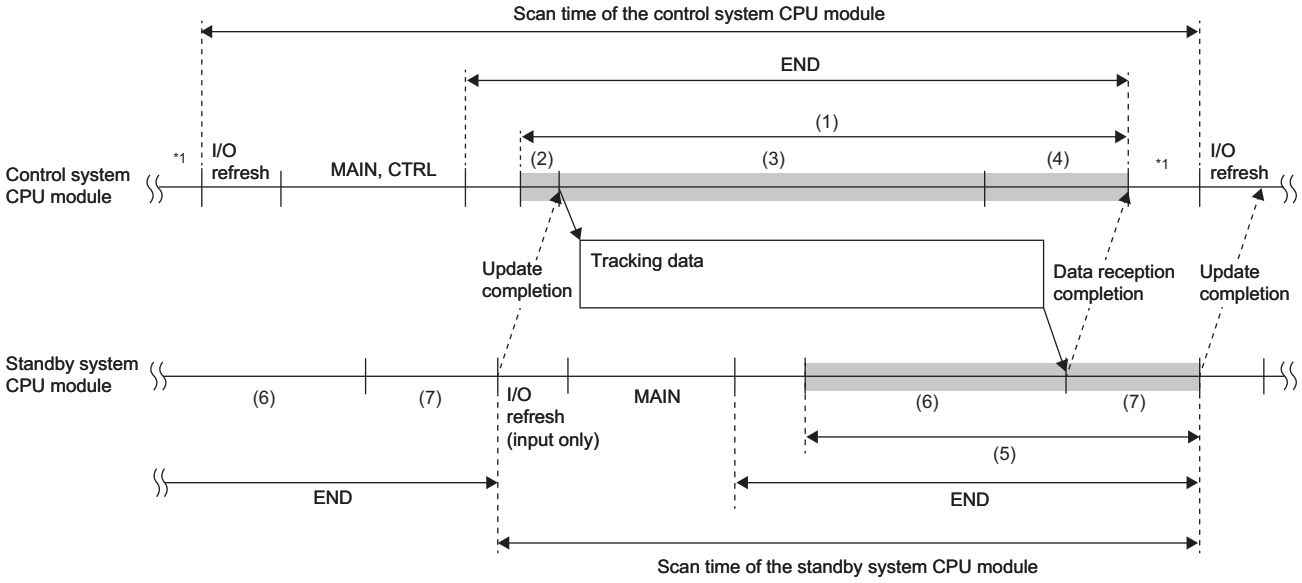
☞ Page 714 Increase in the scan time due to tracking transfer

■Synchronous tracking mode

In the synchronous tracking mode, tracking transfer is always performed once every scan during the END processing. Until the standby system receives the tracking data sent from the control system, the next scan is not started. Thus, the scan time of the control system and standby system increases by the time for tracking data send/receive processing.

Ex.

CTRL: Program executed in the control system, MAIN: Program executed in both systems



*1 If the constant scan is used, waiting time for the constant scan generated.

In the control system, the scan time is extended by the following tracking send processing time.

Item		Description
(1) Tracking send processing	(2) Waiting for completion of tracking data reflection	After receiving a notification of the reflection completion from the standby system, the control system sends the tracking data.
	(3) Sending tracking data	The control system sends the tracking data.
	(4) Waiting for completion of tracking data reception	The control system waits for a notification of receive completion from the standby system. After receiving the notification of the receive completion from the standby system, the control system starts another END processing.

In the standby system, the scan time increases by the following tracking receive processing time.

Item		Description
(5) Tracking receive processing	(6) Waiting for tracking data reception	The standby system receives the tracking data from the control system. After receiving the tracking data, the standby system notifies the control system of the receive completion and reflects the tracking data.
	(7) Reflecting tracking data	The standby system reflects the tracking data. After the reflection completion, the standby system notifies the control system of the reflection completion and starts another END processing.

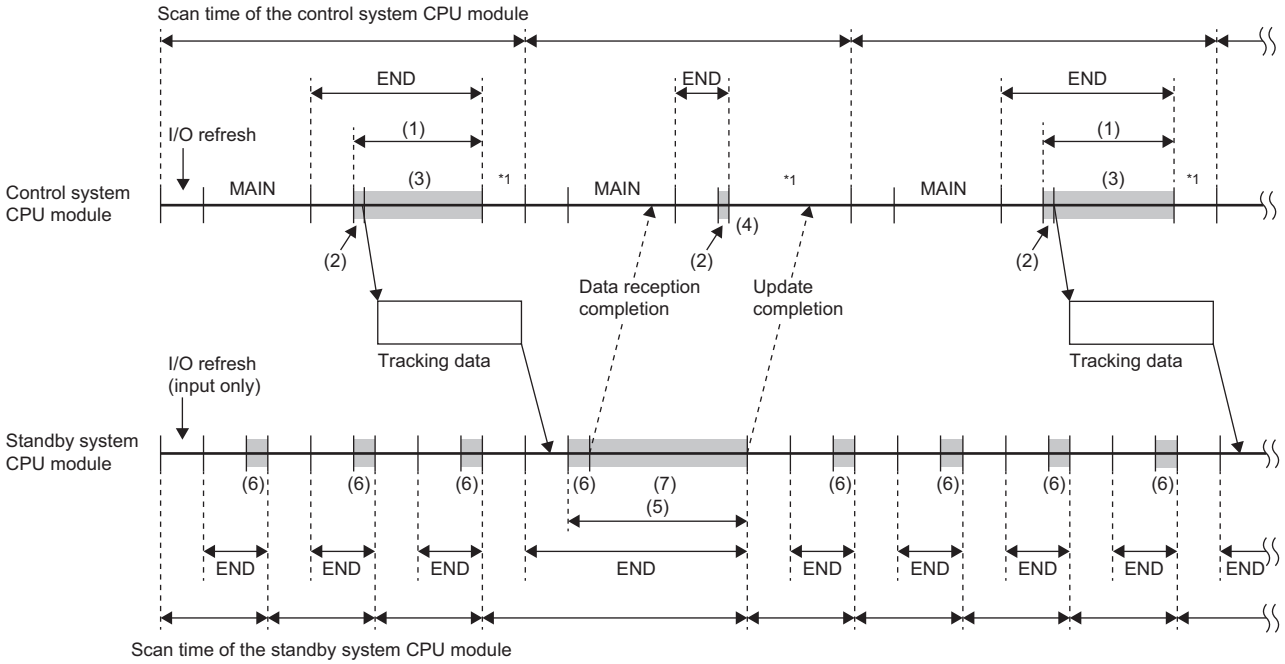
■Asynchronous tracking mode

In the asynchronous tracking mode, the control system starts the next scan without waiting for notifications of data reception/reflection completion from the standby system.

Unlike the synchronous tracking mode, the scan time is not affected by waiting time for data reception/reflection completion. When the standby system does not receive the tracking data from the control system, the standby system starts the next scan.

Ex.

When the control system CPU module is in the RUN state and the standby system CPU module is in the STOP state



*1 If the constant scan is used, waiting time for the constant scan generated.

In the control system, the scan time is extended by the following tracking send processing time.

Item		Description
(1) Tracking send processing	(2) Waiting for completion of tracking data reflection	After receiving a notification of the reflection completion from the standby system, the control system sends the tracking data. When a notification of reflection completion is not arrived (4), the control system does not send tracking data in the scan.
	(3) Sending tracking data	The control system sends the tracking data. After completing the send, the control system starts another END processing without waiting for a notification of receive completion from the standby system.

In the standby system, the scan time increases by the following tracking receive processing time.

Item		Description
(5) Tracking receive processing	(6) Waiting for tracking data reception	The standby system receives the tracking data from the control system. When the standby system does not receive the tracking data, the standby system starts the next scan. After receiving the tracking data, the standby system notifies the control system of the receive completion and reflects the tracking data.
	(7) Reflecting tracking data	The standby system reflects the tracking data. After the reflection completion, the standby system notifies the control system of the reflection completion and starts the next scan.

■When the mode is is switched from the asynchronous tracking mode to the synchronous tracking mode

When the mode is is switched from the asynchronous tracking mode to the synchronous tracking mode, the standby system receives the tracking data twice in one scan. Therefore, the scan time of the standby system is extended by the following time.

Scan time of the standby system: Standby system scan time × 2 + Control system scan time

Precautions

Operation at power-on

When the RUN/STOP/RESET switch of the CPU module of each system is set to the RUN position and both systems are powered on, the control system CPU module starts in the STOP state and switched to the RUN state after reflecting the tracking data is completed in the standby system CPU module. (The same operation is performed when one system is powered on while the other system is waiting for the start-up of the other system.)
Since the operating status of the control system and that of the standby system CPU modules do not march until the control system is switched from the STOP state to RUN state, the BACKUP LED flashes.
When the time of the initial processing and that of the initial processing (when switched to RUN) of the standby system CPU module are longer than those of the control system CPU module, the control system CPU module may not be immediately switched to the RUN state after power-on.

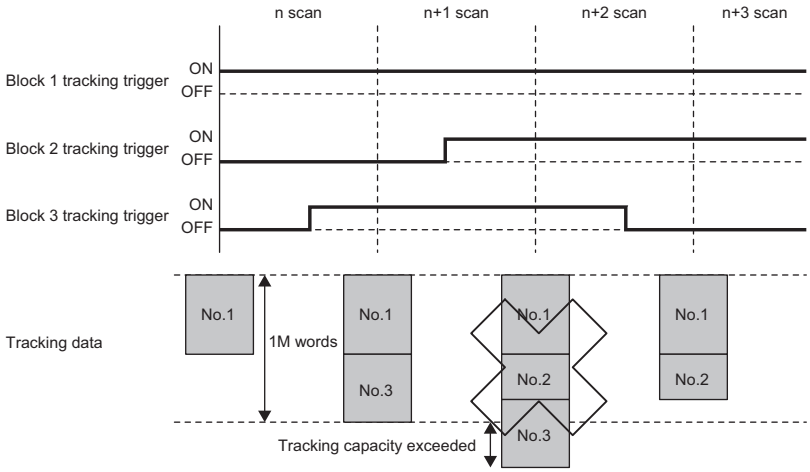
Device data used by the new control system CPU module

After system switching, the new control system CPU module starts operations by using the device/label data received in tracking transfer. The following table shows device/label data used by the new control system CPU module by tracking data receive status at system switching.

Device/label initial value	No tracking data received	Tracking data received once at least
Not set	Operation starts based on the latched device/label data.	Operation starts based on the device/label data that is transferred from the old control system. When the synchronous tracking mode is used as a tracking mode, the data in the old control system at the final scan start is used by the new control system.
Set	Operation starts based on the device/label data that is set with device/label initial values.	

Data capacity for tracking transfer

Up to 1M words of device/label data can be transferred in one scan. Set the tracking data capacity within 1M words.
If the data capacity exceeds 1M words, global devices, local devices, global labels, and local labels are not transferred in the scan. In this case, check which tracking trigger turned on in the event history of the engineering tool and set the tracking data capacity within 1M words.



Even though the size of the tracking data to be transferred is 1M words or less, the size of the data may become larger than 1M words depending on the label type or data type to be used after the data or all the data is converted with the engineering tool. When configuring "Tracking Setting" with the CPU parameter, click the [Size Calculation] button in "Detailed setting" of "Device/Label Detailed Setting" to check if the size of the tracking data 1M words or less and transferred in one scan. (☞ Page 413 Detailed setting)
To reduce the size of the tracking data, consider the following.

- Exclude devices/labels that are not required to continue the system operation.
- Divide the tracking data into multiple blocks to transfer the data in multiple scans.

When data is different between the control system and the standby system

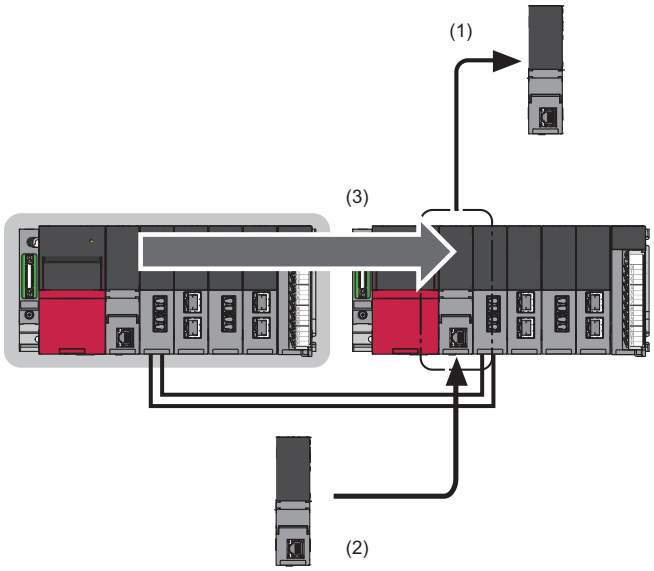
Store the same program, FB file, CPU parameter, and global label setting file in the control system CPU module and the standby system CPU module for tracking transfer. (☞ Page 420 Memory Copy from Control System to Standby System)
If there is any difference, only global devices are transferred.

38.4 Memory Copy from Control System to Standby System

This function transfers data such as parameters and programs in the CPU module of the control system to the CPU module of the standby system to maintain the consistency of the memory in the two CPU modules.

Ex.

Replacement of the standby system CPU module using the memory copy



(1) Remove the standby system CPU module.
(2) Mount a new CPU module.
(3) Execute the memory copy to match the memory contents of the control system CPU module and the standby system CPU module.

The following memories are available for the memory copy.

- Program memory
- Device/label memory (copying only the file area of file register)
- Data memory (copying only the files in the system folder ($MELPRJ$))
- SD memory card (copying only the files in the system folder ($MELPRJ$))

The memory copy is executed on memories that have differences between the control system and the standby system. It is not executed on memories without difference between the systems.

Point

For the module replacement method and maintenance procedure using the memory copy, refer to the following.

☞ Page 466 Module Replacement in the Redundant System

Files copied by the memory copy function

The memory copy function copies files required for starting a redundant system.

The following table lists the files to be copied by the memory copy function.

○: Memory copy possible, ×: Memory copy not possible, —: Storage not possible

File type		Copy			
		CPU built-in memory			SD memory card
		Program memory	Device/label memory	Data memory	
Program		○	—	○	○
FB file		○	—	○	○
CPU parameter		—	—	○	○
System parameter		—	—	○	○
Module parameter		—	—	○	○
Module extension parameter		—	—	○	○
Module-specific backup parameter		—	—	×	×
Memory card parameter		—	—	—	○
Device comment		—	—	○	○
Initial device value		—	—	○	○
Global label setting file		—	—	○	○
Initial label value file	Initial global label value file	—	—	○	○
	Initial local label value file	—	—	○	○
File register		—	○	—	×
Event history		—	—	×	×
Device data storage file		—	—	○	×
General-purpose data		—	—	×	×
Data logging setting file	Common setting file	—	—	—	○
	Individual setting file	—	—	○	○
Remote password		—	—	○	○

Point

When the memory copy is executed, files are copied after the memory of the standby system that has differences is initialized. (☞ Page 427 Automatic memory initialization)

The memory copy cannot be executed to the files such as event history and those files are deleted from the memory of the standby system CPU module. Back up the files before executing the memory copy as required.

■When the security key authentication function is used

The security key of the CPU module is not copied to the standby system. Write the security key from the personal computer where the security key is registered again after completion of the memory copy.

If the memory copy is executed when the security key is written in the CPU module, program files and their security keys are copied but the security key of the CPU module is not copied. Starting up the CPU module without re-writing of the security key causes an error of the CPU module because the security keys do not match between those program files and the CPU module.

Point

If the security key of the CPU module is written in an extended SRAM cassette, the replaced CPU module can take over the security key by simply replacing the extended SRAM cassette at replacement of the CPU module. In this case, re-writing of the security key after completion of the memory copy is not required.

Execution method of memory copy

The following methods are available for memory copy.

Item	Description	Application
Automatic memory copy	Automatically executes the memory copy by the system. CPU parameter settings are required in advance.	Executing the memory copy without using an engineering tool or external devices (such as a GOT)
Memory copy with the engineering tool	Executes the memory copy with an online operation of the engineering tool that is connected to the standby system CPU module.	Using the engineering tool
Memory copy with the special relay and special register	Executes the memory copy with operations of the special relay and special register.	Executing the memory copy from external devices (such as a GOT)

Execution of memory copy

The memory copy can be executed regardless of the operating statuses (RUN/STOP/PAUSE, stop error) of the control system and standby system CPU modules.

The automatic memory copy can be executed only in backup mode.

Automatic memory copy

The following describes the execution procedure of automatic memory copy.
To execute the automatic memory copy, turn off and on or reset the standby system CPU module while the control system is operating.

Point

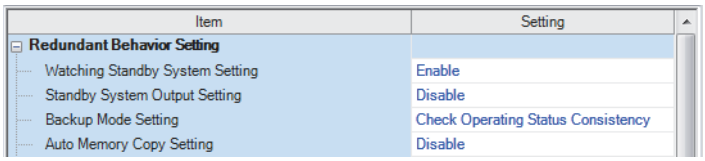
- Set the automatic memory copy in the CPU parameter in advance. Therefore, consider the use of automatic memory copy when determining the maintenance policy in system design.
- The automatic memory copy is executed when the parameter setting has been configured for the control system CPU module. Even if the parameter setting has not been configured for the standby system CPU module, the automatic memory copy is executed when the parameter setting has been configured for the control system CPU module.

Parameter

To execute the automatic memory copy, set "Auto Memory Copy Setting" to "Enable" in the CPU parameter.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Redundant Behavior Setting]

Window



Displayed items

Item	Description	Setting range	Default
Auto Memory Copy Setting	Set whether to enable the automatic memory copy function or not.	• Disable • Enable	Disable

Restriction

- When the control system power supply and the standby system power supply are simultaneously turned on, the automatic memory copy is not executed.
- The standby system CPU module does not operate in boot operation using an SD memory card, but operates based on the files copied by the automatic memory copy function. However, when the automatic memory copy cannot be executed due to a communication error or other causes at power-on or reset of the standby system CPU module, the standby system CPU module operates with the files transferred by the boot operation.

Execution procedure

1. When the operation mode is the separate mode, switch it to the backup mode. (☞ Page 389 Operation Mode Change)
2. Turn off and on or reset the standby system CPU module. The system executes the memory copy. During the memory copy, the MEMORY COPY LEDs of the redundant function modules in both systems flash (at 200ms intervals).

Control system: MEMORY COPY

Standby system: MEMORY COPY

3. The standby system CPU module is automatically reset and restarted, and the memory copy is completed. The MEMORY COPY LEDs of both systems turn off.

Control system: MEMORY COPY

Standby system: MEMORY COPY

Memory copy with the engineering tool

The following describes the execution procedure of memory copy using the engineering tool.

Execution procedure

1. Connect the engineering tool to the control system CPU module.
2. Open the "Redundant Operation" window of the engineering tool.

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

3. Select "Memory Copy" in the "Redundant Operation" window and click [Execute]. During the memory copy, the MEMORY COPY LEDs of the redundant function modules in both systems flash (at 200ms intervals).

Control system

Standby system

4. When the memory copy is completed, the MEMORY COPY LED of the control system turns off and that of the standby system turns on.

Control system

Standby system

5. Turn off and on or reset the standby system CPU module. The MEMORY COPY LED of the standby system turns off.

Control system

Standby system

Memory copy with the special relay and special register

The following describes the execution procedure of memory copy using the special relay and special register.

Special relay to be used

The following table lists the special relay areas used in memory copy.

SM number	Name
SM1653	Memory copy start
SM1654	Memory copy being executed
SM1655	Memory copy completion
SM1656	Auto memory copy

For details on the special relay (SM), refer to the following.

☞ Page 606 List of Special Relay Areas

Special register to be used

The following table lists the special register used in memory copy.

SD number	Name
SD988*1	Memory copy completion status (latch)
SD1653	Memory copy destination I/O number
SD1654	Memory copy completion status

*1 Even after the CPU module is turned off and on or reset, the result of the memory copy executed just before this reset operation can be checked since this register area is a latch area.

For details on the special register (SD), refer to the following.

☞ Page 635 List of Special Register Areas

Execution procedure

To execute the memory copy with the special relay and special register, use the special relay and special register of the standby system CPU module.

1. Check that SM1654 (Memory copy being executed) and SM1655 (Memory copy completion) are off.

- When SM1654 is on, the memory copy is being executed. Another memory copy cannot be started until the memory copy is completed.
- When SM1655 is on, the memory copy with the special relay and special register cannot be started. Execute the memory copy again after turning off SM1655.

2. Store 03D1H (Standby system CPU module) in SD1653 (Memory copy destination I/O number).

3. Turn on SM1653 (Memory copy start) to execute the memory copy. During the memory copy, the MEMORY COPY LEDs of the redundant function modules in both systems flash (at 200ms intervals).

Control system

Standby system

4. When the memory copy is completed, the MEMORY COPY LED of the control system turns off and that of the standby system turns on.

Control system

Standby system

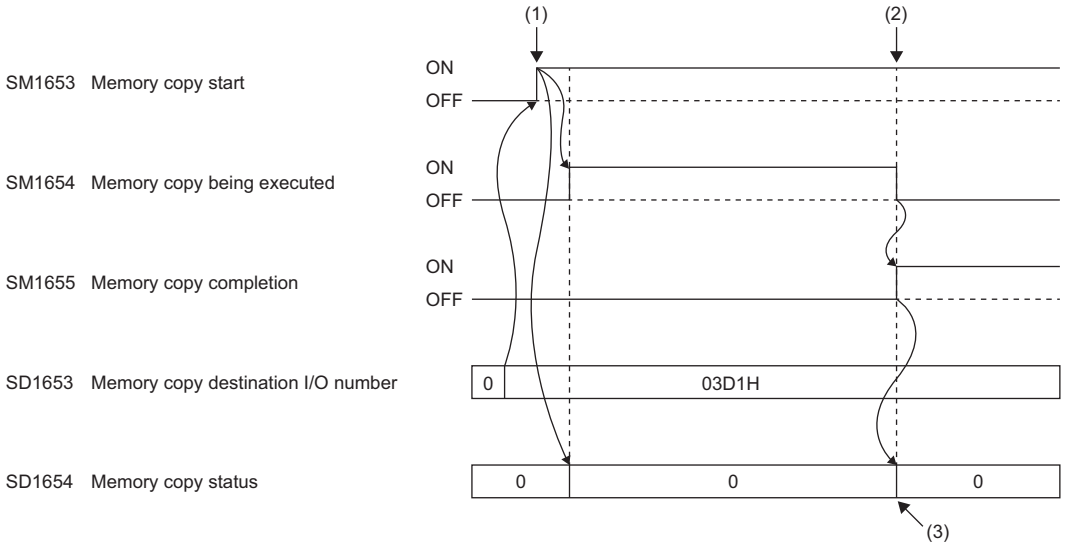
5. Turn off and on or reset the standby system CPU module. The MEMORY COPY LED of the standby system turns off.

Control system

Standby system

Operation of the special relay and special register

The following chart shows the operation of the special relay and special register in memory copy.



(1) The memory copy is started.
(2) The memory copy is completed.
(3) When the memory copy is successfully completed, 0 is stored. When it is completed with an error, an error code is stored.

Precautions

The following describes precautions on the memory copy function.

Automatic memory initialization

If there are differences in the data memory or an SD memory card, the "$MELPRJ$" folder is deleted from the data memory or SD memory card, and the memory copy is executed.
If there are differences in the program memory or device/label memory, the memory is initialized, and the memory copy is executed.
Thus, the ERROR LED of the standby system CPU module may flash when the memory copy with the engineering tool or the memory copy with the special relay and special register is executed. This phenomenon is not an error. Follow the procedure to turn off and on or reset the standby system CPU module.
Also, back up the data that is not supported by the memory copy and is stored in the "$MELPRJ$" folder, such as event history, before executing the memory copy.

Restrictions

■Conditions in which memory copy cannot be executed

In the following conditions, the memory copy cannot be executed.

- The standby system CPU module is off or being reset.
- During initial processing (when switched to RUN)
- The tracking cable has an error or is pulled out.
- An error occurs on a redundant function module.
- A system configuration in which communications via a redundant function module are unavailable is used.
- Different models of CPU module are used in the control system and standby system.
- A memory is being initialized.
- An online change is being performed.
- The operation mode is being switched.
- The systems are being switched.
- An online module change is being performed (for replacing a redundant function module or modules of the standby system).
- In separate mode (for the automatic memory copy)
- The memory copy has been executed on the standby system.
- The memory copy is being executed.
- A value other than 03D1H is stored in SD1653 (for the memory copy with the special relay and special register).

■Functions that cannot be executed during memory copy

The following functions cannot be executed while the memory copy is being executed.

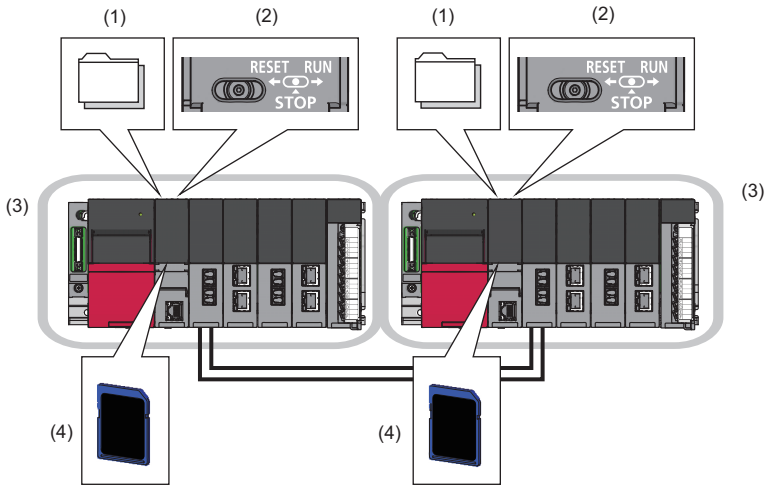
- Online operation from the engineering tool
- Operation of the RUN/STOP/RESET switch
- Removal of the SD memory card
- Online module change
- Powering off or resetting the CPU modules
- Removing tracking cables

Errors during memory copy

When the memory copy is completed with an error, the MEMORY COPY LED of the control system turns off and that of the standby system flashes (at 1s intervals). In this case, the memory copy has not been normally executed on the standby system CPU module.
After checking the error code of memory copy stored in SD1654 (Memory copy completion status) and eliminating the error cause, execute the memory copy again. (☞ Page 635 List of Special Register Areas)

38.5 System Consistency Check

This function checks whether the system configurations and files in the CPU modules are the same between the control system and the standby system when the redundant system is in backup mode.



The following table lists the items to be checked in the system consistency check.

No.	Item	Description	Reference
(1)	File	Parameter file, program file, and other files	Page 430 File
(2)	Operating status	Operating status of CPU module (RUN/STOP/PAUSE)	Page 431 Operating status
(3)	Mounting status of main base unit	CPU module and other modules mounted on the main base unit	Page 431 Mounting status of main base unit
(4)	SD memory card	Installing status of the SD memory card and status of the write protect switch	Page 431 SD memory card

Execution timing

The following table shows the execution timing of the system consistency check.

Item	Execution timing
File	• When both systems are simultaneously turned on or reset • When one system is turned on or reset while waiting for the start-up of the other system • When the standby system is turned on or reset while the control system is operating • When the operation mode is switched to the backup mode • When the operating status of the CPU module is switched from STOP to RUN • At END processing • At online change completion • At system switching • When a tracking cable is reconnected
Operating status*1	• When the standby system is turned on or reset while the control system is operating • When the operation mode is switched to the backup mode • When the operating status of the CPU module is switched from STOP to RUN • At END processing • When a tracking cable is reconnected
Mounting status of main base unit	• When both systems are simultaneously turned on or reset • When one system is turned on or reset while waiting for the start-up of the other system • When the standby system is turned on or reset while the control system is operating • At mode switching to the backup mode • At reconnection of tracking cable
SD memory card	• When both systems are simultaneously turned on or reset • When one system is turned on or reset while waiting for the start-up of the other system • When the standby system is turned on or reset while the control system is operating

*1 Checking this item can be disabled by setting "Backup Mode Setting" of "Redundant Behavior Setting". (☞ Page 439 Redundant System Operation Setting)

In the following conditions, the system consistency check is not performed.

- One of the systems has not started up. (The CPU module is powered off or is reset or during an initial processing.)
- A stop error has occurred in the CPU module.
- The operation mode is the separate mode.
- The memory copy is being executed.

File

Whether both systems have the same files is checked.

The following table shows whether or not to perform the check on each file type.

○: Checked, ×: Not checked, —: Storage not possible

File type		Check target memory	
		Built-in memory of CPU module	SD memory card[*3]
Program[*1]		○	×
FB file[*1]		○	×
CPU parameter		○	×
System parameter		○	×
Module parameter		○	×
Module extension parameter		○	○
Module-specific backup parameter		×	×
Memory card parameter		—	○
Device comment		×	×
Initial device value		○	○
Global label setting file		○	×
Initial label value file	Initial global label value file	○	○
	Initial local label value file	○	○
File register[*2]		○	×
Event history		×	×
Device data storage file		×	×
General-purpose data		×	×
Data logging setting file	Common setting file	—	×
	Individual setting file	×	×
Remote password		○	×

*1 "Allocate Memory for Online Program Change" is also checked. (☞ Page 85 Configuration of a program file)

*2 Whether a file is stored or not is checked. The file contents are not checked.

*3 The file will not be checked at the timing on when SD memory cards are inserted during operation.

Operation of when a mismatch is detected

If a mismatch between the files is detected, a stop error occurs on the standby system CPU module.

Set the CPU modules of both systems to have the same files with either of the following methods.

- Writing the correct project to the standby system (☞ Page 385 Writing Data to the Programmable Controller)
- Executing the memory copy from the control system to the standby system for storing the same files in both systems (☞ Page 420 Memory Copy from Control System to Standby System)

Operating status

Whether the CPU modules of the control system and standby system are in the same operating status (RUN/STOP/PAUSE) is checked.

Operation of when a mismatch is detected

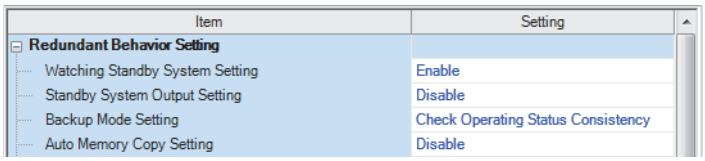
If a mismatch between the operating statuses is detected, a continuation error occurs on the standby system CPU module. The BACKUP LEDs of the redundant function modules of the control system and standby system flash because a cause of the system switching failure has occurred.

Backup mode setting

To prevent a continuation error from being detected when the operating status is changed during operation, set the system consistency check so that the consistency check on the operating status is not performed.
With this setting, a continuation error is not detected even if the CPU modules of both systems are in different statuses.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Redundant Behavior Setting]

Window



Displayed items

Item	Description	Setting range	Default
Backup Mode Setting	Set whether to perform the consistency check for the operating statuses of both systems or not in backup mode.	• Do Not Check Operating Status Consistency • Check Operating Status Consistency	Check Operating Status Consistency

Mounting status of main base unit

The system consistency check is performed for the following items.

- Models of CPU modules
- Models and types of modules mounted on each slot

Operation of when a mismatch is detected

If a mismatch between the mounting statuses of the main base units is detected, a stop error occurs on the standby system CPU module. If a mismatch between the mounting statuses of the main base units is detected when the CPU modules of both systems are simultaneously turned on or reset, a stop error occurs on the control system CPU module as well.
("Simultaneously" here means that one CPU module is started up within three seconds after the other CPU module is started.)

SD memory card

The system consistency check is performed to check the installation of the SD memory cards and the status of the write protect switch. The SD memory card type or capacity is not checked.

Point

The system checks if the SD memory card is inserted while the control system is running even in the case only the standby system is turned off and on or reset. When an SD memory card is used, removing it more than necessary is not recommended.

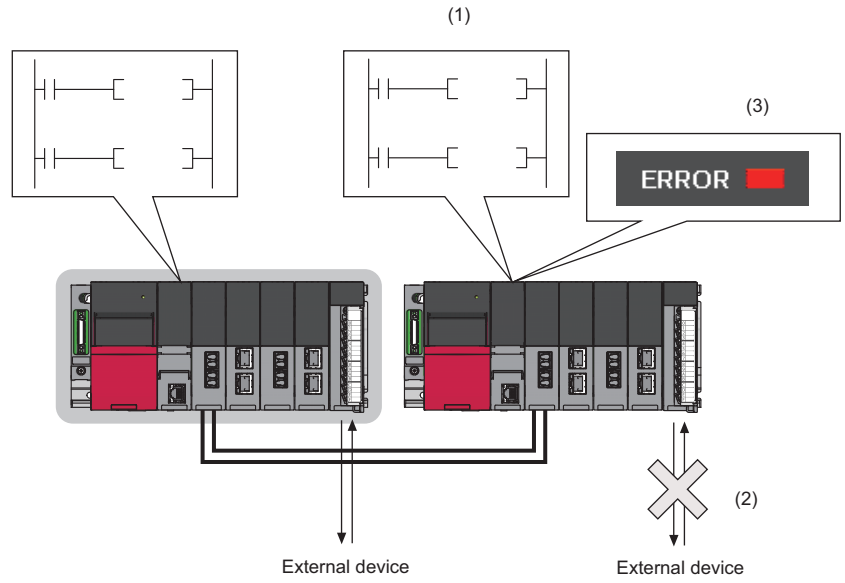
Operation of when a mismatch is detected

If a mismatch of the installation of the SD memory cards or the status of the write protect switch is detected, a stop error occurs on the standby system CPU module. If a mismatch between the mounting statuses of the main base units is detected when the CPU modules of both systems are simultaneously turned on or reset, a stop error occurs on the control system CPU module as well.

38.6 Program Execution in Both Systems

This function etects an error in the external device or network of the systems (control system and standby system) by executing a program that diagnoses external devices or networks of both systems.
The program set in "Both Systems Program Executions Setting" is executed on the CPU modules of both systems.

Ex.
When an error in the external device of the standby system is notified through a continuation error



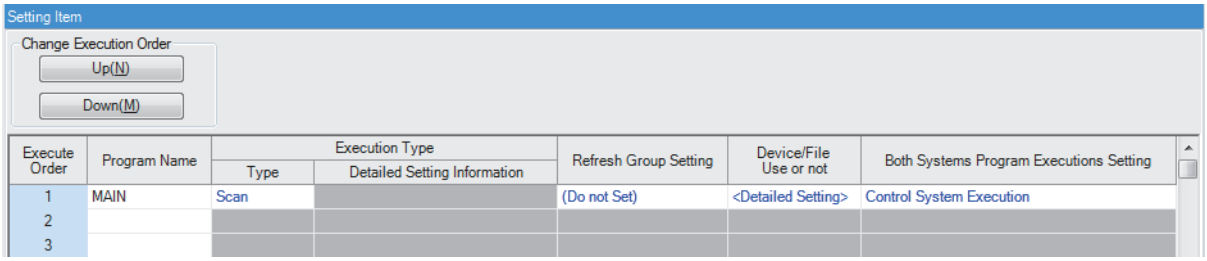
(1) The diagnostic program set as a both systems execution program is executed.
(2) An error is detected on the standby system external device by the diagnostic program.
(3) The PALERT instruction is executed, and the detailed information on a continuation error is displayed.

Setting of Program execution in both systems

Configure the setting for each program to be executed on both systems.

[CPU Parameter] ⇨ [Program Setting]

Window



Displayed items

Item	Description	Setting range	Default
Both Systems Program Executions Setting	Set whether to execute a program only on the control system CPU module or on the CPU modules of both systems. • "Both Systems Executions" can be set for programs whose execution type is the initial execution type, scan execution type, or standby type program. • Only "Control System Execution" can be set for programs whose execution type is the fixed scan execution type or event execution type.	• Control system execution • Both systems executions	Control System Execution

Point

To enable the output (Y) from a standby system external device using a program executed in both systems, configure settings in "Standby System Output Setting" of the CPU parameter. (☞ Page 439 Redundant System Operation Setting)

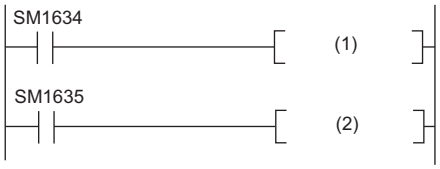
Operation of a program executed in both systems

The following describes the operation of a program executed in both systems.

Control system/ Standby system	Backup mode	Separate mode
Control system	A program is executed according to its execution type.	A program is executed according to its execution type regardless of the setting in "Both Systems Program Executions Setting".
Standby system	A program executed in both systems is executed according to its execution type. When an initial execution type program is set as a program executed in both systems, the program is executed in the first scan while the CPU module are in the RUN state. When an initial execution type program is not set as a program executed in both systems, a program executed in both systems of scan execution type is executed in the first scan while the CPU modules are in the RUN state.	

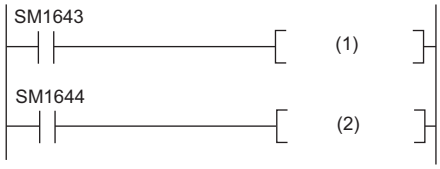
Point

- To perform different diagnostic processing in the control system and standby system by using a program executed in both systems, use the special relay. SM1634 (Control system judgment flag) turns on only in the control system and SM1635 (Standby system judgment flag) only in the standby system. The following shows a program example of when different processing is performed. For (1) and (2) in the following example, use the SET instruction or others to prevent duplication of a coil in the processing.



(1) Diagnostic processing in the control system, (2) Diagnostic processing in the standby system

- When a program executed in both systems is used, device/label values may differ between the control system and the standby system. If the systems are switched in this state, the program is started based on the different data, causing an unintended operation. The device/label values to be used can be initialized by using SM1643 (ON for only one scan after system switching (standby system to control system)) and SM1644 (ON for only one scan after system switching (control system to standby system)). For (1) and (2) in the following example, use the SET instruction or others to prevent duplication of a coil in the processing.



(1) Initialization processing (standby system → control system), (2) Initialization processing (control system → standby system)

Operation at system switching

The operation at system switching is different for a program executed in both systems. The following table shows the operation at system switching.

Item		New control system CPU module	New standby system CPU module
Program execution	Initial execution type program	When an initial execution type program has not been completed on the old control system at system switching, it is executed again from its head.	When an initial execution type program has not been completed on the old control system at system switching, it is not executed on the new standby system because system switching is performed after the program completion on the old control system.
	Scan execution type program	This type of program is executed from the step 0.	
Direct access input (DX)		In the program execution after system switching, data is fetched when an instruction using the direct access input (DX) is executed.	
Direct access output (DY)		Data is not output. However, data is output when "Standby System Output Setting" of the CPU parameter is enabled and an instruction using the direct access output (DY) is executed in the program execution after system switching. (☞ Page 439 Redundant System Operation Setting)	
FROM/TO instructions		In the program execution after system switching, this instruction is executed when the instruction execution condition is satisfied.	
Instruction that requires several scans		• In the program execution after system switching, this instruction is executed when the instruction execution condition is satisfied. • When the systems are switched while the instruction is being executed, the instruction execution continues and the completion device turns on at a completion of the instruction execution.	

For the operation of programs which are not set as both systems execution programs at system switching, refer to the following.

☞ Page 397 Operation at system switching

Precautions

The following lists the precautions for using a program executed in both systems.

Item		Description	Reference
Program execution time		Set a program execution time of the standby system to be shorter than that of the control system.	Page 437 Program execution time
Constant scan		The constant scan function is invalid for the standby system in backup mode.	—
Time required for system switching		If system switching is attempted to be performed while a program executed in both systems is being executed, the system is switched after the END instruction is completed. Thus, a time required for system switching may be extended.	—
Program execution type	Execution type change	The execution type of a program executed in both systems is not taken over at system switching. Even if the systems are switched after the execution type is changed by using a program control instruction (PSCAN(P), PSTOP(P), or POFF(P) instruction) on the old control system, the program is executed on the new control system with the execution type of the old standby system.	—
	Initial execution type	When the systems are switched while an initial execution type program set as a both systems execution program is being executed, the initial execution type program is executed two times on the new control system CPU module. The old standby system is switched to the new control system after the initial execution type program has been completed, and then the initial execution type program is executed again.	—
Interrupt disabled or enabled state		The control system and standby system have individual interrupt disabled or enabled state because the states are not transferred.	—
Tracking transfer		Do not set global devices used in a program executed in both systems as a tracking transfer target. When using labels in a program executed in both systems, use local labels. When using FBs in a program executed in both systems, use local FBs.	Page 438 Tracking transfer
Device	Timer (T)	When the control system is switched to the standby system, the current value of the timer is not updated in the first scan of the new standby system and a timeout does not occur. As a result, an error of the time required for system switching + one scan is produced at system switching.	—
	Long timer (LT) and long retentive timer (LST)	When the long timer (LT) or long retentive timer (LST) is used in the standby system, time is not measured and a timeout does not occur. After the standby system has been switched to the control system, the long timer (LT) or long retentive timer (LST) is started up. To measure time in the standby system, use the timer (T).	—
	Interrupt pointer (I)	The interrupt pointer (I) cannot be used for the standby system in backup mode.	—
Restricted instructions		Some instructions have restrictions when they are used in a program executed in both systems.	Page 438 Restricted instructions
Standard function block	TIMER_□_M	When the control system is switched to the standby system, the current value of the timer is not updated in the first scan of the new standby system and a timeout does not occur. As a result, an error of the time required for system switching + one scan is produced at system switching.	—
	TP(_E), TON(_E), TOF(_E)	When these function blocks are used in the standby system, time is not measured and a timeout does not occur. After the standby system has been switched to the control system, TP(_E), TON(_E), or TOF(_E) are started up.	—
Ethernet	Socket communications	When data is sent to the Ethernet-equipped module of the standby system, the data receive processing is not performed because the Ethernet-equipped module discards the received data.	MELSEC iQ-R Ethernet User's Manual (Application)
	Communications using the fixed buffer		

Program execution time

Set a program execution time of the standby system to be shorter than that of the control system. When a program execution time of the standby system is longer than the total program execution time of the control system, the control system starts the next scan before the standby system has received tracking data, causing a continuation error. If the systems are switched in this state, the latest data may not be reflected on the new control system.

When a program execution time of the standby system cannot be shortened, set SD1662 (Tracking transfer data receive completion wait time) to extend a waiting time for tracking data receive completion of the control system. With this setting, the control system CPU module starts the next scan after the standby system has received tracking data. Even if the systems are switched, the control can be continued based on the latest data.

Tracking transfer

- Do not set global devices used in a program executed in both systems as a tracking transfer target. Due to tracking transfer, the standby system data is overwritten with that of the control system, and the standby system program may operate in an unintended way.
- When using a label in a program executed in both systems, use a local label.
- When using an FB in a program executed in both systems, use a local FB. If a global FB is used, a program execution result of the control system is transferred and written over a program execution result of the standby system, causing an unintended operation.

Restricted instructions

The following table lists the instructions that are restricted in a program executed in both systems.

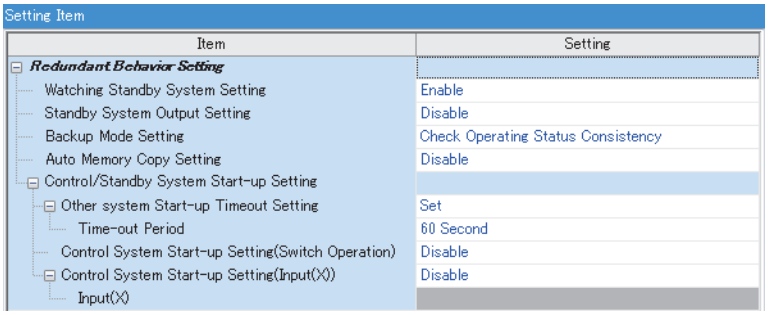
Classification	Notation	Description
Calling a subroutine program	CALL(P)	These instructions may not normally operate if a subroutine with any of a rising instruction, falling instruction, or SCJ instruction in a program which is not set as a both systems execution program is called in the standby system.
Calling a subroutine program and turning the output off	FCALL(P)	
Calling a subroutine program in the specified program file	ECALL(P)	
Calling a subroutine program in the specified program file and turning the output off	EFCALL(P)	
Calling a subroutine program with output off	XCALL	
Program control instruction	PSTOP(P)	No operation is performed if these instructions are executed on a program which is not set as a both systems execution program in the standby system.
	POFF(P)	
	PSCAN(P)	
Redundant system instruction	SP.CONTSW	No operation is performed if this instruction is executed in the standby system.
PID control instructions (Inexact differential)	S(P).PIDINIT	The PID control instruction information of the control system is transferred and written over the PID control instruction information of the standby system. When these instructions are executed in the standby system, a stop error may occur.
	S(P).PIDCONT	
	S(P).PIDSTOP	
	S(P).PIDRUN	
	S(P).PIDPRMW	
PID control instructions (Exact differential)	PIDINIT(P)	
	PIDCONT(P)	
	PIDSTOP(P)	
	PIDRUN(P)	
	PIDPRMW(P)	

38.7 Redundant System Operation Setting

Set the redundant system operation in the redundant settings of the CPU parameter.

[CPU Parameter] ⇨ [Redundant System Settings]

Window



Displayed items

Item		Description	Setting range	Default
Watching Standby System Setting		Set this item not to detect continuation errors when: • A Communication error with the other system has occurred or communications with the other system has been disabled. • The standby system CPU module is off, has been reset, or a stop error has occurred. This item is enabled only for the control system CPU module.	• Disable • Enable	Enable
Standby System Output Setting		Set this item to enable the output (Y) from the standby system for system inspection or adjustment in the standby system in backup mode. Even though this parameter setting has been set to "Disable" in separate mode, the output (Y) is enabled. • To diagnose devices that are connected to the standby system by using a both systems execution program, set "Enable". However, set the standby system output setting to "Disable" when common devices have been connected to both the control system and standby system. • When "Enable" is set, do not specify outputs (Y) to be used in the standby system as the data to be tracked. If such an operation is performed, the outputs (Y) are overwritten with the control system data and output from the standby system. (Even though the standby system CPU module is in the STOP state, the output (Y) received in tracking transfer is output.)	• Disable • Enable	Disable
Backup Mode Setting		☞ Page 431 Operating status		
Auto Memory Copy Setting		☞ Page 423 Automatic memory copy		
Control/Standby System Start-up Setting	Other system Start-up Timeout Setting	Set a timeout time taken until communications with the other system are enabled after own system is started up and the initial processing is completed. • When "Set" is selected, a stop error will occur in the own system unless communications are performed with the other system even though the timeout time has come. • When "Not Set" is selected, the own system will wait until communications with the other system are established.	• Not Set • Set	Set
	Time-out Period		3 to 1800s (in increments of 1s)	60s
	Control System Start-up Setting (Switch Operation)	Set this item to enable the operation to be started in the control system with the switch operation (RUN→STOP→RUN) while own system that has started is waiting for the start-up of the other system.	• Disable • Enable	Disable
	Control System Start-up Setting (Input (X))	Set this item to enable the operation to be started in the control system with the contact input (X) while own system that has started is waiting for the start-up of the other system.	• Disable • Enable	Disable
	Input (X)		X0 to X2FFF	—

Standby system output setting

Output timing

When "Standby System Output Setting" has been set to "Enable", the output timing from the standby system in backup mode is at the completion of the END processing or depends on the refresh group setting or refresh settings of each module. (☞ Page 53 Group setting for refresh)

Therefore, when a control system execution program is set in the refresh group setting, the I/O refresh is not performed while the module is running because the program is not executed in the standby system. (When the CPU module is the STOP/PAUSE state, the I/O refresh is performed at the timing of the END processing.)

To execute a program set in the refresh group setting in the standby system in backup mode, set "Both Systems Program Executions Setting". (☞ Page 433 Program Execution in Both Systems)

Operation at system switching

The operation performed at standby system differs when "Standby System Output Setting" is set to "Enable". The following table shows the operation performed when "Standby System Output Setting" has been set to "Enable".

Item	New control system CPU module	New standby system CPU module
Output (Y)	The status of the old standby system is held and output refresh is performed.	The status of the old control system is held and output refresh is performed.
Direct access output (DY)	In the program execution after system switching, data is output when an instruction using the direct access output (DY) is executed.	■For program executed in both systems In the program execution after system switching, data is output when an instruction using the direct access output (DY) is executed. ■For program executed in the control system No operation is performed because the program does not operate.

For the program execution in both systems, refer to the following.

☞ Page 433 Program Execution in Both Systems

38.8 Redundant Function Module Communication Test

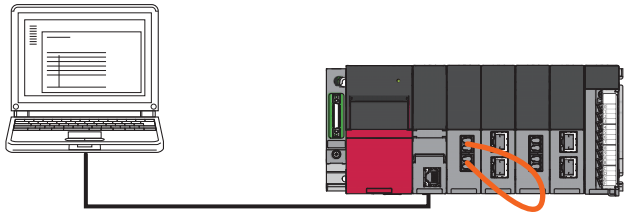
The hardware of the redundant function module is checked for an error when its communication is unstable.

The following table shows the test items included in the module communication test.

Test item	Description
Internal selfloopback test	Checks whether the communication function of the redundant function module normally operates.
External selfloopback test	Checks whether communications can be normally performed with a tracking cable that connects the IN connector and OUT connector of the redundant function modules.

Execution procedure of the module communication test

1. Change the operation mode to the separate mode. (☞ Page 389 Mode switching to the separate mode)
2. Connect the engineering tool to the standby system CPU module.
3. Set the CPU module operating status to the STOP state.
4. Connect the IN and OUT connectors of the standby system redundant function module with the tracking cable. (A cable disconnection error is detected in the control system.)



5. Open the "Redundant Operation" window of the engineering tool.

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

6. Click the [Execute Test] button of "Module Communication Test".

• LED status of the redundant function module during the module communication test

Status	RUN LED	ERR LED
Module communication test in execution	Flashing	Off
Completed successfully	On	Off
Completed with an error	On	On

7. When the test is completed with an error, take actions according to "Corrective Action" in the test result window.
8. When the test is completed successfully, connect the control system and standby system with the tracking cable. (☞ Page 380 Wiring the redundant function modules)
9. Click the [Close] button in the "Redundant Operation" window to exit the module communication test.
10. Set the CPU module operating status to the RUN state.
11. Change the operation mode to the backup mode. (☞ Page 390 Mode switching to the backup mode)

Precautions

- Always connect the IN and OUT connectors of the redundant function module with the tracking cable before performing the module communication test.
- Perform the module communication test on the standby system CPU module. Performing the test on the control system CPU module may cause an unintended operation.

38.9 Clock Function

This section describes the clock function of the redundant system.

Time synchronization

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

38.10 Online Change

When the online change (ladder block) is performed on the CPU module in one system in backup mode, the change is also reflected on the CPU module in the other system.
In separate mode, only the ladder block of the CPU module in the system specified in the transfer setup of the engineering tool is changed.

Precautions

During an online change, avoid the following conditions.

- Changing the operating status of the CPU module from STOP (PAUSE) to RUN
- System switching
- Operation mode switching (switching to the backup mode)
- The tracking cable has been removed.
- The control system or standby system CPU module has been off or reset.

When either of the above conditions is satisfied during an online change, a file mismatch is detected in the system consistency check or the online change fails.

■Action for when an online change has failed

If an online change has failed, take the following actions.

1. Get ready for an online change and execute the online change with the same data as before the failure.

- If the online change succeeds, the action is completed.
- If the online change fails again, take the following actions.

2. Verify the data in the control system CPU module with the programmable controller from the engineering tool, and check whether the online change has been completed successfully.

- When the verification result is mismatched, the online change has not been completed successfully. Connect the engineering tool to the control system CPU module, execute an online change only to the control system, and proceed to step 3.
- When the verification result is matched but the program memory transfer window was not displayed at the online change, the possible cause is that data transfer to the program memory has not been completed successfully. Batch-write files only to the control system, and proceed to step 3.
- When the verification result is matched and the program memory transfer window is displayed (the online change is completed successfully) during online change, proceed to step 3.

3. Execute the memory copy from the control system to the standby system. When the memory copy has failed, power off and on or reset the standby system CPU module and execute the memory copy again. (☞ Page 420 Memory Copy from Control System to Standby System)

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

38.11 Constant scan

The following describes the precautions for setting the constant scan in the redundant system.

Increase in scan time

In the standby system, when the CPU module is powered off, a hardware failure has occurred, or a tracking cable has a failure, the scan time will increase in the control system.

When setting the constant scan, take one of the following measures.

- Set the setting time of the constant scan by adding the increase in the scan time when an error occurs. (☞ Page 714 Increase in the scan time due to tracking transfer)
- When a failure has occurred and a continuation error occurs due to the excess of constant scan time, clear the error. (☞ Page 233 Error Clear)

System switching in separate mode

When the systems are switched in separate mode, the scam time of the control system will increase by the increase of the standby system scam time.

When the constant scan has been set, a continuation error may occur due to the excess of constant scan time because the scan time of the control system increases.

When a continuation error occurs due to the excess of constant scan time after the system switching in separate mode, clear the error.

For program executed in both systems

The constant scan function is invalid for the standby system in backup mode.

38.12 Remote Operation

In a redundant system, the operation target of remote operations (with the engineering tool) depends on the operation mode and method.

Operation mode	Remote operation	
	Remote RUN/Remote STOP/Remote PAUSE	Remote RESET
Backup mode	The CPU module operating status of a system specified in the transfer setup of the engineering tool or both systems can be changed.	The CPU modules of both systems can be reset by performing the remote RESET operation on the control system CPU module. Only the standby system CPU module can be reset by performing the remote RESET operation on the standby system CPU module.
Separate mode	The CPU module operating status of a system specified in the transfer setup of the engineering tool can be changed.	

Remote RUN/Remote STOP/Remote PAUSE

The following describes the remote RUN, remote STOP, and remote PAUSE operations in a redundant system.

In backup mode

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

■Remote operation performed on the CPU module of a system specified in the transfer setup of the engineering tool

When any of the following items is selected in "Execution Target", the remote operation is performed only on the CPU module of a system specified in the transfer setup of the engineering tool.

- Currently Specified Station
- All Stations Specified
- Specify Group No.

[Online] ⇨ [Remote Operation]

■Remote operation performed on both systems

When the following item is selected in "Specify Execution Target", the remote operation is performed on the CPU modules of both systems.

- Specify Both Systems

[Online] ⇨ [Remote Operation]

Point

When "Specify Redundant CPU" is set to "Not Specified" in the transfer setup, select "All Stations Specified" or "Specified Group" to perform a remote operation on the CPU modules of both systems.

In separate mode

The CPU module operating status of a system specified in the transfer setup of the engineering tool can be changed. (☞ Page 443 Remote operation performed on the CPU module of a system specified in the transfer setup of the engineering tool)

Remote RESET

The following describes the remote RESET operation in a redundant system.

In backup mode

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

Select "Control System" or "Standby System" in the transfer setup, and select "Currently Specified Station/Specify Both Systems" in "Specify Execution Target" of "Remote Operation".

■Precautions

The following describes the precautions on the remote RESET operation performed on both systems in backup mode.

- When the control system CPU module is in the STOP state and the standby system CPU module is in the RUN state, performing the remote RESET operation on the control system CPU module causes system switching. To prevent system switching in the remote RESET operation, perform the remote RESET operation after setting both of the CPU modules to the STOP state.
- When a remote operation is performed on the CPU module of the control system or standby system through another path, the standby system CPU module cannot be reset by performing a remote RESET operation on the control system CPU module. To perform a remote RESET operation on the CPU module of the control system or standby system, use the path that was used to perform the remote operation on the standby system CPU module. Cancel the remote operation first and perform the remote RESET operation.
- When performing a remote RESET operation on both systems, select "Control System" in "Specify Redundant CPU". When "Not Specified" is selected, an error may occur depending on the timing of the operating status change of each CPU module.

[Online] ⇨ [Current Connection Destination] ⇨ [Specify Redundant CPU]

In separate mode

Only the CPU module of a system specified in the transfer setup of the engineering tool can be reset. The operating status of the CPU module in a system not specified is not changed.

Precautions

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

- When the control system or standby system is in the initial processing (the READY LED is flashing), do not perform the remote operation of "Specify Both Systems". If such an operation is performed, both systems may be recognized as being mismatched in the system consistency check because the operating statuses may be mismatched.

38.13 Boot Operation

This section describes precautions on using the boot operation in the redundant system.

Boot operation at start-up of the redundant system

Use the boot operation only to simultaneously start up both systems.

1. Attach SD memory cards that have boot setting data to both systems.
2. Simultaneously turn on or reset both systems and perform the boot operation for them.

Do not use the boot operation to restart one system. If only one system is restarted with the boot operation, a system consistency check error may occur. The following operations are recommended.

- Remove the SD memory cards for boot operation after simultaneously starting up both systems.
- When using an SD memory card for other applications, such as the data logging function, prepare another one.

Online change in boot operation

It is recommended to remove SD memory cards for boot operation while the system is operating. Do not perform the online change (online change (ladder block)). To copy project data to an SD memory card for boot operation, set the CPU module in the STOP state and write the project data to the SD memory card.

38.14 SLMP communication

The following describes the precautions on SLMP communications.

System switching

There are the following notes when the system IP address matching function is not used.

■Re-setting of the connection destination

When the relay CPU module is in the communication-disabled state (power-off, reset, or tracking cable disconnection) at system switching, the connection destination needs to be set again for SLMP communications.

■Re-execution of the write command

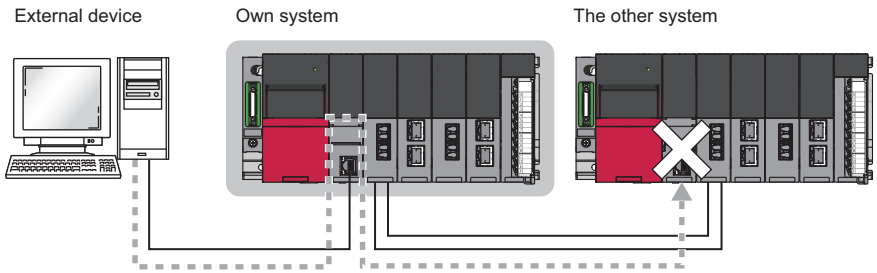
When "Control System" or "Standby System" has been selected in the transfer setup and the systems are switched, a target system mismatch is detected by a command issued during the system switching, and a communication error occurs. If a communication error occurs while a data write command is being issued, the data write command needs to be issued for the new control system.

Remote operation

If a remote operation command is executed, the CPU modules enter different operating statuses and thus systems cannot be switched.

Communicating with other systems

For SLMP communications via the built-in Ethernet port of the CPU module, when communications are performed to the other system that cannot respond (power-off, reset, or tracking cable disconnection), a timeout error may occur.



38.15 Data Logging Function

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

Point

For details on CPU Module Logging Configuration Tool used for the data logging function in a redundant system, refer to the following.

☞ Page 726 How to Use CPU Module Logging Configuration Tool

Data logging states

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

Data logging state	Description
Stop	No data logging settings are registered and data collection is inactive.
Stop (Execute other functions after completing collection)	Transition from "Collection completed" to "Stop" has occurred due to the execution of another function*1.
Stop (Execute other functions after error)	Transition from "Error" to "Stop" has occurred due to the execution of another function*1.
Waiting RUN Not collected	• Data collection has not yet been started because the CPU module is not in the RUN state. • After the operation mode was switched to the separate mode, the standby system CPU module is waiting for the transition to the RUN state.
Waiting start Not collected	Data collection has not yet been started and the CPU module is waiting for the start operation.
Pause	Data logging is suspended and data collection has not yet been started.
Waiting to establish collection conditions Not collected	Waiting for the first collection timing after the start operation
Collecting	Continuous logging is active and data collection is being performed.
Waiting trigger Collecting before trigger	Trigger logging is active, data collection is being performed, and waiting until the trigger condition is met
Collecting after trigger	Trigger logging is active and data collection is being performed after the trigger condition is met
Collection completed	• Continuous logging: Data collection has finished upon reaching "Number of files to be saved" specified as part of the "Stop" setting configured in "Operation when the number of files exceeds the set value". • Trigger logging: Trigger logging has finished collecting data as much as the specified number of records.
Error	Data logging has failed due to the occurrence of an error.
Standby system start waiting	Data collection has not been performed, waiting for the start operation in the standby system.
Standby system pause	Data logging is suspended in the standby system.
Standby system no collection	Data logging is started in the standby system, but data collection is inactive.

*1 The execution of another function includes:
- Execution of data logging with the same trigger conditions (Trigger conditions = Specified conditions)
- Auto logging
- Online change

Data logging states at system switching

The following table lists the data logging states that change at system switching.

■When the control system is switched to the standby system

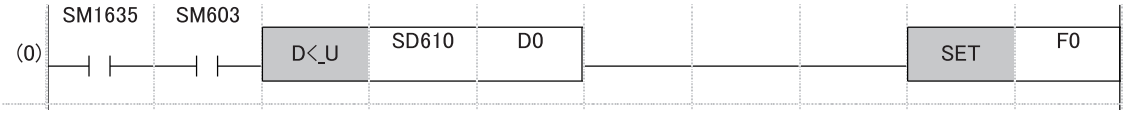
Before system switching (old control system)	After system switching (new standby system)
Waiting start Not collected	Standby system start waiting
Pause	Standby system pause
Waiting to establish collection conditions Not collected	Standby system no collection
Collecting	
Waiting trigger Collecting before trigger	
Collecting after trigger	

■When the standby system is switched to the control system

Before system switching (old standby system)	After system switching (new control system)
Standby system start waiting	Waiting start Not collected
Standby system pause	Pause
Standby system no collection	Waiting to establish collection conditions Not collected

Point

Before using the data logging function, check if the SD memory card inserted in the standby system CPU module has sufficient free space with a program. When the SD memory card inserted in the standby system CPU module does not have sufficient free space and the systems are switched, an error occurs at the data collection in the new control system and the data collection is finished.
To prevent occurrence of an error, use a program that turns on the annunciator when the free space of the SD memory card of the standby system is smaller than the value of the criteria, such as the one shown below.



- For the above program, set "Both Systems Executions" for "Both Systems Program Executions Setting" of the CPU parameter. (☞ Page 433 Program Execution in Both Systems)
- Set the value of the criteria in D0 in increments of K of the double word [unsigned].

LED status

The following table shows the LED status in the data logging states of the standby system.

State of the data logging function		LED status
Execution in progress	All of the registered data logging sessions are in the "Standby System Start Waiting", "Standby System Pause", or "Standby System No Collection" state.	• CARD READY LED: On • CARD ACCESS LED: Off • FUNCTION LED: On

Data collection conditions

When time specification has been set as a data collection condition and the systems are switched during data logging, time measurement starts from 0 in the new control system, and data is not collected at the specified interval.

Trigger logging

If the systems are switched while trigger logging is being executed, logging data collected in the old control system is discarded.

Switching to a storage file

After system switching, the file name in the old control system is not taken over to the new control system. (A number added to the file name is not a serial number.)

An identifier added to the end of the file name indicates whether the systems are switched or not.*3

Identifier	Description
_NC*1*2	The systems have not been switched.
_CS	The control system has been switched to the standby system.
_SC	The standby system has been switched to the control system.
_SS	The standby system has been switched to the control system and back to the standby system. (The system is switched back to the control system while data is being written to the save file.)

*1 With the trigger logging, the identifier "_NC" is added because the logging data is discarded at system switching.

*2 With the auto logging, the identifier "_NC" is added because the auto logging is completed when the systems are switched.

*3 Since three characters of an identifier are added to a file name, the save file name should be specified within 61 one-byte characters (including an underscore (_), a serial number (eight digits), a period, and an extension). (☞ Page 754 Save)

Point

When the simple setting (default) is applied for the save file name, only a serial number is added to the file name. If the systems are switched with this setting, which file is sequentially output cannot be determined. Thus, the following settings are recommended. (☞ Page 754 Save)

- Simple setting: Select "Date" and "Time", and select "Date to establish file switching condition" in the add date type.
- Optional setting: Enter a string indicating a date (YYYYMMDD) and time (hhmmss) in "Format", and select "Date to establish file switching condition" in the add date type.

Storage file switching condition

Switch the storage file at system switching at the following timing.

Storage file switching condition	Storage file switch timing
System switching from control system to standby system	When the data collected before system switching to the standby system is transferred to the SD memory card

Writing the data logging setting

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

- While the data logging setting is being written, do not perform the following operations on both of the control system and standby system.
  - Turning off or resetting the CPU modules
  - Connecting/disconnecting the tracking cable
  - Online module change of redundant function modules
- In the following cases, the setting is written only to own system.
  - The other system CPU module is off or reset.
  - A hardware failure (a failure of the CPU module or redundant function module) has occurred on the other system.
  - A tracking cable is disconnected or incorrectly connected, or failed.
  - Tracking communications stops due to an error in the CPU module.

Auto logging function

- The auto logging function cannot be used in the standby system. Even though an SD memory card that stores the data logging setting file for which the auto logging function has been enabled is inserted, the data logging is not started automatically.
- When the systems are switched during auto logging, the auto logging is completed.

38.16 Clearing an Error on the Standby System CPU Module from the Control System CPU Module

An error on the standby system can be cleared from the control system CPU module by using SM1679 (Error reset (the other system)) in programs or external devices.

Error clearing procedure

Use SM1679 to clear an error.

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

Precautions

- Error clear operation with SM1679 can be performed by the control system CPU only. A continuation error on the standby system CPU module cannot be cleared by turning off and on SM1679 of the standby system CPU module.
- Error clear operation with SM1679 can be performed in backup mode only.
- The cause of an error which has occurred in a module other than the target CPU module for the error clear cannot be eliminated even though the error is cleared using SM1679.
- If the error cause is not eliminated after the error clear operation, the same error is detected again.
- The error clear processing is performed in the END processing. To clear an error, execute the END instruction while SM1679 is on.

39 PRECAUTIONS ON PROGRAMMING

This chapter describes the precautions on programming for a redundant system.

39.1 Instructions Not Available in Redundant System

This section describes the instructions not available in a redundant system.

Instructions that cause stop errors

Do not use the following instructions in backup mode. Doing so causes an error when the operating status of the CPU module is changed from STOP to RUN.

Classification		Instruction symbol
Special counter instructions	Counting up or down the current value (1-phase input)	UDCNT1
	Counting up or down the current value (2-phase input)	UDCNT2
Special timer instructions	Teaching timer	TTMR
	Special function timer	STMR
Shortcut control instruction	Rotary table shortest direction control	ROTC
Ramp signal instruction	Ramp signal	RAMPQ
Pulse related instructions	Measuring the density of pulses	SPD
	Outputting pulses at regular intervals	PLSY
	Performing the pulse width modulation	PWM
Matrix input instruction	Matrix input	MTR
Multiple CPU dedicated instructions	Reading device data from another CPU module	D.DDRD
		DP.DDRD
		M.DDRD
		MP.DDRD
	Writing device data to another CPU module	D.DDWR
		DP.DDWR
		M.DDWR
		MP.DDWR

Instructions that need to be executed again in a new control system

For an instruction that requires several scans for completing the processing, the instruction will be continuously executed when the system switching is performed during execution of the instruction. When a completion device has been used in an execution program of the control system, the completion device will not turn on even though the instruction is completed after the system switching from the control system to the standby system. However, the completion device will turn on after the system switching from the control system to the standby system and then to the control system again. Note that the completion state of the completion device is not reflected to the tracking device. When the system switching is performed during execution of an instruction, execute the instruction again as required.

Classification	Instruction symbol
Data processing instructions	SORTD(_U), DSORTD(_U)
Reading/writing data instructions	SP.DEVST, SP.FREAD, SP.FWRITE
Open/close processing instructions	SP.SOCOPEN, SP.SOCCLOSE
Socket communications instructions	SP.SOCRCV, S.SOCRCVS, SP.SOCSND, SP.SOCCINF, SP.SOCCSET, SP.SOCRMODE, S(P).SOCRDATA
Module dedicated instructions	Instructions that require several scans for completing processing

For some instructions, an error will occur if an instruction is executed during execution of the same instruction. For the operation for each instruction that is executed again during execution of the same instruction, refer to the following.

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

Re-execution of instruction when systems are switched during instruction execution

When the system switching is performed while an instruction that requires several scans is being executed, the instruction can be executed again in the new control system after the system switching by using such as the following programs.

■REMFR instruction

When the system switching is performed while the instruction is being executed (M201 = ON), SM1643 (ON only in one scan after system switching (standby to control)) will turn on for one scan in the new control system and the REMFR instruction will be executed again on the station number 10 of the network number 1.

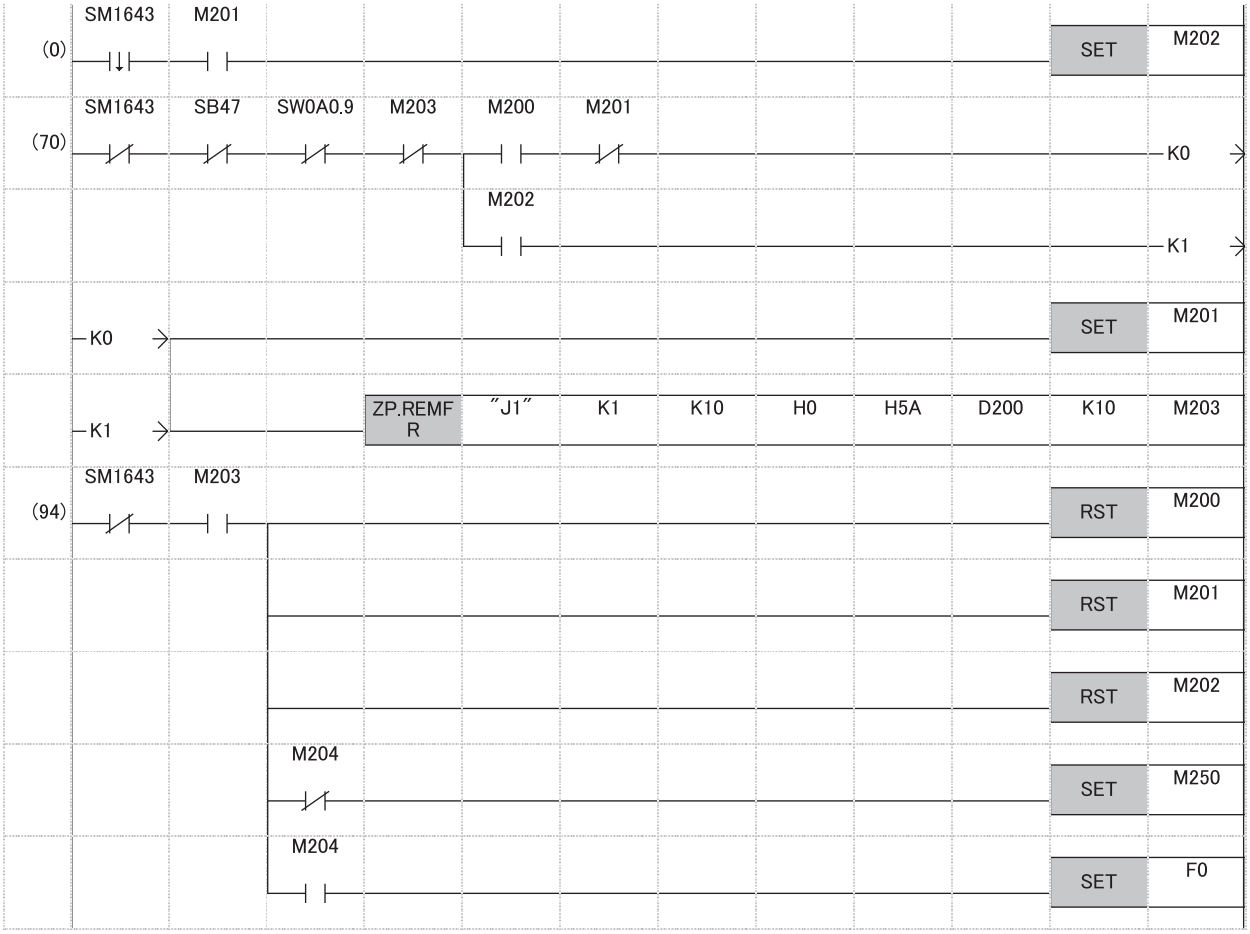
• DEVICES

DEVICES	Data stored
SM1643	System switching check flag (standby system to control system)
SB47*1	Baton pass status of own station
SW0A0.9*1	Baton pass status of each station
M200*2	On: Reading request
M201*2	At instruction execution
M202*2	ON at instruction re-execution request due to system switching
M203*2	Completed without an error
M204*2	Completed with an error

*1 For details on the link special relays (SB) and link special registers (SW), refer to the manuals for the network used.

*2 Change the device number according to the system.

• Program example



■REMTO instruction

When the system switching is performed while the instruction is being executed (M101 = ON), SM1643 (ON only in one scan after system switching (standby to control)) will turn on for one scan in the new control system and the REMTO instruction will be executed again on the station number 10 of the network number 1. When the system switching is performed while a write instruction such as the REMTO instruction is being executed, execution of the instruction may have been suspended before completion of writing data to the target module. Thus, insert an interlock in the new control system to read the X signals and buffer memory status of the target module and to determine whether or not to execute the instruction again.

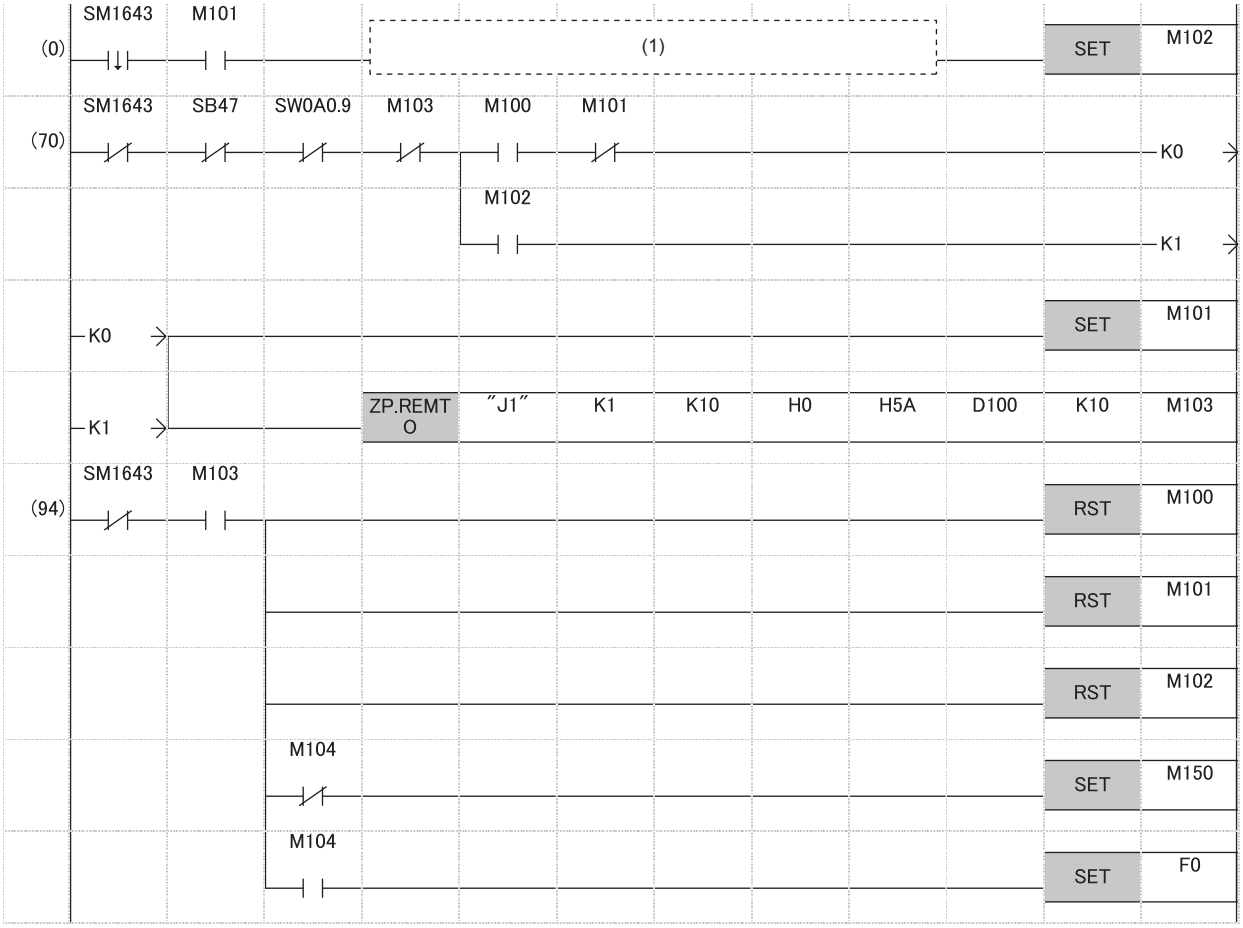
• DEVICES

DEVICES	Data stored
SM1643	System switching check flag (standby system to control system)
SB47*1	Baton pass status of own station
SW0A0.9*1	Baton pass status of each station
M100*2	Write request
M101*2	At instruction execution
M102*2	ON at instruction re-execution request due to system switching
M103*2	Completed without an error
M104*2	Completed with an error

*1 For details on the link special relays (SB) and link special registers (SW), refer to the manuals for the network used.

*2 Change the device number according to the system.

• Program example



(1) Add an interlock circuit for determining whether or not to execute the instruction again (according to the X signals and buffer memory status of the target module) as required.

Instructions whose operations vary depending on tracking of the signal flow memory

The following describes the instructions whose operations after the system switching vary depending on whether the signal flow memory is tracked or not in backup mode. The operations vary when one of the following instructions is executed among program organization units that have the signal flow memory, memory to which tracking can be performed.

Classification/type of special relay	Instruction symbol
Rising instruction	PLS, MEP, SET F, RST F, FF, LEDR, DUTY, LOGTRG, LOGTRGR, □P (including LDP, ANDP, ORP, MOVP, and INCP)
Falling instruction	PLF, MEF, LDF, ANDF, ORF
SCJ instruction	SCJ
Data processing instructions	SORTD(_U), DSORTD(_U)
TIMCHK instruction	TIMCHK
XCALL instruction	XCALL
Rising instruction using SM1643 as an execution condition	—

Rising instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, a rising instruction whose execution condition turned on during system switching will not be executed.

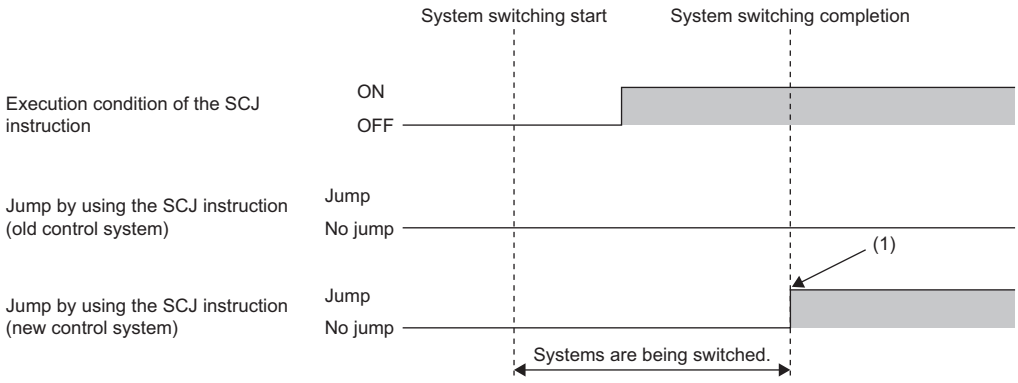
Falling instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, a falling instruction whose execution condition turned off before the system switching will be executed.

SCJ instruction

■When the signal flow memory is not tracked

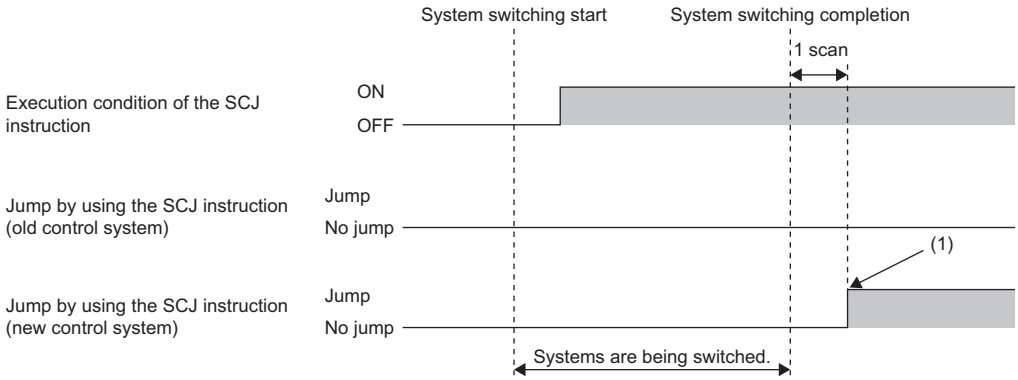
Once the system switching is performed, the execution condition of the SCJ instruction turns on after the signal flow memory of the new control system turns off. In the new control system, processing jumps to the pointer specified by the SCJ instruction in the first scan.



(1) After system switching, the processing jumps in the first and later scans.

■When the signal flow memory is tracked

When the system switching is performed, the execution condition of the SCJ instruction turns on while the signal flow memory remains off. In the new control system, the processing jumps to the pointer specified by the SCJ instruction in the second scan.



(1) After system switching, the processing jumps in the second and later scans.

Data processing instructions

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, the SORTD(_U)/DSORTD(_U) instruction in the first scan after the system switching will be executed not as the first execution but as continuous execution (continuous processing). When the instruction is executed for the first time, the instruction is executed without data to be stored in the devices used by the system, causing an unintended operation.

TIMCHK instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, the SORTD(_U)/DSORTD(_U) instruction in the first scan after the system switching will be executed not as the first execution but as continuous execution (continuous processing). At the first execution, the current value is not cleared and the device that turns on at timeout is not turned off. The instruction is executed with the status at the previous measurement.

XCALL instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, the subroutine program will not be executed when the execution condition of the XCALL instruction remains off in the first scan after system switching.

Rising instruction using SM1643 as an execution condition

SM1643 is the special relay that turns on for one scan in the new control system after system switching. If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on and a rising instruction cannot be executed. To execute a rising instruction where SM1643 has been set as an execution condition, use a falling edge pulse operation contact (LDF/AND/ORF instruction) and create a program in which the rising instruction is to be executed at the falling edge of SM1643, as shown below. However, when the falling edge of SM1643 has been set as the execution condition, the target instruction will be executed in the second scan after the system switching.

(1) The instruction will be executed in the second scan after the system switching.

Instructions that affect the status of another instruction when executed

When one of the following instructions is executed and the status of another instruction changes, the new status will not be tracked to the other system. When the system switching is performed during execution of an instruction, execute the instruction again as required.

Classification		Instruction symbol
Program execution control instructions	Disabling interrupt programs	DI
	Enabling interrupt programs	EI
	Disabling interrupt programs with specified priority or lower	DI
	Interrupt program mask	IMASK
	Disabling/enabling the specified interrupt pointer	SIMASK
File register operation instructions	Switching the file register block number	RSET(P)
	Changing the file register file name	QDRSET(P)
Timing check instruction	Generating timing pulses	DUTY

Instruction that causes different operation results between the control system and standby system

The following describes the instruction that causes different operation results in both systems after system switching.

PID control instruction

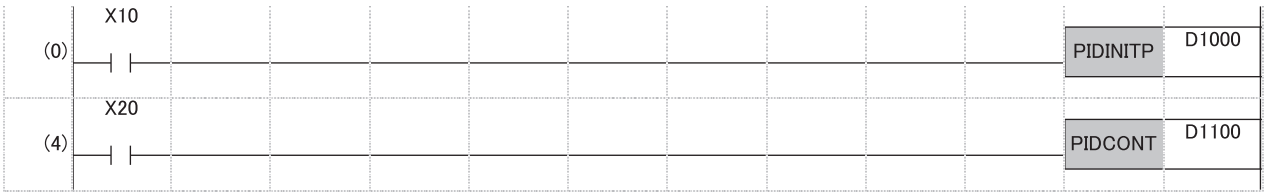
When using the following PID control instructions, include the number of device points used by the systems in tracking target data. Otherwise, the instructions cause different operation results between the control system and standby system after system switching.

n: Total number of loops

Classification		Instruction symbol	Number of device points used
PID control instructions (Inexact differential)	Registering the PID control data to the CPU module	S(P).PIDINIT	2 + n × 4
	PID operation	S(P).PIDCONT	10 + n × 23
PID control instructions (Exact differential)	Registering the PID control data to the CPU module	PIDINIT(P)	2 + n × 10
	PID operation	PIDCONT(P)	10 + n × 18

Ex.

When the number of loops is eight, the PIDINITP instruction needs 2 + 8 × 10 = 82 words and the PIDCONTP instruction needs 10 + 8 × 18 = 154 words. Thus, include D1000 to D1081 and D1100 to D1253 in tracking target data.



Precautions for using the COM or ZCOM instruction

When the network module link refresh is performed by using the COM or ZCOM instruction, output from the remote I/O station may change after system switching. To prevent this, do not perform the network module link refresh with the COM or ZCOM instruction. For the COM instruction, whether or not to perform the network module link refresh can be set using SM775 (Selection of refresh processing during the COM instruction execution) and SD775 (Selection of refresh processing during the COM instruction execution). Set SM775 and SD775 and perform only the I/O refresh and device/label access service processing with the COM instruction.

○: Selectable, ×: Not selectable

Instruction	Refresh processing	In a redundant system
COM instruction	I/O refresh	○
	Network module link refresh	○*1
	Intelligent function module refresh	×*2
	Refresh using the CPU buffer memory of the multiple CPU system (END)	×*3
	Device/label access service processing (communications with the engineering tool, GOT, or other external devices)	○
ZCOM instruction	Network module link refresh	○*1
	Intelligent function module refresh	×*2

*1 When the COM or ZCOM instruction is executed, tracking transfer is not performed. Thus, when the system switching is attempted after execution of the instruction and before completion of tracking transfer, the system switching is performed without tracking transfer. Thus, even though output to the network module is changed by the COM or ZCOM instruction with the CPU module in the control system, the change will not be reflected to the CPU module in the standby system. After the system switching, the CPU module in the new control system outputs old values to the network module, and output from the network module may vary.

*2 This processing is not selectable because no intelligent function modules can be mounted on main base units in a redundant system.

*3 This processing is not selectable because a multiple CPU system cannot be built in a redundant system.

Precautions for using the ADRSET instruction

Even though a file is written to both the control system and standby system, different addresses are assigned to the systems. To continue the processing in the new control system, use the ADRSET instruction to obtain indirect addresses again.

39.2 Interrupt from Modules

The following describes the precautions for interrupts from modules.

System switching in backup mode

When the old control system is switched to the new standby system

The old control system retains the interrupt factors that have occurred even after the system is switched to the new standby system by system switching before execution of an interrupt program. After the systems are switched again, the interrupt program of an interrupt factor that the old control system has retained will be executed.
Since the interrupt factor that has occurred on the old control system is not inherited to the new control system, the interrupt program of an interrupt factor that has occurred on the old control system will not be executed on the new control system.

When the old standby system is switched to the new control system

The old standby system retains the interrupt factors that have occurred. After the old standby system is switched to the new control system by system switching, the interrupt program of an interrupt factor that the old standby system has retained will be executed on the new control system.
When the old standby system has retained multiple interrupt factors, the scan time may greatly increase.

System switching in separate mode

When the old control system is switched to the new standby system

The old control system executes the interrupt factors that the old control system has retained even after the system is switched to the new standby system by system switching before execution of an interrupt program.

When the old standby system is switched to the new control system

The old standby system retains the interrupt factors that have occurred. The new control system executes the interrupt program of an interrupt factor that the old standby system has retained regardless of the system switching operation.

When the operation mode is switched to the separate mode

For the control system

The control system retains the interrupt factors before switching to the separate mode. After the operation mode is switched to the separate mode, the control system executes the interrupt programs of the interrupt factors that have occurred in the backup mode.

For the standby system

The interrupt factors that the standby system has retained before switching to the separate mode will be discarded. Therefore, after the operation mode is switched to the separate mode, the interrupt programs of the interrupt factors that have occurred before switching to the separate mode will not be executed.

When the operation mode is switched to the backup mode

For the control system

When the interrupt programs of the interrupt factors before switching to the backup mode have not been executed yet, the control system will execute the programs that the system has retained in the separate mode.

For the standby system

When the interrupt programs of the interrupt factors before switching to the backup mode have not been executed yet, the standby system will retain the interrupt factors that the system has retained in the separate mode. Interrupt programs will not be executed.

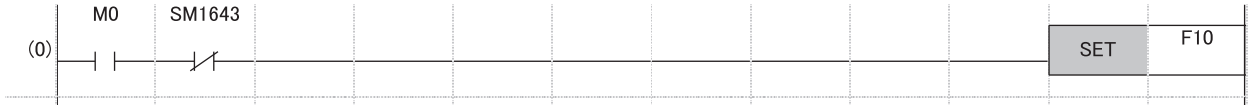
39.3 Precautions for Using the Annunciator (F)

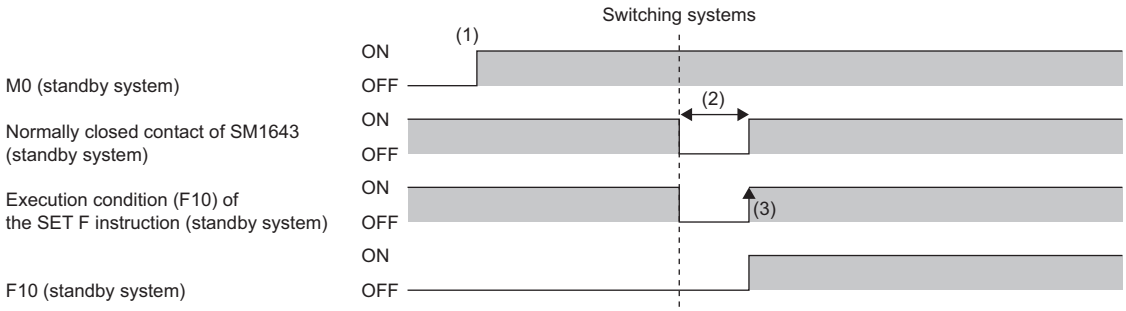
This section describes the precautions for using the annunciator (F).

When the SET F instruction is used to register the annunciator

When the execution condition of the SET F☐ instruction has been satisfied at system switching, the annunciator that is turned on by the SET F☐ instruction can be registered in the new control system. The SET F☐ instruction registers the annunciator at the rising edge of the execution condition. To register the annunciator in the new control system at system switching, add a normally closed contact of SM1643 (ON for only one scan after system switching (standby system to control system)) as the AND condition in the execution conditions of the SET F☐ instruction.

Ex.





(1) When M0 turns on in the control system before system switching, M0 of the standby system also turns on as a result of tracking transfer.
(2) The contact turns off for one scan after system switching.
(3) When the execution condition turns on, annunciator information is registered by the SET F☐ instruction.

When the OUT F instruction is used to register the annunciator

When the execution condition of the OUT F☐ instruction has been satisfied at system switching, annunciator information is registered in the new control system at system switching.

39.4 Precautions on Timers and Timer Function Blocks

The following describes the precautions on timers and timer function blocks at system switching.

Current values at system switching

For the timer (T), retentive timer (ST), and a timer function block TIMER_☐_M, the current values of the timers in the first scan of the CPU module of the new control system will not be updated after system switching.

Timeout before system switching

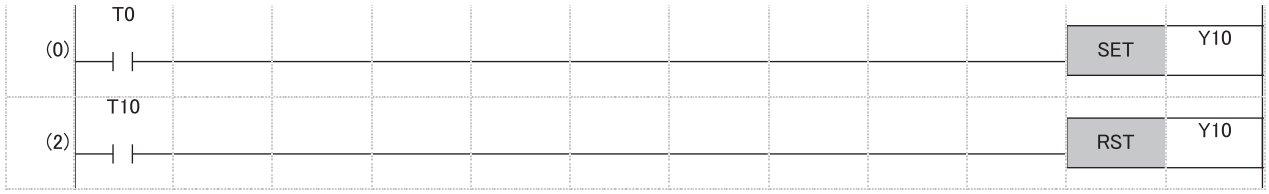
Depending on the timing to perform system switching such as power-off, tracking transfer processing is suspended and tracking data may not be reflected to the CPU module in the new control system. For the timer (T), retentive timer (ST), long timer (LT), long retentive timer (LST), and timer function blocks TIMER_☐_M, TP(_E), TON(_E), and TOF(_E), the timer whose time has been up before system switching may go into the state in which no timeout has occurred in the first scan after system switching.

When values are output (writing values to the buffer memory and the output (Y)) with a timer contact or an output variable, the timer statues may go into the state in which no timeout has occurred as described above, causing chattering of the output. To transfer data with modules or external devices using the output (Y) or buffer memory, a program may not properly function due to chattering of output after the system switching. To output values (writing data to the buffer memory and the output (Y)) to modules or external devices with a timer contact or an output variable, consider the time taken for data to be transferred from the CPU module in the control system to the CPU module in the standby system after the time is up.

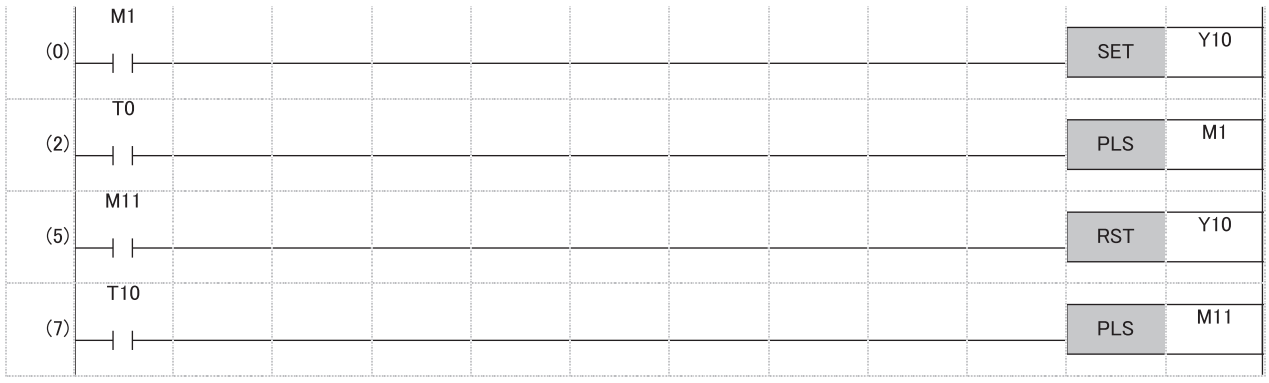
Ex.

Program that delays outputting values by one scan after the time of the timer (T) is up

[Without measures]



[With measures]



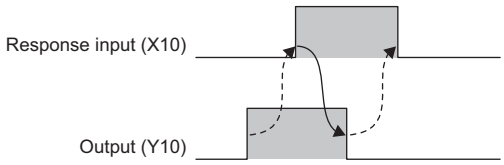
Point

In the CPU parameter, select "Transfer" (default setting) in "Signal Flow Memory Tracking Setting" of "Redundant System Settings". (☞ Page 412 Tracking transfer setting for the signal flow memory)

39.5 Precautions on Access to Intelligent Function Module or External Devices

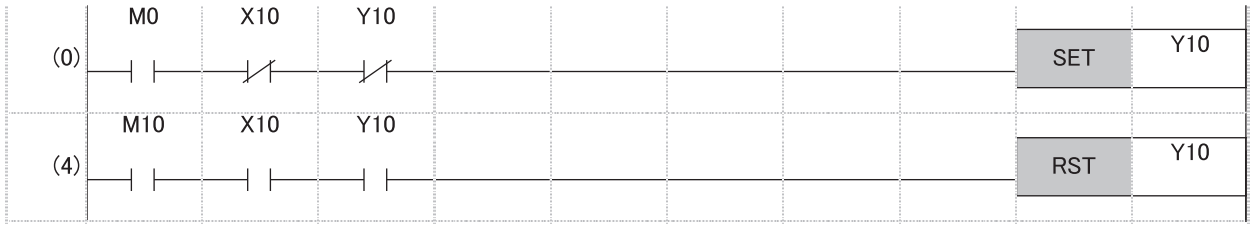
Depending on the timing of system switching cause to be caused, such as power-off, tracking processing is suspended and device/label data may not be applied to the CPU module in the new control system after the system switching.
Consequently, output data may differ from device/label data of the CPU module of the new control system. In communication with intelligent function modules or external devices using the output (Y) or buffer memory, programs may not properly function due to a mismatch in the device data after system switching.
For command output (such as output (Y), startup by writing data to the buffer memory, and clear) to the intelligent function module and external devices, consider the time taken for tracking the execution condition of command output from the CPU module in the control system to the CPU module in the standby system.
The following example shows a program that outputs data one scan later after the command output condition is satisfied.

Ex.
When response input is returned to output

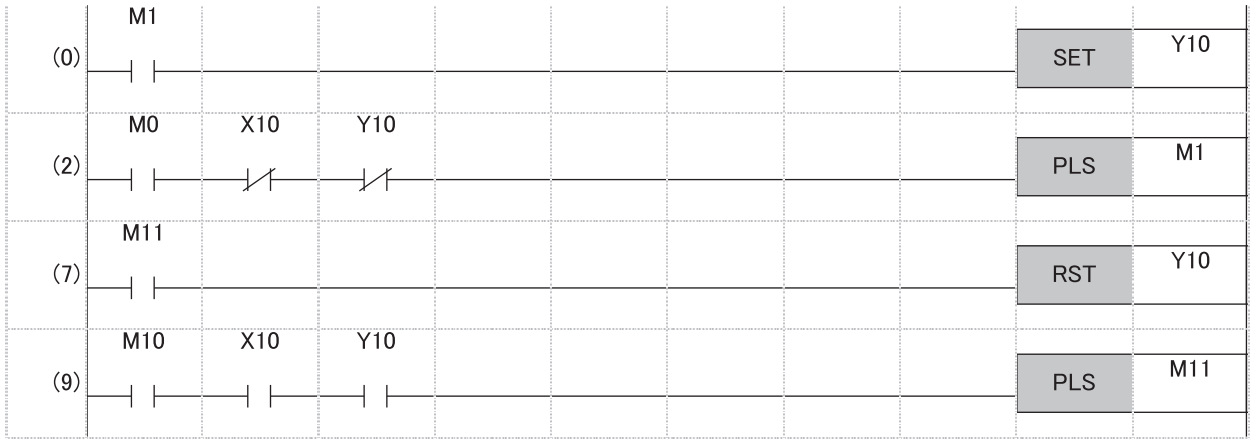


In the following program, turning on M0 turns on the output (Y10) and turning on M10 turns off the output (Y10).

[Without measures]



[With measures]



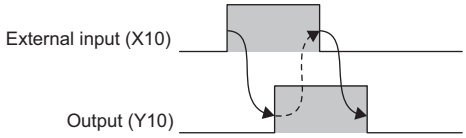
(2) PLS M1 delays SET Y0 by one scan.
(9) PLS M11 delays RST Y10 by one scan.

Point

In the CPU parameter, select "Transfer" (default setting) in "Signal Flow Memory Tracking Setting" of "Redundant System Settings". (☞ Page 412 Tracking transfer setting for the signal flow memory)

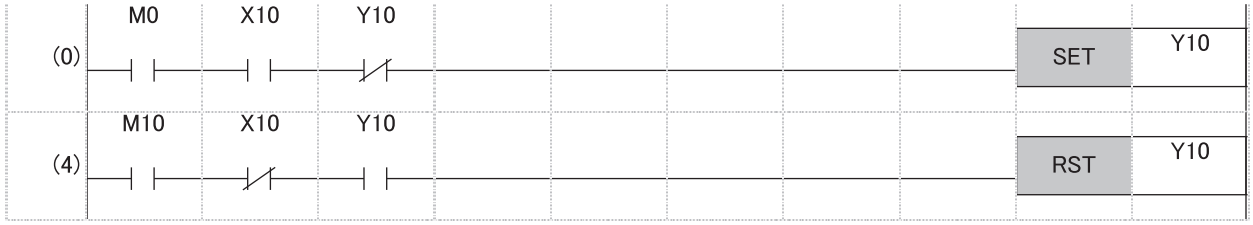
Ex.

When output is returned to external input

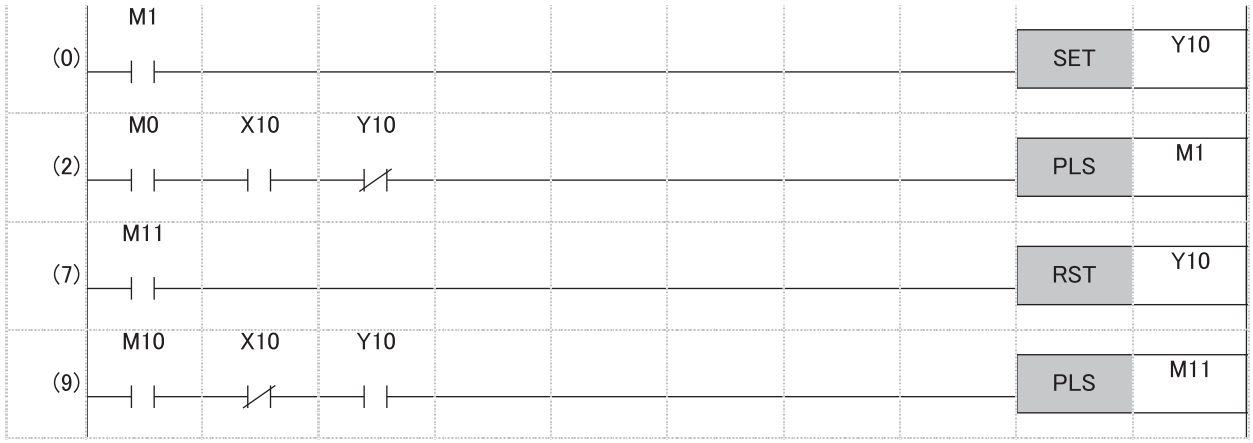


In the following program, turning on M0 turns on the output (Y10) and turning on M10 turns off the output (Y10).

[Without measures]



[With measures]



(2) PLS M1 delays SET Y10 by one scan.

(9) PLS M11 delays RST Y10 by one scan.

Point

In the CPU parameter, select "Transfer" (default setting) in "Signal Flow Memory Tracking Setting" of "Redundant System Settings". (☞ Page 412 Tracking transfer setting for the signal flow memory)

39.6 Precautions on Writing Data from GOT or External Devices

When data is written from the GOT or external devices, the tracking data may not be applied to the CPU module of the new control system depending on the timing of system switching cause to be caused, such as power-off.

Consequently, data that is written from the GOT or external devices before system switching may be lost. Write the data again after the system switching.

39.7 Precautions on Outputting in the Middle of the Scan

When values are output during execution of a program with the following devices or setting, the program will be executed again from the step 0 in the new control system after system switching. Therefore, values may be output twice, before and after system switching. Output results may differ between the two outputs.

- Link direct device (Jn\Y)
- Direct access output (DY)
- Refresh at execution of a specified program ("Refresh Group Setting" of "Program Setting")

Before outputting values with one of the above devices or setting, check that the above operation does not cause any problem in the system design phase. If any problem occurs, do not perform the external output until the status of the received output signal becomes stable, or take other measures with external circuits.

Point

In the program of the redundant system, performing the output refresh by the END processing is recommended instead of outputting with the link direct device (Jn\Y), direct access output (DY), or refresh at execution of a specified program ("Refresh Group Setting") in the middle of the scan.

40 MAINTENANCE AND INSPECTION FOR A REDUNDANT SYSTEM

This chapter describes the maintenance and inspection for a redundant system.

40.1 Module Replacement in the Redundant System

The following describes the methods of replacing modules in a redundant system while the system is operating.

- Replacing modules in a control system with the online module change function
- Replacing modules in a standby system (To replace modules in a standby system, power off the standby system or use the online module change function.)

○: Replaceable, ×: Not replaceable

Replacement target	Replacement		Reference
	Control system	Standby system	
CPU module	×	○	Page 467 Replacing a CPU module
Power supply module	×	○	Page 469 Replacing a power supply module
Redundant power supply module	○*1	○	Page 469 Replacing a redundant power supply module
Redundant function module	○*3	○*2	Page 470 Replacing a redundant function module
I/O modules	○*4	○*2	Page 470 Replacing I/O modules
Network module	×	○	Page 471 Replacing a network module
Main base unit	×	○	Page 471 Replacing a main base unit

*1 This module in a redundant power supply system can be replaced.

*2 To replace this module without powering off the system, use the online module change function. (MELSEC iQ-R Online Module Change Manual)

*3 When an error has been detected on a redundant function module in the control system, a continuation error occurs on the CPU module, and the control system and the standby system continue operating without being switched. Perform online module change if the redundant function module has failed. (MELSEC iQ-R Online Module Change Manual)

*4 Use the online module change function. (MELSEC iQ-R Online Module Change Manual)

Point

Since the standby system needs to be powered off for replacing their modules, a continuation error occurs on the CPU module of the control system. After the replacement, clear the continuation error that has occurred on the CPU module of the control system. (Page 233 Error Clear)

The following describes how to replace modules in the standby system (by powering off the standby system).

Replacing a CPU module

The following describes the procedure for replacing the CPU module.

The replacement procedure differs depending on whether the automatic memory copy function is enabled or disabled. (☞ Page 423 Automatic memory copy)

When the automatic memory copy function is enabled

1. Check the operation mode.

Check the BACKUP LED of the redundant function module in the control system to check if the CPU module in the control system is in the backup mode.

- The BACKUP LED of the redundant function module in the control system is on or flashes.

2. Check the system of the replacement target CPU module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target CPU module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

3. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes.

4. Replace the CPU module in the standby system.

Replace the CPU module in the standby system with a new CPU module whose model is the same as that of the CPU module in the control system. If an SD memory card or an extended SRAM cassette has been inserted to the replacement target CPU module, insert it to the new CPU module.

5. Change the switch status of the CPU module in the standby system.

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

6. Power on the standby system.

Power on the standby system.

7. Memory copy from control system to standby system

The memory is automatically copied. When the memory copy is completed, the MEMORY COPY LEDs of the redundant function modules of both systems turn off. Then, the CPU module in the standby system is automatically reset and restarted. (☞ Page 423 Automatic memory copy)

The BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

When the automatic memory copy function is disabled

1. Check the system of the replacement target CPU module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target CPU module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace the CPU module in the standby system.

Replace the CPU module in the standby system with a new CPU module whose model is the same as that of the CPU module in the control system. If an SD memory card or an extended SRAM cassette has been inserted to the replacement target CPU module, insert it to the new CPU module.

4. Change the switch status of the CPU module in the standby system.

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

5. Power on the standby system.

Power on the standby system.

6. Memory copy from control system to standby system

The memory is copied with the engineering tool or the special relay and special register. (☞ Page 424 Memory copy with the engineering tool, ☞ Page 425 Memory copy with the special relay and special register)

When the memory copy is completed, the MEMORY COPY LED of the redundant function module in the control system turns off and that in the standby system turns on.

7. Restart the standby system.

Restart the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

Replacing a power supply module

The following describes the procedure for replacing the power supply module.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace the power supply module in the standby system.

Replace the power supply module in the standby system.

4. Power on the standby system.

Check if wiring to the power supply module and the power supply voltage are correct, and power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

Replacing a redundant power supply module

Select the standby system or control system and power off and replace the redundant power supply module in the system. After that, power off and replace the module in the other system. Since the redundant power supply module that is not the replacement target supplies power to the modules on the base unit, controls can be continuously performed during replacement of the other redundant power supply module.

For the replacement procedure, refer to the following.

MELSEC iQ-R Module Configuration Manual

Replacing a redundant function module

The following describes the procedure for replacing the redundant function module.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace the redundant function module in the standby system.

Replace the redundant function module in the standby system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

5. Check the modules for errors.

Check that no error has occurred on the CPU module or redundant function module in the standby system. If the ERROR LED of either of the modules is on or flashes, check the cause of an error and eliminate the error cause.

Replacing I/O modules

The following describes the procedure for replacing I/O modules.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace the I/O modules in the standby system.

Replace the I/O modules in the standby system with new I/O modules whose models are the same as the ones of the I/O modules in the control system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

Replacing a network module

The following describes the procedure for replacing the network module.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace the network module in the standby system.

Replace the network module in the standby system with a new network module whose model is the same as that of the network module in the control system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

5. Perform network diagnostics on the standby system.

Check that no error has occurred on the network module.

Replacing a main base unit

The following describes the procedure for replacing the main base unit.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace the main base unit in the standby system.

Replace the main base unit in the standby system with a new main base unit whose model is the same as that of the main base unit in the control system. When mounting modules to the new base unit, mount each module to the slots of the unit in the same order as that of the base unit of the control system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

5. Check the modules for errors.

Check that no error has occurred on the CPU module or redundant function module in the standby system. If the ERROR LED of either of the modules is on or flashes, check the cause of an error and eliminate the error cause.

MEMO

PART 5 WHEN USING THE SAFETY CPU

This part consists of the following chapters. Please read these chapters when using the Safety CPU. Since information same as that of the standard CPU is not described in these chapters, refer to Part 1 to Part 3.

41 RUNNING A SAFETY PROGRAM

42 MEMORY SPECIFICATIONS

43 FUNCTIONS

44 SAFETY DEVICES AND CONSTANTS

41 RUNNING A SAFETY PROGRAM

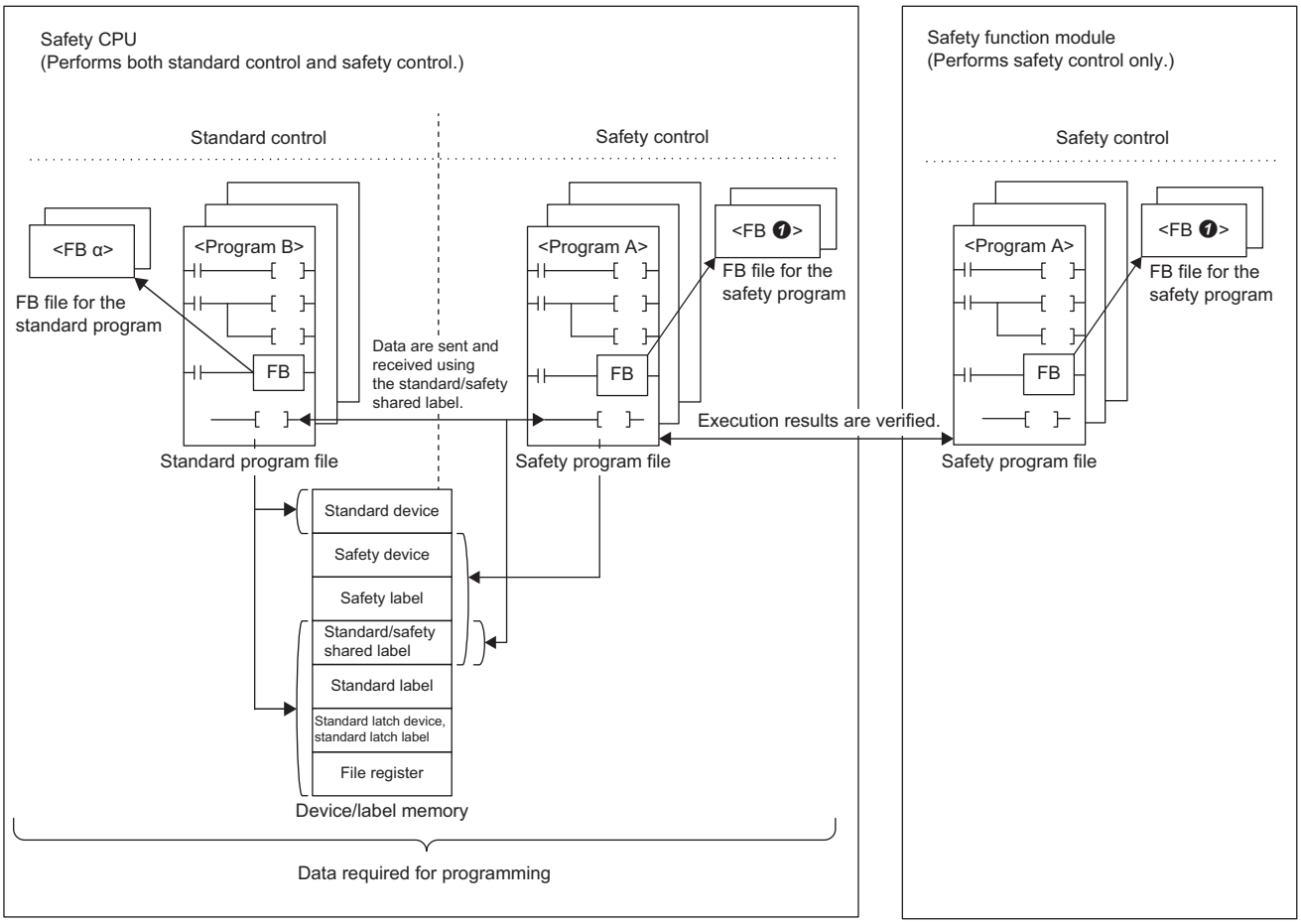
This section describes how to execute a safety program. Information not described in this chapter is same as that of the standard CPU. (☞ Page 32 RUNNING A PROGRAM to ☞ Page 75 MEMORY CONFIGURATION OF THE CPU MODULE)

Point

For details on the setting method (registration procedure) of safety programs, refer to the following.
📖 GX Works3 Operating Manual

41.1 Overview

The Safety CPU executes standard programs and safety programs, and the safety function module executes safety programs only. During operations, the Safety CPU verifies the execution results of safety programs with those of the safety function module.



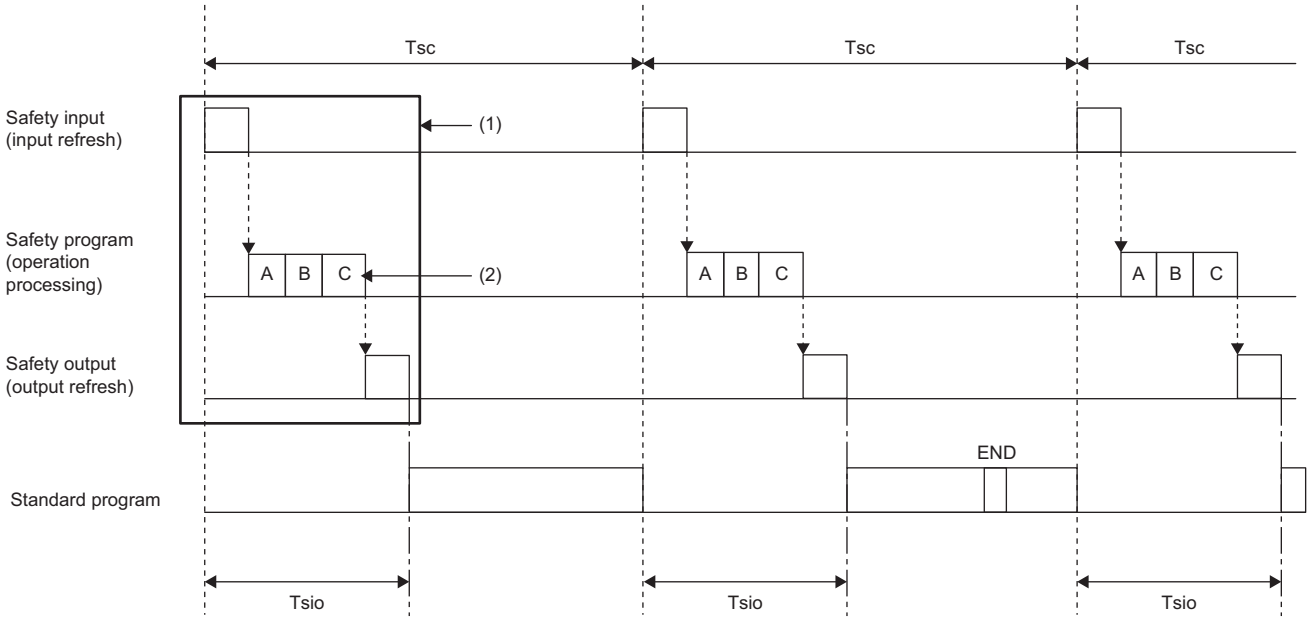
Devices/labels that can be used in safety programs

Only safety devices and safety labels can be used in safety programs.

- Safety device: ☞ Page 504 Safety Devices
- Safety label: 📖 MELSEC iQ-R Programming Manual (Program Design)

41.2 Safety Program

Safety programs are executed at every safety cycle. Safety cycle processing is performed in the order safety input (refresh) processing followed by safety output (refresh) processing. (☞ Page 477 Safety Cycle Time)
Standard programs (+ END processing) are executed within the remaining time of the safety cycle time after safety programs are executed. (Standard programs are executed until next safety cycle time starts.)



Tsc: Safety cycle time
Tsio: Safety program + Safety input/output processing time
(1): Store the processing times below.
- Safety CPU: SD1890, SD1891
- Safety function module: Un\G62, Un\G63

(2): Execute all safety programs. (Execute safety programs A, B, and C.)

Point

- After executing a rising/falling instruction, this will be valid until the safety program for the next safety cycle processing is executed. (MELSEC iQ-R Programming Manual (Instructions, Standard Functions/ Function Blocks))
- In safety programs, the block number of the file register (R) is not saved or restored. Data in the index register (Z, LZ) are not saved or restored, either.
- Even if the operating status of Safety CPUs for which no safety program has been registered is STOP or PAUSE, safety program operations other than operation processing are performed. Consequently, the standard control processing time and device/label access service processing time will be longer.

Safety Program Execution Type

The standard program and safety program execution types are as follows.

Program	Execution type
Standard program	• Initial • Scan • Fixed scan • Event • Standby
Safety program	Fixed scan

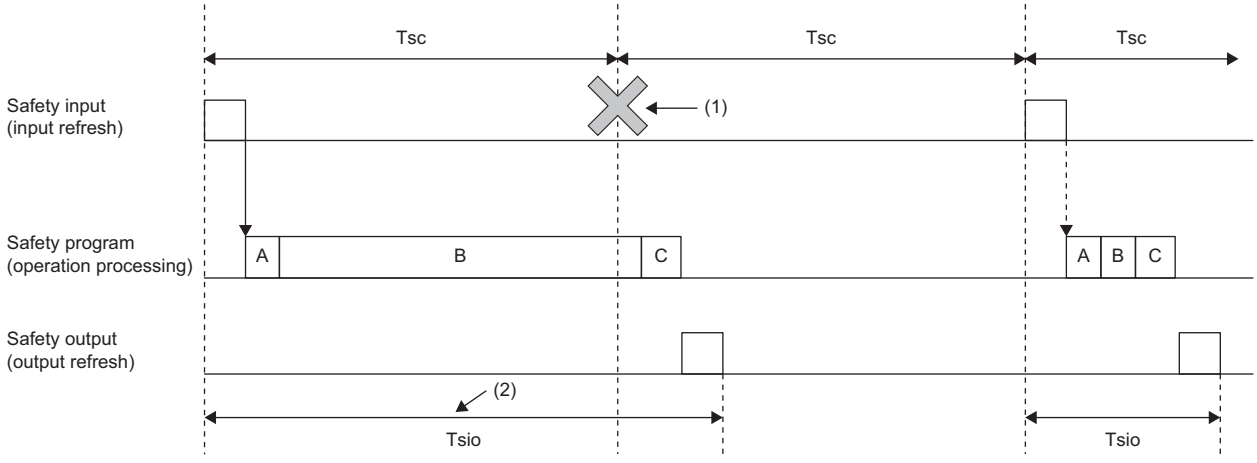
Safety programs are executed as a fixed scan execution type program. However, safety programs perform safety control, and therefore operation differs from standard fixed scan execution type programs in the following ways.

- They are unaffected by disabling interrupt instructions (EI instruction, DI instruction, and MASK instruction), and are executed based on the safety cycle time.
- Of all interrupt programs that can be created by customers, the execution of safety programs is given highest priority.
- Safety cycle processing including safety programs prioritizes fixed periodicity. If the safety cycle time is exceeded, and safety cycle processing is executed, processing is not performed in succession, but based on the interrupt timing of the next cycle time.

When the program does not complete within the safety cycle time

Monitors whether execution of the safety program is completed within the safety cycle time, and if the program execution time exceeds the next safety cycle time, an error occurs at the Safety CPU and safety function module. Note that if the next safety cycle time is exceeded, the safety input → safety program is not executed following program completion, a safety cycle processing error (error code: 1A01H) is detected at the Safety CPU, and the safety program is executed at the next safety cycle time. Also, if execution of the program is completed after the safety cycle time has been exceeded, the safety cycle processing time execution cycle error flag turns on, and the safety cycle processing time execution cycle error count is incremented by one as follows.

Item	Safety cycle processing time execution cycle error flag	Safety cycle processing time execution cycle error count	Cause of safety cycle processing time execution cycle error count being incremented by one
Safety CPU	SM1888	SD1888	If error codes 1A00H, 1A01H occur
Safety function module	—	Un\G54	If error code 1A00H occurs



Tsc: Safety cycle time

Tsio: Safety program + Safety input/output processing time

(1): If safety input/output and safety program execution are not completed within the safety cycle time, interrupts during this period are ignored. Execute safety input/output and safety programs from the next interrupt.

(2): If the time from safety input to safety output exceeds the safety cycle time, a continuation error will occur after safety processing is complete.

- Safety CPU: SM1888 turns ON, and the value in SD1888 is incremented by one.
- Safety function module: The value in Un\G54 is incremented by one.

41.3 Safety Cycle Time

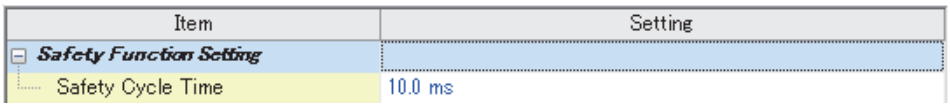
A safety cycle time is a timing for executing safety programs and performing safety input/output processing.

Setting method

Set a safety cycle time in the CPU parameter.

[CPU parameter] ⇨ [Safety Function Setting]

Window



Displayed items

Item	Description	Setting range	Default
Safety Cycle Time	Set a timing (safety cycle time) for executing safety programs and safety input/output processing.	1.0 to 1000.0ms (unit: 0.1ms)	10.0ms

Point

If an instruction with long processing time is executed in the standard program, the start of safety cycle processing is delayed. To prioritize safety cycle processing, set "Enable" for the enabling interrupt setting during command execution. (☞ Page 65 Interrupt enabled during instruction execution)

41.4 Safety CPU Operating Status

The following is a list of Safety CPU operating statuses.

- RUN state
- STOP state
- PAUSE state

Operation processing based on safety CPU operating status

Operation processing based on the Safety CPU operating status is the same as that of the standard CPU. (☞ Page 71 Operation Processing According to Operating Status of the CPU Module)

Safety communications processing based on safety CPU operating status

Safety communications processing based on the Safety CPU operating status is as follows.

Safety CPU operating status	Safety communications processing
RUN state	Executes the set safety communications.
STOP state	Executes the set safety communications, however, send data is turned off (0).*1*2
PAUSE state	Executes the set safety communications, however, send data is turned off (0).*1

*1 If the safety operation mode is TEST MODE, output is based on the output hold and clear settings when the status of network modules (RJ71GF11-T2) for which safety communication settings are specified is CPU STOP. (📖Manual for network module used)

*2 Safety communications are not performed if a stop error occurs at the Safety CPU.

Operation processing when operating status of the Safety CPU changes

This displays operation processing with the safety program when the operating status of the Safety CPU is changed.

Safety CPU Operating Status	Safety CPU processing			
	Safety program	External output	Safety device data*1	
			Other than SA\Y	SA\Y
STOP to RUN state	Executes safety programs.	Outputs the value after safety program execution.	Retains the safety device data status immediately before the RUN status. Clears safety local devices.	Retains the SA\Y value immediately before the RUN status.
RUN to STOP state	Executes up to the END instruction and then stops.	Turns off all safety communication outputs.*3	Retains the device data status immediately before the STOP status.	Turns off all SA\Y values.*2
RUN → PAUSE	Stops operation after the execution of one scan.	Turns off all safety communication outputs.*3	Retains the device memory status immediately before the PAUSE state.	Turns off all SA\Y values.*2
PAUSE → RUN	Executes the program from the start.	Outputs the value after safety program execution.	Retains the device data status immediately before the RUN status. Clears safety local devices.*2	Retains the device data status immediately before the RUN status.
PAUSE → STOP	Operation remains stopped.	All safety communication outputs remain off.*3	Retains the device data status immediately before the STOP status.	Retains the SA\Y value immediately before the STOP state.
STOP → PAUSE	Operation remains stopped.	All safety communication outputs remain off.*3	Retains the device data status immediately before the PAUSE state.	Retains the SA\Y value immediately before the PAUSE state.

*1 Assignment device data for standard/safety shared labels is processed in the same way as operation processing with standard programs.

*2 Values for safety devices when the Safety CPU operating status changes are cleared or turn off, regardless of the safety operation mode.

*3 If the safety operation mode is TEST MODE, output is based on the output hold and clear settings when the status of network modules (RJ71GF11-T2) for which safety communication settings are specified is CPU STOP. (📖Manual for network module used)

Operation processing with standard programs is the same as the following. (☞ Page 72 Operation Processing When Operating Status of the CPU Module Changed)

42 MEMORY SPECIFICATIONS

This chapter describes the memory specifications of the Safety CPU. Information not described in this chapter is same as that of the standard CPU. (☞ Page 75 MEMORY CONFIGURATION OF THE CPU MODULE)

42.1 Memory Configuration

Specifications of the device/label memory differ from the standard CPU in the memory configuration of the Safety CPU.

Device/label memory

In addition to standard devices and standard labels, the data such as safety devices and safety labels are allocated to each data area of the device/label memory.

The safety device area, safety label area, safety local device area, and standard/safety shared label area are allocated between the standard device area and standard label area. The capacities for each area can be changed in the CPU parameters by using the engineering tool. (☞ Page 495 Device/Label Memory Area Setting)

<table>
<tr><td colspan="2">Standard device area</td></tr>
<tr><td rowspan="3">Safety device/label area</td><td>Safety device area</td></tr>
<tr><td>Safety label area</td></tr>
<tr><td>Safety local device area</td></tr>
<tr><td colspan="2">Standard/safety shared label area</td></tr>
<tr><td rowspan="2">Standard label area</td><td>Standard label area</td></tr>
<tr><td>Standard latch label area</td></tr>
<tr><td colspan="2">Standard local device area</td></tr>
<tr><td colspan="2">File storage area</td></tr>
</table>

Data to be allocated

The following table lists the data allocated to each area.

Area		Application
Standard device area		Standard user devices
Safety device/label area	Safety device area	Safety user devices
	Safety label area	Safety global labels and safety local labels
	Safety local device area	Safety local devices
Standard/safety shared label area		Standard/safety shared labels
Standard label area	Standard label area	Standard global labels and standard local labels
	Standard latch label area	Standard global labels and standard local labels with latch specified
Standard local device area		Standard local devices (excluding index register)
File storage area		File register files and other data

Point

Safety devices and safety labels cannot be latched, and therefore there is no latch area for safety devices and safety labels.

42.2 File Size Unit in Memory

The following table lists the unit of the file size (cluster size) of the Safety CPU memory.

Safety CPU	File size unit		
	Program memory	Device/label memory	Data memory
R08SFCPU	128 bytes	512 bytes	2048 bytes
R16SFCPU			4096 bytes
R32SFCPU			8192 bytes
R120SFCPU			16384 bytes

Point

Data is written in the unit of the file size (cluster size). For example, when 464 bytes of CPU parameter is written to the data memory on R08SFCPU, it is written as 2048 bytes of data because the file size unit of the data memory is 2048 bytes.

42.3 Memory Operation

Using an engineering tool, each memory can be initialized and cleared to zero. For details on the operation method, refer to the following.

GX Works3 Operating Manual

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

Point

- If the power goes off during initialization or zero clear, the memory is left in the state of that point, and it is necessary to re-execute the memory operation.
- Following initialization, write the file required to run the Safety CPU. (GX Works3 Operating Manual)
- Following data memory initialization, if the power is turned off and on or reset, the device/label memory will revert to the default status (default capacity for each area).

42.4 Files

This section lists the files used by the Safety CPU.

File types and storage memory

The following table summarizes the types of files stored in the Safety CPU and storage memory.

◎: Required, ○: Can be stored, ×: Cannot be stored

File type		CPU built-in memory			SD memory card	File name and extension
		Program memory	Device/label memory	Data memory		
		Drive 0	Drive 3	Drive 4	Drive 2	
Standard program		◎*4*6	×	◎*4*6	×	ANY_STRING.PRG
Standard FB file		○*4	×	○*4	×	ANY_STRING.PFB
Standard CPU parameter		×	×	◎	×	CPU.PRM
System parameter		×	×	◎	×	SYSTEM.PRM
Standard module parameter		×	×	○	×	UNIT.PRM
Module extension parameter		×	×	○	○	• UEXmmmnn.PRM*1 • UEXmmm00.PPR*5
Memory card parameter		×	×	×	○	MEMCARD.PRM
Device comment		×	×	○	○	ANY_STRING.DCM
Initial device value		×	×	○	×	ANY_STRING.DID
Standard global label setting file		×	×	○	×	GLBLINF.IFG
Initial label value file	Initial global label value file	×	×	○	×	GLBLINF.LID
	Initial local label value file	×	×	○	×	PROGRAM_NAME.LID
File register		×	○	×	○*3	ANY_STRING.QDR
Event history		×	×	○	○	EVENT.LOG
Device data storage file		×	×	○	○*3	DEVSTORE.QST
General-purpose data		×	×	○	○	ANY_STRING.CSV/BIN
Data logging setting file	Common setting file	×	×	×	○	LOGCOM.LCS
	Individual setting file	×	×	○	○	LOGnn.LIS*2
Remote password		×	×	○	○	00000001.SYP
Safety program*7		◎*4*6	×	◎*4*6	×	ANY_STRING.SPG
Safety FB file*7		○*4	×	○*4	×	ANY_STRING.SPB
Safety CPU parameter*7*8		×	×	◎	×	CPU.SPR
Safety module parameter*7		×	×	○	×	UNIT.SPR
Safety global label setting file*7		×	×	○	×	GLBLINF.SIF
Standard/safety shared label setting file*7		×	×	○	×	S_GLBLINF.SIF

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

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

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

*4 When this file is stored in the built-in memory of the Safety CPU, it is divided into program memory and data memory and stored. (☞ Page 85 Configuration of a program file)

*5 Module extension parameter for the protocol setting, storing protocol setting information in the predefined protocol support function

*6 One or more of either standard program or safety program is required.

*7 The file is also stored in the safety function module.

*8 This parameter relates to safety control inside the CPU parameters. (☞ Page 723 Safety CPU parameters)
This item is yellow in the engineering tool CPU parameters.

File operation available

The following lists file operations which are available for each file type.

In TEST MODE

The following lists file operations which are available for each file type in TEST MODE.

○: Available, ×: Not available, —: N/A

File type		Operation using engineering tool			Operation with SLMP and FTP server function			Operation with instructions in standard programs	
		Write	Read	Delete	Write	Read	Delete	Write	Read
Standard program		○*1	○	○*2	×	×	×	—	—
Standard FB file		○*2	○	○*2	×	×	×	—	—
Standard CPU parameter		○*2	○	○*2	×	×	×	—	—
System parameter		○*2	○	○*2	×	×	×	—	—
Standard module parameter		○*2	○	○*2	×	×	×	—	—
Module extension parameter		○*2	○	○*2	×	×	×	—	—
Memory card parameter		○*2	○	○*2	×	×	×	—	—
Device comment		○	○	○*2	×	×	×	—	—
Initial device value		○	○	○*2	×	×	×	—	—
Standard global label setting file		○*2	○	○*2	×	×	×	—	—
Label initial value file	Global label initial value file	○	○	○*2	×	×	×	—	—
	Local label initial value file	○	○	○*2	×	×	×	—	—
File register		○	○	○*2	×	×	×	○	○
Event history		—	—	—	×	×	×	—	—
Device data storage file		—	—	—	×	×	×	○	○
General-purpose data		○	○	○	○	○	○	○	○
Data logging setting file	Common setting file	○*3	○*3	○*3	○	○	○	—	—
	Individual setting file	○*3	○*3	○*3	○	○	○	—	—
Remote password		○*2	○	○*2	×	×	×	—	—
Safety program		○*1	○	○*2	×	×	×	—	—
Safety FB file		○*2	○	○*2	×	×	×	—	—
Safety CPU parameter*4		○*2	○	○*2	×	×	×	—	—
Safety module parameter		○*2	○	○*2	×	×	×	—	—
Safety global label setting file		○*2	○	○*2	×	×	×	—	—
Standard/safety shared label setting file		○*2	○	○*2	×	×	×	—	—

*1 To write files to the programmable controller, use the write function when the operating status of the Safety CPU is STOP. When the operating status is RUN, use the online change function. When the Safety CPU operating status is RUN, only the standard programs registered in the CPU parameter can be written to the programmable controller.

*2 Operation is possible only when the operating status of the Safety CPU is STOP/PAUSE. If performed in RUN state, the operating status of the Safety CPU is changed by the remote STOP function, and then the operation continues.

*3 This indicates operations performed using the CPU Module Logging Configuration Tool.

*4 This parameter relates to safety control inside the CPU parameters. (☞ Page 723 Safety CPU parameters)
This item is yellow in the engineering tool CPU parameters.

In SAFETY MODE

The following lists file operations which are available for each file type in SAFETY MODE.

○: Available, ×: Not available, —: N/A

File type		Operation using engineering tool			Operation with SLMP and FTP server function			Operation with instructions in standard programs	
		Write	Read	Delete	Write	Read	Delete	Write	Read
Standard program		○*1*5	○	○*2	×	×	×	—	—
Standard FB file		○*2*5	○	○*2	×	×	×	—	—
Standard CPU parameter		×	○	×	×	×	×	—	—
System parameter		×	○	×	×	×	×	—	—
Standard module parameter		×	○	×	×	×	×	—	—
Module extension parameter		○*2	○	○*2	×	×	×	—	—
Memory card parameter		○*2	○	○*2	×	×	×	—	—
Device comment		○	○	○*2	×	×	×	—	—
Initial device value		○	○	○*2	×	×	×	—	—
Standard global label setting file		○*2*5	○	○*2	×	×	×	—	—
Label initial value file	Global label initial value file	○	○	○*2	×	×	×	—	—
	Local label initial value file	○	○	○*2	×	×	×	—	—
File register		○	○	○*2	×	×	×	○	○
Event history		—	—	—	×	×	×	—	—
Device data storage file		—	—	—	×	×	×	○	○
General-purpose data		○	○	○	○	○	○	○	○
Data logging setting file	Common setting file	○*3	○*3	○*3	○	○	○	—	—
	Individual setting file	○*3	○*3	○*3	○	○	○	—	—
Remote password		○*2	○	○*2	×	×	×	—	—
Safety program		×	○	×	×	×	×	—	—
Safety FB file		×	○	×	×	×	×	—	—
Safety CPU parameter*4		×	○	×	×	×	×	—	—
Safety module parameter		×	○	×	×	×	×	—	—
Safety global label setting file		×	○	×	×	×	×	—	—
Standard/safety shared label setting file		×	○	×	×	×	×	—	—

*1 To write files to the programmable controller, use the write function when the operating status of the Safety CPU is STOP. When the operating status is RUN, use the online change function. When the Safety CPU operating status is RUN, only the standard programs registered in the CPU parameter can be written to the programmable controller.

*2 Operation is possible only when the operating status of the Safety CPU is STOP/PAUSE. If performed in RUN state, the operating status of the Safety CPU is changed by the remote STOP function, and then the operation continues.

*3 This indicates operations performed using the CPU Module Logging Configuration Tool.

*4 This parameter relates to safety control inside the CPU parameters. (☞ Page 723 Safety CPU parameters)
This item is yellow in the engineering tool CPU parameters.

*5 Writing is possible in file units, however, it is not possible to select files together with parameter files or files relating to safety and then write to the programmable controller CPU. If writing in file units, write after clearing the selection of other parameter files or files relating to safety, or write after switching to TEST MODE.

File size

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

File type		File size
Standard program		Approx. 4248 bytes minimum (only END instruction + 500 steps reserved for online program change)
Standard FB file		Approx. 4984 bytes minimum (non-processing FB + 500 steps reserved for online program change)
Standard CPU parameter		824 bytes minimum
System parameter		112 bytes minimum
Standard module parameter		The size differs depending on the module used. For example, the size is 1036 bytes at minimum when the Ethernet function is used.
Module extension parameter		The size differs depending on the module used. For example, the size is 65572 bytes when the predefined protocol support function is used with the Ethernet function.
Memory card parameter		124 bytes minimum
Device comment		220 bytes minimum (when a device comment with 20 characters is set in a device)
Initial device value		Approx. 192 bytes minimum (when the initial value of a word device is set)
Standard global label setting file		Approx. 716 bytes minimum (when a word-type global label is set)
Initial label value file	Initial global label value file	Approx. 140 bytes minimum (when the initial value of a word type global label is set)
	Initial local label value file	
File register		2048 bytes minimum (when the setting capacity is 1K words)
Event history		1024 bytes minimum (when the setting capacity is 1K words (default setting is 131072 bytes))
Device data storage file		2048 bytes minimum (when the setting capacity is 1K words)
General-purpose data		The size differs depending on the size of the file to be written.
Data logging setting file	Common setting file	142 bytes (when the setting is configured so that one-word data is collected in a binary file at each scan in continuous logging)
	Individual setting file	1192 bytes (when the setting is configured so that one-word data is collected in a binary file at each scan in continuous logging only for setting No.1)
Remote password		224 bytes minimum
Safety program		Approx. 4288 bytes minimum (only END instruction + 500 steps reserved for online change)
Safety FB file		Approx. 5028 bytes minimum (non-processing FB + 500 steps reserved for online change)
Safety CPU parameter		292 bytes minimum
Safety module parameter		The size differs depending on the module used. (Example: When the safety communication setting is used with a remote I/O station: 860 bytes)
Safety global label setting file		Approx. 752 bytes minimum (when a word-type safety global label is set)
Standard/safety shared label setting file		Approx. 748 bytes minimum (when a word-type standard/safety shared label is set)

43 FUNCTIONS

This chapter describes the functions added to the Safety CPU and the functions different (modified or restricted) from the standard CPU. Functions not described in this chapter are same as those of the standard CPU. (☞ Functions in Part 2)
For the availability of Safety CPU functions, refer to the following.
📖 MELSEC iQ-R CPU Module User's Manual (Startup)

Function		Description	Reference
Function added to the Safety CPU	Safety operation mode	There are two safety operation modes: TEST MODE and SAFETY MODE. One mode is for changing the safety program or safety parameters, and the other is for operating the system as a safety system.	Page 488 Safety Operation Mode
	Continuous RUN prevention in TEST MODE	Prevents the Safety CPU from running continuously for a long time in TEST MODE.	Page 492 Continuous RUN Prevention in TEST MODE
	Safety diagnostic function	Self-diagnostic function of the Safety CPU	Page 493 Safety Diagnostic Function
	Safety data identify check	Checks if the project data created using the engineering tool and the data in the Safety CPU are the same, and confirms that the program executed in SAFETY MODE is the one written by the user.	Page 494 Safety Data Identify Check
	Safety communication function	Communicates data between the Safety CPU and modules supporting safety functions using safety protocols.	Page 494 Safety Communication Function

Function			Description	Reference
Function different (modified or restricted) from the standard CPU	Clock function		When setting clock data for the Safety CPU at the "Set Clock" window of the engineering tool, "All Stations Specified" cannot be selected as an execution target.	—
	Online change		• Online changes (ladder block) are not possible in safety programs. • Safety program files, safety global label setting files, and standard/safety shared global label setting files cannot be changed online. However, if standard/safety shared labels inside standard programs are edited, standard programs can be changed online. • If a disconnection, power-off or reset of the CPU module is detected while the program restoration information is being written, the processing will be suspended and a message will be displayed. If the processing is suspended by a disconnection, click the [Retry] button in the message after reconnecting cables. If the processing is suspended by power-off or reset, login to the user authentication function is disabled. Log in to the function and write data again to the programmable controller.	—
	Interrupt function	Interrupt period setting	The interrupt pointer (I) cannot be used in safety programs. Also, the following cannot be used in standard programs. • Inter-module synchronous interrupt (I44) • Multiple CPU synchronous interrupt (I45) • High-speed internal timer interrupts (I48, I49)	MELSEC iQ-R CPU Module User's Manual (Startup)
		Multiple interrupt function	Safety cycle processing is performed as a higher priority interrupt than other processing, and therefore even when set to "Enable" in the multiple interrupt settings, interrupts will never occur during safety cycle processing.	—
	PID control function		The PID control instructions cannot be used in safety programs.	MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)
	Remote operation		When specifying the execution destination at the "Remote Operation" window of the engineering tool, the specification of all stations and group No. is not supported, and CPU module cannot issue the command if being issued by the own station. All stations and group No. specification is not executed by CPU module if those in which all stations and group No. have been specified are received. Also, if the Safety CPU is in SAFETY MODE when clearing device/label memory when performing remote RUN, safety global devices, safety global devices, safety global labels, safety local labels, and standard/safety shared labels are not cleared.	—
	Device/label initial value setting		An initial value cannot be set for safety global devices, safety local devices, safety global labels, safety local labels, and standard/safety shared labels. To set an initial value to those devices and labels, create a program so that the safety state can be secured and the value is set only at the start-up.	—
	Output Mode Setting of STOP to RUN		The safety output (SA\Y) for the safety output device (SA\Y) is off, regardless of this setting. If the value is changed with a device test or others while the CPU module is in the STOP state, the value will be applied when the status of the CPU module changes from STOP → RUN.	Page 505 Safety output (SA\Y)
	Latch function		Safety devices, safety labels, and standard/safety shared labels cannot be latched.	—

Function			Description	Reference
Function different (modified or restricted) from the standard CPU	Device/label access service processing setting		Even though this function is set, the scan time of some functions (operations) becomes longer than the specified time. The following are added to those functions. • Safety local devices (label) batch/registration monitor • Current value change of the safety labels/safety devices • Switching safety operation mode • Safety data identify check	—
	Data logging function		There are restrictions on the sampling data, execution conditions, and trigger conditions.	Page 500 Data Logging Function
	RAS function	Self-diagnostic function	The behavior for each setting differs.	Page 501 How to check errors
		Error clear	Errors in the Safety CPU are added. Errors of the safety function module can also be cleared.	Page 502 Error Clear
		Event history function	Events in the Safety CPU are added. Note that events that occur at the safety function module are not retained in the event history.	Page 596 Event list
	Multiple CPU system function	Data communications between CPU modules	The Safety CPU supports only data communications using the CPU buffer memory. Data communications using the fixed scan communication area is not supported. The data communications using the fixed scan communication area is available between the CPU modules other than the Safety CPU.	—
		Multiple CPU synchronous interrupt	Multiple CPU synchronous interrupt (I45) is not supported since the execution of safety programs is prioritized.	—
	SLMP communication function		Safety devices cannot be accessed. Also, the command execution status differs.	Page 483 File operation available
	Ethernet function	File transfer function (FTP server)		

43.1 Safety Operation Mode

There are two safety operation modes: TEST MODE and SAFETY MODE. Running normally as a safety system or changing a safety program and safety parameter can be selected.

Safety operation mode	Description
SAFETY MODE	This is a mode for operating the safety system controlled by the Safety CPU. In this mode, Safety CPU safety programs and safety parameters cannot be changed. Only device data values in safety programs can be changed.
TEST MODE	This is a mode for performing maintenance (such as setting changes and tests) of the safety system controlled by the Safety CPU. In this mode, Safety CPU safety programs and safety parameters can be changed. Device data can be changed by performing a device test.
SAFETY MODE (wait-for-restart)*1	The system switches to this mode during the period from immediately after safety operation mode is changed to SAFETY MODE when in TEST MODE until the CPU module is turned off or is reset. Even if safety operation mode is switched to SAFETY MODE, safety operation mode becomes this mode until the CPU module is powered on or is reset. Programs are not executed in this mode. An error occurs if the CPU module is the RUN state.

*1 The same restrictions apply to this mode as those for SAFETY MODE.

Checking the safety operation mode

The safety operation mode can be checked in three ways.

- TEST LED of the safety function module (MELSEC iQ-R CPU Module User's Manual (Startup))
- "Safety Operation Mode Switch" window of the engineering tool (GX Works3 Operating Manual)
- Module information list of the engineering tool (GX Works3 Operating Manual)
- Safety special register of the Safety CPU (SA\SD205) (Page 682 List of Safety Special Register Areas)

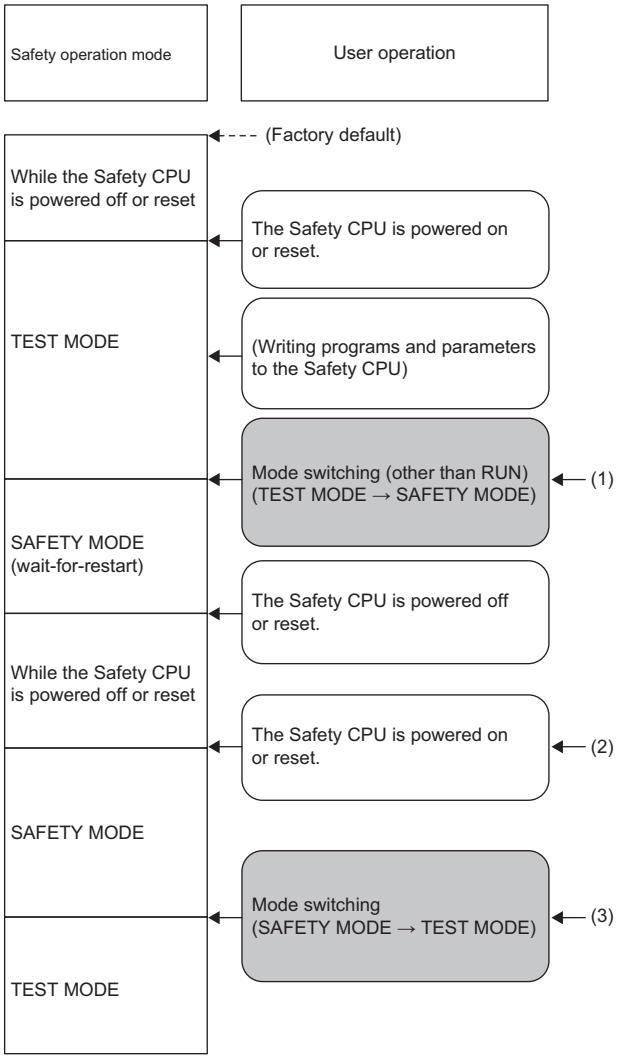
Switching the safety operation mode

This section describes how to switch the safety operation mode.

Safety operation mode transition timing

The timing at which safety operation mode switches is shown below.

Safety operation mode switching direction	Safety operation mode switch timing
TEST MODE → SAFETY MODE	When the CPU module is powered off and on or is reset:
SAFETY MODE → TEST MODE	When "safety operation mode changes" are made



(1) After switching from TEST MODE to SAFETY MODE, the safety operation mode will be SAFETY MODE (wait-for-restart). (The status will be wait-for-restart until the CPU module is powered off or is reset.) An error occur if the Safety CPU is switched to RUN status when in SAFETY MODE (wait-for-restart).

(2) After switching the safety operation mode (TEST MODE → SAFETY MODE), the mode will switch to SAFETY MODE when the CPU module is powered off and on or is reset.

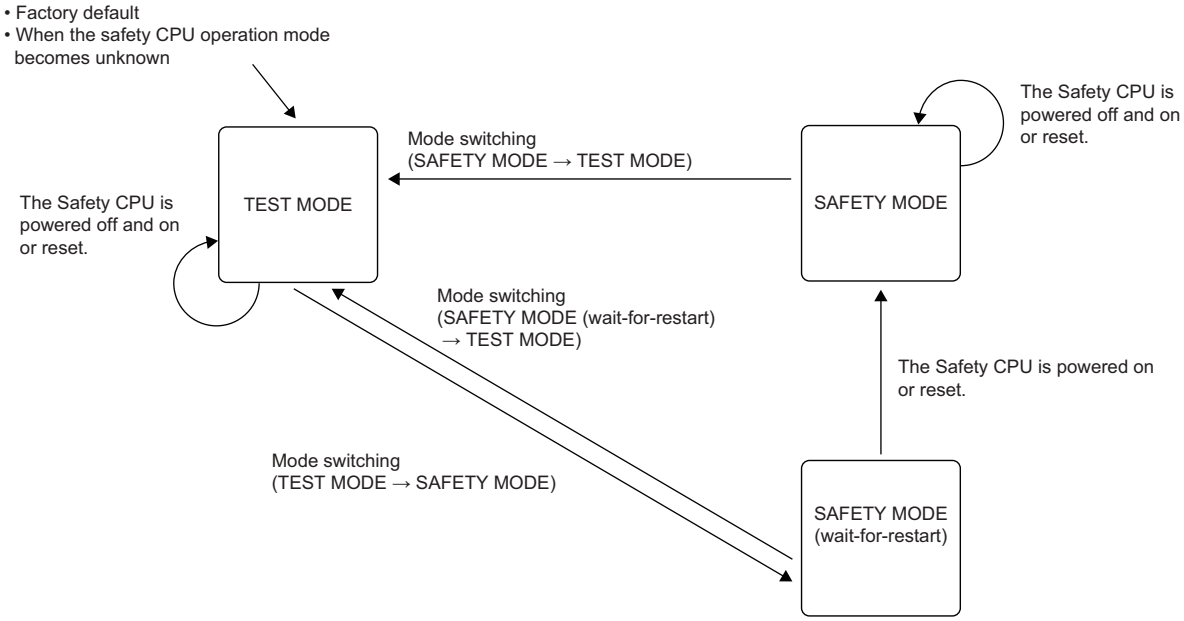
(3) After switching the safety operation mode (SAFETY MODE → TEST MODE), the mode will switch to TEST MODE.

Point

The Safety CPU retains the safety operation mode status even when the power is turned off, or a reset is performed.

Safety operation mode state transition

The Safety CPU retains the safety operation mode even in the event of a power failure, and therefore the current safety operation mode state will remain unchanged even if the CPU module is powered off and on or is reset. Safety function modules do not retain the safety operation mode in the event of a power failure, and the safety operation mode will be the same as that of the Safety CPU when the CPU module is powered off and on or is reset.



Safety operation mode state	Status transition by operation		
	Power off → on or reset	Safety operation mode switching operation	
		TEST MODE → SAFETY MODE	SAFETY MODE → TEST MODE
TEST MODE	→ TEST MODE	→ SAFETY MODE (wait-for-restart)	—
SAFETY MODE	→ SAFETY MODE	—	→ TEST MODE
SAFETY MODE (wait-for-restart)	→ SAFETY MODE	—	→ SAFETY MODE (wait-for-restart)

Point

If [Online] ⇨ [User Authentication] ⇨ [Initialization of all PLC Data] is performed, the safety operation mode will revert to the default mode (TEST MODE). All programmable controller information can be initialized regardless of the safety operation mode.

Safety operation mode switching conditions

The conditions under which the safety operation mode can be changed are shown below.

Safety operation mode switching direction	Condition
TEST MODE → SAFETY MODE	The user who is making the switch has "Developers" access level or higher, and is currently logged on.
	The safety operation mode is currently set to the TEST MODE.
	The Safety CPU operating status is other than RUN.
	The following files held by the engineering tool, Safety CPU, and safety function module match. • Safety CPU parameter • Safety module parameter • Safety program • Safety FB file • Safety global label setting file • Standard/safety shared label setting file
	The following files are not being written from another engineering tool. • Safety CPU parameter • Safety module parameter • Safety program • Safety FB file • Safety global label setting file • Standard/safety shared label setting file
SAFETY MODE → TEST MODE	The user who is making the switch has "Developers" access level or higher, and is currently logged on.
	The safety operation mode is currently set to SAFETY MODE or SAFETY MODE (wait-for-restart).

How to switch the safety operation mode

The safety operation mode can be switched using the engineering tool.

[Online] ⇨ [Safety PLC Operation] ⇨ [Safety Operation Mode Switch]

For details, refer to the following.

GX Works3 Operating Manual

Operations restricted in SAFETY MODE

When the safety operation mode is set to SAFETY MODE, the following operations cannot be performed.

Operation		Restrictions
Writing data to the programmable controller		Data cannot be written to files relating to safety control such as safety programs or safety CPU parameters. (☞ Page 483 File operation available)
Device/label test		Device/label tests (value changes) cannot be conducted from external devices for the following. (This includes value changes from the test function or monitor windows, and change requests from other external devices.) • Safety global devices • Safety local devices • Safety global labels • Safety local labels • Standard/safety shared label
Memory operation		Memory operations for the CPU built-in memory (including extended SRAM cassette) cannot be performed from the engineering tool. (☞ Page 481 Memory Operation) However, SD memory card initialization is possible.
Security function	Security key authentication function	It is not possible to register a key for the Safety CPU or extended SRAM cassette.
	File password function	It is not possible to register a password for the Safety CPU.

43.2 Continuous RUN Prevention in TEST MODE

This function prevents the Safety CPU from running continuously for a long time in TEST MODE. When the continuous RUN time has exceeded the allowed time, a continuation error occurs.

Measuring the continuous RUN time in TEST MODE

When the Safety CPU enters RUN state in TEST MODE, measurement of RUN time starts. When any of the following operations is performed, the Safety CPU stops the measurement and clears the measured value.

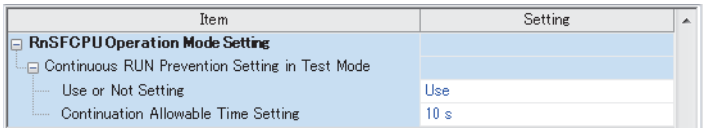
- The operating status of the Safety CPU is changed to STOP or PAUSE.
- The Safety CPU is powered off or is reset.
- The mode is changed to SAFETY MODE.
- An error is cleared.

Setting method

Set the continuous RUN allowed time in TEST MODE.

[CPU Parameter] ⇨ [RAS Setting] ⇨ [RnSFCPU Operation Mode Setting]

Window



Displayed items

Item		Description	Setting range	Default
Continuous RUN prevention setting in TEST MODE	Use or Not Setting	Select whether to use the function or not.	• Use • Not Use	Use
	Continuation Allowable Time Setting	Set the continuous RUN allowed time in TEST MODE.	1 to 86400 seconds	10 seconds

Checking the continuous RUN time in TEST MODE

The continuous RUN time in TEST MODE can be checked at SD1840 and SD1841 (Continuous RUN time in TEST MODE). (☞ Page 635 List of Special Register Areas)

Also, if the permissible TEST MODE continuous RUN time set in the parameters is exceeded, this can be checked by ensuring that SM1840 (Allowed time over flag for continuous RUN in TEST MODE) is on. (☞ Page 606 List of Special Relay Areas)

43.3 Safety Diagnostic Function

The following lists self-diagnostic functions specific to the Safety CPU.

Item[*1]		Description	Diagnostic timing	Error code
Memory diagnosis	RAM diagnosis	Detects errors occurring at program memory, device memory, and memory used by the system.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3C20, 3C21, 3C22, 3C2F, 3C33
	F/W diagnosis	Diagnoses whether the firmware stored in the ROM is corrupt.	• When power is turned off and on • When the CPU module is reset	3C10
Program diagnosis	Operation circuit diagnosis	Detects errors at operation circuits used to operate safety programs.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3C15
	File verification	Detects errors in stored safety programs and safety parameters.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	2180
		Diagnoses whether safety programs and safety parameters stored in both the Safety CPU and safety function module are the same.	• When power is turned off and on • When the CPU module is reset	2180
Operation result verification	Output data verification	Verifies output results calculated in the Safety CPU and safety function module.	When executing safety cycle processing	3E60
System diagnosis	Time monitoring	Monitors the execution status of the respective Safety CPU and safety function module processing.	Always	3E61
	Microcomputer diagnosis	Detects microcomputer internal register errors.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3C00, 3C01
Power supply voltage monitoring	Power supply voltage monitoring	Detects voltage errors to prevent operation with voltage outside the guaranteed operating range.	Always	— (Performs a shutdown.)
	Power supply voltage monitoring/ shutdown circuit diagnosis	Monitors whether the circuit monitoring the power supply voltage and circuit used to perform shutdown are functioning properly.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3E10, 3E11
Reset circuit monitoring	Reset circuit monitoring diagnosis	Diagnoses whether the reset has been performed correctly.	• When power is turned off and on • When the CPU module is reset	3E12
Clock stop detection	Clock diagnosis	Detects clock errors (frequency errors).	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3C16
CRC calculation circuit diagnosis	CRC calculation circuit diagnosis	Diagnoses whether the CRC calculation circuit is capable of correctly calculating CRC.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3E01
Module	Module OS diagnosis	Verifies whether the Safety CPU and safety function module are functioning normally without running out of control.	• When power is turned off and on • When the CPU module is reset • When END processing is performed	3C11, 3C17

*1 Self-diagnostics is performed in both Safety CPU and safety function module.

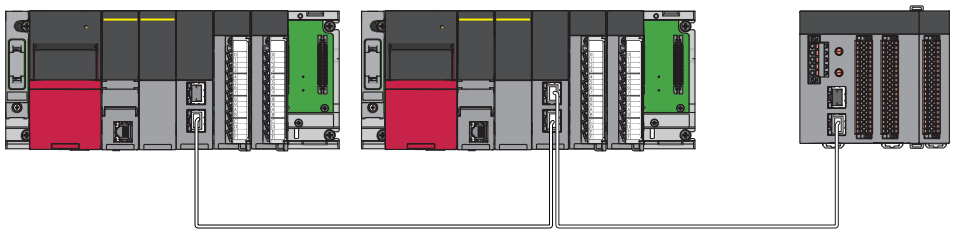
43.4 Safety Data Identify Check

This function checks if the project data created using the engineering tool and the data in the Safety CPU are the same, and confirms that the program executed in SAFETY MODE is the one written by the user.
This function compares files in the engineering tool with files stored in the Safety CPU, and shows the comparison result.

[Online] ⇨ [Safety PLC Operation] ⇨ [Safety Data Identify Check]
For details, refer to the following.
GX Works3 Operating Manual

43.5 Safety Communication Function

This function communicates data between the Safety CPU and modules supporting safety functions using safety protocols.



The Safety CPU performs safety communications between safety stations using the following network.

- CC-Link IE Field Network (safety communication function)

For details on the function and settings required for safety communications, refer to the manuals for the network used.

43.6 Device/Label Memory Area Setting

The capacity of each area on the device/label memory can be specified.

Default capacity

The default capacity of each area is as follows.

Item	R08SFCPU	R16SFCPU	R32SFCPU	R120SFCPU
Standard device area	40K words	40K words	40K words	40K words
Safety device area	20K words	20K words	20K words	20K words
Safety label area	20K words	20K words	20K words	20K words
Safety local device area	0K words	0K words	0K words	0K words
Standard/safety shared label area	10K words	10K words	10K words	10K words
Standard label area	40K words	50K words	90K words	110K words
Standard latch label area	2K words	2K words	4K words	4K words
Standard local device area	0K words	0K words	0K words	0K words
File storage area	457K words	713K words	969K words	1481K words

Point

The standard local device area capacity obtained by subtracting the total capacity of the standard device area, standard label area, standard latch label area, file storage area, and safety devices/safety labels, and the total standard/safety shared label area from the total device/label memory is set. Note, however, that even if the total capacity of the standard device area, standard label area, and safety devices/safety labels is smaller than the following, the remaining capacity cannot be assigned to the standard local device area. (The remaining area will be an unused area.)

- R08SFCPU: 50K words
- R16SFCPU: 60K words
- R32SFCPU: 70K words
- R120SFCPU: 90K words

Point

The safety local device area capacity obtained by subtracting the total capacity of the safety device area and safety label area from the safety device/label memory capacity is set.

Setting range of capacity of each area

The setting range of the capacity of each area on the device/label memory is as follows.*1

*1 The remaining capacity of other areas is automatically set as the capacity of the standard local device area and safety local device area. (☞ Page 495 Default capacity)

R08SFCPU

Area	Setting range of capacity of each area				
	Without an extended SRAM cassette	With an extended SRAM cassette (1MB)	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (4MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 588K words	2 to 1100K words	2 to 1612K words	2 to 2636K words	2 to 4684K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 586K words	0 to 1098K words	0 to 1610K words	0 to 2634K words	0 to 4682K words
Standard latch label area	0 to 544K words	0 to 1056K words	0 to 1568K words	0 to 2592K words	0 to 4640K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 544K words	0 to 1056K words	0 to 1568K words	0 to 2592K words	0 to 4640K words

R16SFCPU

Area	Setting range of capacity of each area				
	Without an extended SRAM cassette	With an extended SRAM cassette (1MB)	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (4MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 854K words	2 to 1366K words	2 to 1878K words	2 to 2902K words	2 to 4950K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 852K words	0 to 1364K words	0 to 1876K words	0 to 2900K words	0 to 4948K words
Standard latch label area	0 to 800K words	0 to 1312K words	0 to 1824K words	0 to 2848K words	0 to 4896K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 800K words	0 to 1312K words	0 to 1824K words	0 to 2848K words	0 to 4896K words

R32SFCPU

Area	Setting range of capacity of each area				
	Without an extended SRAM cassette	With an extended SRAM cassette (1MB)	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (4MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 1152K words	2 to 1664K words	2 to 2176K words	2 to 3200K words	2 to 5248K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 1150K words	0 to 1662K words	0 to 2714K words	0 to 3198K words	0 to 5246K words
Standard latch label area	0 to 1088K words	0 to 1600K words	0 to 2112K words	0 to 3136K words	0 to 5184K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 1088K words	0 to 1600K words	0 to 2112K words	0 to 3136K words	0 to 5184K words

R120SFCPU

Area	Setting range of capacity of each area				
	Without an extended SRAM cassette	With an extended SRAM cassette (1MB)	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (4MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 1684K words	2 to 2196K words	2 to 2708K words	2 to 3732K words	2 to 5780K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 1682K words	0 to 2194K words	0 to 2706K words	0 to 3730K words	0 to 5778K words
Standard latch label area	0 to 1600K words	0 to 2112K words	0 to 2624K words	0 to 3648K words	0 to 5696K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 1600K words	0 to 2112K words	0 to 2624K words	0 to 3648K words	0 to 5696K words

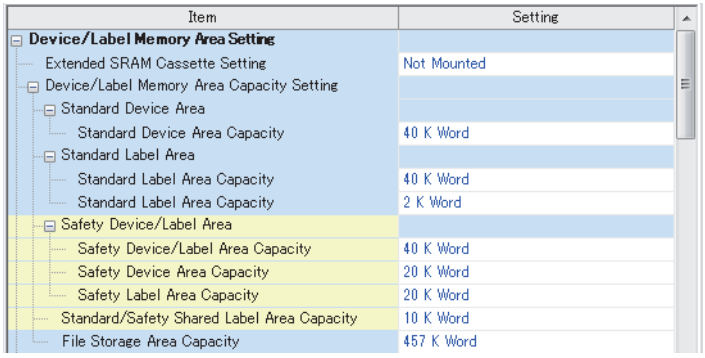
Setting method

The capacity in each data area of the device/label memory can be changed.

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

Operating procedure

"Device/Label Memory Area Setting" window



1. Set whether to use an extended SRAM cassette or not in "Extended SRAM Cassette Setting".
2. Set the capacity of each area in "Device/Label Memory Area Capacity Setting".

Displayed items

Item			Description	Setting range	Default
Extended SRAM Cassette Setting			Set the capacity of the extended SRAM cassette if used.	• Not Mounted • 1MB • 2MB • 4MB • 8MB	Not Mounted
Device/Label Memory Area Capacity Setting	Standard Device Area	Standard Device Area Capacity	Set the capacity of the device area used for standard global devices.	☞ Page 496 Setting range of capacity of each area	☞ Page 495 Default capacity
	Standard Label Area	Standard Label Area Capacity	Set the capacity of the label area used for standard global labels and standard local labels.		
		Standard Latch Label Area Capacity	Set the capacity of the standard latch label area used for latch-type labels.		
	Safety Device/ Label Area	Safety Device/ Label Area Capacity	Set the total capacity for the safety device area and safety label area.		
		Safety Device Area Capacity	Set the capacity of the device area used for safety global devices.		
		Safety Label Area Capacity	Set the capacity of the label area used for safety global labels and safety local labels.		
	Standard/Safety Shared Label Area Capacity		Set the capacity of the label area used for standard/safety shared labels.		
	File Storage Area Capacity		Set the capacity of the file storage area used to store files, such as file register files.		

Point

Note that the total of the capacity of each area (including the capacity of the local device area/safety local device area) and safety special relay/safety special register area capacity should not exceed the capacity of the device/label memory. (MELSEC iQ-R CPU Module User's Manual (Startup))

Standard device area setting range

The number of points of each device used in standard programs and capacity in which the total number of device points can be stored are set. Set the total number of device points within the device area range.

Type	Device name	Symbol	Range of use*1	Setting unit
Bit	Input	X	X0 to X2FFF	—
	Output	Y	Y0 to Y2FFF	—
	Internal relay	M	M0 to M94674943	64 points
	Link relay	B	B0 to B5A49FFF	64 points
	Annunciator	F	F0 to F32767	64 points
	Link special relay	SB	SB0 to SB5A49FFF	64 points
	Edge relay	V	V0 to V32767	64 points
	Latch relay	L	L0 to L32767	64 points
Word	Timer	T	T0 to T5259711	32 points
	Retentive timer	ST	ST0 to ST5259711	32 points
	Long timer	LT	LT0 to LT1479295	1 point
	Long retentive timer	LST	LST0 to LST1479295	1 point
	Counter	C	C0 to C5259711	32 points
	Long counter	LC	LC0 to LC2784543	32 points
	Data register	D	D0 to D5917183	4 points
	Link register	W	W0 to W5A49FF	4 points
	Link special register	SW	SW0 to SW5A49FF	4 points

*1 This is the maximum range when the R120SFCPU with an extended SRAM cassette (8MB) (NZ2MC-8MBS) is used. The number of points varies depending on the model of the Safety CPU, whether to use an extended SRAM cassette, and the type of the cassette.

Safety device area setting range

The number of points of each safety device used in safety programs and capacity in which the total number of safety device points can be stored are set. Set the total number of safety device points within the safety device area range.

Type	Device name	Symbol	Range of use	Setting unit
Bit	Safety input	SA\X	SA\X0 to SA\X2FFF*1	—
	Safety output	SA\Y	SA\Y0 to SA\Y2FFF*1	—
	Safety internal relay	SA\M	SA\M0 to SA\M638975	64 points
	Safety link relay	SA\B	SA\B0 to SA\B9BFFF	64 points
Word	Safety timer	SA\T	SA\T0 to SA\T35487	32 points
	Safety retentive timer	SA\ST	SA\ST0 to SA\ST35487	32 points
	Safety counter	SA\C	SA\C0 to SA\C35487	32 points
	Safety data register	SA\D	SA\D0 to SA\D39935	4 points
	Safety link register	SA\W	SA\W0 to SA\W9BFF	4 points

*1 If using the range from 2000 to 2FFF, check the versions of the CPU module and engineering tool. (☞ Page 779 Added and Enhanced Functions)

43.7 Data Logging Function

This section describes restrictions when the data logging function is used.

Data to be collected

Only data in standard devices can be collected. (☞ Page 188 Data to be collected)

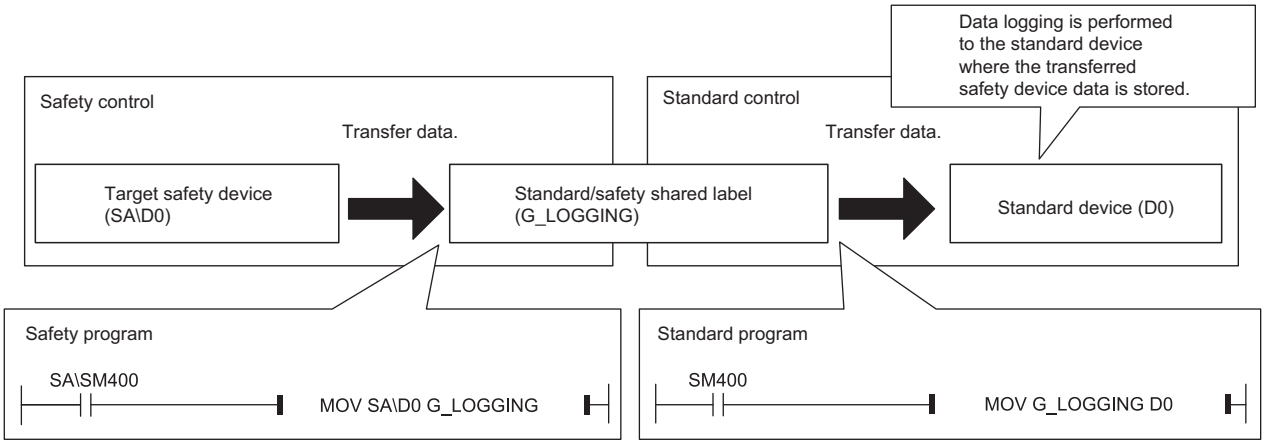
Point

Although data logging cannot be executed by specifying safety devices and safety labels, data in safety devices and safety labels can be collected by taking the following steps (❶ to ❸).

❶: Transfer the values of the target safety device or safety label to a standard/safety shared label.

❷: Transfer the values of the standard/safety shared label specified in Step ❶ to a standard device.

❸: Logs standard devices specified in Step ❷.



Data collection conditions

As a sampling interval, only "Each scanning cycle" can be set.

Trigger conditions

Only standard devices can be specified as monitoring data of when "Device change specification" is selected as the trigger condition. "Step No. specification" can be selected only for standard programs.

Output settings

With the Safety CPU, even if set to "Output with device comment" in the output settings "Device comment output" item, device comments are not output to the file.

Auto logging

If a sampling interval of the data logging function is set to the one other than "Each scanning cycle", data logging registration fails at the start of auto logging and an error occurs.

43.8 Self-Diagnostic Function

This section describes the self-diagnostic function of the Safety CPU module and safety function module. As for the Safety CPU, only the parts that differ from the standard CPU are described.

How to check errors

Errors in the Safety CPU can be checked in the same way as for the standard CPU. (☞ Page 229 Self-Diagnostics Function)
Errors in the safety function module can be checked in the following ways.

Using the buffer memory

When the safety function module detects an error, the corresponding error code is stored in Un\G0 (Latest self-diagnostic error code). If multiple errors are detected, the latest error code is stored in Un\G0. Up to 16 error codes can be stored in Un\G10 (Self-diagnostic error code 1) to Un\G25 (Self-diagnostic error code 16). (The error codes of the 17th error and later are not stored.)

Using the LED of the safety function module

Like intelligent function modules, the error status can be checked with the ERROR LED. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

Using the engineering tool

Like intelligent function modules, the error status of the entire system and the history of errors currently occurred or events can be checked on the "Module Diagnostics" window. (📖 GX Works3 Operating Manual)

Operation setting in the CPU parameter (CPU Module Operation Setting at Error Detected)

The "Instruction Execution Error" setting under the "CPU Module Operation Setting at Error Detected" is applied only for standard programs. A stop error always occurs in safety programs. (Even if "Continue" is selected, a stop error occurs.)

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

43.9 Error Clear

This function clears all the existing continuation errors occurring at the Safety CPU or safety function module at once.

Errors that can be cleared

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

■Safety CPU

Error code	Error name
1000H	Power shutoff
1080H	ROM write count error
1090H	Battery error
1100H	Memory card access error
1120H	SNTP clock setting error
1124H	Default gateway/gateway IP address error
1128H	Own node port number error
1129H	Open specification port number error
112DH	Specified IP address error
112EH	Connection establishment failed
1130H	IP address duplication error
1133H	Socket communications response send error
1134H	TCP connection timeout
1152H	IP address error
1155H	Connection number acquisition error
1157H	Receive buffer securement error
1165H	UDP/IP transmission failed
1166H	TCP/IP transmission failed
1167H	Unsend data transmission error
1200H, 1210H	Module moderate error
1220H	Another CPU module moderate error
1800H	Annunciator ON
1830H	Receive queue full
1831H	Receive processing error
1832H	Transient data error
1900H	Constant scan time exceeded
1A00H	Safety cycle time exceeded
1A01H	Safety cycle processing error
1A20H	Continuous RUN time in TEST MODE exceeded
1A40H*1	Safety communications partner station error
1A50H*1	Safety communications partner station parameter mismatch
1A51H*1	Safety communications partner station and device mismatch
1A52H*1	Safety communications partner station and device version mismatch
1A60H*1, 1A61H*1, 1A62H*1, 1A63H*1, 1A64H*1, 1A65H*1, 1A66H*1	Safety communications timeout
1A70H*1, 1A71H*1, 1A72H*1, 1A73H*1	Safety communications receipt data error
2120H, 2121H	Memory card error
2400H, 2401H	Module verification error
2420H	Fuse blown error
2441H, 2442H	Major error in module
2450H	Module major error
2461H, 2462H	Another CPU module major error
2470H	Another CPU module major error
2610H	Inter-module synchronization signal error
2630H	Multiple CPU synchronization signal error

Error code	Error name
2800H, 2801H, 2802H, 2803H, 2804H, 2805H, 2806H, 2807H, 2810H	I/O number or network number specification error
2820H, 2821H, 2822H, 2823H, 2824H	Device, label, or buffer memory specification error
2840H, 2841H	File name specification error
3400H, 3401H, 3402H, 3403H, 3404H, 3405H, 3406H, 3420H, 3421H, 3422H, 3440H, 3441H, 34A0	Operation error

*1 If a safety communications error (minor error) occurs between safety stations, a safety station interlock will be triggered for safety communications between relevant stations, and safety communications will not be resumed until the safety station interlock is released. The safety station interlock is released with a safety special register (safety station interlock release request for each safety connection). (☞ Page 682 List of Safety Special Register Areas)
For details on the safety station interlock status, refer to the following.
📖 MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

■Safety function module

Error code	Error name
1080H	ROM write count error
1A00H	Safety cycle time exceeded
1A40H	Safety communications partner station error
1A50H	Safety communications partner station parameter mismatch
1A51H	Safety communications partner station and device mismatch
1A52H	Safety communications partner station and device version mismatch
1A60H, 1A61H, 1A62H, 1A63H, 1A64H, 1A65H, 1A66H	Safety communications timeout
1A70H, 1A71H, 1A72H, 1A73H	Safety communications receipt data error

How to clear errors

Errors in the Safety CPU can be cleared in the same way as for the standard CPU. The following describes how to clear errors of the safety function module.

■Using the engineering tool

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

■Using the buffer memory

Clear errors using operations of buffer memory.

1. Check the standard program and identify the detected continuation error using Un\G0 (Latest self-diagnostic error code).
2. Eliminate the cause of the detected continuation error.
3. Set 1 in Un\G50 (Error clear) in the standard program to clear the error.*1

*1 After the error is cleared, 0 is automatically set in Un\G50 by the system.

44 SAFETY DEVICES AND CONSTANTS

This chapter describes the safety devices.

44.1 Safety Devices

A safety device is a device used in safety programs. Data can be read from or written to safety devices when used in safety programs. Data cannot be read from or written to safety devices when used in standard programs.

List of safety devices

The following table lists the safety devices.

Classification	Type	Device name	Symbol	Number of points (default)	Parameter-set range	Notation
Safety user devices	Bit	Safety input	SA\X	8K points	Selectable either 8K or 12K points*1	Hexadecimal
	Bit	Safety output	SA\Y	8K points		Hexadecimal
	Bit	Safety internal relay	SA\M	6K points	Changeable (☞ Page 499 Safety device area setting range)	Decimal
	Bit	Safety link relay	SA\B	4K points		Hexadecimal
	Bit/word	Safety timer	SA\T	512 points		Decimal
	Bit/word	Safety retentive timer	SA\ST	0 points		Decimal
	Bit/word	Safety counter	SA\C	512 points		Decimal
	Word	Safety data register	SA\D	12K points		Decimal
	Word	Safety link register	SA\W	4K points		Hexadecimal
Safety system devices	Bit	Safety special relay	SA\SM	4K points	Unchangeable	Decimal
	Word	Safety special register	SA\SD	4K points		Decimal

*1 If selecting 12K points, check the version of the CPU module and engineering tool. (☞ Page 779 Added and Enhanced Functions)

Point

For details on standard devices used for the Safety CPU, refer to the following.
☞ Page 312 DEVICES

Safety user devices

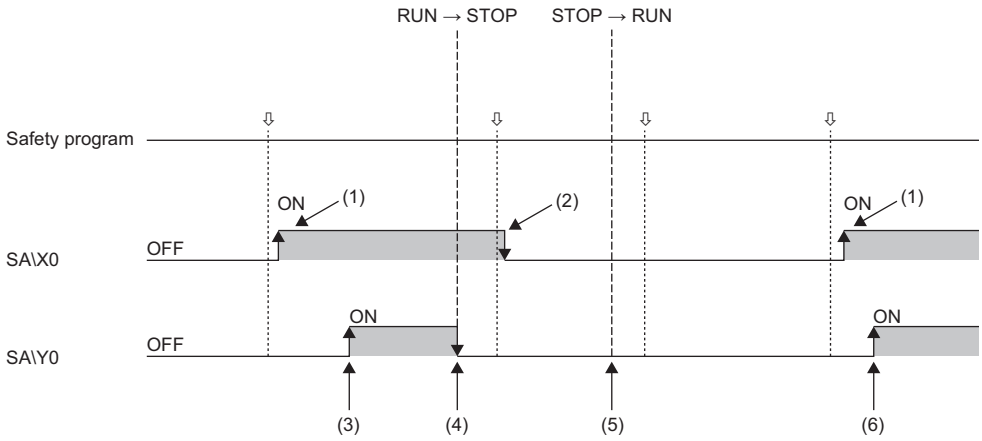
This section describes the safety user devices.

Safety input (SA\X)

Safety inputs are used to give instructions or data to the Safety CPU with external devices such as emergency stop buttons, safety plugs, door switches, and light curtains.

Safety output (SA\Y)

Safety outputs are used to output safety program control results to external relays, contactors, robots, and motion, etc.
When the status changes from STOP → RUN, the value immediately before the RUN status is retained, and the value is updated the moment safety cycle processing is executed for the first time after changing to the RUN status.



⇩: Safety cycle processing start
(1) Turns on when safety communications are received.
(2) Turns off when safety communications are received.
(3) SA\Y0 turns on when SA\X0 turns on in the safety program.
(4) SA\Y0 turns off when the status of the CPU module is switched to STOP.
(5) The value is maintained when the status changes from STOP → RUN.
(6) Turns on when a safety program is executed.

Safety internal relay (SA\M)

This device is used as an auxiliary relay within the Safety CPU. The following operations turn off all safety internal relays.

- Safety CPU power-off → on
- Reset

Safety link relay (SA\B)

The safety link relay is used as a device when sending and receiving safety data between Safety CPUs over CC-Link IE Field Network.

■Refreshing network modules using safety link relay

Data is sent and received between Safety CPUs. The safety communications send/receive device range is set in the CC-Link IE Field Network safety communication settings. Locations that are not used for safety communications send/receive devices can be used for other applications.

Safety timer (SA\T)/safety retentive timer (SA\ST)

This device starts measurement when the safety timer coil is turned on. When the current value reaches a setting value, time is up and the contact is turned on. This safety timer is an up-timing type device and therefore the current value matches a setting value when the safety timer time is up. Operations other than the following are the same as those for the timer. (☞ Page 319 Timer)

■Safety timer types

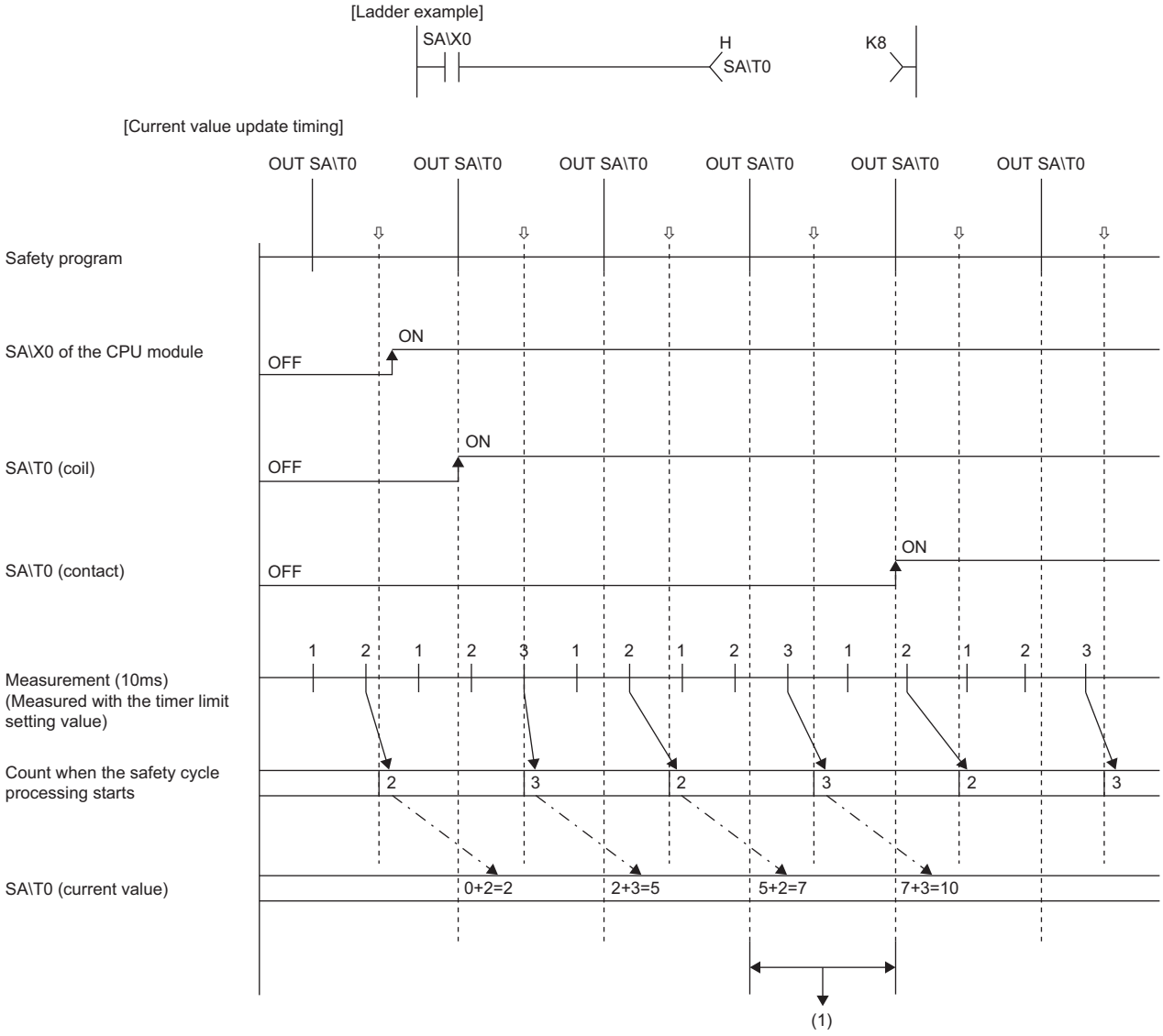
There is a safety timer (SA\T) that retains the current value in 16-bit units. In addition, there is a safety retentive timer (SA\ST) that retains the current value even if the coil turns off.

■Safety timer accuracy

The current value is measured when starting safety cycle processing. The value for the elapsed time since the previous safety cycle processing was started until the present is added to the current value when the OUT SA\T☐ instruction is executed. If the safety timer coil is off at the execution of the OUT SA\T☐ instruction, the current value is not updated. The maximum response accuracy of the timer is the "elapsed time since previous safety cycle processing was started until the present + timer limit setting".

Ex.

Timer limit setting = 10ms, SA\T0 setting value = 8



⇩: Safety cycle processing start

(1) Time accuracy - (Elapsed time since the previous safety cycle processing was started until the present + Timer limit setting) to (Elapsed time since the previous safety cycle processing was started until the present)

(2) Sets the coefficient when starting safety cycle processing.

Safety counter (SA\C)

This device counts the number of rising operation of the input condition in the program. The safety counter is an up-timing type device and therefore when the count value matches a setting value, the count reaches its upper limit and the contact is turned on. Operations other than the following are the same as those for the counter. (☞ Page 325 Counter)

■Safety counter types

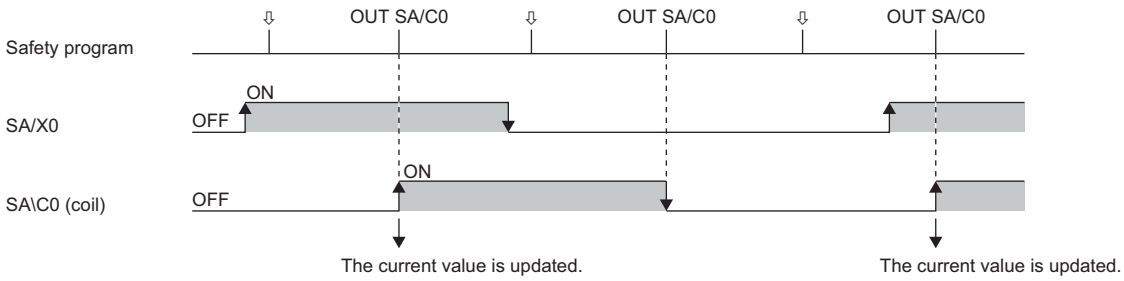
There is a safety counter (SA\C) that retains the counter value in 16-bit units.

■Counting process

When executing the safety counter coil, the safety counter coil is turned on/off, the current value is updated (count value +1), and the contact is turned on/off. The current value is updated (count value +1) when the safety counter coil input is rising (off to on). The current vale is not updated when the coil input is off, on to on, and on to off.

[Ladder example]

[Current value update timing]



⇩: Safety cycle processing start

■Resetting the safety counter

The safety counter current value is not cleared even when the counter coil input is turned off. To clear (reset) the safety counter current value and turn off the contact, issue the RST SA\C☐ instruction. When the RST SA\C☐ instruction is executed, the counter value is cleared and the contact is turned off.

Safety data register (SA\D)

This device can store numerical values.

Safety link register (SA\W)

This register is used as a device when sending and receiving safety data between Safety CPUs over CC-Link IE Field Network.

■Refreshing network modules using safety link register

Data is sent and received between Safety CPUs. The safety communications send/receive device range is set in the CC-Link IE Field Network safety communication settings. Locations that are not used for safety communications send/receive devices can be used for other applications.

Safety system devices

Details of the safety system devices are described below.

Safety special relay (SA\SM)

This relay stores the Safety CPU status relating to safety control. (☞ Page 680 List of Safety Special Relay Areas)

Safety special register (SA\SD)

This register stores the Safety CPU status relating to safety control. (☞ Page 682 List of Safety Special Register Areas)

44.2 Constants

The decimal constant (K) and hexadecimal constant (H) can be used in safety programs. The specification method is the same as that for standard programs. (Example: K1234, H1FFF) (☞ Page 347 CONSTANTS)

44.3 How to Use Safety Devices Conveniently

This section describes how to use safety devices conveniently. For details on how to use standard devices conveniently, refer to the following.

☞ Page 350 HOW TO USE DEVICES CONVENIENTLY

Restriction

An index modification and indirect specification cannot be performed in safety programs.

Safety global devices

A safety global device is a device that can be shared by all the safety programs.

Safety local devices

A safety local device is a device used individually by each safety program.

Devices that can be used as a safety local device

The following devices can be used as a safety local device.

- Safety internal relay (SA\M)
- Safety timer (SA\T)
- Safety retentive timer (SA\ST)
- Safety counter (SA\C)
- Safety data register (SA\D)

How to set safety local devices

The range and usability of the safety local devices can be set in the same way for the standard local devices. (☞ Page 353 Setting method for the local devices)

Point

Set safety local devices within the number of device points set in the CPU parameter. The number of safety local device points used is calculated by the following calculation formula. Set the number of safety local device points used so that the number is equal to or less than the capacity of the safety local device area.

Total number of safety local device points used = ((A ÷ 16) + B + (C × 2) ÷ 16)) × D

- A: Number of points of the safety local device (SA\M)
- B: Number of points of the safety local devices (SA\D, SA\T (current value), SA\ST (current value), and SA\C (current value))
- C: Number of points of the safety local devices (SA\T (contact/coil), SA\ST (contact/coil), and SA\C (contact/coil))
- D: Number of programs using the safety local device

How to specify safety local devices

To specify safety local devices in a safety program, add "#".

Ex.
SA\#D100, SA\K4#M0

Point

In the program, safety local devices are displayed with "#" in front of the device name. This helps users to distinguish local devices from global devices.

SM777 setting

Safety programs are executed only as fixed scan execution type programs, and use storage location program file safety local devices, regardless of the SM777 (Local device setting in interrupt programs) setting.

APPENDICES

Appendix 1 Error Codes

The CPU module stores the corresponding error code in the special register (SD) upon detection of an error by the self-diagnostic function. If an error occurs when the data communications are requested from the engineering tool, intelligent function module, or network system connected, the CPU module returns the corresponding error code to the request source. The error details and cause can be identified by checking the error code. The error code can be checked in either of the following ways:

- Module diagnostics by the engineering tool (MELSEC iQ-R CPU Module User's Manual (Startup))
- Special register (SD0 (Latest self-diagnostics error code), SD10 to SD25 (Self-diagnostic error number)) (Page 635 List of Special Register Areas)

This section describes errors that may occur in the CPU module and actions to be taken for the errors.

Error code system

All error codes are given in hexadecimal format (4 digits) (16-bit unsigned integer). The type of error includes the error, which is detected through the self-diagnostic function of each module, and the common error, which is detected during data communications between modules. The following table lists the error detection type and the error code ranges.

Error detection type		Range	Description
By the self-diagnostic function of each module	Minor error	1000H to 1FFFH	Error code specific to each module, such as self-diagnostic errors
	Moderate error	2000H to 3BFFH	
	Major error	3C00H to 3FFFH	
During data communications between modules		4000H to 4FFFH	Error in the CPU module
		6F00H to 6FFFH	Error in the redundant function module
		7000H to 7FFFH	Error in the serial communication module
		B000H to BFFFH	Error in the CC-Link module
		C000H to CFBFH	Error in the Ethernet-equipped module
		CFC0H to CFFFH	Error in CC-Link IE Field Network Basic
		D000H to DFFFH	Error in the CC-Link IE Field Network module
		E000H to EFFFH	Error in the CC-Link IE Controller Network module
		F000H to FFFFH	Error in the MELSECNET/H network module or MELSECNET/10 network module

Detailed information

Upon detection of an error by the self-diagnostic function, the detailed information of the error cause is stored together with an error code. The detailed information can be checked using the engineering tool. The following detailed information is added to each error code. (Up to two types of information are stored together with an each error code. The types differ depending on the error code.) Detailed information 1 and 2 for the latest error code can also be checked in the special register (SD). (☞ Page 635 List of Special Register Areas)

Detailed information	Item	Description
Detailed information 1	Error location information*1	Information on the location in a program
	Drive/file information	Information on the corresponding drive name and file name
	Parameter information	Information on the parameter, such as parameter storage location and parameter type
	System configuration information	Information on the system configuration, such as I/O numbers and power supply numbers
	Frequency information	Information on the frequency, such as the number of writes to memory
	Time information	Information on time
	TEST MODE continuous RUN prevention setting	Continuous RUN allowable time in TEST MODE
	Failure information	Information on failures
	System switching information	Information on the system switching cause or system switching failure cause
	Data type (tracking transfer) information	Information on tracking transfer setting data
	Tracking transfer trigger information	Information on tracking block numbers when data are transferred
Detailed information 2	Drive/file information	Information on the corresponding drive name and file name
	Annunciator information	Information on the annunciator areas
	Parameter information	Information on the parameter, such as parameter storage location and parameter type
	System configuration information	Information on the system configuration, such as I/O numbers and power supply numbers
	Process control instruction processing information	Information on processing blocks of the process control instructions
	Error information of other stations (CC-Link IE Field)	Information on errors occurred in other stations during safety communications*2
	Program error information	Program error code

*1 The step number which is displayed in the error location information, is the one that is counted from the start of the file. It may differ from the step number of the program which is displayed in error jump of the engineering tool.

*2 For details on error classifications and error codes, refer to the manual of each device connected. Note that if the connected device is a Safety CPU, and the error classification is "350", the error code that occurred at the connection destination is displayed for the error code in decimal notation. Furthermore, "0s" in the occurrence date and time information, and in high-order digits are omitted. For example, if the occurrence time (hhmmss) is 09:10:05, the displayed value will be 91005.

Operation when an error occurs

There are two types of errors: continuation errors and stop errors.

Stop error

If a stop error occurs, the CPU module stops its operation and the operating status changes to STOP. Modules can communicate with the CPU module even after the stop error occurs in the CPU module. The external output of each module is controlled in accordance with the output mode setting in error. (☞ Page 230 CPU module operation upon error detection setting) Besides, if the stop error occurs when the multiple CPU system is configured, the stop error target CPU module (all CPU modules or only the corresponding CPU module) can be set in parameter. (☞ Page 276 Stop setting)

Continuation error

If a continuation error occurs, the CPU module continues its operation. (The operating status remains the same.)

Safety output operation when error occurs

If a stop error occurs, safety output turns off, and safety communications do not function. Other safety input device/safety label (including internal safety devices) data is retained.

How to clear errors

Continuation errors can be cleared. For the error clear method, refer to the following.

- CPU module: ☞ Page 233 Error Clear
- Safety function module: ☞ Page 502 Error Clear

In a redundant system, an error of the CPU module in the standby system can be cleared from the CPU module in the control system. (☞ Page 451 Clearing an Error on the Standby System CPU Module from the Control System CPU Module)

List of error codes

Codes of errors detected by the self-diagnostic function (1000H to 3FFFH)

The following table lists error codes detected by the self-diagnostic function of the CPU module, redundant function module, and safety function module.

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1000H	Power shutoff	• A momentary power failure has occurred. • The power supply has been shut off.	• Check the power supply status.	—	Always
1010H	Power shutoff (either of the redundant power supply modules)	• Power has been shut off or the power supply voltage has dropped in either of the redundant power supply modules on the redundant power supply base unit. Or, only one redundant power supply module is mounted.	• Check the power supplied to the power supply modules on the base unit. • Check that two power supply modules are mounted on the base unit. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.	System configuration information	Always
1020H	Failure (either of the redundant power supply modules)	• A failure has been detected in either of the redundant power supply modules on the redundant power supply base unit.	• The possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.	System configuration information	Always
1030H	Invalid power supply module	• An invalid power supply module has been mounted on the redundant power supply base unit.	• Mount only applicable power supply modules. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.	System configuration information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1031H	Power supply module configuration error	• A power supply module other than the redundant power supply module has been mounted on the redundant power supply base unit.	• Mount only applicable power supply modules. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.	System configuration information	Always
1080H	ROM write count error	■CPU module • The number of writes to the flash ROM (data memory, program memory, and system memory (memory used by the system when the CPU module executes its function)) exceeded 100000 times. ■Safety function module • The number of writes to the flash ROM exceeded 100000 times. (Number of writes > 100000)	■CPU module • Replace the CPU module. ■Safety function module • Replace the safety function module.	Frequency information	At power-on, at RESET, at write
1090H	Battery error	• The voltage of the battery built in the CPU module has dropped below the specified value. • The connector of the battery built in the CPU module is disconnected. • The connector of the battery built in the CPU module is not securely connected.	• Replace the battery. • Connect the battery connector to use the file storage area in the device/label memory or the latch function. • Check the connection status of the battery connector. If it is loose, securely connect the connector.	—	Always
1100H	Memory card access error	• Data cannot be written to the memory card because the write protect switch is locked.	• Unlock the write protect switch of the memory card.	—	Always
1120H	SNTP clock setting error	• Time setting has failed when the CPU module is powered on or reset.	• Check if the time is correctly set in parameter. • Check if the specified SNTP server is operating normally and there is no failure on the network accessing the SNTP server computer.	—	At power-on, at RESET
1124H	Default gateway/ gateway IP address error	• The default gateway is not set correctly. • The gateway IP address is not set correctly. • The default gateway/gateway IP address (network address after the subnet mask) is different from that of the IP address of the own node.	• Correct the default gateway IP address. • Set the same network address as that of the IP address.	Parameter information	Always
1128H	Own node port number error	• The port number is incorrect.	• Correct the port number.	—	Always
1129H	Open specification port number error	• The port number setting of the external device is incorrect.	• Correct the port number of the external device.	—	Always
112DH	Specified IP address error	• The IP address setting of the external device for the open processing is incorrect.	• Correct the IP addresses. • Check if the class of the IP address is set to A/B/C.	—	Always
112EH	Connection establishment failed	• A connection could not be established in the open processing.	• Check the operation of the external device. • Check if the open processing has been performed in the external device. • Check and correct the port number of the module, IP address/port number of the external device, and opening method. • When the firewall is set in the external device, check if the access is permitted. • Check if the Ethernet cable is disconnected.	—	Always
1133H	Socket communications response send error	• The response send failed during socket communications.	• Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub.	—	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1134H	TCP connection timeout	• A TCP ULP timeout error has occurred in the TCP/IP communication. (The external device does not send an ACK response.)	• Check the operation of the external device. • Check and correct the TCP ULP timeout value. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected.	—	Always
1152H	IP address error	• The IP address is not set correctly.	• Correct the IP addresses.	Parameter information	Always
1155H	Connection number acquisition error	• The specified connection was already closed in TCP/IP communications. • Open processing is not performed.	• Perform the open processing for the specified connection. • Check if the open processing has been performed in the external device.	—	Always
1157H	Receive buffer securement error	• The specified connection was already closed in UDP/IP communications. • Open processing is not performed.	• Perform the open processing for the specified connection. • Check if the open processing has been performed in the external device.	—	Always
1165H	UDP/IP send failed	• Data was not sent correctly with UDP/IP.	• Check the settings for connection with the external device. • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute a PING test. If the test completes with an error, take an action to correct the error.	—	Always
1166H	TCP/IP send failed	• Data was not sent correctly with TCP/IP.	• Check the settings for connection with the external device. • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute a PING test. If the test completes with an error, take an action to correct the error.	—	Always
1167H	Unsend data send error	• Unsent data found, but could not be sent.	• Check the settings for connection with the external device. • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute a PING test. If the test completes with an error, take an action to correct the error.	—	Always
1180H	Redundant IP address error	• The same IP address has been set as the system A IP address, system B IP address, and/or control system IP address • Network addresses of the system A IP address, system B IP address, and control system IP address are different.	• Set different IP addresses for the system A IP address, system B IP address, and control system IP address. • Set the same network address for the system A IP address, system B IP address, and control system IP address.	Parameter information	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
11A0H	PID operation error	• A value outside the range ($T_S \leq 0$) was specified for the sampling time (T_S).	• Check and correct the sampling time (T_S) value.	Error location information	At instruction execution
11A1H	PID operation error	• A value outside the range ($\alpha < 0$ or $100 \leq \alpha$) was specified for the input filter constant (α).	• Check and correct the input filter constant (α) value.	Error location information	At instruction execution
11A2H	PID operation error	• A value outside the range ($K_P < 0$) was specified for the proportional gain (K<1075>P</1075>).A value outside the range ($K_P < 0$) was specified for the proportional gain (K<1075>P</1075>).	• Check and correct the proportional gain (K_P) value.	Error location information	At instruction execution
11A3H	PID operation error	• A value outside the range ($T_I < 0$) was specified for the integral time (T_I).	• Check and correct the integral time (T_I) value.	Error location information	At instruction execution
11A4H	PID operation error	• A value outside the range ($K_D < 0$ or $201 \leq K_D$) was specified for the derivative gain (K_D).	• Check and correct the derivative gain (K_D) value.	Error location information	At instruction execution
11A5H	PID operation error	• A value outside the range ($T_D < 0$) was specified for the derivative time (T_D).	• Check and correct the derivative time (T_D) value.	Error location information	At instruction execution
11A6H	PID operation error	• A value less than or equal to the operation cycle was specified for the sampling time (T_S).	• Check and correct the sampling time (T_S) value.	Error location information	At instruction execution
11A7H	PID operation error	• A process value variation (ΔPV) overflow occurred.	• Check and correct the control data setting values.	Error location information	At instruction execution
11A8H	PID operation error	• A deviation (EV) overflow occurred.	• Check and correct the control data setting values.	Error location information	At instruction execution
11A9H	PID operation error	• An integral calculation value overflow occurred.	• Check and correct the control data setting values.	Error location information	At instruction execution
11AAH	PID operation error	• A derivative gain (K_D) value overflow occurred.	• Check and correct the control data setting values.	Error location information	At instruction execution
11ABH	PID operation error	• A differential calculation value overflow occurred.	• Check and correct the control data setting values.	Error location information	At instruction execution
11ACH	PID operation error	• A PID operation result overflow occurred.	• Check and correct the control data setting values.	Error location information	At instruction execution
11ADH	PID operation error	• A value smaller than the output lower limit setting value was specified for the output upper limit setting value.	• Check and correct the output upper limit setting value or output lower limit setting value.	Error location information	At instruction execution
11AEH	PID operation error	• A value smaller than 0 was specified for the input variation amount warning setting value or output variation amount warning setting value.	• Change the input variation amount warning setting value or output variation amount warning setting value to a value greater than 0.	Error location information	At instruction execution
11AFH	PID operation error	• The deviation when starting auto tuning (step response method) became 150 or less. • The deviation when finishing auto tuning (step response method) was 1/3 or greater than the deviation when starting.	• Check and correct the control data. • If the deviation when starting auto tuning (step response method) is not 150 or more, specify a set value for auto tuning.	Error location information	At instruction execution
11B0H	PID operation error	• The movement direction assumed from the relationship between the set value and process value when starting auto tuning (step response method) and the output value movement direction do not match.	• Perform auto tuning again after correcting the relationship between the set value, auto tuning (step response method) output value, and process value. • Check and correct the control data.	Error location information	At instruction execution
11B1H	PID operation error	• The input value (PV) did not change correctly, and auto tuning (step response method) did not function normally.	• Set a larger value for the sampling time (T_S). • Set a long time for the input filter constant (α). • Check the input value (PV).	Error location information	At instruction execution
11B2H	PID operation error	• A value equal to or lower than the output lower limit (LLV) was specified for the auto tuning (limit cycle method) output upper limit (ULV).	• Check and correct the output upper limit (ULV) and output lower limit (LLV) values.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
11B3H	PID operation error	• A value outside the range (SHPV < 0) was specified for the auto tuning (limit cycle method) process value threshold (hysteresis) width (SHPV).	• Check and correct the process value threshold (hysteresis) width (SHPV) setting value.	Error location information	At instruction execution
11B4H	PID operation error	• The system area used for auto tuning (limit cycle method) was rewritten.	• Check whether the occupied system area has been rewritten with the PID instruction.	Error location information	At instruction execution
11B5H	PID operation error	• As the auto tuning (limit cycle method) process time was exceeded, it was not possible to properly acquire the τ and τon time. (τon > τ, τon < 0, τ < 0)	• Increase the difference (ULV-LLV) between the auto tuning output value upper and lower limits, or reduce the input filter constant (α) and auto tuning process value threshold (hysteresis) width (SHPV) values.	Error location information	At instruction execution
11B6H	PID operation error	• An overflow occurred with the proportional gain (K_P) calculated when performing auto tuning (limit cycle method).	• Ensure that the process value (PV) change increases. • Reduce the difference (ULV-LLV) between the auto tuning output value upper and lower limits.	Error location information	At instruction execution
11B7H	PID operation error	• The integral time (T_I) calculated when performing auto tuning (limit cycle method) was outside the 0 to 32767 range.	• Increase the difference (ULV-LLV) between the auto tuning output value upper and lower limits, or reduce the input filter constant (α) and auto tuning process value threshold (hysteresis) width (SHPV) values.	Error location information	At instruction execution
11B8H	PID operation error	• The derivative time (T_D) calculated when performing auto tuning (limit cycle method) was outside the 0 to 32767 range.	• Increase the difference (ULV-LLV) between the auto tuning output value upper and lower limits, or reduce the input filter constant (α) and auto tuning process value threshold (hysteresis) width (SHPV) values.	Error location information	At instruction execution
1200H	Module moderate error	• A moderate error has been notified from the intelligent function module connected.	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, identify the error module, and eliminate the error cause.	System configuration information	Always
1210H	Module moderate error	• An inter-module synchronous signal error has been notified from the intelligent function module connected.	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, identify the error module, and eliminate the error cause.	System configuration information	Always
1220H	Another CPU module moderate error	• A moderate error has been notified from another CPU module.	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, identify the error module, and eliminate the error cause. • Check the mounting status and reset status of other CPU modules.	System configuration information	Always
1240H	Inter-module synchronization processing error	• The execution interval of a synchronous interrupt program has exceeded the set value. • The inter-module synchronous interrupt program (I44) did not complete within the inter-module synchronization cycle.	• Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take the following actions: (1) Check and correct the processing of the interrupt program (I44) so that it can be completed within the period specified in the fixed scan interval setting. (2) Check and correct the fixed scan interval setting value.	Time information	At interrupt occurrence
1241H	Inter-module synchronization processing error	• The execution interval of a synchronous interrupt program has exceeded the set value. • A cycle where the inter-module synchronous interrupt program (I44) was not executed was detected.	• Check and correct the interrupt disable sections and the interrupt programs with a high priority so that the inter-module synchronous interrupt program can be executed.	—	At interrupt occurrence

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1260H	Multiple CPU synchronization processing error	• The execution interval of a synchronous interrupt program has exceeded the set value. • The multiple CPU synchronous interrupt program (I45) did not complete within the multiple CPU fixed scan communication cycle.	• Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take the following actions: (1) Check and correct the processing of the interrupt program (I45) so that it can be completed within the period specified in the fixed scan interval setting. (2) Check and correct the fixed scan interval setting value.	Time information	At interrupt occurrence
1262H	Multiple CPU synchronization processing error	• The program execution section of a synchronous interrupt program has been exceeded. • The multiple CPU synchronous interrupt program (I45) did not complete within the program execution section.	• Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take the following actions: (1) Check and correct the processing of the interrupt program (I45) so that it can be completed within the program execution section. (2) Check and correct the fixed scan interval setting value.	Time information	At interrupt occurrence
1800H	Annunciator ON	• The annunciator (F) on status has been detected.	• Check the detailed error information (annunciator information) by executing module diagnostics using the engineering tool, and correct the program corresponding to the displayed annunciator number.	Error location information and annunciator information	At instruction execution
1810H	Operation continuation error	• The PALERT(P) instruction was executed.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, and check the corresponding program - one that outputs the program error code.	Error location information and program error information	At instruction execution
1811H	Control CPU error	• An error has been detected in the CPU module.	• Check the error details on the "Module Diagnostics" window, and take an action.	—	Always
1812H	Module error	• Tracking communications could not be performed successfully due to an error in the redundant function module.	• The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	—	Always
1830H	Receive queue full	• The number of transient receive request exceeded the upper limit of simultaneously processable requests.	• Reduce the frequency, and perform transmissions again. • Increase the frequency by using the COM instruction.	—	Always
1831H	Receive processing error	• Transient reception failed.	• Reduce the frequency, and perform transmissions again.	System configuration information	Always
1832H	Transient data error	• Too much transient transmission processing exists and transient transmission cannot be performed.	• Correct the number of transient transmission executions.	—	Always
1845H	Transient data buffer full	• Too much transient transmission processing exists and transient transmission cannot be performed.	• Correct the number of transient transmission executions.	—	Always
1860H	Network error	• Tracking communications stopped due to an error in the network or in the redundant function module.	• Check the network status on the "Module Diagnostics" window, and take an action. • If the error code is displayed again even after taking an action, please consult your local Mitsubishi representative.	—	Always
1861H	Tracking communication error	• Tracking communications could not be performed successfully due to an error in the network or in the redundant function module.	• Check the network status on the "Module Diagnostics" window, and take an action. • If the error code is displayed again even after taking an action, please consult your local Mitsubishi representative.	—	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1900H	Constant scan time error	• The scan time exceeded the constant scan time set in the CPU parameters.	• Check and correct the constant scan time setting.	Time information	At END instruction execution
1A00H	Safety cycle time exceeded	• Processing of the safety program, from safety input to safety output, did not complete within the safety cycle time.	• Check and correct the safety cycle time.	Time information	At interrupt occurrence
1A01H	Safety cycle processing error	• Safety cycle processing was not executed because of either of the following reasons: (1) The safety cycle processing takes too much time, and the safety cycle time was exceeded. (2) The safety cycle processing interrupt was not executed in the standard program due to an instruction with excessively long processing time.	• Check and correct the safety cycle time.	—	At interrupt occurrence
1A20H	Continuous RUN time in TEST MODE exceeded	• The continuous RUN time in TEST MODE was exceeded.	• Change the mode to SAFETY MODE.	TEST MODE continuous RUN prevention setting	Always
1A40H	Safety communications partner station	• Error information was received from the safety communication target station.	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the displayed station number. If the target station is a remote device station or remote I/O station, read error history data of the target station by executing a command from the CC IE Field Configuration window, and check the details of the error. If the target station is a master/local station, execute system monitor to the target station, and check the details of the error.	System configuration information, other station error information (CC-Link IE Field)	At interrupt occurrence
1A50H	Safety communications partner station parameter mismatch	• The parameters written to the CPU module and parameters written to the safety communication target station do not match.	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the displayed station number. Check the safety communication setting method and procedure referring to the manual of the safety communication target station, and write parameters to the safety communication target station and Safety CPU. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU data memory or the module performing safety communications. Please consult your local Mitsubishi representative.	System configuration information	At interrupt occurrence
1A51H	Safety communications partner station device mismatch	• The model or production information of the safety communication target station is incorrect. • The device actually connected to perform safety communications and the device set in the network configuration settings do not match.	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the displayed station number. Then, match the connected device and the device set in the network configuration settings. • Write parameters to the safety communication target station and Safety CPU. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU data memory or the module performing safety communications. Please consult your local Mitsubishi representative.	System configuration information	At interrupt occurrence

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1A52H	Safety communications partner station device version mismatch	• The version of the device actually connected to perform safety communications and the version of the device set in the network configuration settings do not match.	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the displayed station number. Then, match the versions of the connected device and the device set in the network configuration settings. • Write parameters to the safety communication target station and Safety CPU. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU data memory or the module performing safety communications. Please consult your local Mitsubishi representative.	System configuration information	At interrupt occurrence
1A60H to 1A66H	Safety communications timeout	• A timeout error occurred during safety communications.	• Check and correct the safety refresh monitoring time and the transmission interval monitoring time setting values. • The possible cause is malfunction due to noise or a cable error. Check that there is no error in the transmission path by executing CC-Link IE Field diagnostics. Check the distance of wires and cables and the grounding status of each device as well. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the displayed station number. If the target station is a remote device station or remote I/O station, read error history data of the target station by executing a command from the CC IE Field Configuration window, and check the details of the error. If the target station is a master/local station, execute system monitor to the target station, and check the details of the error. • Check that no online operation is being performed from a peripheral such as the engineering tool, or no program or parameter is being written to the safety communication target station. • The possible cause is a hardware failure of the module performing safety communications. Please consult your local Mitsubishi representative.	System configuration information	At interrupt occurrence
1A70H to 1A73H	Safety communications receipt data error	• The received data is abnormal.	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the displayed station number. If the target station is a remote device station or remote I/O station, read error history data of the target station by executing a command from the CC IE Field Configuration window, and check the details of the error. If the target station is a master/local station, execute system monitor to the target station, and check the details of the error. • The possible cause is a hardware failure of the module performing safety communications. Please consult your local Mitsubishi representative.	System configuration information	At interrupt occurrence

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1B00H	Network configuration mismatch	• The CC-Link IE Field Network configuration differs between the systems A and B. (The error is detected in the standby system.)	• Check that all network cables are connected correctly. • Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the network cable. Please consult your local Mitsubishi representative.	System configuration information	Always
1B20H	System consistency check error (operating status)	• The operating status of the CPU module differs between the systems A and B. (The error is detected in the standby system.)	• Set the same operating status to the CPU modules in both systems.	—	Always
1B40H	Redundant system error	• Both systems were found to be set as system A or system B when tracking cables were connected. (The error is detected in the control system or the standby system.)	• Set one system as system A and the other system as system B by performing online operation. Then, restart the CPU modules in both systems.	—	Always
1B42H	Redundant system error	• The CPU module operated in redundant mode was connected to the CPU module set to process mode with tracking cables. (The error is detected in the control system or the standby system.)	• Read the project stored in the CPU module operated in redundant mode, and write it to the CPU module set to process mode. Then, reset the CPU module set to process mode, and run it again.	—	Always
1B43H	Redundant system error	• The firmware version of the CPU module connected is not compatible with the other one.	• Replace one CPU module with the one having a compatible firmware version, referring to the MELSEC iQ-R Module Configuration Manual. Then, restart the system.	—	Always
1B60H	Standby system CPU module error	• In backup mode, the standby system CPU module has not started up. (The error is detected in the control system.)	• If the standby system is powered off, power on the system. • If the standby system CPU module is in the RESET state, clear the RESET state. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	—	Always
1B61H	Standby system CPU module error	• In backup mode, a stop error of the standby system CPU module was detected. (The error is detected in the control system.)	• Eliminate the error cause, and restart the system. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	—	Always
1B70H	Tracking communications disabled	• Data communications with the other system cannot be performed even in backup mode. (The error is detected in the control system or the standby system.)	• If the standby system is powered off, power on the system. • If a WDT error has occurred in the other system CPU module, eliminate the error cause, and restart the system. • Connect tracking cables securely and properly to the connectors of the redundant function modules in both systems. (One cable shall be connected between the IN connector of the module in system A and the OUT connector of the module in system B. The other cable shall be connected between the OUT connector of the module in system A and the IN connector of the module in system B.) • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.	—	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1B71H	Tracking communication error	• Data communications with the other system cannot be performed in one of the two tracking cables even in backup mode. (The error is detected in the control system or the standby system.)	• Connect tracking cables securely and properly to the connectors of the redundant function modules in both systems. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module or tracking cable. Please consult your local Mitsubishi representative.	—	Always
1B78H	Tracking communication error	• During tracking transfer, a tracking communication error occurred. (The error is detected in the control system or the standby system. Note that if the error is detected in the standby system, the detailed information is not stored.)	• If the other system is powered off, power on the system. • If a WDT error has occurred in the other system CPU module, eliminate the error cause, and restart the system. • Connect tracking cables securely and properly to the connectors of the redundant function modules in both systems. • Set the program execution time of the standby system shorter than that of the control system. If the shorter value cannot be set, correct the SD1662 value. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.	Data type (tracking transfer) information	At END instruction execution
1B80H	Tracking transfer error	• The tracking data set in the CPU parameters exceeds the allowable tracking transfer range. (The error is detected in the control system.)	• Set the device/label data within the allowable range in the CPU parameters ("Tracking Device/Label Setting").	Tracking transfer trigger information	At END instruction execution
1B81H	Tracking transfer error	• The file register capacity has been set in the CPU parameters of the control system CPU module is less than the file register capacity set for tracking transfer. (The error is detected in the control system.)	• Check and correct the tracking transfer settings in the CPU parameters so that they will be within the device range or the file register capacity. • Check and correct the device settings or the file register capacity setting in the CPU parameters.	—	At END instruction execution
1B82H	Tracking transfer error	• The file register data is transferred from the control system CPU module to the standby system CPU module exceeding the capacity of the receive side. (The error is detected in the standby system.)	• Check and correct the tracking transfer settings in the CPU parameters so that they will be within the device range or the file register capacity. • Check and correct the device settings or the file register capacity setting in the CPU parameters.	—	At END instruction execution
1BA0H	Redundant function module error	• An error has been detected in the redundant function module.	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the status of the redundant function module. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	Always
1BD0H	System switching error	• The systems were not switched. There was a cause of system switching failure. (The error is detected in the control system.)	• Check the cause of system switching failure in the detailed information (system switching information) of the error by executing module diagnostics using the engineering tool, eliminate the error cause, and switch the systems from the engineering tool again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or tracking cable. Please consult your local Mitsubishi representative.	System switching information	At system switching execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1BD1H	System switching error	• The systems were not switched by using the SP.CONTSW instruction because SM1646 (System switching by a user) was off. (The error is detected in the control system.)	• To switch the systems by using the SP.CONTSW instruction, turn on SM1646 and then execute the instruction.	Error location information	At instruction execution
2000H	Module configuration error	• The module type set in the system parameters ("I/O Assignment Setting") differs from that of the module actually mounted.	• Re-set the module type in the system parameters in accordance with the CPU module or intelligent function module actually mounted.	System configuration information	At power-on, at RESET
2001H	Module configuration error	• The I/O numbers set in the system parameters ("I/O Assignment Setting") are overlapping between modules.	• Re-set the I/O numbers in the system parameters in accordance with the intelligent function module or I/O module actually mounted.	System configuration information	At power-on, at RESET
2002H	Module configuration error	• The number of points assigned to the intelligent function module in the system parameters ("I/O Assignment Setting") is smaller than that of the module actually mounted.	• Re-set the number of points in the system parameters in accordance with the intelligent function module actually mounted.	System configuration information	At power-on, at RESET
2004H	Module configuration error	• Nine or more CC-Link IE Controller Network modules and/or MELSECNET/H network modules are mounted in the entire system. (The CC-Link IE built-in Ethernet interface module is included if the module is used as a CC-Link IE Controller Network module.) • Five or more MELSECNET/H network modules are mounted in the entire system.	• Reduce the number of CC-Link IE Controller Network modules and/or MELSECNET/H network modules to eight or less in the entire system. (The CC-Link IE built-in Ethernet interface module is included if the module is used as a CC-Link IE Controller Network module.) • Reduce the number of MELSECNET/H network modules to four or less in the entire system.	System configuration information	At power-on, at RESET
2005H	Module configuration error	• Two or more interrupt modules (QI60) with no interrupt pointer setting are mounted. • The interrupt pointer numbers are overlapping in the interrupt module (QI60) with no interrupt pointer setting and a module with an interrupt pointer setting.	• Mount only one QI60 in the entire system. • Configure the interrupt pointer setting for the QI60. • Correct the interrupt pointer setting.	System configuration information	At power-on, at RESET, at STOP → RUN state
2006H	Module configuration error	• A module is mounted on the 65th slot or later.	• Remove the module mounted on the 65th slot or later.	System configuration information	At power-on, at RESET
2007H	Module configuration error	• A module is mounted on the slot whose number is later than that specified in the system parameters ("I/O Assignment Setting").	• Remove the module mounted on the slot whose number is later than that specified in the system parameters.	System configuration information	At power-on, at RESET
2008H	Module configuration error	• A module is mounted over or across the maximum number of I/O points (4096).	• Remove the module mounted over or across the maximum number of I/O points (4096). • Replace the module mounted on the last slot to the one that does not exceed the maximum number of I/O points (4096).	System configuration information	At power-on, at RESET
2009H	Module configuration error	• There is no response from the I/O module or intelligent function module accessed.	• Check and correct the I/O assignment setting in the system parameters. • Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the I/O module or intelligent function module. Please consult your local Mitsubishi representative.	System configuration information	Always
2020H	Module configuration error	• There is a mounted module that is not supported, or there is a mounted module that is not supported by the network type (module model name) set in system parameters ("I/O Assignment Setting").	• Remove the unsupported module if any. • Check whether modules are supported by the network type (module model name) set in the system parameters. • If all the modules are supported, the possible cause is a hardware failure of the CPU module, base unit, I/O module, or intelligent function module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2021H	Module configuration error	• In a multiple CPU system, the control CPU of the Q series intelligent function module incompatible with the multiple CPU system is set to other than CPU No.1.	• Replace the Q series intelligent function module with the one (function version B) compatible with the multiple CPU system. • Set the control CPU of the Q series intelligent function module incompatible with the multiple CPU system to CPU No.1.	System configuration information	At power-on, at RESET
2022H	Module configuration error	• A power supply module other than the redundant power supply module has been mounted on the redundant power supply base unit.	• Mount only the redundant power supply module. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module, CPU module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2040H	CPU module configuration error	• The number of CPU modules set in the system parameters ("I/O Assignment Setting") differs from the number of CPU modules actually mounted. • The CPU module is mounted on the slot different from the one specified in the system parameters (I/O assignment setting).	• Correctly set the number of CPU modules (including the empty setting) in the system parameters in accordance with the number of CPU modules actually mounted. • Correctly set the system parameters so that the setting and actual CPU module mounting status will be the same.	System configuration information	At power-on, at RESET
2041H	CPU module configuration error	• The CPU module is not mounted on the slot that is set for the CPU module in the system parameters ("I/O Assignment Setting"). • The CPU module is mounted on the slot that is set for empty in the system parameters ("I/O Assignment Setting"). • An I/O module or intelligent function module is mounted between the CPU modules. • Arrangement of CPU modules mounted is not appropriate to configure a multiple CPU system.	• Check and correct the I/O assignment setting in the system parameters. • Correctly set the number of CPU modules (including the empty setting) in the system parameters in accordance with the number of CPU modules actually mounted. • Remove the I/O module or intelligent function module mounted between the CPU modules.	System configuration information	At power-on, at RESET
2043H	CPU module configuration error	• The CPU module is mounted on the inapplicable slot.	• Mount the CPU module on the applicable slot (CPU slot or I/O slot 0 to 6). • Remove the CPU module from the inapplicable slot.	System configuration information	At power-on, at RESET
2044H	CPU module configuration error	• The host CPU No. set in the system parameters ("I/O Assignment Setting") differs from the one determined by the mounting position of the CPU module.	• Re-set the host CPU No. in the system parameters in accordance with the mounting position of the CPU module.	System configuration information	At power-on, at RESET, at STOP → RUN state
2046H	CPU module configuration error	• Another CPU module was accessed, but there was no response.	• Check and correct the I/O assignment setting in the system parameters. • Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of another CPU module. Please consult your local Mitsubishi representative.	System configuration information	Always
2050H	CPU module configuration error	• An unsupported CPU module is mounted.	• Remove the unsupported CPU module. If all the CPU modules are supported, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2051H	CPU module configuration error	• A CPU module that does not support the online module change (direct change) function is used in a multiple CPU system.	• Refer to the MELSEC iQ-R Online Module Change Manual, and correct the system configuration. • To disable the online module change (direct change) function, set "Disable" to the "Direct change setting" in the CPU parameters. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2052H	CPU module configuration error	• A multiple CPU system was configured with CPU modules incompatible with multiple CPU systems.	• Check and correct the system configuration.	System configuration information	At power-on, at RESET
2060H	Base unit configuration error	• Eight or more extension base units are connected.	• Reduce the number of extension base units to seven or less.	System configuration information	At power-on, at RESET
2061H	Base unit configuration error	• Any of the following base units is connected: QA1S3□B, QA1S5□B/QA1S6□B, QA6□B, QA6ADP+A5□B/A6□B, or QA1S6ADP-S1+A1S5□B/A1S6□B.	• Remove the inapplicable base unit: QA1S3□B, QA1S5□B/QA1S6□B, QA6□B, QA6ADP+A5□B/A6□B, and QA1S6ADPS1+A1S5□B/A1S6□B.	System configuration information	At power-on, at RESET
2063H	Base unit configuration error	• Extension base unit levels are overlapping.	• Check and correct the level setting of the extension base units.	System configuration information	At power-on, at RESET
2070H	Base unit configuration error	• An unsupported base unit is connected. • A GOT is bus-connected to the Q series extension base unit.	• Disconnect the unsupported base unit. If all the base units are supported, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative. • Disconnect the GOT bus-connected to the Q series extension base unit.	System configuration information	At power-on, at RESET
2080H	Inter-module synchronization configuration error	• An inter-module synchronization signal error has been detected.	• The possible cause is a hardware failure of the CPU module, base unit, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
20E0H	Module unrecognized	■CPU module • A module that the CPU module cannot recognize is mounted. • In a multiple CPU system, the module cannot be recognized because the control CPU setting in the system parameters is different from that of other CPU modules. ■Redundant function module • Data communications with the CPU module cannot be performed.	■CPU module • Mount only applicable modules. • Correct the system parameter settings for the CPU No.2 and later. The settings need to be the same between all the CPU modules. • The possible cause is a hardware failure of the I/O module or intelligent function module accessed. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	System configuration information	Always
2100H	Memory error (extended SRAM cassette)	• The extended SRAM cassette set in the CPU parameters differs from the one actually inserted.	• Re-set the extended SRAM cassette in the CPU parameters in accordance with the one actually inserted. If the same error code is displayed again, the possible cause is a hardware failure of the extended SRAM cassette. Please consult your local Mitsubishi representative.	Drive/file information Parameter information	At power-on, at RESET
2101H	Memory error (extended SRAM cassette)	• The extended SRAM cassette is inserted or removed while the system is powered on.	• Do not insert or remove the extended SRAM cassette during operation. • Check if the extended SRAM cassette is securely inserted to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the extended SRAM cassette. Please consult your local Mitsubishi representative.	Drive/file information	Always
2102H	Memory error (extended SRAM cassette)	• An error has been detected in the extended SRAM cassette.	• Check if the extended SRAM cassette is securely inserted to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the extended SRAM cassette. Please consult your local Mitsubishi representative.	Drive/file information	Always
2103H	Memory error (extended SRAM cassette)	• An unsupported extended SRAM cassette is inserted.	• Replace the extended SRAM cassette with the one supported by the CPU module.	Drive/file information	Always
2120H	Memory card error	• The memory card was removed without the card being disabled.	• Disable the memory card, and then remove it.	Drive/file information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2121H	Memory card error	• An error has been detected in the memory card.	• Format the memory card, or re-insert the memory card. If the same error code is displayed again, the possible cause is a hardware failure of the memory card. Replace the memory card.	Drive/file information	Always
2122H	Memory card error	• The CPU module did not start because the memory card was not restored during the startup processing.	• Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the memory card. Replace the memory card.	Drive/file information	At power-on, at RESET
2150H	Initial processing time error	• Since the initial processing takes time, the host CPU in a multiple CPU system cannot respond to communication requests from other CPU modules in initial communications. (Other CPU modules cannot start up.)	Take the following action to shorten the initial processing time. • Check and correct the restoration setting and the number of files to be restored if the automatic data restoration function is being executed.	—	At power-on, at RESET
2160H	IP address duplication error	• Overlapping IP addresses have been detected.	• Check and correct the IP addresses.	—	Always
2180H	Invalid file	• An invalid file has been detected.	■CPU module • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, select the correct file name, and write the specified file to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, select the correct file name, and write the specified file to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Drive/file information	At power-on, at RESET, at STOP → RUN state
21A0H	File specification error	• The file specified in the CPU parameters does not exist. • The memory card is disabled by tuning on SM606 (SD memory card forced disable instruction). • The file register file does not exist in the specified memory when the file register setting is set to "Use Common File Register in All Programs" and the file capacity is not set in the CPU parameters ("File Setting"). • The file specified in the memory card parameters ("Boot Setting") does not exist in the memory card.	• Turn off SM606. (Cancel the disabled state.) • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, select the correct file name, and write the specified file to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the device/label memory in the CPU module or the memory card. Please consult your local Mitsubishi representative.	Drive/file information Parameter information	At instruction execution, at interrupt occurrence, at power-on, at RESET, at STOP → RUN state, at END instruction execution
21A1H	File specification error	• The file specified in parameter cannot be created.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the name and size of the file corresponding to the displayed parameter number. • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and take the following actions: (1) Format the corresponding drive. (2) Delete unnecessary files on the corresponding drive to increase free space. (3) Unlock the corresponding drive if it is locked.	Drive/file information Parameter information	At write, at power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
21A2H	File specification error	• The CPU module model set to the file using the engineering tool differs from that of the CPU module actually mounted.	• Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and correct the CPU module model set to the file in accordance with that of the CPU module actually mounted.	Drive/file information	At write, at power-on, at RESET, at STOP → RUN state
21A3H	File specification error	The files names are overlapping.	• Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and take the following actions: (1) Format the corresponding drive. Write the program(s) and FB program(s) to the CPU module again. Then, reset the CPU module, and run it again. (2) Correct the file names so that they do not overlap.	Drive/file information	At write, at power-on, at RESET, at STOP → RUN state
2200H	Parameter error	• The system parameter file and CPU parameter file do not exist. • The memory card parameter file or module extension parameter file stored in the memory card cannot be accessed because the memory card is disabled by turning on SM606 (SD memory card forced disable instruction).	• Write the system parameter file and CPU parameter file to the CPU module. • Turn off SM606. (Cancel the disabled state.)	Parameter information	At power-on, at RESET, at STOP → RUN state
2220H	Parameter error	• The parameter setting is corrupted.	■CPU module • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and write the displayed parameter setting to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, the memory card, or the module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative. ■Safety function module • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and write the displayed parameter setting to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module, I/O module, or intelligent function module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2221H	Parameter error	• The set value is out of range.	■CPU module • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, the memory card, or the module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative. ■Safety function module • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module, I/O module, or intelligent function module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state, at END instruction execution, at instruction execution, at module access
		An unsupported parameter exists. • Other than 1 is set to the group number in the CC-Link IEF Basic setting.	Check the CPU module version and use the supported product. Use the product in a supported category. • Set the number of occupied stations so that the total number is 16 stations or less, and set 1 to the group number.	Parameter information	At power-on, at RESET, at STOP → RUN state, at END instruction execution, at instruction execution, at module access
2222H	Parameter error	• Use of the function that is not supported by the module is enabled in parameter.	■CPU module • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, the memory card, or the module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative. ■Safety function module • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module, I/O module, or intelligent function module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state
2223H	Parameter error	• The parameter that needs a reset of the CPU module was modified and overwritten.	• Reset the CPU module, and run it again.	Parameter information	At write, at power-on, at RESET, at STOP → RUN state

A

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2224H	Parameter error	• A memory area cannot be ensured. • A global label setting file exists when the label assignment area set in the CPU parameters is 0K word.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, display the error-detected area by clicking the [Error Jump] button, and increase the capacity of the area. (If the capacity of the area cannot be increased, decrease the capacity of other areas.) • Reduce the number of labels or local devices used. • If no global label is used, delete the global label setting file.	Parameter information	At write, at power-on, at RESET, at STOP → RUN state
2225H	Parameter error	• The CPU module model set to the project using the engineering tool differs from that of the CPU module actually mounted. • The operation set in the memory card parameters cannot be performed. (The boot function cannot be executed.)	• Correct the CPU module model set to the project in accordance with the CPU module actually mounted. • Delete the memory card parameter settings. • Remove the memory card so that the operation set in the memory card parameters will not be performed. (Do not execute the boot operation.)	Parameter information	At write, at power-on, at RESET, at STOP → RUN state
2226H	Parameter error	• The SFC setting in the CPU parameters is incorrect. (Block 0 was set to start automatically, however, block 0 does not exist.)	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number.	Parameter information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
2227H	Parameter error	• The execution type of the SFC program set in the CPU parameter program settings is other than the scan execution type.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number.	Parameter information	At power-on, at RESET, at STOP → RUN state
2240H	Parameter error (module)	• In a multiple CPU system, the I/O module or intelligent function module controlled by another CPU module is specified in the module parameters.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state
2241H	Parameter error (module)	• The I/O numbers set in the system parameters differ from those of the module actually mounted. • The target module is not mounted on the slot where the system parameters and module parameters are set. • The module type set in parameter differs from that of the module actually mounted.	• Check if the system configuration displayed on the system monitor window of the engineering tool match the actual system configuration. • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state, at END instruction execution, at instruction execution, at module access
2242H	Parameter error (module)	• The intelligent function module has detected a module parameter error.	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed I/O number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the intelligent function module connected. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2260H	Parameter error (network)	• Network numbers are overlapping.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the intelligent function module connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET
2261H	Parameter error (network)	• Different network types (CC IE Control extended mode/normal mode) are set between the control station and the normal station.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the intelligent function module connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state
2262H	Parameter error (network)	■Programmable controller CPU, Safety CPU • When the MELSECNET/H network module station number is 0, parameters of PLC to PLC network are set. • The station type set in the module parameters differs from that of the module actually mounted. ■Process CPU • When the MELSECNET/H network module station number is 0, parameters of PLC to PLC network are set. • The station type set in the module parameters differs from that of the module actually mounted. • In the CPU module in redundant mode, "RJ71GF11-T2" (the model name cannot be set for the redundant system) is selected to the model name in the I/O assignment setting of the system parameter.	■Programmable controller CPU, Safety CPU • Correct the MELSECNET/H network module station number. • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the intelligent function module connected. Please consult your local Mitsubishi representative. ■Process CPU • Correct the MELSECNET/H network module station number. • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. • In the CPU module in redundant mode, select "RJ71GF11-T2(MR)", "RJ71GF11-T2(SR)", or "RJ71GF11-T2(LR)" (model names can be set for the redundant system) to the model name in the I/O assignment setting of the system parameter. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the intelligent function module connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state
2263H	Parameter error (network)	• Even though the CC-Link IE module or MELSECNET/H network module is mounted, a different CC-Link IE module or MELSECNET/H network module is set in the system parameters ("I/O Assignment Setting"), or CC-Link IE module or MELSECNET/H network module parameters have not been set.	• Set the system parameters and module parameters. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the intelligent function module connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2280H	Parameter error (refresh)	• The refresh setting is not set correctly. (Data were refreshed exceeding the file register capacity.) • The refresh settings (number of points) are different from those of other numbered CPU modules.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number so that the data are refreshed within the specified device range. (Take the following actions: increase the number of file register points, create a file register file having a capacity for all of the target data to be refreshed, or reduce the refresh device range.) • Rewrite the refresh settings (number of points) in the CPU parameters for all the CPU modules. (Use the same number of points in the refresh settings for all the CPU modules.)	Parameter information	At power-on, at RESET, at STOP → RUN state, at END instruction execution, at instruction execution, at module access
2281H	Parameter error (refresh)	• A device that cannot be used as a refresh device is specified.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number.	Parameter information	At power-on, at RESET, at STOP → RUN state
2282H	Parameter error (refresh)	• The number of specified refresh points is invalid.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number.	Parameter information	At power-on, at RESET, at STOP → RUN state
2283H	Parameter error (refresh)	• The total number of refresh points exceeded the maximum limit.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number.	Parameter information	At power-on, at RESET, at STOP → RUN state
22E0H	Parameter verification error	• In a multiple CPU system, the system parameter settings of the host CPU module differ from those of other CPU modules. • In a multiple CPU system, the system parameter settings are overwritten only to the host CPU module, and the settings differ from those of other CPU modules.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the system parameter settings corresponding to the displayed number for the CPU No.2 and later. The settings need to be the same among all the CPU modules. (The module synchronization setting and fixed scan communication setting need to be the same between the CPU modules that use these functions.) • When the system parameter settings are changed, update the settings of all the CPU modules connected. (The system parameter settings must be same in all the CPU modules.)	Parameter information System configuration information	At write, at power-on, at RESET, at STOP → RUN state
2300H	Security key authentication error	• The security key set to the program does not match the one registered to the CPU module (or extended SRAM cassette).	• Check and correct the security key setting.	Drive/file information	At power-on, at RESET, at STOP → RUN state
2301H	Security key authentication error	• The security key is set to the program, but it is not registered to the CPU module (or extended SRAM cassette).	• Check and correct the security key setting.	Drive/file information	At power-on, at RESET, at STOP → RUN state
2302H	Security key authentication error	• The security key set to the file is corrupted and does not match the one registered to the CPU module. • The security key registered to the CPU module is corrupted and does not match the one set to the file.	• Write the file to the CPU module again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	—	At power-on, at RESET, at STOP → RUN state
2303H	Security key authentication error	• The security key is registered to the CPU module and extended SRAM cassette.	• Check and correct the security key setting.	—	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2320H	Remote password setting error	• The start I/O number of the remote password target module is set to other than 0H to 0FF0H. • There is a problem on the slot specified by the start I/O number of the remote password setting. (1) No module is mounted. (2) The mounted intelligent function module does not support the remote password setting.	• Set the start I/O number of the remote password target module within the range 0H to 0FF0H. • On the specified slot, mount an intelligent function module that supports the remote password setting.	System configuration information	At power-on, at RESET, at STOP → RUN state
2321H	Remote password setting error	• In a multiple CPU system, the module controlled by another CPU module is specified by the start I/O number of the remote password setting.	• Check and correct the remote password setting.	System configuration information	At power-on, at RESET, at STOP → RUN state
2400H	Module verification error	• The module information at power-on differs from the information of modules actually mounted. • The I/O module or intelligent function module is not mounted properly or was removed during operation.	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed slot number. • Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	System configuration information	Always
2401H	Module verification error	• A CPU module, I/O module, or intelligent function module was mounted on the base unit during operation.	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed slot number. • Do not mount a CPU module, I/O module, nor intelligent function module during operation. • Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	System configuration information	Always
2420H	Fuse blown error	• The output module with a blown fuse has been detected.	• Check the FUSE LED of each output module, and replace the one with the FUSE LED on. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and replace the module corresponding to the displayed slot number.	System configuration information	Always
2440H	Module major error	• In a multiple CPU system, the control CPU setting in the system parameters is different from that of other CPU modules. • In a multiple CPU system, other CPU modules (Safety CPUs) have detected a parameter verification error. • An error has been detected in the I/O module or intelligent function module during the initial processing.	• Correct the system parameter settings for the CPU No.2 and later. The settings need to be the same between all the CPU modules. • Clear another CPU module (Safety CPU) errors. • Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2441H	Module major error	• An error has been detected in the I/O module or intelligent function module when the instruction was executed.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	Error location information System configuration information	At instruction execution
2442H	Module major error	• An error has been detected in the I/O module or intelligent function module during the END processing.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	System configuration information	At module access

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2443H	Module major error	• An error has been detected in the I/O module or intelligent function module.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	System configuration information	At module access
2450H	Module major error	• A major error has been notified from the intelligent function module connected. • The I/O module or intelligent function module is not mounted properly or was removed during operation.	• Take measures to reduce noise. • Check the connection status of the extension cable. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed slot number. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	System configuration information	Always
2460H	Another CPU module major error	• An error has been detected in another CPU module during the initial processing.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the host CPU module or another CPU module where the error has been detected. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2461H	Another CPU module major error	• An error has been detected in another CPU module when the instruction was executed.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the host CPU module or another CPU module where the error has been detected. Please consult your local Mitsubishi representative.	Error location information System configuration information	At instruction execution
2462H	Another CPU module major error	• An error has been detected in another CPU module during the END processing.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the host CPU module or another CPU module where the error has been detected. Please consult your local Mitsubishi representative.	System configuration information	At END instruction execution
2463H	Another CPU module major error	• An error has been detected in another CPU module.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the host CPU module or another CPU module where the error has been detected. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2470H	Another CPU module major error	• A major error has been notified from another CPU module.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the host CPU module or another CPU module where the error has been detected. Please consult your local Mitsubishi representative.	System configuration information	Always
2480H	Multiple CPU error	• In a multiple CPU system, an error has been detected in the CPU module where "Stop" is set in the operation mode setting parameter. • Any CPU module other than CPU No.1 is mounted in the inapplicable slot. (An error occurs in the CPU module mounted in the inapplicable slot.)	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, identify the error module, and eliminate the error cause. • Remove the CPU module from the inapplicable slot.	System configuration information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2481H	Multiple CPU error	• In a multiple CPU system, any CPU module other than CPU No.1 was disconnected from the base unit during operation. Or, any CPU module other than CPU No.1 was reset.	• Check the mounting status and reset status of the CPU modules other than CPU No.1.	System configuration information	Always
24C0H	System bus error	• An error was detected on the system bus.	■CPU module • Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C1H	System bus error	• An error was detected on the system bus.	■CPU module • Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C2H	System bus error	• The I/O module or intelligent function module is not mounted properly or was removed during operation. • An error was detected on the system bus.	■CPU module • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed slot number. • Check the connection status of the extension cable. • Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	System configuration information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
24C3H	System bus error	• An error was detected on the system bus.	■CPU module • Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C4H	System bus error	• An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C5H	System bus error	• An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	—	At module access
24C6H	System bus error	• An error was detected on the system bus.	■CPU module • Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or extension cable. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	—	At module access
24C8H	System bus error	• An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	—	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
24D0H	System bus error	• The extension level setting of the Q series extension base unit is overlapping with that of any other extension base units. • An error was detected on the system bus.	• Check and correct the level setting of the Q series extension base unit. • Check the connection status of the extension cable. • Check that the 10m mark is printed on the base unit connected when using the ten-meter extended cables (RC100B). (MELSEC iQ-R Module Configuration Manual) • Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or extension cable. Please consult your local Mitsubishi representative.	System configuration information	Always
24E0H	System bus error	• An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	System configuration information	Always
2500H	WDT error	■CPU module • The scan time exceeded the execution monitoring time set in parameter. • The initial (1st) scan time exceeded the execution monitoring time set in the CPU parameters. • The execution time of the fixed scan interrupt program exceeded the interrupt execution interval. ■Safety function module • The scan time exceeded the execution monitoring time set in parameter. • The initial (1st) scan time exceeded the execution monitoring time set in the CPU parameters.	■CPU module • Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take the following actions: (1) Check and correct the program so that it can be executed within the execution monitoring time set in parameter. (2) Change the execution monitoring time setting to an appropriate value. • Check and correct the fixed scan interrupt program so that the processing completes within the interrupt execution interval. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Check and correct the program so that it can be executed within the execution monitoring time set in parameter. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU. Please consult your local Mitsubishi representative.	Time information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2501H	WDT error	■CPU module • The scan time exceeded the execution monitoring time set in parameter. • The 2nd or later scan time exceeded the execution monitoring time set in the CPU parameters. • The execution time of the fixed scan interrupt program exceeded the interrupt execution interval. ■Safety function module • The scan time exceeded the execution monitoring time set in parameter.	■CPU module • Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take the following actions: (1) Check and correct the program so that it can be executed within the execution monitoring time set in parameter. (2) Change the execution monitoring time setting to an appropriate value. • Check and correct the fixed scan interrupt program so that the processing completes within the interrupt execution interval. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Check and correct the program so that it can be executed within the execution monitoring time set in parameter. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU. Please consult your local Mitsubishi representative.	Time information	Always
2520H	Invalid interrupt	• Even though an interrupt was requested, there is no interrupt factor.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	System configuration information	At interrupt occurrence
2521H	Invalid interrupt	• Even though an interrupt was requested, there is no interrupt factor.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	—	At interrupt occurrence
2522H	Invalid interrupt	• An interrupt was requested from the module with no interrupt pointer setting.	• Check and correct the interrupt pointer setting in the module parameters. • Take measures so that no interrupt is requested from the module with no interrupt pointer setting. • Check and correct the interrupt setting in the buffer memory of the intelligent function module. • Correct the BASIC program executed in the QD51.	System configuration information	At interrupt occurrence
2610H	Inter-module synchronization signal error	• An execution interval error of the synchronization interrupt program has been detected. • An inter-module synchronization error has been detected.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	—	Always
2611H	Inter-module synchronization signal error	• An inter-module synchronization error has been detected.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET, at END instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2630H	Multiple CPU synchronization signal error	• An execution interval error of the synchronization interrupt program has been detected. • A multiple CPU synchronization error has been detected.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	—	Always
2631H	Multiple CPU synchronization signal error	• A multiple CPU synchronization error has been detected.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET, at END instruction execution
2800H	I/O number or network number specification error	• The specified I/O number is out of range (other than 0 to FFH, 3E0 to 3E3H).	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
2801H	I/O number or network number specification error	• The I/O number of the module that does not exist was specified.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information System configuration information	At instruction execution
2802H	I/O number or network number specification error	• The I/O number of the module that does not support the instruction was specified. • The dedicated instruction specified in the program cannot be executed in the specified module or mode.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the execution propriety (including support status and execution mode) of the dedicated instruction, referring to the manual for the target module.	Error location information System configuration information	At instruction execution
2803H	I/O number or network number specification error	• The I/O number of the module that cannot be specified in the instruction was specified.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information System configuration information	At instruction execution
2804H	I/O number or network number specification error	• The specified network number is out of range (other than 1 to 239).	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
2805H	I/O number or network number specification error	• The network number that does not exist was specified.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
2806H	I/O number or network number specification error	• An I/O module or intelligent function module controlled by another CPU module was specified.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Delete the link direct device that specifies a network module controlled by another CPU module from the program. • Specify a network module controlled by host CPU module by using the link direct device.	Error location information System configuration information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2807H	I/O number or network number specification error	• The module cannot be identified in the instruction that requires a specification of the I/O module or intelligent function module. (There is a mistake in the string specifying the module.)	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
2810H	I/O number or network number specification error	• The I/O module or intelligent function module specified in the instruction cannot execute the instruction.	• The possible cause is a hardware failure of the I/O module or intelligent function module specified in the instruction. Please consult your local Mitsubishi representative.	Error location information System configuration information	At instruction execution
2820H	Device, label, or buffer memory specification error	■CPU module • The device or label area used in the instruction exceeded the specified range. • The file register file is not set or was accessed without setting it in the CPU parameters (file setting). ■Safety function module • The device or label area used in the instruction exceeded the specified range.	■CPU module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Set the file register file in parameter, and access the file. ■Safety function module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information Process control instruction processing information	At instruction execution, at END instruction execution
2821H	Device, label, or buffer memory specification error	• The device or label areas used in the instruction to store data are overlapping.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information Process control instruction processing information	At instruction execution
2822H	Device, label, or buffer memory specification error	• The device or label that cannot be used in the instruction was specified.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
2823H	Device, label, or buffer memory specification error	• The buffer memory area of the module specified in the instruction exceeded the specified range. • The module specified in the instruction does not have buffer memory.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
2824H	Device, label, or buffer memory specification error	• The access prohibited area in the buffer memory was accessed.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
2840H	File name specification error	• The file specified in the instruction does not exist.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, create the specified file, and write it to the CPU module. Or, set all the required files in the CPU parameters (file setting).	Error location information Drive/file information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2841H	File name specification error	• The program file specified in the instruction is not set in the CPU parameters (program setting).	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and set the specified program file in the CPU parameter.	Error location information Drive/file information	At instruction execution
2842H	File name specification error	• A file that cannot be specified with the instruction was specified.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check detailed information (drive/file information) for the error by executing module diagnostics using the engineering tool, and check the specified file.	Error location information Drive/file information	At instruction execution
3000H	Boot function execution error	• The boot setting in the memory card parameters is incorrect.	• Check and correct the boot setting in the memory card parameters.	Drive/file information	At power-on, at RESET
3001H	Boot function execution error	• When the boot function was executed, the file format processing failed.	• Reset the CPU module, and execute the boot function again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information	At power-on, at RESET
3003H	Boot function execution error	• When the boot function was executed, the file passwords did not match.	• Check and correct the file password settings of the transfer source and transfer destination files. • Delete the boot setting.	Drive/file information	At power-on, at RESET
3004H	Boot function execution error	• When the boot function was executed, the CPU built-in memory capacity was exceeded.	• Check and correct the boot setting. • Delete unnecessary files in the CPU built-in memory. • Clear the CPU built-in memory by selecting "Clear" to "Operation Setting at CPU Built-in Memory Boot" in the memory card parameters, and execute the boot function.	Drive/file information	At power-on, at RESET
3005H	Boot function execution error	• When the boot function is executed, the security key registered in the CPU module (or extended SRAM cassette) does not match the one that locks the boot source program. • When the boot function is executed, the security key is not registered in the CPU module (or extended SRAM cassette) even though the boot source program is locked with the security key.	• Check and correct the security key setting. • Delete the boot setting.	Drive/file information	At power-on, at RESET
3010H	Data restoration function execution error	• The model of the restoration target CPU module differs from the model of the backup source CPU module.	• Execute the data restoration function to the CPU module whose model is the same as that of the backup source CPU module. • Turn off bit0 of SD955 (Automatic restoration execution) to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET
3011H	Data restoration function execution error	• Reading of backup data from an SD memory card completed with an error.	• Replace an SD memory card, and execute the function again. • The backup data may have been corrupted. Execute the data restoration function using another backup data. • Turn off bit0 of SD955 (Automatic restoration execution) to disable the automatic data restoration function.	CPU module data backup/ restoration folder information Drive/file information	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3012H	Data restoration function execution error	• Writing of backup data to the CPU built-in memory completed with an error.	• The possible cause is a hardware failure of the restoration target CPU module. Execute the data restoration function to another CPU module.	CPU module data backup/ restoration folder information Drive/file information	At power-on, at RESET
3013H	Data restoration function execution error	• The system file does not exist in the backup data to be restored. • File(s) in the system file information does not exist in the folder of the backed up data.	• The backup data may have been corrupted. Execute the data restoration function using another backup data. • Turn off bit0 of SD955 (Automatic restoration execution) to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET
3014H	Data restoration function execution error	• Data was restored to the CPU module where the same data with a file password has already been stored.	• Delete file passwords, and execute the CPU module data backup/restoration function. • Turn off bit0 of SD955 (Automatic restoration execution) to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET
3015H	Data restoration function execution error	• A folder with a value that matches the restoration target date folder setting value or number folder setting value does not exist in the SD memory card. • The restoration target data setting value is out of range. • The restoration target date folder setting value or number folder setting value is out of range.	• Check and correct the restoration target date folder setting value or number folder setting value, and execute the function again. • Check and correct the restoration target data setting value, and execute the function again. • Turn off bit0 of SD955 (Automatic restoration execution) to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET
3016H	Data restoration function execution error	• The automatic data restoration function was executed with the CPU module where an SD memory card was not inserted.	• Insert or re-insert an SD memory card, and execute the function again. • Turn off bit0 of SD955 (Automatic restoration execution) to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET
3017H	Data restoration function execution error	• The automatic data restoration function was executed exceeding the maximum memory capacity of the CPU module. • The automatic data restoration function was executed exceeding the maximum number of files that can be stored in the CPU module.	• Execute the function so that the maximum memory capacity will not be exceeded. • Execute the function so that the maximum number of storable files will not be exceeded. • Turn off bit0 of SD955 (Automatic restoration execution) to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET
3018H	Data restoration function execution error	• The status (such as programs, parameters, and file structure) of the CPU module differs from that of when the data backup function was executed.	• Match the CPU module status to the one at the time of backup, and execute the function again. • Set all data as the restoration target data, and execute the data restoration function. • Turn off bit0 of SD955 (Automatic restoration execution) to disable the automatic data restoration function.	CPU module data backup/ restoration folder information	At power-on, at RESET
3070H	Operation stop error	• The PABORT instruction was executed.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, and check the corresponding program - one that outputs the program error code.	Error location information Program error information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3100H	Program error	• The program includes any instruction that cannot be used or decoded in the CPU module.	■CPU module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the program to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the program to the Safety CPU again. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Error location information	At instruction execution
3101H	Program error	• The program contains a dedicated SFC program instruction even although it is not an SFC program.	■CPU module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Error location information	At write, at power-on, at RESET, at STOP → RUN state
3120H	Program error	• The CPU module does not support the dedicated instruction executed.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Replace the CPU module with the one that supports the instruction.	Error location information	At power-on, at RESET, at STOP → RUN state, at instruction execution
3121H	Program error	• The number of devices used in the dedicated instruction specified in the program is incorrect.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3122H	Program error	• The function block or function specified in the program does not exist.	■CPU module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) again. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Error location information	At instruction execution
3140H	END instruction error	• The END (FEND) instruction does not exist in the program.	■CPU module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) again. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Error location information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3141H	FB/FUN program error	• The structure of FB/FUN program is incorrect.	■CPU module • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) again. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Error location information	At instruction execution
3142H	Temporary area error	• The temporary area was used incorrectly.	■CPU module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and check the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and check the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) again. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Error location information	At instruction execution
3143H	FB/FUN program error	• The FB/FUN program file is incorrect.	■Safety CPU • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) again. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) again. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3160H to 3163H	SFC program block, step error	• The SFC program configuration is illegal.	• Take measures to reduce noise. • Write the SFC program to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Error location information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
3170H	SFC program block, step error	• The number of SFC program steps exceeds the total number of step relays (S).	• Correct the program so that the number of SFC program steps does not exceed the total number of step relays (S). • Check and correct the number of device setting step relays (S) in the CPU parameters.	Error location information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
3171H	SFC program block, step error	• The total number of SFC program blocks (max. step No. + 1) exceeds the total number of step relays (S).	• Correct the program so that the total number of SFC program blocks (max. step No. + 1) does not exceed the total number of step relays (S). • Check and correct the number of device setting step relays (S) in the CPU parameters.	Error location information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
3180H	SFC program configuration error	• The SFC program configuration is illegal.	• Take measures to reduce noise. • Write the SFC program to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Error location information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
3190H	SFC program configuration error	• The SFC program configuration is illegal.	• Take measures to reduce noise. • Write the SFC program to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Error location information	At SFC program execution
3191H	SFC program configuration error	• The SFC program configuration is illegal.	• Take measures to reduce noise. • Write the SFC program to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Error location information	At SFC program execution
3192H	SFC program configuration error	• A self step number was specified for the specification destination step number for the jump transition.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the SFC program to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Error location information	At SFC program execution
3193H	SFC program configuration error	• A self step number was specified for the specification destination step number for the reset step.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the SFC program to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Error location information	At SFC program execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
31A0H	SFC program block, step specification error	• An attempt was made to start an SFC program block that was already running.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Turn on Start/stop SFC program (SM321) if it is off.	Error location information	At SFC program execution
31A1H	SFC program block, step specification error	• A non-existent SFC program block was specified.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Turn on SM321 (Start/stop SFC program) if it is off. • Check the SFC program has existed. • Check the execution status of the SFC program.	Error location information	At instruction execution, at SFC program execution
31A2H	SFC program block, step specification error	• The specified block exceeds the range that can be used in the SFC program.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Turn on SM321 (Start/stop SFC program) if it is off.	Error location information	At instruction execution
31B1H	SFC program block, step specification error	• A non-existent SFC program step was specified.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Turn on SM321 (Start/stop SFC program) if it is off. • Check the SFC program has existed. • Check the execution status of the SFC program.	Error location information	At instruction execution, at SFC program execution
31B2H	SFC program block, step specification error	• The specified step exceeds the range that can be used in the SFC program.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Turn on SM321 (Start/stop SFC program) if it is off.	Error location information	At instruction execution
31B3H	SFC program block, step specification error	• The number of simultaneous active block steps that can be specified in the SFC program exceeds the permissible value.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution, at SFC program execution
31B4H	SFC program block, step specification error	• The total number of simultaneous active steps that can be specified in the SFC program exceeds the permissible value.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution, at SFC program execution
31B5H	SFC program block, step specification error	• A SET Sn/BLm/Sn and RST Sn/BLm/Sn instruction were specified for the self step in the step operation output.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3200H	Program execution error	■Programmable controller CPU, Process CPU • Memory/Device Setting set in the CPU parameters differ from those assigned to the sequence programs, FB programs, and global label setting file. (After Memory/Device Setting were modified, only the CPU parameters were written to the CPU module.) ■Safety CPU • Memory/Device Setting set in the CPU parameters differ from those assigned to the sequence programs, FB programs, and global label setting file. (After Memory/Device Setting were modified, only the CPU parameters were written to the CPU module.) • The safety program, safety FB file, safety global label setting file, standard/safety shared label setting file device/label assignment does not match the Safety CPU CPU parameter memory/device settings. ■Safety function module • The safety program, safety FB file, safety global label setting file, standard/safety shared label setting file device/label assignment does not match the Safety CPU CPU parameter memory/device settings. • Parameters that require resetting after being rewritten at the CPU module were rewritten.	■Programmable controller CPU, Process CPU • After Memory/Device Setting are modified, write the sequence program file(s), FB file(s), and global label setting file together with the CPU parameter file to the CPU module. • If no global label is used, delete the global label setting file. ■Safety CPU • After Memory/Device Setting are modified, write the sequence program file(s), FB file(s), and global label setting file together with the CPU parameter file to the CPU module. • If no global label is used, delete the global label setting file. • Batch write the CPU parameters, safety CPU parameters, safety program, safety FB file, safety global label setting file, and standard/safety shared label setting file. • If not using safety global labels or standard/safety shared labels, delete the safety global label setting file and standard/safety shared label setting file. ■Safety function module • Batch write the CPU parameters, safety CPU parameters, safety program, safety FB file, safety global label setting file, and standard/safety shared label setting file. • If not using safety global labels or standard/safety shared labels, delete the safety global label setting file and standard/safety shared label setting file. • Reset the CPU module, and run it again.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3201H	Program execution error	• Even though no program is set in the CPU parameters, multiple program files exist.	• Set a program in the CPU parameter. • Delete unnecessary program files.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3202H	Program execution error	• The program file is incorrect. Or, the program file is not written properly. • For the subroutine type FB, the unique property "Use MC/MCR in EN Control" is set to "Yes".	■CPU module • Write the program file to the CPU built-in memory again. • Change the unique property of the subroutine type FB "Use MC/MCR in EN Control" to "No" and write the program again. • Replace the CPU module to the one whose firmware version can execute the subroutine type FB in which the unique property "Use MC/MCR in EN Control" is set to "Yes". ■Safety function module • Write the program again.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3203H	Program execution error	• No program file exists.	• Check if the system parameter file, CPU parameter file, and program file exist. • Write the system parameter file, CPU parameter file, and program file to the CPU built-in memory.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3204H	Program execution error	• Two or more SFC programs were executed.	• Ensure that only one SFC program is executed.	Drive/file information	At SFC program execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3205H	Program execution error	• After the global label setting file was modified, only the modified file was written to the programmable controller. Or, without writing the modified global label setting file, only the sequence program file(s) and FB file(s) were written to the programmable controller. (The global label setting file was not written to the programmable controller.) • After an FB file was modified, only the modified file was written to the programmable controller. Or, without writing the modified FB file, only the sequence program file(s) and global label setting file were written to the programmable controller. (The modified FB file was not written to the programmable controller.) • After the "Access from External Device" setting of the global label setting was changed, only the global label assignment information was written to the programmable controller. Or, after the "Access from External Device" setting was changed, only the sequence program file(s) and FB file(s) were written to the programmable controller. (The global label setting file was not written.) • When the "Access from External Device" setting is not selected, the sequence program file(s), FB file(s), and global label setting file are written to the programmable controller without initializing the memory where the global label assignment information file is stored.	• Write all the sequence program file(s), FB file(s), global label setting file, and global label assignment information file to the programmable controller. • If no global label is used, delete the global label setting file. • If the "Access from External Device" setting is not selected, initialize the memory where the global label assignment information file is stored, and write the sequence program file(s), FB file(s), and global label setting file to the programmable controller.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3206H	Program execution error	• After the sequence program was modified, only the modified sequence program file was written to the programmable controller. Or, without writing the modified sequence program file, only the initial local label value file was written to the programmable controller. • After the global label setting file was modified, only the modified file was written to the programmable controller. Or, without writing the modified global label setting file, only the initial global label value file was written.	• Write both the sequence program file(s) and initial local label value file to the programmable controller. • Write both the global label setting file and initial global label value file to the programmable controller. • If no initial global label value is used, delete the initial global label value file. • If no initial local label value is used, delete the initial local label value file.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3207H	Program execution error	• The signal flow area to be used in the FB program cannot be secured.	• Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and identify the error program file. Then, reduce the number of instructions that use the signal flow area of the FB program used in the sequence program file.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3208H	Program execution error	• The label file is incorrect.	• Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, select the correct file name, and write the specified file to the CPU module.	Drive/file information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3220H	SFC program execution error	• Unable to execute the SFC program.	• Take measures to reduce noise. • Write the SFC program and CPU parameters to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
3221H	SFC program execution error	• Unable to execute the SFC program.	• Take measures to reduce noise. • Write the SFC program to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
3222H	SFC program execution error	• Unable to execute the SFC program.	• Take measures to reduce noise. • Write the SFC program to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
32FEH	License authentication error	• A device or a label that is required for the license authentication does not exist.	• Please consult your local Mitsubishi representative.	Error location information	At instruction execution
32FFH	License authentication error	• The specified license key is incorrect.	• Check the license key and specify correct one.	Error location information	At instruction execution
3300H	Pointer setting error	• The total number of points of local or global pointers used in the program exceeded the points set in the CPU parameters (pointer device area).	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check and correct the pointer device area point setting in the CPU parameters.	Error location information	At power-on, at RESET, at STOP → RUN state
3301H	Pointer setting error	• The total number of points of pointer-type labels used in the program exceeded the points set in the CPU parameters (pointer-type label assignment area).	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check and correct the pointer-type label assignment area point setting in the CPU parameters.	Error location information	At power-on, at RESET, at STOP → RUN state
3302H	Pointer setting error	• Multiple global pointers with the same number or pointer-type global labels are used in the program. (The pointer numbers or the labels are overlapping.) (The pointer numbers are overlapping.)	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At power-on, at RESET, at STOP → RUN state
3303H	Pointer setting error	• Multiple local pointers with the same number or pointer-type local labels are used in the program. (The pointer numbers or the labels are overlapping.) (The pointer numbers are overlapping.)	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At power-on, at RESET, at STOP → RUN state
3320H	Interrupt pointer setting error	• The interrupt pointer numbers used in the files are overlapping.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3340H	FOR-NEXT instruction error	• The NEXT instruction was not executed even though the FOR instruction was executed. Or, there are more FOR instructions than NEXT instructions.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At END instruction execution
3341H	FOR-NEXT instruction error	• The NEXT instruction was executed even though the FOR instruction was not executed. Or, there are more NEXT instructions than FOR instructions.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
3342H	FOR-NEXT instruction error	• The BREAK instruction was executed even though the FOR instruction was not executed.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
3360H	Nesting depth error	• The number of nesting levels in the subroutine program exceeded its limit (16).	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the number of nesting levels (16 or less).	Error location information	At instruction execution
3361H	Nesting depth error	• The number of nesting levels in the FOR instruction exceeded its limit (16).	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the number of nesting levels (16 or less).	Error location information	At instruction execution
3362H	Nesting depth error	• The number of nesting levels in the DI instruction exceeded its limit (16).	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the number of nesting levels (16 or less).	Error location information	At instruction execution
3363H	Nesting depth error	• The number of nesting levels in the function block or function exceeded its limit (32).	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the number of nesting levels (32 or less).	Error location information	At instruction execution
3380H	Pointer execution error	• The pointer specified in the instruction does not exist.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
3381H	Pointer execution error	• The RET instruction does not exist in the executed subroutine program.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At END instruction execution
3382H	Pointer execution error	• The RET instruction exists before the FEND instruction in the main program.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
33A0H	Interrupt pointer execution error	• The interrupt pointer corresponding to the interrupt input does not exist.	• Check if the program corresponding to the interrupt pointer number set in the module parameters exists.	—	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
33A1H	Interrupt pointer execution error	• The IRET instruction does not exist in the executed interrupt program. • The STOP instruction has been executed in an interrupt program.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
33A2H	Interrupt pointer execution error	• The IRET instruction exists before the FEND instruction in the main program.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
33A3H	Interrupt pointer execution error	• The IRET instruction or STOP instruction was executed in the fixed scan execution type program.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
33A4H	Interrupt pointer execution error	• The IRET instruction or STOP instruction has been executed in an event execution type program.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
33C0H	FB/FUN execution error	• Before the FB/FUN program ends, the call source program ended.	■CPU module • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the program to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Write the program to the Safety CPU again. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Error location information	At instruction execution
33D0H	Temporary area exceeded	• The secured temporary area size exceeded its limit.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the number of nesting levels in the function.	Error location information	At instruction execution
3400H	Operation error	• Division where the divisor is zero was performed.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information Process control instruction processing information	At instruction execution
3401H	Operation error	• Data that cannot be converted by using the data conversion instruction was input.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3402H	Operation error	• The operation was performed with the invalid data (-0, denormalized number, NaN (not a number), or ±∞).	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information Process control instruction processing information	At instruction execution
3403H	Operation error	• An overflow has occurred during the operation.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information Process control instruction processing information	At instruction execution
3404H	Operation error	• A string that is not supported in the instruction was specified.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
3405H	Operation error	• The input data was out of range.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information Process control instruction processing information	At instruction execution
3406H	Operation error	• The operation result is out of the output range. (The operation result of the instruction that concatenate character strings exceeded the allowable number of characters.)	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information Process control instruction processing information	At instruction execution
3420H	Operation error	• The link direct device, module access device, or CPU buffer memory access device is specified for both (s) and (d) used in the BMOV instruction.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
3421H	Operation error	• When writing data to the data memory by using the SP.DEVST instruction, the number of writes per day exceeded the number set in SD771. • The value set in SD771 is out of range.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and check if the SP.DEVST instruction is used correctly in the program. • Execute the SP.DEVST instruction again on another day, or change the value in SD771. • Set the value in SD771 within the settable range.	Error location information	At instruction execution
3422H	Operation error	• The structure of the PID control instruction is incorrect.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the PID control instruction structure.	Error location information	At instruction execution
3423H	Operation error	• The size of data to be sent/received by the socket communications instruction exceeds the allowable range.	• Check and change the send data size of the CPU module or the external device. • If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3424H	Operation error	• A second SFC program was started with an instruction while an SFC program was running.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and check the SFC program execution status.	Error location information	At instruction execution
3425H	Operation error	• The clock data specified with the DATEWR(P) instruction is less than 1 hour from the daylight saving time starting time.	• Check the detailed error information (error location information) by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the clock data specified with the instruction or the daylight saving time setting starting time.	Error location information	At instruction execution
3426H	Operation error	• The specified file name (before a period) or extension includes two or more "*". • The specified file name (before a period) or extension includes "*" and "?" • A wild card ("*", "?") is used in a location where it cannot not be used. • A file that cannot be transfered has been specified. • A file name is not specified. • A delimiter for a drive number has been specified with characters other than ":\" or ":/".	• Check the usage of wild card characters. • Use a file that can be transferred. • Specify a file name. • Specify a delimiter for a drive number with ":\" or ":/".	Error location information	At instruction execution
3427H	Operation error	The control data (d1) of the SP.FREAD and SP.FWRITE instruction are specified with an invalid combination between the execution/completion type and the data type specification.	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and change the combination to a valid one.	Error location information	At instruction execution
3430H	Operation error	• An instruction has been executed without setting parameters which are required when the instruction is executed.	• Set parameters required to execute the instruction.	Error location information	At instruction execution
3440H	Operation error	• In a multiple CPU system, the multiple CPU dedicated instruction (the one whose symbol starts with D(P)) was executed when "Do Not Use" was set to "Fixed scan communication function" in the system parameters ("Multiple CPU Setting").	• Change "Fixed scan communication function" to "Use". • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and change the multiple CPU dedicated instruction to the one whose symbol starts with M(P).	Error location information	At instruction execution
3441H	Operation error	• In a multiple CPU system, the number of data points was specified exceeding the multiple CPU dedicated instruction areas applicable for each CPU module.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and change the number of data points for the multiple CPU dedicated instruction.	Error location information	At instruction execution
34A0H	Operation error	• Response data of the socket communications instruction cannot be created.	• Increase the request interval. • Decrease the number of request nodes. • Wait for a response to the previous request before sending the next request. • Correct the timeout value.	Error location information	At END instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3600H	Safety system configuration error	• The safety function module for self management has not been mounted to the right of the Safety CPU. • There is a vacant setting to the right of the Safety CPU in the system parameter I/O assignment settings.	• Mount a safety function module for self management to the right of the Safety CPU. • Re-set the system parameter I/O assignment settings so that there is a safety function module to the right of the Safety CPU. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU, safety function module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
3610H	Safety system configuration error	• The safety function module with a pair version that differs from the one of the Safety CPU is mounted.	• Mount the safety function module with the same pair version as the one of the Safety CPU.	System configuration information	At power-on, at RESET
3620H	Control CPU module error	• An error has been detected in the control CPU module.	• Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the control CPU module or this module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
3640H	Safety mutual monitoring error	A file mismatch was detected during mutual monitoring between the Safety CPU and safety function module.	• Format the memory. Write all files to the Safety CPU. Then, reset the Safety CPU, and run it again.	—	At power-on, at RESET
3641H	Safety mutual monitoring error	A Safety CPU or safety function module error was detected during mutual monitoring between the Safety CPU and safety function module. The Safety CPU or safety function module was not mounted properly or was removed during operation.	• Take measures to reduce noise. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed slot number. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU, safety function module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	Always
3680H	Parameter error (safety function)	• A safety function module for self management has not been mounted, and therefore parameter communications with the safety function module are not possible.	• Mount a safety function module for self management. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU or safety function module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET
3681H	Parameter error (safety function)	• Connection between self-managed modules is set in the safety communication settings.	■Safety CPU • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU data memory or intelligent function module. Please consult your local Mitsubishi representative. ■Safety function module • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module or intelligent function module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET
3682H	Parameter error (safety function)	• The safety communication setting transmission interval monitoring time is shorter than the safety cycle time, or is set to the same value.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter settings corresponding to the displayed number so that the transmission interval monitoring time in the safety communication settings will not be smaller than or equal to the safety cycle time.	Parameter information	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3683H	Parameter error (safety function)	• The safety communication setting safety refresh monitoring time is shorter than the transmission interval monitoring time, or is set to the same value.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter settings corresponding to the displayed number so that the safety refresh monitoring time in the safety communication settings will not be smaller than or equal to the transmission interval monitoring time.	Parameter information	At power-on, at RESET
3684H	Parameter error (safety function)	• The safety CPU parameters do not exist.	• Write the safety CPU parameters.	Parameter information	At power-on, at RESET, at STOP → RUN state
3690H	Parameter error (safety function)	• The corresponding Safety CPU does not exist. • Multiple control CPUs were detected.	• Remove the safety function module. • Use a control CPU for a single Safety CPU. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU, safety function module, or base unit. Please consult your local Mitsubishi representative.	—	At power-on, at RESET
36A0H	Parameter error (safety function)	• A value exceeding the maximum number of safety programs to be executed is set in the program settings.	• Correct the number of safety programs so that it is within the maximum number to be executed.	Parameter information	At power-on, at RESET
36A1H	Parameter error (safety function)	• A value exceeding the maximum number of safety FB files to be executed is set in the FB/FUN file settings.	• Correct the number of safety FB files so that it is within the maximum number to be executed.	Parameter information	At power-on, at RESET
36C0H	Safety operation mode switching error	• An attempt was made to change the CPU operating status to RUN when the safety operation mode is SAFETY MODE (wait-for-restart).	• Reset the Safety CPU, and run it again.	—	Always
3700H	System consistency check error (system configuration)	• The module mounting status of each slot or the model of the modules used differs between the systems A and B. • The module model name set in the system parameters ("I/O Assignment Setting") differs between the systems A and B. (The error is detected in the control system or the standby system.)	• Match the module mounting status of each slot and the model of the modules used between the systems, and restart the CPU module. • Set the same module model name in the system parameters ("I/O Assignment Setting") for both systems.	System configuration information	At power-on, at RESET, at system switching execution, at tracking cable connection
3701H	System consistency check error (system configuration)	• The CPU module model differs between the systems A and B. (The error is detected in the control system or the standby system.)	• Mount the same CPU module, and restart the CPU module.	—	At power-on, at RESET, at system switching execution, at tracking cable connection
3702H	System consistency check error (system configuration)	• The number of slots on the main base unit differs between the systems A and B. (The error is detected in the control system or the standby system.)	• Use the same main base unit (with the same number of slots), and restart the CPU module.	—	At power-on, at RESET
3710H	System consistency check error (SD memory card mounting status)	• The memory card mounting status differs between the systems A and B. (The error is detected in the control system or the standby system.)	• Match the memory card mounting status between the systems, and restart the CPU module.	—	At power-on, at RESET
3711H	System consistency check error (memory card write protect switch status)	• The write protect switch status of the memory card differs between the systems A and B. (The error is detected in the control system or the standby system.)	• Match the write protect switch status between the systems, and restart the CPU module.	—	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3714H	System consistency check error (file)	• A file mismatch was detected in the system consistency check. (The error is detected in the standby system.)	• Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, format the corresponding drive of the standby system CPU module, write all files to the CPU module, and restart the standby system CPU module. (Set the same number of steps as that of the control system CPU module in "Allocate Memory for Online Program Change" on the "Program Detail Setting" window.) • Execute the memory copy function to match files in the control system and the standby system. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information	At write, at power-on, at RESET, at STOP → RUN, at system switching execution, at tracking cable connection
3730H	Redundant function module communication error	• Data communications with the redundant function module has failed.	• Check that there is no error in the CPU module, base unit, or redundant function module. • Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	Always
3740H	Redundant system error	• The operation mode differs between the systems A and B. (The error is detected in the standby system.)	• Restart the standby system CPU module.	—	At power-on, at reset, at tracking cable connection
3741H	Redundant system error	• Both systems were set as system A or system B at start-up. • The CPU module waiting for a start-up of the other system detected that the system A/B setting is not set to the CPU module of the other system.	• Set one system as system A and the other system as system B by performing online operation. Then, restart the CPU modules in both systems. • Set system A or B to the CPU module with no system setting, and restart the CPU modules in both systems.	—	At power-on, at reset, at tracking cable connection
3742H	Redundant system error	• The systems were started without system A/B setting.	• Set one system as system A and the other system as system B by performing online operation. Then, restart the CPU modules in both systems. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, or tracking cable. Please consult your local Mitsubishi representative.	—	At power-on, at RESET
3743H	Redundant system error	• The system setting (system A or B) was not determined.	• Set one system as system A and the other system as system B by performing online operation. Then, restart the CPU modules in both systems.	—	At power-on, at RESET
3744H	Redundant system error	• Both systems are set as a control system. (The error is detected in the system B (control system).)	• Restart the CPU module in the system B.	—	At tracking cable connection
3745H	Redundant system error	• The systems were started with one CPU module set to redundant mode and the other CPU module set to process mode. • The systems were started in a state where the firmware versions of the CPU modules were not compatible.	• Read the project stored in the CPU module operated in redundant mode, and write it to the CPU module operated in process mode. Then, restart the CPU modules in both systems. • Replace one CPU module with the one having a compatible firmware version, referring to the MELSEC iQ-R Module Configuration Manual. Then, restart the system.	—	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3746H	Tracking communications disabled	• Data communications with the other system could not be performed within the period of time set in the CPU parameters ("Other System Start-up Timeout Setting").	• If the other system CPU module is powered off, restart the CPU modules in both systems. • If a WDT error has occurred in the other system CPU module, eliminate the error cause, and restart the system. • Connect tracking cables securely and properly to the connectors of the redundant function modules in both systems, and restart the CPU module(s) with the error. (One cable shall be connected between the IN connector of the module in system A and the OUT connector of the module in system B. The other cable shall be connected between the OUT connector of the module in system A and the IN connector of the module in system B.) • Set a longer time in the CPU parameters ("Other System Start-up Timeout Setting"). • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.	—	At power-on, at RESET
3747H	Redundant system error	• The system cannot be started as a control system because the CPU module of the other system has been stopped due to a stop error.	• Eliminate the error cause, and restart the CPU modules in both systems.	—	At power-on, at RESET
3748H	Redundant system error	• The systems were started in a state where the firmware versions of the CPU modules were not compatible.	• Replace one CPU module with the one having a compatible firmware version, referring to the MELSEC iQ-R Module Configuration Manual. Then, restart the system.	—	At power-on, at RESET
3760H	Control system CPU module error	• In backup mode, a stop error of the control system CPU module was detected. (The error is detected in the standby system.)	• Eliminate the error cause, and restart the CPU modules in both systems.	—	Always
3780H	Redundant system configuration error	• A network module that does not support the redundant system function is mounted. • A network module with a firmware version that does not support the redundant system function is mounted. • A network type other than "Ethernet" is set to port 1 and/or port 2 of the Ethernet interface module.	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed slot number. If the module does not support the redundant system function, remove the module from the base unit. • Check the firmware version of the corresponding module. If the module does not support the redundant system function, remove the module from the base unit. • Set the network type of both port 1 and port 2 to "Ethernet" in the system parameters ("I/O Assignment Setting"). If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or network module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
3781H	Redundant system configuration error	• Two or more redundant function modules are mounted.	• Mount only one redundant function module.	System configuration information	At power-on, at RESET
3782H	Redundant system configuration error	• An extension base unit is connected.	• Remove all extension base units.	System configuration information	At power-on, at RESET
3783H	Redundant system configuration error	• In redundant mode, nine or more CC-Link IE Field Network modules or CC-Link modules are mounted.	• Reduce the number of CC-Link IE Field Network modules or CC-Link modules to eight or less.	System configuration information	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3784H	Redundant system configuration error	• The redundant function module is not mounted next to the control CPU set to redundant mode.	• Mount the redundant function module, and restart the system. Or, set the CPU module to process mode, and restart the system. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	—	At power-on, at RESET, at STOP → RUN state
3790H	Parameter error (redundant function)	• The redundant function module was mounted next to the control CPU set to process mode.	• Set the CPU module to redundant mode, and restart the system. Or, remove the redundant function module, and restart the system. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET, at STOP → RUN state
37A0H	Parameter error (redundant function)	• The range set for tracking transfer in the CPU parameters exceeds the device range or the file register capacity set in the CPU parameters.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, correct the parameter setting corresponding to the displayed number, and take the following actions: (1) Check and correct the tracking transfer settings in the CPU parameters so that they will be within the device range or the file register capacity. (2) Check and correct the device settings or the file register capacity setting in the CPU parameters.	Parameter information	At power-on, at RESET
37A1H	Parameter error (redundant function)	• A device that cannot be used is specified for tracking transfer in the CPU parameters.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, correct the parameter setting corresponding to the displayed number, and write parameters to the CPU module again. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or redundant function module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET
37A2H	Parameter error (redundant function)	• The number of points set for tracking transfer in the CPU parameters is invalid.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, correct the parameter setting corresponding to the displayed number, and write parameters to the CPU module again. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or redundant function module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET
37A3H	Parameter error (redundant function)	• Total amount of data set for tracking transfer in the CPU parameters exceeds the maximum limit.	• Set the tracking capacity of each block within the allowable range in the CPU parameters.	Parameter information	At power-on, at RESET
37A4H	Parameter error (redundant function)	• In the CPU module which operates in redundant mode, parameters for the unsupported function have been set or parameters related to a redundant system have not been set.	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, correct the parameter setting corresponding to the displayed number, and write parameters to the CPU module again. • Correct the operation mode set in the engineering tool. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at write
37A5H	Parameter error (redundant function)	• In the CPU module which operates in redundant mode, the number of points of the step relay (S) is set to other than 0 in the CPU parameters.	• Set the number of points of the step relay (S) to 0 in the CPU parameters.	Parameter information	At power-on, at RESET, at write

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
37C0H	Program error	• The program includes an instruction that cannot be used in the operation mode set in the engineering tool.	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and delete the program. Then, write the program to the CPU module again.	Error location information	At write, at power-on, at RESET, at STOP → RUN state
37C1H	Program execution error	• In the CPU module which operates in redundant mode, an SFC program is set in the CPU parameters ("Program Setting").	• Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and delete all SFC programs registered. Then, write CPU parameters to the CPU module again.	Drive/file information	At write, at power-on, at RESET, at STOP → RUN state
37E0H	Memory format by memory copy	• The memory copy function was executed, and the memory in the standby system has been formatted.	• Power off and on or reset the CPU module after the memory data has been copied from the control system to the standby system.	—	At memory copy
3C00H	Hardware failure	• A hardware failure has been detected.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	Failure information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3C01H	Hardware failure	• A hardware failure has been detected.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	Failure information	Always
3C02H	Hardware failure	• A hardware failure has been detected.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET, at END instruction execution, at interrupt occurrence

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3C03H	Hardware failure	• A hardware failure has been detected.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	Always
3C0FH	Hardware failure	• A hardware failure has been detected.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or extended SRAM cassette. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	Failure information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3C10H	Hardware failure	• A hardware failure has been detected.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET
3C11H	Hardware failure	• A hardware failure has been detected.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution, at instruction execution
3C12H	Hardware failure	• The waveform of the voltage out of the specified range has been detected in the power supply module. • A hardware failure has been detected in the power supply module, CPU module, base unit, or extension cable. (In a redundant power supply system, the error is detected when two power supply modules failed.)	• Check the waveform of the voltage applied to the power supply module. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module, CPU module, base unit, or extension cable. Please consult your local Mitsubishi representative.	Failure information	Always
3C13H	Hardware failure	• A hardware failure has been detected.	• Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3C14H	Hardware failure	• A hardware failure has been detected.	• Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3C15H	Hardware failure	• A hardware failure has been detected.	■Safety CPU • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution, at power-on, at RESET
3C16H	Hardware failure	• A hardware failure has been detected.	■Safety CPU • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution, at power-on, at RESET
3C17H	Hardware failure	• A hardware failure has been detected.	■Safety CPU • Take measures to reduce noise. • Check the mounting status of the CPU module. • Format the memory. Write all files to the Safety CPU. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Format the memory. Write all files to the Safety CPU. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3C20H	Memory error	• An error has been detected in the memory.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET
3C21H	Memory error	• An error has been detected in the memory.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution, at power-on, at RESET
3C22H	Memory error	• An error has been detected in the memory.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution, at power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3C2FH	Memory error	• An error has been detected in the memory.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative. ■Redundant function module • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	Failure information	Always
3C30H	Memory error	• An error has been detected in the memory.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module, and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Format the memory. Write all files to the Safety CPU. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At instruction execution
3C31H	Memory error	• An error has been detected in the memory.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module, and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Format the memory. Write all files to the Safety CPU. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3C32H	Memory error	• An error has been detected in the memory.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	Always
3C33H	Memory error	• An error has been detected in the memory.	■Safety CPU • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET
3E00H	Operation circuit error	■CPU module • An error has been detected in the CPU module. ■Safety function module • An error has been detected in the safety function module. ■Redundant function module • An error has been detected in the redundant function module.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module, and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Format the memory. Write all files to the Safety CPU. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative. ■Redundant function module • Reset the CPU module. • If the error code is displayed again even after taking an action, please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3E01H	Operation circuit error	■Safety CPU • An error has been detected in the Safety CPU. ■Safety function module • An error has been detected in the safety function module.	■Safety CPU • Take measures to reduce noise. • Check the mounting status of the CPU module. • Format the memory. Write all files to the Safety CPU. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Format the memory. Write all files to the Safety CPU. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution, at power-on, at RESET
3E10H	Circuit error	■Safety CPU • An error has been detected in the Safety CPU. ■Safety function module • An error has been detected in the safety function module.	■Safety CPU • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution, at power-on, at RESET
3E11H	Circuit error	■Safety CPU • An error has been detected in the Safety CPU. ■Safety function module • An error has been detected in the safety function module.	■Safety CPU • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution, at power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3E12H	Circuit error	■Safety CPU • An error has been detected in the Safety CPU. ■Safety function module • An error has been detected in the safety function module.	■Safety CPU • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET
3E20H	Program execution error	• The entire program was executed without executing the END instruction.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module, and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Format the memory. Write all files to the Safety CPU. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution
3E22H	Program execution error	• The FB/FUN program did not complete successfully.	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module, and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Format the memory. Write all files to the Safety CPU. Then, reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3E60H	Safety mutual monitoring error	• A hardware error was detected when performing mutual monitoring between the Safety CPU and safety function module.	■Safety CPU • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET, at interrupt occurrence
3E61H	Safety mutual monitoring error	• A hardware error was detected when performing mutual monitoring between the Safety CPU and safety function module.	■Safety CPU • Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the Safety CPU. Please consult your local Mitsubishi representative. ■Safety function module • Take measures to reduce noise. • Check the mounting status of the safety function module. • Reset the Safety CPU, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the safety function module. Please consult your local Mitsubishi representative.	Failure information	Always

Codes of errors detected by other than the self-diagnostic function (4000H to 4FFFH)

The following table lists the codes of errors, other than those detected by the self-diagnostic function of the CPU module.

■Error codes returned to request source during communications with the CPU module

- Errors generated when the data communications are requested from the engineering tool, intelligent function module, or network system connected
- Errors generated with the data logging function

These error codes are not stored in SD0 because they are not detected by the self-diagnostic function of the CPU module.

Error code	Error name	Error details and cause	Action
4000H	Common error	• Serial communication sum check error	• Connect the serial communication cable correctly. • Take measures to reduce noise.
4001H	Common error	• An unsupported request was executed. (The request was executed to the CPU module that does not support the request.)	• Check the command data of the SLMP/MC protocol. • Check the CPU module model name selected in the engineering tool. • Check the target CPU module model name.
4002H	Common error	• An unsupported request was executed.	• Check the command data of the SLMP/MC protocol. • Check the CPU module model name selected in the engineering tool. • Execute the request again. • If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.
4003H	Common error	• Command for which a global request cannot be performed was executed.	• Check the command data of the SLMP/MC protocol.
4004H	Common error	All the operations to the CPU module are disabled because of the following reason: • The CPU module is starting up.	• Perform operations to the CPU module again after the start-up processing ends.
4005H	Common error	• The volume of data handled according to the specified request is too large.	• Check the command data of the SLMP/MC protocol.
4006H	Common error	• Initial communication has failed.	• When using serial communication, inquire of the external device manufacturer for support conditions. • When using serial communication, check the CPU module model name selected in the engineering tool. • When using Ethernet communication, shift the communication start timing.
4008H	Common error	• The CPU module is BUSY. (The buffer is not vacant.)	• Execute the request again after the specified period of time has elapsed.
4010H	CPU module operation error	• The request cannot be executed because the CPU module is running.	• Set the operating status of the CPU module to STOP, and execute the request again.
4013H	CPU module operation error	• The request cannot be executed because the CPU module is not in a STOP state.	• Set the operating status of the CPU module to STOP, and execute the request again.
4021H	File related error	• The specified drive (memory) does not exist or there is an error.	• Check the specified drive (memory) status. • Back up data in the CPU module, and then initialize the memory.
4022H	File related error	• The file with the specified file name or file No. does not exist. ■When CPU Module Logging Configuration Tool is used • The data logging specifying a program name that does not exist was started.	• Check the specified file name and file No.
4023H	File related error	• The file name and file No. of the specified file do not match.	• Delete the file and then recreate the file.
4024H	File related error	• The specified file cannot be handled by a user.	• Do not access the specified file.
4025H	File related error	• The specified file is processing the request from another engineering tool.	• Forcibly execute the request. Or, execute the request again after the processing being performed ends.
4026H	File related error	• The file password set in advance to the target drive (memory) must be specified.	• Specify the file password set in advance, and then access to the drive (memory).
4027H	File related error	• The specified range is larger than the file size.	• Check the specified range and access within that range.
4028H	File related error	• The same file already exists.	• Forcibly execute the request. Or, change the file name and execute the request again.
4029H	File related error	• The specified file capacity cannot be obtained.	• Review the specified file capacity, and execute the request again.
402AH	File related error	• The specified file is abnormal.	• Back up data in the CPU module, and then initialize the memory.

Error code	Error name	Error details and cause	Action
402BH	File related error	• The request cannot be executed in the specified drive (memory).	• Set the operating status of the CPU module to STOP, and execute the request again.
402CH	File related error	• The request cannot be executed currently.	• Execute the request again after a while.
402FH	File related error	• Writing of a file did not complete.	• Back up data internally in the CPU module, and delete the corresponding file or initialize the drive 4. Then, write the file to the programmable controller again.
4030H	Device specification error	• The specified device is not supported. ■When CPU Module Logging Configuration Tool is used • The data logging specifying a device that is not supported was started.	• Check the specified device.
4031H	Device specification error	• The specified device number is out of range. • The CPU module does not support the specified device. ■When CPU Module Logging Configuration Tool is used • The data logging specifying a device number that does not exist was started.	• Check the specified device number. • Check the device assignment of the CPU module. • Check the specified device.
4032H	Device specification error	• The device modification was incorrectly specified. Or, the unusable device (TS, TC, SS, SC, CS, or CC) was specified in any of the following SLMP/MC protocol commands; Read random, Write random (in units of words), Entry monitor device, or Execute monitor. ■When CPU Module Logging Configuration Tool is used • The data logging specifying a device modification that is not supported was started.	• Check the device modification method. • Check the specified device.
4033H	Device specification error	• Writing cannot be done because the specified device is for system use.	• Do not write the data in the specified device, and do not turn on or off.
4034H	Device specification error	• The dedicated instruction cannot be executed since the completion device for the dedicated instruction does not turn on.	• Since the completion device for the SREAD or SWRITE instruction does not turn on in the CPU module on the target station, execute the instruction again after setting the operating status of the CPU module on the target station to the RUN status.
4040H	Intelligent function module specification error	• The request cannot be executed to the specified intelligent function module.	• Check whether the specified module is the intelligent function module having the buffer memory.
4041H	Intelligent function module specification error	• The access range exceeds the buffer memory range of the specified intelligent function module.	• Check the start address and access number of points and access using a range that exists in the intelligent function module.
4042H	Intelligent function module specification error	• The specified intelligent function module cannot be accessed.	• Check that the specified intelligent function module is operating normally. • Check the specified module for a hardware fault.
4043H	Intelligent function module specification error	• The intelligent function module does not exist in the specified position. ■When CPU Module Logging Configuration Tool is used • The data logging specifying a device that does not exist or cannot be accessed was started.	• Check the I/O number of the specified intelligent function module.
4044H	Intelligent function module specification error	• A control bus error occurred during access to the intelligent function module.	• Check the specified intelligent function module and other modules and base units for a hardware fault.
4049H	Intelligent function module specification error	• A request cannot be processed because the module extension parameter of the positioning module is being used for the positioning control.	Turn off module ready (Yn+0) of the positioning module or execute the processing again after removing the extension parameter of the positioning module from the target data.
4050H	Protect error	• The request cannot be executed because the write protect switch of the SD memory card is on.	• Turn off the write protect switch.
4051H	Protect error	• The specified drive (memory) cannot be accessed.	Check the following and take action. • Is the drive (memory) the one that can be used? • Is the specified drive (memory) correctly installed?
4052H	Protect error	• The specified file attribute is read only, so the data cannot be written.	• Do not write data in the specified file. Or, change the file attribute.

Error code	Error name	Error details and cause	Action
4053H	Protect error	• An error occurred when writing data to the specified drive (memory).	• Check the specified drive (memory). Or, write data again after changing the corresponding drive (memory).
4054H	Protect error	• An error occurred when deleting the data in the specified drive (memory).	• Check the specified drive (memory). Or, delete data again after replacing the specified drive (memory).
4060H	Online registration error	• The online debug function (such as online change) and the data logging function are being executed with another engineering tool. ■When CPU Module Logging Configuration Tool is used • An attempt was made to write or delete data logging settings or to execute data logging to the setting registered by another request source.	• Execute the function again after the processing of the function executed from another engineering tool ends. • If the operation of another engineering tool is on hold, resume and finish the operation of another engineering tool, and then execute the function again.
4061H	Online registration error	• Settings for the online debug function (such as online change) are incorrect.	• Register an online debug function (such as online change and external input/output forced on/off), and then execute the function. • Execute again after checking the communication route such as the communication cable.
4063H	Online registration error	• The registered number of locked files exceeded the maximum value.	• Execute the request again after the file access from another engineering tool ends.
4064H	Online registration error	• The specified contents of the online debug function (such as the online program change), data logging function, memory dump function, or real-time monitor function are incorrect. ■When CPU Module Logging Configuration Tool is used • The trigger logging was started in a state that the trigger condition has already been satisfied.	• Check the set data of the online debug function (such as the online program change), data logging function, memory dump function, and real-time monitor function. • Execute again after checking the communication route such as the communication cable. ■When CPU Module Logging Configuration Tool is used • Clear the satisfied trigger condition, and execute the trigger logging again.
4065H	Online registration error	• The device assignment information differs from the parameter.	• Check the device assignment of the CPU module or the device assignment of the request data.
4066H	Online registration error	• The specified file password is incorrect.	• Check and specify the correct file password.
4067H	Online registration error	• Monitor communication has failed.	• Execute again after checking the communication route such as the communication cable.
4068H	Online registration error	• Operation is disabled because it is being performed with another engineering tool.	• Execute the request again after the processing of the function executed from another engineering tool ends.
406AH	Online registration error	• The drive (memory) number other than 0 to 4 was specified.	• Check the drive and specify the drive number correctly.
406BH	Online registration error	• Online operation was interrupted due to a CPU module error.	• Check the status of the CPU module by executing the module diagnostics. Identify the error, and take a corrective action referring to the troubleshooting section.
4070H	Circuit verification error	• The program not yet corrected and the one corrected by online program change are different.	• Read the program from the CPU module to match it with that of the engineering tool, and then execute online change again.
4080H	Any other error	• Request data error ■When CPU Module Logging Configuration Tool is used • Request or setting data error	• Check the request data that has been specified. ■When CPU Module Logging Configuration Tool is used • Check the specified data, and write it to the CPU module again.
4081H	Any other error	• The search target data cannot be detected.	• Check the data to be searched.
4082H	Any other error	• The specified command cannot be executed because it is being executed.	• Execute the command again after the processing of the command from another engineering tool ends.
4083H	Any other error	• An attempt was made to perform operation to the program not registered in parameter. ■When CPU Module Logging Configuration Tool is used • The data logging specifying the program that is stored in the CPU module but not registered in the CPU parameters was started.	• Register the program in parameter.
4084H	Any other error	• The specified pointer (P or I) does not exist.	• Check if the pointer (P or I) exist in the data.
4085H	Any other error	• The pointer (P or I) cannot be specified because the program is not specified in parameter.	• Register the program to be executed in parameter first, and specify the pointer (P or I).
4086H	Any other error	• The specified pointer (P or I) has already been added.	• Check and correct the pointer number to be added.
4087H	Any other error	• The number of pointers (P or I) exceeds its limit.	• Check and correct the specified pointer (P or I).

Error code	Error name	Error details and cause	Action
4088H	Any other error	• The specified step number is not at the head of the instruction. • The program differs from that stored in the CPU module. ■When CPU Module Logging Configuration Tool is used • The data logging specifying the step number that does not exist or is not specified as the start number of the instruction was started.	• Check and correct the specified step number. • Read the program from the CPU module to match it with that of the engineering tool, and then execute online change again.
4089H	Any other error	• An attempt was made to insert/delete the END instruction by online program change.	• Check the specified program file contents. • Set the operating status of the CPU module to STOP, and write the program.
408AH	Any other error	• The file capacity exceeded after the online change was executed.	• Check the capacity of the specified program file. • Set the operating status of the CPU module to STOP, and write the program.
408BH	Any other error	• The remote request cannot be executed.	• Change the status of the CPU module so that the remote request can be executed, and execute the request again. • For remote operation, set the parameter to "Enable remote reset".
408DH	Any other error	• The instruction code that cannot be handled exists.	• Check whether the model of the used CPU module is correct or not. • The program where online change was attempted includes the instruction that cannot be handled by the CPU module specified for the project. Check the program and delete the instruction.
408EH	Any other error	• The write step is illegal. • The program differs from that stored in the CPU module.	• Set the operating status of the CPU module to STOP, and write the program. • The starting position of online program change is not specified with the correct program step number. Check whether the engineering tool supports the model and version of the CPU module that is specified for the project. • Read the program from the CPU module to match it with that of the engineering tool, and then execute online the program change again.
40A0H	SFC device specification error	• A block No. outside the range was specified.	• Check the settings and make a correction.
40A1H	SFC device specification error	• The number of blocks exceeds the range.	• Check the set quantity and correct.
40A2H	SFC device specification error	• A step No. outside the range was specified.	• Check the settings and make a correction.
40A3H	SFC device specification error	• The number of steps exceeds the range.	• Check the set quantity and correct.
40A4H	SFC device specification error	• A sequence step No. outside the range was specified.	• Check the settings and make a correction.
40A5H	SFC device specification error	• The specified device is outside the range.	• Check the set quantity and correct.
40A6H	SFC device specification error	• The block specification pattern and step specification pattern are incorrect.	• Check the settings and make a correction.
40A7H	SFC device specification error	• A block No. that does not exist in the 0 to 319 range was specified.	• Check the settings and make a correction.
40A8H	SFC device specification error	• A step No. that does not exist in the 0 to 511 range was specified.	• Check the settings and make a correction.
40B0H	SFC file related error	• The drive (memory) specified with the SFC program file operation is incorrect.	• Check the settings and make a correction.
40B1H	SFC file related error	• The SFC program specified with the SFC program file operation is incorrect.	• Check the specified file name and correct.

Error code	Error name	Error details and cause	Action
40B2H	SFC file related error	• The program specified with the SFC program file operation is not an SFC program.	• Check the specified file name and correct.
40B3H	SFC file related error	• An attempt was made to rewrite a dedicated SFC instruction by changing an SFC program online.	• Set the operating status of the CPU module to STOP, and write the SFC program.
40B4H	SFC file related error	• An attempt was made to change or delete the active block.	• Deactivate the target block and then change or delete it by using the online change.
40B5H	SFC file related error	• The number of SFC steps after the program change exceeded the maximum.	• Reduce the number of SFC steps to be added and execute the online change again.
40B6H	SFC file related error	• The specified block does not exist.	• Read the program from the CPU module to match it with that of the engineering tool, and then execute the online program change again.
40B7H	SFC file related error	• The online change cannot be executed to the standby type SFC program.	• Execute the online change (inactive SFC block) to the scan type SFC program.
40B8H	SFC file related error	• The device number of SFC information device is out of range.	• Check and correct the setting of the block information.
40B9H	SFC file related error	• The changed SFC program is incorrect.	• Execute again after checking the communication route such as the communication cable.
40BAH	SFC file related error	• The online change for each block cannot be executed to the SFC block whose number of sequential steps exceeds 32K steps.	• To execute the online change for each block, the number of sequential steps of the target SFC block must be within 32K steps. • After setting the operating status of the CPU module to STOP, write the SFC program to the programmable controller.
40BBH	SFC file related error	• The online change cannot be executed because writing to the programmable controller has just completed or a program execution error has occurred.	• After the operating status of the CPU module is changed from STOP to RUN, execute the online change (inactive SFC block). (The program can be changed while the operating status of the CPU module is STOP by writing the program to the programmable controller.) • When a program execution error (error code: 3204H) occurs, set the parameter so that one or less SFC program is set as the scan execution type.
40BDH	SFC file related error	• Online change (inactive SFC block) execution error	• Read the program from the CPU module to match it with that of the engineering tool, and then execute the online program change again. • After setting the operating status of the CPU module to STOP, write the SFC program to the programmable controller.
40C0H	Label communication error	• The specified label name does not exist.	• Check the label setting. If the "Access from External Device" checkbox is not selected, check the checkbox.
40C1H	Label communication error	• Label access with a label name has failed because the specified array element number is out of range.	• Specify the array element number within the set range.
40C2H	Label communication error	• Label access with a label name has failed because the bit array type label is not specified by bit specification.	• Specify the label by bit specification, and access again.
40C3H	Label communication error	• Label access with a label name has failed because the word array type label is not specified by word specification.	• Specify the label by word specification, and access again.
40C4H	Label communication error	• Label access with a label name has failed because the number of labels used in the program exceeded its limit.	• Reduce the number of labels in the program, and access the labels multiple times.
40C5H	Label communication error	• Label access with a label name has failed because the global label setting file and global label assignment information do not match.	• Check the access from an external device of the specified global label setting, and write the global setting file and the global label assignment information to the programmable controller together.
40C6H	Label communication error	• The request execution has failed because the global label is being modified by the CPU module specified.	• Wait until modification of the global label of the CPU module specified is completed, write files to the programmable controller, and execute the online change function again.

Error code	Error name	Error details and cause	Action
40C7H	Label communication error	• Although changes of the global label were written to the programmable controller, the reflection operation (STOP → RUN or Power-on and reset) of the global label setting file is not performed. • The request cannot be executed because the data in the global label setting file being processed and the specified consistency check data are not the same.	• Perform the reflection operation (STOP → RUN or Power-on and reset) of the global label setting file. • Correct the details of the global label setting, and write the file to the programmable controller again.
40C8H	Label communication error	• Registration of a label definition has failed because the number of registered labels reached its limit.	• Reduce the number of registered labels with the "Access from External Device" checkbox selected.
40C9H	Label communication error	• A label definition could not be registered because the registration target memory capacity has been exceeded its limit.	• Reduce the number of registered labels with the "Access from External Device" checkbox selected. • If the data memory is specified as the registration target memory, change the memory to the SD memory card. • Change the settings of the functions that use the registration target memory.
40CAH	Label communication error	• A label definition could not be changed, added, or deleted because the label communication data does not exist in the CPU module. ■When CPU Module Logging Configuration Tool is used • An attempt was made to change, add, or delete labels without creating the label communication data.	• Write the label communication data to the programmable controller.
40CBH	Label communication error	• Data are not written to the CPU module because the data type of the specified label does not match the size of the write data.	• Change the size of data written from the external device (SLMP/MC protocol device) so that it matches the data type of the label in the specified CPU module.
40CCH	Label communication error	• Online program change has failed because the global setting file before modification and the global label assignment information do not match.	• Write the global setting file and the global label assignment information to the programmable controller together after modification.
40CEH	Label communication error	• An attempt was made to access a label that cannot be accessed with a label name.	• Change the data type of the specified label to the one other than "Function Block" or "Pointer". • Change the class of the specified label to the one other than "VAR_GLOBAL_CONSTANT". • Specify a device that is not being modified. (Bit-specified word devices and digit-specified bit devices can be specified.)
4100H	Any other error	• Hardware failure of the CPU module.	• Replace the CPU module.
4101H	Any other error	• Serial communication connection was executed for a different CPU module series.	• Check the CPU module series.
4103H	Any other error	• The instruction written by online program change is incorrect or invalid.	• Execute online program change again. Or, set the operating status of the CPU module to STOP, and write the program.
4104H	Any other error	• An instruction(s) that cannot be executed by the CPU module set to process mode or redundant mode is included in the instructions targeted for online program change.	• Delete the instruction(s) that cannot be executed, and execute the online program change function again. Or, set the operating status of the CPU module to STOP, and write the program.
4105H	Any other error	• Hardware failure of the CPU module internal memory	• The possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.
4108H	Any other error	• The device monitor/test cannot be performed normally.	• Execute the function again. Check that the access prohibited area is not accessed, and execute the function again.
410AH	Any other error	• The specified command cannot be executed because the online program change is be ing executed. ■When CPU Module Logging Configuration Tool is used • The data logging specifying the step number as a sampling or trigger condition was started during online program change.	• Execute the request again after the online program change.
410BH	Any other error	• The monitor condition registration was cleared after the online program change was executed.	• Execute the registration of monitoring condition again after the online program change.
410CH	Any other error	• Writing to the specified data is not supported.	• Check that the version of the engineering tool used is correct. • Check the settings and make a correction.
410EH	Any other error	• When the execution status of the online program change is in error, the online program change command was issued.	• Issue the online program change cancel command, and execute the function again.
410FH	Any other error	• During the execution of the online program change function, the cancel command was issued by the same request source.	• Issue the command again after the currently-performed processing ends.

Error code	Error name	Error details and cause	Action
4110H	CPU module error	• The request cannot be executed because the CPU module is in a stop error state.	• Reset the CPU module, and execute the request again.
4111H	CPU module error	• The request cannot be executed because the other CPU modules, except the host CPU module, have not yet started in a multiple CPU system.	• Execute the request again after all the other CPU modules have started.
4121H	File related error	• The specified drive (memory) or file does not exist.	• Execute again after checking the specified drive (memory) or file.
4122H	File related error	• The specified drive (memory) or file does not exist.	• Execute again after checking the specified drive (memory) or file.
4123H	File related error	• The specified drive (memory) is abnormal. ■When CPU Module Logging Configuration Tool is used • The data logging was started to the memory having an error.	• Initialize the memory, and restore the drive (memory) to its normal state.
4124H	File related error	• The specified drive (memory) is abnormal.	• Initialize the memory, and restore the drive (memory) to its normal state.
4125H	File related error	• The specified drive (memory) or file is performing processing.	• Execute again after a while.
4126H	File related error	• The specified drive (memory) or file is performing processing.	• Execute again after a while.
4127H	File related error	• File password mismatch	• Execute again after checking the file password.
4128H	File related error	• File password mismatch with copy destination	• Execute again after checking the file password.
4129H	File related error	• The request cannot be executed since the specified drive (memory) is ROM.	• Execute again after changing the target drive (memory).
412AH	File related error	• The request cannot be executed since the specified drive (memory) is ROM.	• Execute again after changing the target drive (memory).
412BH	File related error	• The specified drive (memory) is write-inhibited.	• Execute again after changing the write inhibit condition or drive (memory).
412CH	File related error	• The specified drive (memory) is write-inhibited.	• Execute again after changing the write inhibit condition or drive (memory).
412DH	File related error	• The specified drive (memory) does not have enough free space.	• Execute again after increasing the free space of the drive (memory).
412EH	File related error	• The specified drive (memory) does not have enough free space.	• Execute again after increasing the free space of the drive (memory).
412FH	File related error	• The drive (memory) capacity differs between the drive (memory) copy destination and copy source.	• Execute again after checking the drive (memory) copy destination and copy source.
4130H	File related error	• The drive (memory) type differs between the drive (memory) copy destination and copy source.	• Execute again after checking the drive (memory) copy destination and copy source.
4131H	File related error	• The file name of the file copy destination is the same as that of the copy source.	• Execute again after checking the file names.
4132H	File related error	• The specified number of files does not exist.	• Execute again after checking the specified data.
4133H	File related error	• The specified drive (memory) has no free space.	• Execute again after increasing the free space of the drive (memory).
4134H	File related error	• The attribute specification data of the file is wrong.	• Execute again after checking the specified data.
4135H	File related error	• The date/time data of the engineering tool (personal computer) is out of range.	• Execute again after checking the clock setting of the engineering tool (personal computer).
4136H	File related error	• The specified file already exists.	• Execute again after checking the specified file name.
4137H	File related error	• The specified file is read-only.	• Execute again after changing the condition of the specified file.
4138H	File related error	• Simultaneously accessible files exceeded the maximum.	• Execute again after decreasing file operations.
4139H	File related error	• The size of the specified file has exceeded that of the existing file.	• Execute again after checking the size of the specified file.
413AH	File related error	• The specified file has exceeded the already existing file size.	• Execute again after checking the size of the specified file.
413BH	File related error	• The same file was simultaneously accessed from different engineering tools. ■When CPU Module Logging Configuration Tool is used • An operation was performed to a file being accessed.	• Execute again after a while.
413CH	File related error	• The specified file is write-inhibited.	• Execute again after changing the file condition.
413DH	File related error	• The specified file capacity cannot be secured.	• Execute again after increasing the capacity of the specified drive (memory).
413EH	File related error	• Operation is disabled for the specified drive (memory).	• Execute again after changing the target drive (memory).

Error code	Error name	Error details and cause	Action
413FH	File related error	• The file is inhibited to be written to the file storage area.	• Execute again after changing the specified drive (memory).
414AH	Intelligent function module specification error	• In a multiple CPU system, operation was performed to a non-controlled intelligent function module or network module.	• Execute the operation again from the control CPU of the target module.
414CH	Intelligent function module specification error	• The specified buffer memory address cannot be accessed.	• Execute again after checking the buffer address.
4150H	File related error	• An attempt was made to initialize the drive (memory) protected by the system.	• Do not initialize the target drive (memory) as it cannot be initialized.
4151H	File related error	• An attempt was made to delete the file/folder protected by the system.	• Do not delete the target file as it cannot be deleted.
4160H	Online registration error	• The registered number of I/O devices of the forced on/off target exceeded the maximum.	• Cancel the registration of I/O devices of the forced on/off target that is not used.
41C1H	File related error	• The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41C2H	File related error	• File open specification data for file access is wrong.	• Execute again after checking the specification data.
41C3H	File related error	• Simultaneously accessible files exceeded the maximum.	• Execute again after decreasing file operations.
41C4H	File related error	• Simultaneously accessible files exceeded the maximum.	• Execute again after decreasing file operations.
41C5H	File related error	• The specified file does not exist. ■When CPU Module Logging Configuration Tool is used When an attempt was made to re-register the data logging with the previous settings, the corresponding file did not exist.	• Execute again after checking the file.
41C7H	File related error	• The specified file or drive (memory) does not exist.	• Execute again after checking the file or drive (memory).
41C8H	File related error	• The size of the specified file has exceeded that of the existing file.	• Execute again after checking the size of the specified file. • If the error recurs after re-execution, the file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41C9H	File related error	• Access to the file sector has failed. • The format information data of the target drive (memory) is abnormal.	• Back up data in the CPU module, and then initialize the memory.
41CAH	File related error	• Access to the file sector has failed. • The format information data of the target drive (memory) is abnormal.	• Back up data in the CPU module, and then initialize the memory.
41CBH	File related error	• The file name is specified in a wrong method.	• Execute again after checking the file name.
41CCH	File related error	• The specified file does not exist. Or, the specified subdirectory does not exist. ■When CPU Module Logging Configuration Tool is used • The data logging was started in a state that sub-folders for storing data logging files (or folders) cannot be created or accessed. Or, sub-folders cannot be created or accessed while the data logging is being performed or the logged data is being saved.	• Execute again after checking the name of the file and subdirectory.
41CDH	File related error	• An access to the file is prohibited in the system. ■When CPU Module Logging Configuration Tool is used • The data logging was started in a state that files (or folders) cannot be created or accessed because a file (or folder) with the same name exists. Or, files (folders) cannot be created or accessed while the data logging is being performed or the logged data is being saved.	• Do not access the specified file or subdirectory. • Execute again after checking the file and subdirectory.
41CEH	File related error	• The file cannot be written because the specified file is read-only.	• Execute again after checking the attribute of the specified file.
41CFH	File related error	• The specified drive (memory) has been used exceeding the capacity.	• Execute again after checking the drive (memory) capacity.
41D0H	File related error	• The specified drive (memory) has no free space. Or, the number of files in the directory of the specified drive (memory) has exceeded the maximum.	• Execute again after increasing the free space of the drive (memory). • Delete files in the drive (memory), and execute the function again.

Error code	Error name	Error details and cause	Action
41D1H	File related error	• The file name is specified in a wrong method. • The SD memory card is being disabled by SM606 (SD memory card forced disable instruction).	• Execute again after checking the file name. • If the error recurs after re-execution, the file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory. • Cancel the SD memory card forced disable instruction.
41D5H	File related error	• The file of the same name exists.	• Forcibly execute the request, or execute after changing the file name.
41D6H	File related error	• The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41D7H	File related error	• The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41D8H	File related error	• The specified file is being accessed.	• Execute again after a while.
41DFH	File related error	• The specified drive (memory) is write-protected.	• Execute again after canceling the write protect of the specified drive (memory).
41E0H	File related error	• The specified drive (memory) is abnormal or does not exist.	• Check that an SD memory card is inserted, and execute the function again. • Back up data, and then initialize the memory.
41E1H	File related error	• Access to the SD memory card has failed.	• Back up data, and then write the data to the data memory.
41E2H	File related error	• Access to the SD memory card has failed. No SD memory card is inserted. Or, the data in the SD memory card has been corrupted when the CPU module is powered off or reset or the SD memory card is removed during an access to the SD memory card. Or, the SD memory card has reached the end of life.	• Check that an SD memory card is inserted, and execute the function again. • Back up data, and then initialize the SD memory card. • Execute the operation again after replacing the SD memory card.
41E4H	File related error	• Access to the SD memory card has failed.	• Execute the operation again after checking that the SD memory card has been inserted. • Execute the operation again after replacing the SD memory card. • Back up data, and then initialize the memory.
41E7H	File related error	• The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41E8H	File related error	• The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41E9H	File related error	• The specified file is being accessed.	• Execute again after a while.
41EBH	File related error	• The file name is specified in a wrong method.	• Execute again after checking the file name.
41ECH	File related error	• The file system of the specified drive (memory) is logically corrupted.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41EDH	File related error	• The specified drive (memory) does not have continuous free space. (The free space for file is sufficient but the continuous free space is insufficient.)	• Execute again after deleting unnecessary files.
41EFH	File related error	• Creation of the power failure backup data in the specified drive (memory) has failed.	• Execute the operation again after checking that the SD memory card has been inserted.
41F0H	File related error	• The power failure backup data of the specified drive (memory) are corrupted.	• Execute the operation again after checking that the SD memory card has been inserted.
41F3H	File related error	• The file size is larger than the following: • the value to be acquired when 2 bytes are subtracted from 4G bytes.	• Specify a smaller value for the file size when creating a file or changing the file size. Alternatively, divide the file so that each file size is smaller.
41F4H	File related error	• The request cannot be executed because the operation is prohibited by the system.	• Do not request the file operation because it is prohibited by the system.
41F5H	File related error	• The command for the (split storage) program was executed to the file other than the (split storage) program file. Or, the command not for the (split storage) program was executed to the (split storage) program file.	• Check the command, and request the command applicable to the target file.
41F8H	File related error	• The data is being accessed from another engineering tool.	• Data are being written to the program memory or being transferred to the backup memory. Access the file after the currently-performed processing ends.

Error code	Error name	Error details and cause	Action
41FAH	File related error	• Program was written beyond the area where the program can be executed.	• Execute again after reducing either the already written program or newly written program.
41FBH	File related error	• The specified file is already being manipulated from the engineering tool.	• Execute again after the currently performed operation is completed.
41FCH	File related error	• An attempt was made to initialize the drive (memory) being used.	• Stop all accesses to the specified drive (memory), and execute the request again.
41FDH	File related error	• There are no data written to the data memory.	• Write all the required files to the programmable controller.
41FEH	File related error	• The SD memory card has not been inserted. • The SD memory card is being disabled. • The SD memory card is being disabled by SM606 (SD memory card forced disable instruction). ■When CPU Module Logging Configuration Tool is used • The data logging was started when the CPU module is in the following state: no SD memory card is inserted; the CARD READY LED is not on (green); or the SD memory card is forcibly disabled.	• Insert the SD memory card. • Remove the SD memory card, and insert it again. • Cancel the SD memory card forced disable instruction.
41FFH	File related error	• The type of the SD memory card differs.	• Check the type of the SD memory card.
4202H	Online module change function error	• The request cannot be executed because the specified module is being changed online.	• Execute the request again after the online module change processing ends. • If the online module change operation cannot be completed, power off the system, and change the module.
421DH	Online module change function error	• Redundant operations cannot be performed because the redundant function module is being changed online.	• Perform redundant operations after the online module change processing ends.
4240H	Redundant system related error	Any of the following operations is requested to the CPU module in the standby system, but cannot be performed because they are not supported. • Operation mode change • System switching • Memory copy from control system to standby system • Control system forced start-up while waiting for a start-up of the other system	• Perform these operations to the CPU module in the control system.
4241H	Redundant system related error	Data communications cannot be performed with the other system because of any of the following reasons: • The CPU module of the other system is powered off or in a reset state. • The tracking communication stops due to an error in the CPU module of the own or other system. • Tracking cables are disconnected or incorrectly connected, or failed. • The redundant function module of the own or other system is being changed online, a module communication test is being performed for the module, or a hardware failure has occurred in the module.	• If a WDT error has occurred in the CPU module of the own or other system, eliminate the error cause and then execute the operation again. • Check that there is no error or failure in the CPU module, tracking cables, redundant function modules, and perform a retry. Or, perform a retry after the online module change processing or the module communication test ends. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.
4243H	Redundant system related error	• The request cannot be executed because a stop error has occurred in the CPU module of the standby system.	• Check the stop error, eliminate the error cause, and then execute the request again.
4244H	Redundant system related error	• The request cannot be executed because the operating status of the CPU module in the control system differs from that of the CPU module in the standby system.	• Match the operating status of the CPU module between the systems, and execute the request again.
4246H	Redundant system related error	• The request cannot be executed because the operation mode is being changed or the systems are being switched.	• Execute the request again after the operation mode change processing or the system switching processing ends.
4247H	Redundant system related error	• The request cannot be executed because the memory copy function is being executed.	Execute the request again after the memory copy processing ends. The completion status can be checked in the following special relay areas. • SM1654 (Memory copy being executed): Off • SM1655 (Memory copy completion): On
4248H	Redundant system related error	Data communications cannot be performed with the system specified as a connection destination (request destination module I/O number) because of either of the following reasons: • The request was issued during the system switching processing. • The system specified as a connection destination (request destination module I/O number) does not exist.	Check the following, and execute the request again. • The system switching processing has completed. • The redundant system has started up successfully.

Error code	Error name	Error details and cause	Action
4249H	Redundant system related error	• The request cannot be executed because the system setting (system A or B, control system or standby system) has not been determined.	• Set one system as system A or control system and the other system as system B or standby system, and execute the request again. Or, do not specify any connection destination system (request destination module I/O number), and execute the request again.
424AH	Redundant system related error	• The request cannot be executed because the system A, system B, control system, or standby system is set as a connection destination (request destination module I/O number).	• Do not specify any connection destination system (request destination module I/O number), and execute the request again.
424BH	Redundant system related error	The request cannot be executed because system switching is disabled because of either of the following reasons: • SM1646 (System switching by a user) is off. • The DCONTSW instruction is being executed.	Perform either of the following operations to enable system switching, and execute the request again. • Turn on SM1646 (System switching by a user). • Execute the ECONTSW instruction.
424CH	Redundant system related error	• The request cannot be executed because the online program change function is being executed.	• Execute the request again after the online program change processing ends.
424EH	Redundant system related error	• The request cannot be executed because an unsupported system switching is specified.	• Take measures to reduce noise, and execute the request again. If the same error code is displayed again, the possible cause is a hardware failure of the target module. Please consult your local Mitsubishi representative.
424FH	Redundant system related error	• The request cannot be executed because system switching due to a different cause was executed during execution of system switching by the engineering tool.	• Check that the systems are switched successfully. If not, monitor SD1644 (Cause of system switching failure), eliminate the error cause, and then execute the function again.
4251H	Redundant system related error	• The request cannot be executed because the redundant system is in separate mode.	• Change the operation mode to backup mode, and execute the request again.
4252H	Redundant system related error	• The systems cannot be switched because an error has occurred in an intelligent function module of the standby system.	• Identify an error module by monitoring SD1646 (System switching request status from a network module of the other system), eliminate the error cause, and then switch the systems again.
4256H	Redundant system related error	• The request cannot be executed because a timeout error has occurred in tracking communications.	• Check if the tracking cables have been properly connected. If the same error code is displayed again even after the tracking cables are connected properly, the possible cause is a hardware failure of the CPU module, redundant function module, or tracking cable. Please consult your local Mitsubishi representative.
4258H	Redundant system related error	• The operation mode cannot be changed because the control system in separate mode is in a state waiting for transition to RUN.	• Change the operating status of the CPU module whose PROGRAM RUN LED is flashing to RUN by using the RUN/STOP/RESET switch or remote operation, and change the operation mode again.
4259H	Redundant system related error	• The operation mode cannot be changed from separate mode to backup mode because the communication route differs from that of when the mode was changed from backup mode to separate mode.	• Change the operation mode using the same communication route of when the mode was changed from backup mode to separate mode.
425AH	Redundant system related error	The system cannot be started as a control system because of either of the following reasons: • The system A/B setting has not been determined. • The system is being changed to the control system.	Take the following actions: • Set one system as system A and the other system as system B, restart the CPU module. • Check that the system has been changed to the control system.
425EH	Redundant system related error	• Data communications cannot be performed with the other system because a module communication test is being executed on a redundant function module.	• Retry data communications after the module communication test ends.
425FH	Redundant system related error	• The memory copy function cannot be executed because the CPU module models differ between the control system and the standby system.	• Match the CPU module model between the systems, and execute the function again.
4269H	Any other error	• The remote RUN (function) cannot be executed.	• Execute the function again after a while.
426AH	Any other error	• The date and time data specified by an engineering tool (personal computer) are less than one hour from the start time of daylight saving time.	• Specify the time to other than one less than one hour from the start time of daylight saving time using an engineering tool (personal computer). Then, execute the function again. • Check and Correct the start time of the daylight saving time setting.

Error code	Error name	Error details and cause	Action
4270H	Debug related function error	• Data logging is being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data) to another memory.	• Register data logging to the memory where the data logging is being performed. Or, stop the data logging being performed and register again. ■When CPU Module Logging Configuration Tool is used • Start the data logging to the memory where the data logging is being performed. Or, stop the data logging being performed, and start the data logging.
4271H	Debug related function error	• The specified data logging is already being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).	• Stop the data logging. Or, write, delete, or register data logging to the setting number where no data logging is being performed.
4272H	Debug related function error	• The trigger logging specifying the device as a trigger condition is being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).	• Change the trigger condition. Or, stop the trigger logging being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data), and register another trigger logging. ■When CPU Module Logging Configuration Tool is used • Change the trigger condition. Or, stop the trigger logging being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data), and start another trigger logging.
4275H	Debug related function error	• Auto logging is being performed.	• After the auto logging, replace the SD memory card and execute again.
4276H	Debug related function error	• The function that cannot be executed during the data logging (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data) was executed.	• Stop the data logging, and then execute the function.
4277H	Debug related function error	• The number of saved files exceeded the specified number. ■When CPU Module Logging Configuration Tool is used • The data logging was started in a state where the number of saved files has exceeded the specified number. (The operation when the number of saved files exceeded is set to "Stop".) Or, the data logging was started in a state where the number of saved files has exceeded the specified number. (The operation when the number of saved files exceeded is set to "Overwrite".)	• The number of files saved in the storage destination memory has exceeded the setting value. Delete files, or change the storage destination and then register. ■When CPU Module Logging Configuration Tool is used • The number of files saved in the storage destination memory has exceeded the setting value. Delete files or change the storage destination, and then start the data logging.
4278H	Debug related function error	• An attempt was made to register data logging in a state where the saved file number has reached its maximum, FFFFFFFF. Or, the number reached to the maximum during the execution. ■When CPU Module Logging Configuration Tool is used • The data logging was started in a state where the saved file number has reached its maximum, FFFFFFFF. Or, the number reached to the maximum during the execution.	• The saved file number in the storage target memory has reached its maximum, FFFFFFFF. Delete files, or change the storage destination and then register. ■When CPU Module Logging Configuration Tool is used • The saved file number in the storage target memory has reached its maximum, FFFFFFFF. Delete files or change the storage target memory, and then perform the data logging.
427AH	Debug related function error	• Common setting file does not exist. ■When CPU Module Logging Configuration Tool is used • The data logging was started to the memory where the common settings are not stored.	• Write the common settings to the target memory. • Register data logging to the memory where the common settings are stored. ■When CPU Module Logging Configuration Tool is used • Write the common settings to the target memory. • Start the data logging to the memory where the common settings are stored.

Error code	Error name	Error details and cause	Action
427BH	Debug related function error	• The data logging with the same file storage destination is being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data). ■When CPU Module Logging Configuration Tool is used • The data logging with the same file storage destination is being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).	• Stop the data logging destined for the same storage, and then register. Or, change the storage destination of the file, and then register. ■When CPU Module Logging Configuration Tool is used • Stop the data logging destined for the same storage, and then perform another data logging. Or, change the storage destination of the file, and then register.
4282H	Debug related function error	• The registration was performed with the internal buffer capacity set to 0.	• Check and correct the internal buffer capacity setting.
4283H	Debug related function error	• An attempt was made to register trigger logging in a state that the specified number of records before trigger has exceeded the number of records that can be collected within the internal buffer capacity.	• Check and correct the internal buffer capacity setting. • Reduce the number of records before trigger.
4284H	Debug related function error	• The error codes registered for memory dump are incorrect.	• Check and correct the error codes.
4285H	Debug related function error	• A non-executable function has been executed during collection or save in memory dump.	• Execute the function again after the completion of save in memory dump.
4286H	Debug related function error	• An item that cannot be set in the Safety CPU is specified in the data logging settings.	• Correct the data logging settings.
4287H	Debug related function error	• A safety program is specified in the trigger logging step No. specification.	• Ensure that no safety program is specified in the trigger logging step No. specification.
4288H	Debug related function error	• The request cannot be executed because the number of characters used in the name of the specified file exceeded its limit.	• Rename the file within 61 one-byte characters (including underscore, serial number (8 digits), period, and extension), and execute the request again.
4291H	Debug related function error	• Online program change has been executed during the execution of the real-time monitor where a step number is specified as "Timing" or "Trigger condition".	• After the completion of the online program change, check the step number specified and execute real-time monitor again.
4292H	Debug related function error	• Real-time monitor does not begin during the execution of real-time monitor.	• Start real-time monitor after the stop of the real-time monitor being executed.
4293H	Debug related function error	• Execution fails because the internal buffer exceeds its maximum capacity.	• Check and correct the settings of internal buffer capacity, and then try again.
4294H	Debug related function error	• Execution fails because the settings of internal buffer capacity has been changed during the execution of functions that consume the internal buffer.	• Deactivate the functions that consume the internal buffer, and then try again. Or, restore the internal buffer capacity to the previous state, and then try again.
42A3H	Safety related error	• The safety operation mode specification details lie outside the range.	• Change the safety operation mode again.
42A4H	Safety related error	• An attempt was made to change the safety operation mode to TEST MODE while in TEST MODE, or to change the safety operation mode to SAFETY MODE while in SAFETY MODE.	• Check whether an attempt was made to change the current safety operation mode to the same mode.
42A5H	Safety related error	• An operation that cannot be executed in SAFETY MODE was performed.	• Change the safety operation mode to TEST MODE and execute again.
42A7H	Safety related error	• The safety operation mode cannot be changed to SAFETY MODE.	• Change the CPU module operating status to other than RUN and execute again. • Execute again after completing writing of the safety program, safety FB file, safety CPU parameters, safety module parameters, safety global label setting file, and standard/safety shared label setting file.
42A8H	Safety related error	• The safety operation mode cannot be changed to TEST MODE.	• Change the safety operation mode again.
42A9H	Safety related error	• A CRC error occurred during communication with the Safety CPU.	• Take measures to reduce noise, and retry data communications. • If the same error code is displayed again, the possible cause is a hardware failure at the applicable module. Please consult your local Mitsubishi representative.

Error code	Error name	Error details and cause	Action
42B0H	Safety related error	• Communication between the Safety CPU and safety function module failed.	• Take measures to reduce noise, and retry data communications. • If the same error code is displayed again, the possible cause is a hardware failure at the applicable module. Please consult your local Mitsubishi representative.
42B4H	Safety related error	• An error was detected when performing safety mutual monitoring between the Safety CPU and safety function module.	• Take measures to reduce noise, and retry data communications. • If the same error code is displayed again, the possible cause is a hardware failure at the applicable module. Please consult your local Mitsubishi representative.
42B8H	Safety related error	• The size of the program stored in the program memory and the size of the program being executed differ.	• Restart the Safety CPU. • If the same error code is displayed again, the possible cause is a hardware failure at the applicable module. Please consult your local Mitsubishi representative.
42B9H	Safety related error	• The Safety CPU received a request from an unsupported device.	• Send the request to the Safety CPU again from a supported device.
42BAH	Safety related error	• An SD memory card was specified for the write destination for the safety sequence program, safety FB file, safety CPU parameters, safety module parameters, safety global label setting file, and standard/safety shared label setting file.	• Try the write operation again.
42BBH	Safety related error	• The maximum capacity for safety programs and safety FB files that can be stored in the specified drive (memory) has been exceeding.	• Delete the safety programs or safety FB files stored in the Safety CPU drive (memory), and execute again. • Execute again after formatting Safety CPU drive (memory). • Reduce the safety program and safety FB file to be written to less than 40K steps and execute again.
42BCH	Safety related error	• A file mismatch was detected between the Safety CPU and safety function module.	• Run again after ensuring that the safety function module has been mounted. • Format the memory if the safety function module has not been mounted. Write all files, reset the safety CPU module, and run it again.
42BEH	Safety related error	• The number of safety sequence programs, safety FB files, safety CPU parameters, safety module parameters, safety global label setting files, and standard/safety shared label setting files exceeded the maximum number of files that can be stored on the specified drive (memory).	• Execute again after deleting the safety sequence programs, safety FB files, safety CPU parameters, safety module parameters, safety global label setting files, and standard/safety shared label setting files on the drive (memory).
42C0H	Redundant system related error	• An attempt was made to write an SFC program to the CPU module set to redundant mode by executing the online program change function.	• Check that an SFC program is not written to the CPU module set to redundant mode.
42C1H	Redundant system related error	• Since the operation mode is changed or the systems were switched during the initial processing or the initial processing (when switched to RUN), it takes time to complete the processing.	• Check whether the processing ends or not after a while. • Take measures to reduce noise. • If the same error code is displayed again even after the redundant system is restarted, the possible cause is a hardware failure of the CPU module, redundant function module, or tracking cable. Please consult your local Mitsubishi representative.
4300H	Security function error (user authentication)	• An operation requiring authentication was executed when no authentication was received from the CPU module. • An operation requiring authentication was executed from a startup source or function that does not support user authentication.	• Log in to the CPU module. • Perform operation from a device which supports the user authentication function.
4309H	Security function error (user authentication)	• The specified user name has not been registered in the CPU module.	• Register user control information in the CPU module. • Log in with a registered user name.
430CH	Security function error (user authentication)	• An attempt was made to access an unauthorized file at operator access level.	• Access files within the authorized scope. • Files should be accessed by authorized users.
430DH	Security function error (user authentication)	• An operation requiring authentication was executed from a startup source that does not support the user authentication function with the user authentication function enabled.	• Perform the operation from a device that supports the user authentication function.
4314H	Security function error (user authentication)	• A user authentication operation was performed with the user authentication function disabled.	• Perform the operation after enabling the user authentication function.
4317H	Security function error (user authentication)	• An attempt was made to change the user control information of the logged on user.	• The operation should be performed after the logged on user logs off.

Error code	Error name	Error details and cause	Action
4318H	Security function error (user authentication)	• An attempt was made to initialize all programmable controller information (operation that disables user authentication function) with a user logged on.	• The operation should be performed after the logged on user logs off.
4319H	Security function error (user authentication)	• An attempt was made to initialize all programmable controller information (operation that disables user authentication function) while logged on.	• The operation should be performed after logging off.
431BH	Security function error (user authentication)	• An attempt was made to perform an unauthorized operation at operator access level.	• Check the operator access level.
431CH	Security function error (user authentication)	• The number of simultaneously logged on users exceeds the maximum value, and therefore the system is unable to authenticate the applicable user.	• Unnecessary logged on users should log off. • Adjust the number of log ons to within the maximum value.
431DH	Security function error (user authentication)	• A file write operation was performed with no user control information registered in the CPU module.	• Register user control information in the CPU module.
433CH	Maintenance and inspection error	• The error was not cleared. (Error clear was performed during execution of error clear.)	• Execute again after a while. If the same error code is displayed again, the possible cause is a hardware failure of the target module. Please consult your local Mitsubishi representative.
433DH	Maintenance and inspection error	• The target module cannot handle the error clear.	• Check the target module. (Check the module in which the error occurred.)
4400H	Security function error	• A file protected by a password has been opened without unlocking the password.	• Enter a correct password and perform password authentication.
4401H	Security function error	• Read password authentication has failed when required. • The file password format is incorrect.	• Set a correct read password and perform password authentication. • Access the file with the correct method.
4402H	Security function error	• Write password authentication has failed when required. • The file password format is incorrect.	• Set a correct write password and perform password authentication. • Access the file with the correct method.
4403H	Security function error	• Both passwords for reading and for writing that are set upon Create, Change, Delete, or Disable do not match the previous ones.	• Set correct passwords for both reading and writing, and perform password authentication.
4404H	Security function error	• A file error was detected before or after performing Create, Change, or Delete.	• Initialize the drive including the target file by initializing the memory. • Write the target file to the programmable controller again, and then register or cancel the file password.
4408H	Security function error	• Password authentication failed when access was required.	• Set a correct password and perform password authentication again.
4409H	Security function error	• Password authentication failed when access was required.	• Set a correct password and perform password authentication again 1 minutes later.
440AH	Security function error	• Password authentication failed when access was required.	• Set a correct password and perform password authentication again 5 minutes later.
440BH	Security function error	• Password authentication failed when access was required.	• Set a correct password and perform password authentication again 15 minutes later.
440CH	Security function error	• Password authentication failed when access was required.	• Set a correct password and perform password authentication again 60 minutes later.
440DH	Security function error	• Password authentication failed when access was required.	• Set a correct password and perform password authentication again 60 minutes later.
440EH	Security function error	• The security function was activated and password authentication cannot be performed.	• Set a correct password and perform password authentication again after a certain period of time.
440FH	Security function error	• An operation was performed to the firmware update prohibited file with a file password set.	• Disable the file password setting.

Error code	Error name	Error details and cause	Action
4410H	Security function error	• The file of the locked CPU module is accessed without the security key authentication.	• Register the security key which locks the CPU module to the engineering tool. • When the project is opened, lock the project with the security key which locks the CPU module. • When the CPU module is locked, the access control target file cannot be accessed using the following functions or external devices. · FTP server function · SLMP/MC protocol · GOT · EZSocket
4412H	Security function error	• The security key cannot be registered to the CPU module due to the failure of the internal memory where the security key is registered. Or, the security key of the CPU module cannot be deleted.	• Hardware failure of the CPU module. Replace the CPU module.
4413H	Security function error	• Since the CPU module is locked and 32 engineering tools are reading and writing programs simultaneously, another engineering tool cannot read or write programs.	• Wait until the number of engineering tools which are reading and writing programs decreases to 31 or less, and read or write programs.
4414H	Security function error	• The request cannot be executed because the CPU module is locked.	• Do not execute the request because it cannot be executed by the CPU module which is locked.
4415H	Security function error	• The request cannot be executed because the CPU module is not locked.	• Do not execute the request because it cannot be executed by the CPU module which is not locked.
4416H	Security function error	• The request cannot be executed because the CPU module lock or unlock processing is being performed.	• Execute the request again after the lock or unlock processing ends.
4417H	Security function error	• When an attempt was made to write or delete the security key, no extended SRAM cassette was inserted to the CPU module.	• Check that the extended SRAM cassette is correctly inserted, and then perform the operation again.
4418H	Security function error	• The security key cannot be changed or deleted because any locked program exists in the CPU module.	• Unlock all the programs in the CPU module.
4422H	Security function error	• The access target CPU module does not support the security key information stored in the engineering tool.	• Change the security key information version of the engineering tool in accordance with the version supported by the target CPU module.
4424H	Security function error	• The security key set to the file batch-written by executing the online change function does not match the one registered to the CPU module (or extended SRAM cassette).	• Set the same security key registered to the CPU module to the file, and then execute the online change function.
4425H	Security function error	• The file with a security key was batch-written by executing the online change function, but no security key is registered to the CPU module (or extended SRAM cassette).	• Clear the security key set to the file, and then execute the online change function.
4800H	iQ Sensor Solution related error	• The specified command cannot be executed because the iQ Sensor Solution data backup/restoration function is being executed.	• Execute the command again after the processing of the data backup/restoration function ends.
4801H	iQ Sensor Solution related error	• When the iQ Sensor Solution data backup/restoration is requested, the target module does not exist. • When the iQ Sensor Solution data restoration is requested, the specified backup folder does not exist. • The value in the special register (SD) related to the iQ Sensor Solution data backup/restoration is out of range.	• Check and correct the value in the special register (SD), and execute the function again. • Check that the backup data created by the MELSEC iQ-R series module is specified.
4802H	iQ Sensor Solution related error	• The number of backup folders used by the iQ Sensor Solution data backup/restoration function has reached to its limit.	• Delete an existing backup folder(s), and execute the function again. • Set "Automatic specification (Folder deletion supported)" in the special register for setting a backup target folder number.
4803H	iQ Sensor Solution related error	• The system file does not exist in the specified backup data, or the system file is corrupted.	• Specify another backup data.
4804H	iQ Sensor Solution related error	• The backup file does not exist in the specified backup data, or the backup file is corrupted.	• Specify another backup data.
4805H	iQ Sensor Solution related error	• When the iQ Sensor Solution data backup function is executed, no backup target device exists. • The data backup function was executed to a device which does not support the function.	• Check and correct the value in the special register (SD), and execute the function again. • Execute the function to a device which supports the function.

Error code	Error name	Error details and cause	Action
4806H	iQ Sensor Solution related error	An SD memory card is in the following state while the function is being executed. • An SD memory card is not inserted. • An SD memory card is disabled by SM606 (SD memory card forced disable instruction).	• Insert or re-insert an SD memory card, and execute the function again. • Enable the SD memory card operation, and execute the function again.
4807H	iQ Sensor Solution related error	• Data communications cannot be performed with a target device of the iQ Sensor Solution data backup/restoration function.	• Check the operation of the target device. • Check the network status of the target device, such as a cable, hub, and router. • The line may be busy. Perform a retry after a while.
4808H	iQ Sensor Solution related error	• When restoring backup data to a device supporting iQ Sensor Solution, a device of the specified backup data and a restoration target device do not match.	• Check and correct the specified backup data and the device supporting iQ Sensor Solution (manufacturer, model name, version) specified as the restoration target device.
4809H	iQ Sensor Solution related error	• The data backup/restoration function of iQ Sensor Solution was executed to the module which does not support the function.	• Replace the module to the one which supports the function. • Execute the function to the module which supports the function.
480CH	iQ Sensor Solution related error	• The specified command cannot be executed because the automatic detection of connected device function of iQ Sensor Solution is being executed.	• Execute the command again after the automatic detection processing ends.
480DH	iQ Sensor Solution related error	• The specified command cannot be executed because the communication setting reflection function of iQ Sensor Solution is being executed. • A communication timeout occurred in a command of a device supporting iQ Sensor Solution.	• Execute the command again after the communication setting reflection processing ends. • Check and correct the communication time check setting value using the engineering tool.
480EH	iQ Sensor Solution related error	• The specified command cannot be executed because the monitor function of iQ Sensor Solution is being executed. • The specified command cannot be executed because the sensor parameter read/write function of iQ Sensor Solution is being executed. • A communication timeout occurred in a command of a device supporting iQ Sensor Solution.	• Execute the function again after a while. • Execute the command again after the sensor parameter read/write processing ends. • Check and correct the communication time check setting value using the engineering tool.
480FH	iQ Sensor Solution related error	• The iQ Sensor Solution data restoration function cannot be executed because the CPU module cannot recognize the system file for the iQ Sensor Solution data backup/ restoration function. (The file version differs.)	• Execute the function using the CPU module that can recognize the system file. (Use the CPU module that can recognize the system file version.)
4810H	iQ Sensor Solution related error	• The restoration target module differs from the module where the iQ Sensor Solution data backup function was executed.	• Execute the data restoration function to the module where the data backup function was executed.
4811H	iQ Sensor Solution related error	• The backup target module this time differs from that last time.	• Execute the data backup function to the module where the data backup function was executed last time. • Set a different folder number.
4812H	iQ Sensor Solution related error	• The functions that cannot be executed simultaneously with the data backup/restoration function of iQ Sensor Solution, such as the file transfer function (FTP server/client), are being executed.	• Execute the function again after a while.
4905H	Any other error	• The capacity of label area used exceeded its limit.	• Check the program, delete the unused local and global label definitions, compile the program, and then write the program to the programmable controller. • Change the label area size in parameter, compile the program, and then write the program to the programmable controller.
4906H	Any other error	The file register data cannot be cleared by either of the following reasons: • The QDRSET instruction was executed more than once after the CPU module was powered on or reset. • The file register setting parameter is set to the one other than "Use Common File Register in All Programs".	• If the QDRSET instruction is being executed, power on or reset the CPU module. • The operation cannot be performed if the file register setting parameter is not set to "Use Common File Register in All Programs".

Error code	Error name	Error details and cause	Action
4A00H	Network error	• Access to the specified station cannot be made since the routing parameters are not set to the start source CPU module and/or relay CPU module. • For routing via a multiple CPU system, the control CPU of the network module for data routing has not started. • The third byte of the IP address (network number) specified for the IP communication test is the same as that of the CPU module where the test is executed. • The CPU module that performs IP packet transfer is not the control CPU of the CC-Link IE module, which is on the path that IP packets travel.	• Set to the related stations the routing parameters for access to the specified station. • Retry after a while. Or, start communication after checking that the system for data routing has started. • Do not use the same third byte of the IP address (network number) specified for the IP communication test with that of the CPU module where the test is executed. • Set the CPU module that performs IP packet transfer as the control CPU of the CC-Link IE module, which is on the path that IP packets travel.
4A01H	Network error	• The network of the number set to the routing parameters does not exist. • The specified CPU module cannot be communicated through the network that is not supported by the CPU module.	• Check and correct the routing parameters set to the related stations. • Set communication through the network that is supported by the specified CPU module.
4A02H	Network error	• Access to the specified station cannot be made.	• Check the network module for error, or check that the modules are not in offline. • Check if the network numbers/PC numbers are correctly set.
4A03H	Network error	• A request for network test was issued.	• Check the request of the SLMP/MC protocol.
4A20H	IP communication test error	• The upper 2 bytes of the IP addresses do not match between the CPU module and the request destination device on the same path that IP packets travel. • The upper 2 bytes of the IP addresses do not match between the CPU module and the CC-Link IE module on the same path that IP packets travel. • The upper 2 bytes of the IP addresses do not match between the CC-Link IE modules on the same path that IP packets travel. • The upper 2 bytes of the IP addresses do not match between the request source device and the CPU module connected to the request destination device by Ethernet.	• Check and correct the IP address settings of the CPU module. • Check and correct the IP address of the request destination device. • Check and correct the IP address of the CC-Link IE module. • Check and correct the IP address of the request source device.
4A21H	IP communication test error	• The 3rd byte (Network No.) or 4th byte (Station No.) of the IP address of the CPU module is the number that is not available for CC-Link IE. • The 3rd byte (Network No.) or 4th byte (Station No.) of the IP address of the request destination device is the number that is not available for CC-Link IE.	• Check and correct the IP address settings of the CPU module. • Check and correct the IP address of the request destination device.
4A22H	IP communication test error	• The IP address is not set to the CC-Link IE module on the path that IP packets travel.	• Set the IP address to the CC-Link IE module used as a master station. • Check the communication status with the master station when the CC-Link IE module is used as a local station. • Replace the CC-Link IE module (master station) with the one that supports the IP packet transfer function. • Conduct the IP communication test again after the CC-Link IE module is started up. • Check and correct the IP address of the request destination device.
4A23H	IP communication test error	• The CPU module on the path that IP packets travel does not support the IP packet transfer function. • Routing parameters are set so that IP packets are routed to CPU modules that are incompatible with IP packet transfer.	• Replace the CPU module with the one supporting the IP packet transfer function. • Correct routing parameter so that IP packets are routed to the CPU module that supports the IP packet transfer function. • Check and correct the IP address of the request destination device.

Error code	Error name	Error details and cause	Action
4A24H	IP communication test error	• The CC-Link IE module on the path that IP packets travel does not support the IP packet transfer function. • Routing parameters are set so that IP packets are routed to the network module that does not support the IP packet transfer function. • The 3rd byte (Network No.) of the IP address of the device on the path that IP packets travel is overlapping with the network No. of the module connected to the CPU module and does not support the IP packet transfer function. • The 3rd byte (Network No.) of the IP address of the request destination device is overlapping with the network No. of the module connected to the CPU module and does not support the IP packet transfer function.	• Replace the CC-Link IE module with the one supporting the IP packet transfer function. • Correct routing parameter so that IP packets are routed to the CC-Link IE module that supports the IP packet transfer function. • Check and correct the setting so that the 3rd byte (Network No.) of the IP address of the device on the path that IP packets travel does not overlap with the network No. of the module connected to the CPU module and does not support the IP packet transfer function. • Check and correct the setting so that the 3rd byte (Network No.) of the IP address of the request destination device does not overlap with the network No. of the module connected to the CPU module and does not support the IP packet transfer function. • Check and correct the IP address of the request destination device.
4A25H	IP communication test error	• The IP packet transfer setting is not set. • Routing parameters are set so that IP packets are routed to the CPU module where IP packet transfer setting is not set.	• Select "Use" for the IP packet transfer function setting in the CPU parameters. • Correct routing parameters so that IP packets are routed to the CPU module where IP packet transfer setting is set. • Check and correct the IP address of the request destination device.
4A27H	IP communication test error	• The CPU module that performs IP packet transfer is not the control CPU of the CC-Link IE module, which is on the path that IP packets travel.	• Set the CPU module that performs IP packet transfer as the control CPU of the CC-Link IE module, which is on the path that IP packets travel.
4A28H	IP communication test error	• In the system where the CPU module is connected to the request destination device over Ethernet, the request path and the response path of IP packets differ. • When multiple CC-Link IE modules with the same network number are connected in a multiple CPU system, the CC-Link IE module mounted on the lowest number slot is not set as a controlled module of the CPU module that transfers IP packets. • When multiple CC-Link IE modules with the same network number are connected in a single or multiple CPU system, the station number of the CC-Link IE module mounted on the lowest number slot is not set as a relay station number in routing parameter.	• Correct the routing parameter setting so that IP packets travel the same path for both request and response transmission. • When multiple CC-Link IE modules with the same network number are connected in a multiple CPU system, set the module mounted on the lowest number slot as a controlled module that transfers IP packets. • When multiple CC-Link IE modules with the same network number are connected in a single or multiple CPU system, set the station number of the module mounted on the lowest number slot as a relay station number in routing parameter.
4A29H	IP communication test error	• The 3rd byte (network No.) of the IP address of the request destination device is overlapping with the 3rd byte of the IP address of the CPU module connected to the request source device over Ethernet.	• Check and correct the IP address settings of the CPU module. • Check and correct the IP address of the request destination device.
4A2AH	IP communication test error	• An IP address of a device on CC-Link IE network or the CPU module is not specified.	• Specify an IP address of a device on CC-Link IE network or the CPU module.
4B00H	Target module error	• An error occurred in the access destination or the relay station. • The specified connection destination (request destination module I/O number) is incorrect. • The access destination CPU module has not started up.	• Check the error occurred in the specified access destination or the relay station, and take an action. • Check the connection destination (request destination module I/O number or PC number) in the request data of SLMP/MC protocol. • Check the stop error, and take an action.
4B02H	Target module error	• The request is not addressed to the CPU module.	• Perform operation for the module that can execute the specified function.
4B03H	Target module error	• The specified route is not supported by the specified CPU module version. • The communication target CPU module is not mounted. • A device mounted on the specified route does not support communications.	• Check whether the specified route is supported or not. • Check the mounting status of the CPU module. • Check the stop error, and take an action.
4B04H	Target module error	• The specified connection destination (request destination module I/O number) is not supported.	• An invalid value is set as the start I/O number of the connection target module. Set the start I/O number of the target module correctly, and retry data communications.
4C00H	Data logging function error	• There is not enough free space for storing the result file in the target memory.	• Increase the free space, and create the result file again.

Error code	Error name	Error details and cause	Action
4C01H	Data logging function error	• The result file cannot be written to the target memory because the SD memory card is write-protected or the folder/file structure is incorrect.	• Unlock the write protect switch of the SD memory card, and write the result file again. • Check that the SD memory card is not damaged. • Check that files/folders to be used in the SD memory card are not deleted.
4C02H	Data logging function error	• The SD memory card was removed while the data logging function was being executed (data logging status: Waiting RUN Not collected, Waiting to establish collection conditions Not collected, Waiting start Not collected, Pause, Collecting, Waiting trigger Collecting before trigger, Collecting after trigger, or Saving in progress).	• Insert the SD memory card, and execute the function again.
4C03H	Data logging function error	• The number of files in the root directory and subdirectory in the target memory exceeded the limit.	• Increase the free space of the drive (memory), and execute the function again. • Delete files in the drive (memory), and execute the function again.
4C04H	Data logging function error	• During auto logging, a data logging was not registered due to a registration failure of the data logging with another setting number.	• Clear the error, and start auto logging.
4C05H	Data logging function error	• The online change function was executed while the data logging function specifying the step number as a sampling or trigger condition was being executed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).	• Do not execute the online change function while the data logging function specifying the step number is being executed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data). • Stop the data logging function specifying the step number.
4C06H	Data logging function error	• System error	• Check the specified data, and write it to the CPU module again.
4C10H	CPU function dedicated error	• The maximum allowable capacity is exceeded. • The maximum allowable number of files is exceeded. • The upper limit of the backup folder number is exceeded. • The maximum length (255 characters) of the file path is exceeded.	• Increase the free space of the SD memory card and CPU module, and execute the function again. • Delete files in the SD memory card and CPU module, and execute the function again. • Delete backup data in the SD memory card and CPU module, and execute the function again. • Check and correct the folder structure or folder/file names of backup target data, and execute the data backup function again.
4C11H	CPU function dedicated error	• An SD memory card is not inserted. • An SD memory card is disabled by SM606 (SD memory card forced disable instruction).	• Insert or re-insert an SD memory card, and execute the function again. • Enable the SD memory card operation, and execute the function again.
4C12H	CPU function dedicated error	• Reading/writing of data from/to an SD memory card completed with an error.	• Check that an SD memory card is inserted, and execute the function again. • Replace an SD memory card, and execute the function again. • The backup data may have been corrupted. Execute the data restoration function using another backup data.
4C13H	CPU function dedicated error	• Reading/writing of data from/to the CPU built-in memory completed with an error.	• Back up data in the CPU built-in memory, initialize the memory, and write the data back to the original memory. Then, execute the data backup/restoration function. • The possible cause is a hardware failure of the restoration target CPU module. Execute the data restoration function to another CPU module.
4C14H	CPU function dedicated error	• The CPU module data backup/restoration function cannot be executed because a file password is set to the data. • Data was restored to the CPU module where the same data with a file password has already been stored.	• Delete file passwords, and execute the CPU module data backup/restoration function.
4C15H	CPU function dedicated error	• The function that cannot be executed simultaneously with the CPU module data backup/restoration function, such as the file transfer function (FTP), iQ Sensor Solution data backup/restoration function, and IP address change function, is being executed.	• Execute the function again after a while.

Error code	Error name	Error details and cause	Action
4C16H	CPU function dedicated error	• The automatic backup setting by specification of day and time is turned on while the setting value (automatic backup date and time) are out of range. • The automatic backup setting by specification of time and day of the week is turned on while the setting value (automatic backup time and day of the week) are out of range.	• Review the setting value (automatic backup date and time) and turn on the automatic backup setting by specification of day and time. • Review the setting value (automatic backup time and day of the week) and turn on the automatic backup setting by specification of time and day of the week.
4C17H	CPU function dedicated error	• The model of the restoration target CPU module differs from the model of the backup source CPU module.	• Execute the data restoration function to the CPU module whose model is the same as that of the backup source CPU module.
4C18H	CPU function dedicated error	• Data was restored while the operating status of the CPU module is in RUN or PAUSE.	• Change the operating status of the CPU module to STOP, and execute the function again.
4C19H	CPU function dedicated error	• The data restoration function was executed with backup files ($BKUP_CPU_INF.BSC and BKUP_CPU.BKD) not structured properly. • Data (file(s)) is missing in the backup file ($BKUP_CPU_INF.BSC) in the backup data folder. • The data restoration function was executed with a folder where no backup files ($BKUP_CPU_INF.BSC, BKUP_CPU.BKD, and BKUP_CPU_DEVLAB.BKD) are stored.	• The backup data may have been corrupted. Execute the data restoration function using another backup data.
4C1AH	CPU function dedicated error	• A folder with a value that matches the restoration target date folder setting value or number folder setting value does not exist in the SD memory card. • The restoration target data setting value is out of range. • The restoration target date folder setting value or number folder setting value is out of range.	• Check and correct the restoration target date folder setting value or number folder setting value, and execute the function again. • Check and correct the restoration target data setting value, and execute the function again.
4C1BH	CPU function dedicated error	• The data restoration function was executed to the CPU module whose status (such as programs, parameters, and file structure) differs from that of when the data backup function was executed.	• Match the CPU module status to the one at the time of backup, and execute the function again. • Set all data as the restoration target data, and execute the automatic data restoration function.
4C1CH	CPU function dedicated error	• An SD memory card is not inserted. • An SD memory card is disabled by SM606 (SD memory card forced disable instruction). • An SD memory card is write-protected.	• Insert or re-insert an SD memory card, and execute the function again. • Enable the SD memory card operation, and execute the function again. • Cancel the write protection, and execute the function again.
4C1EH	CPU function dedicated error	• When an SFC program start mode is set to "Resume start", the status of the SFC program (such as step status and transition conditions) has been changed during execution of the data backup function.	• Take measures so that the status of the SFC program does not change during execution of the data backup function, and execute the function again.
4C1FH	CPU function dedicated error	• The specified command cannot be executed because the CPU module data backup/restoration function is being executed.	• Execute the command again after the data backup/ restoration processing ends.
4C20H	CPU function dedicated error	• The data backup/restoration function was executed while the CPU module is in a state where this function cannot be executed.	• Deselect the "Access from External Device" checkbox in the label setting window or delete labels with this item selected, and initialize the memory which is written the global label assignment information. Then, execute the data backup function again. • Initialize the CPU built-in memory, and execute the data restoration function again.
4C21H	CPU function dedicated error	• The number of backup data stored in an SD memory card exceeds the upper limit value.	• Delete backup data in the SD memory card, and execute the function again. • Check and correct the settings of the upper limit value of the number of backup data, and execute the function again.
4C22H	CPU function dedicated error	• Bit5 of SD944 (Upper limit status for the number of the CPU module backup data) has been turned on while the set value of SD1353 (Upper limit value setting for the number of CPU module backup data) is out of the allowable range.	• Check and correct the set value of SD1353, and turn on bit5 of SD944.
4C23H	CPU function dedicated error	• Upper limit value of the number of the backup data cannot be changed because a CPU data folder has already been in an SD memory card.	• After deleting the CPU data folder in the SD memory card and turning off bit5 of SD944 (Upper limit status for the number of the CPU module backup data), turn on bit5 of SD944 again.

Error code	Error name	Error details and cause	Action
4C24H	CPU function dedicated error	• The data cannot be backed up because more backup data than the upper limit value of the number of backup data exists in the SD memory card.	• Delete the backup data exceeding the upper limit value, and execute the function again. • Check and correct the settings of the upper limit value of the number of backup data, and execute the function again.
4C40H	File transfer function (FTP client) error	• When files are specified by using wild card characters for the file transfer function instruction, the number of files matched exceeds the upper limit of the transferable number of files. • When files are specified by using wild card characters for the file transfer function instruction, no files are matched.	• Check and correct the wild card specification. • Check if the specified folder path exists.
4C41H	File transfer function (FTP client) error	• Timeout error of DNS client response	• Check the DNS server address. • Check if communications with the DNS server is valid by using the PING command. • Check if IP addresses of own station and the DNS server belong to the same class. (If the class differs, check the router setting.)
4C42H	File transfer function (FTP client) error	• Timeout error of FTP response	• Check if the FTP server name is registered in the DNS. • Delete the FTP server name, change the specification setting to the IP address, and check the operation. • Check if communications with FTP server is valid by using the PING command.
4C43H	File transfer function (FTP client) error	• The number of processing completed files for sending or acquiring FTP client file is mismatched with the total number of processing files.	• Execute the function again.
4C44H	File transfer function (FTP client) error	The file transfer function (FTP client) is executed while the following functions are being executed. • CPU module data backup/restoration function • iQ Sensor Solution data backup/restoration function	• Execute the File transfer function (FTP client) again after the mentioned function is completed.
C000H to CFBFH	📖 MELSEC iQ-R Ethernet User's Manual (Application)		
CFC0H	Cyclic transmission error (master station)	• Unable to execute cyclic transmission because multiple master stations exist in the same network address.	• Check the existence status of master station in network.
CFC1H	Cyclic transmission error (master station)	• Unable to execute cyclic transmission because the error occurred in cyclic transmission.	• Take measures to reduce noise. • If the same error is displayed again, please contact your local Mitsubishi representative.
CFC8H	Cyclic transmission error (master station)	• Unable to execute cyclic transmission because the slave station controlled by other master station exists.	• Check the existence status of master station in network. • Check the slave station where the error occurred.
CFC9H	Cyclic transmission error (master station)	• Unable to execute cyclic transmission because the slave station of the same IP address exists in the same network address.	• Check the existence status of slave station in network. • Check the slave station where the error occurred.
CFD0H	Master station error	• The port No. (61450) used in CC-Link IE Field Network Basic has already been used.	• Check the port No. used in Ethernet function.
CFD1H	Master station error	• Invalid value has been set in subnet mask.	• Check the parameter setting.
CFE0H	Cyclic transmission error (slave station)	• The cyclic transmission was executed for the slave station controlled by other master station.	• Check the existence status of master station in network. • Check the slave station where the error occurred.
CFE1H	Cyclic transmission error (slave station)	• The unusable number of occupied stations has been specified from master station.	• Check the number of occupied stations setting in master station parameter (Network Configuration Settings).
CFE8H	Cyclic transmission error (slave station)	• There is no response from slave station.	• Check the slave station disconnection detection setting in master station parameter (Network Configuration Settings). • Check the existence status of slave station in network. • Check the slave station which is disconnected. • Take measures to reduce noise.

Error code	Error name	Error details and cause	Action
CFE9H	Cyclic transmission error (slave station)	• The slave station of the same IP address has existed in the same network address.	• Check the slave station where the error occurred
CFF0H	Slave station error	• The error occurred in slave station.	• Check the slave station where the error occurred.

■Error codes related to the online module change function

The following table lists the codes of errors related to the online module change function.

The code of an error detected during online module change is stored in SD1618 (Online module change error code).

The code of an error when a disable request is executed during online module change is stored in SD1619 (Disable request error code during online module change).

Error code	Error name	Error details and cause	Action
4110H	Online module change error	The online module change function was executed while the CPU module was in a error state (stop error).	The module cannot be changed online. Power off the programmable controller, and replace the module.
4111H		In a multiple CPU system, the online module change function was executed even though other CPU modules have not started up.	Execute the function again after all the CPU modules in the system start up.
4202H		• An attempt was made to change two modules directly at the same time. • A module that is not ready to be removed was removed.	• Continue processing for the module being changed online. Two modules cannot be changed online at the same time. For the other module, power off the programmable controller, and replace the module. • Turn on SM1602 (Module removal request flag) and check that the value, 5, is stored in SD1617 (Online module change progress status). Then, continue processing from the step for mounting a module.
4205H		The online module change function was executed in a multiple CPU system where a CPU module that does not support this function is included.	The module cannot be changed online. Power off the programmable controller, and replace the module.
4206H		An attempt was made to change the inter-module synchronization target module online.	The module cannot be changed online. Power off the programmable controller, and replace the module.
4210H		The selected module is out of the valid range.	Check and correct the target base unit number and/or slot number, and turn on SM1600 (Module selection request flag) again.
4214H		The model of the newly-mounted module differs from that of the module before online change. Or, the newly-mounted module is not upward compatible (module mismatch).	Mount the same model or the upward compatible module, and execute the request again. • Mount a module of the same model, and perform the module recognition processing again.
4215H		• The module recognition processing is performed without the module being mounted. • The module selection processing is performed to the slot that is set as "Empty". • The module newly-mounted to the system is not mounted correctly.	• Mount a module, and perform the module recognition processing again. • Check and correct the target base unit number and/or slot number, and turn on SM1600 (Module selection request flag) again. • Mount the module correctly.
4216H		The module newly-mounted to the system online has failed. (The CPU module cannot access to the module.) • The module has failed. • The module is not mounted correctly.	Use another module, and execute the request again. Or, power off the system, and replace the module with the one that supports the online module change function. • Replace the module with the normal one. • Mount the module correctly.
4218H		• The online change target module (before change) does not support the online module change function. • The module newly-mounted to the system online (after change) does not support the online module change function. • The online module change function was executed to change the MELSEC-Q series module.	• If an error occurs at the stage of selecting a module, the module cannot be changed online. Power off the programmable controller, and replace the module. • If an error occurs at the stage of recognizing a module, replace the module with the one that supports the online module change function, and continue the processing. • The module cannot be changed online. Power off the programmable controller, and replace the module.
421AH		The online module change function was executed to change a module controlled by another CPU module.	Execute the function from the control CPU module.

Codes of errors detected by other than the self-diagnostic function (6F00H to 6FFFH)

The following table lists the codes of errors, other than those detected by the self-diagnostic function of the CPU module.

Error code	Error name	Error details and cause	Action
6F00H	Transient execution error	• The internal buffer area for transient processing is used to the maximum.	• Temporarily stop transient transmissions, or reduce the frequency. Then, perform transmissions again. Or, add the COM instruction to the programmable controller CPU and increase the frequency. If the same error occurs again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.
6F01H	Transient transmission timeout error	• In transient transmission, timeout has occurred without transmission completion.	• Review the network status. If the same error occurs again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.
6F02H	Module error	• An error has been detected in the redundant function module.	• Please consult your local Mitsubishi representative.
6F23H	Module error	• An error has been detected in the redundant function module.	• Please consult your local Mitsubishi representative.
6F24H	Module error	• An error has been detected in the redundant function module.	• Please consult your local Mitsubishi representative.
6F25H	Module communication test error	• The operating status of the CPU module is in a state other than STOP. (A test cannot be executed.)	• Change the operating status of the CPU module to STOP, and execute a module communication test.
6F27H	Module communication test being executed	• A module communication test execution request was received during the test. • A module communication test termination request was received during the test.	• Since the module communication test is being executed, execute the request again after the completion of the test.
6F28H	Module being changed online	• A module communication test execution request was received during online module change. • A module communication test termination request was received during online module change.	• Since a module is being changed online, execute the request again after the completion of the change.
6F29H	Module communication test error	• A module communication test execution request was received in the state where tracking cables are connected incorrectly.	• Connect one end of a tracking cable to the IN connector and the other end to the OUT connector of the test target redundant function module, and execute the request again.
6F30H	Transient execution error (redundant function module communication being stopped)	Data communications cannot be performed with the other system because of any of the following reasons: • The CPU module of the other system is powered off or in a reset state. • The tracking communication stops due to an error in the CPU module of the own or other system. • Tracking cables are disconnected or incorrectly connected, or failed. • The redundant function module of the own or other system is being changed online, a module communication test is being performed for the module, or a hardware failure has occurred in the module.	• If a WDT error has occurred in the CPU module of the own or other system, eliminate the error cause and then execute the operation again. • Check that there is no error or failure in the CPU module, tracking cables, redundant function modules, and perform a retry. Or, perform a retry after the online module change processing or the module communication test ends. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.
6F40H	Transient execution error	The number of transient transmission requests exceeded the upper limit of simultaneously processable requests.	• Temporarily stop transient transmissions, or reduce the frequency. Then, perform transmissions again.
6F44H	Transient frame error	• An invalid transient frame was received.	• Check and correct the request data (send data) on the request source side, and perform transmissions again.
6F4AH	Transient frame error	• An invalid transient frame was received.	• Check and correct the request data (send data) on the request source side, and perform transmissions again.
6F4BH	Transient frame error	• An invalid transient frame was received.	• Check and correct the request data (send data) on the request source side, and perform transmissions again.
6F4CH	Transient frame error	• An invalid transient frame was received.	• Check and correct the request data (send data) on the request source side, and perform transmissions again.
6F4DH	Transient frame error	• An invalid transient frame was received.	• Check and correct the request data (send data) on the request source side, and perform transmissions again.
6F4EH	Transient frame error	• An invalid transient frame was received.	• Check and correct the request data (send data) on the request source side, and perform transmissions again.
6F60H	Module error	• An error has been detected in the redundant function module.	• Please consult your local Mitsubishi representative.

Appendix 2 Event List

The CPU module collects information, such as errors detected by the module; operations performed for the module; and network errors, from modules, and stores the collected data into the data memory or an SD memory card. (☞ Page 236 Event History Function)
When an event occurs, its event code and details can be read by using an engineering tool.

Point

For details on events occurred in each module, refer to the manual for the module used.

How to read the event list

The event list contains the following information.

Item	Description
Event code	ID number assigned to an event
Event type	Type of an event
Event category	Category of an event
Detected event	Description of a detected event
Detailed information 1 to 3	Details of a detected event

Detailed information

The following table lists the details of information displayed in the detailed information 1 to 3.

Detailed information	Item	Description
Detailed information 1	Operation source information	Information on the operation source • Connection port (such as Ethernet and USB) • I/O number • CPU number (a number assigned to CPU modules in a multiple CPU system) • Network number • Station number • IP address • User name (when the user authentication function is enabled)
	Event history file information	Information on the event history file
	Daylight saving time status	Information on the daylight saving time status (start/end)
	CPU module data backup/restoration information	Information on the CPU module data backup/restoration function (operation, result, error, data specification, initialization, latest data, special relay, special register, operation mode)
	CPU module data backup setting information	Information on the CPU module backup setting (backup function setting, error codes)
	Data backup/restoration information of iQ Sensor Solution	Information on the data backup/restoration function of iQ Sensor Solution (operation, target device, execution unit, target module, folder number setting method, total number of executions, number of successful executions, number of executions completed with an error, folder number)
	Information on the execution status of the data backup/restoration function of iQ Sensor Solution	Information on the execution status of the data backup/restoration function of iQ Sensor Solution (result, error category, error code)
	Information on the right to use the data backup/restoration function of iQ Sensor Solution	Information on the right to use the data backup/restoration function of iQ Sensor Solution (right-to-use number, operation)
	System switching information	Information on the system switching cause, the cause of system switching failure, and the transition state of the systems (control system and standby system)
	Start-up information	Information on the start-up • Normal start-up • Start-up with SD memory card diagnostics
	Information on the initial processing cancellation	Information on the initial processing cancellation • SD memory card diagnostics
	Firmware update information	Information on the firmware update

Detailed information	Item	Description
Detailed information 2	Communication speed and communication mode	Information on the communication speed and the communication mode
	Communication status	Information on the communication status
	Security key operation information	Information on the corresponding security key
	Remote password information	Information on the corresponding remote password
	File password information	Information on the corresponding file password
	Blocked IP address information	Information on the blocked IP address
	Drive/file information	Information on the corresponding drive name and file name
	Copy source drive/file information	Information on the corresponding drive name and file name
	Operation target information	Information on the operation target: I/O number
	Clock information (before change)	Clock information before change
	Remote operation type information	Information on the remote operation type
	User authentication information	Information on the user authentication function status (enabled/disabled)
	Login/logout information	Login/logout information of the user authentication function
	File access control information	Information on file access control: access level
	Device/label information	Information on the corresponding device and label
	System configuration information	Information on the system configuration
	Target folder information of the CPU module data backup/restoration function	Information on the data backup/restoration target folder (folder specification, date, number)
	Target device information of the data backup/restoration function of iQ Sensor Solution	Information on the data backup/restoration target device (station number, station sub-ID number, ID number, IP address)
	System A/B setting information	Information on the system A/B setting
	System switching information	Information on the system switching cause, the cause of system switching failure, and the transition state of the systems (control system and standby system)
	Control system start-up cause information	Information on the start-up cause of the control system
	Error description (the other system)	Description of the error when an error was detected in the other system
	Cause of tracking communication stop	Cause of tracking communication stop
	Target station information	Error description (error codes) of target station
Detailed information 3	Clock information (after change)	Clock information after change
	Copy destination drive/file information	Information on the corresponding drive name and file name

Event list

The following table lists events related to the CPU module.

Event code	Event type	Event category	Detected event	Description	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
00100	System	Info	Link-up	■CPU module The CPU module has entered into the link-up state as a result of an operation such as connecting a network cable between the CPU module and an external device. ■Redundant function module The redundant function module has entered into the communication enable state. *This event code does not indicate the start of tracking communication.	Operation source information	Communication speed and communication mode	—
00110			Communication start/end (TCP)	Data communications with an external device through TCP connection started. Or, data communications with an external device through TCP connection ended.		Communication status	
00120			FTP connection start/end	FTP connection with an external device started. Or, FTP connection with an external device ended.			
00130			Receive frame error	A receive frame error was detected.		—	
00140			SNTP server time synchronization failure	Time setting by the time synchronization function failed because there was no response from the SNTP server.			
00400			Power-on/reset	The CPU module has been power-on or reset.	Start-up information*2		
00401			Initial processing cancelled	The CPU module has been powered off or reset during initial processing.	Information on the initial processing cancellation		
00410			Boot operation	Boot operation was performed.	—		
00411			SD memory card diagnostics completed	SD memory card diagnostics have completed.	—		
00420			Event history file generation	An event history file was generated.	Event history file information		
00430			SFC program continue start not possible	A continue start of an SFC program could not be performed, and an initial start was performed.	—		
00440			Safety operation mode undefined	The safety operation mode was undefined, and the system started in TEST MODE.	—		
00450			Daylight saving time start/end	Daylight saving time started. Or, daylight saving time ended.	Daylight saving time status		
00460			Label initialization	After the data is rebuild (reassigned), labels were initialized when the system was powered off and on or the operating status of the CPU module was changed from STOP to RUN. (Initial values were set or values were cleared to zero.)	—		
00700			Tracking communication start	Tracking communication started.			
			Auto system A/B setting	The system A/B setting was automatically set by the redundant system. (The setting was overwritten.)	System A/B setting information		

Event code	Event type	Event category	Detected event	Description	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
00800	System	Warning	Link-down	■CPU module The CPU module has entered into the link-down state as a result of an operation such as disconnecting a network cable between the CPU module and an external device. ■Redundant function module The redundant function module has entered into the communication disable state as a result of any of the following: • Any tracking cable was disconnected. • The other system was powered off. • An error occurred in the cable, connector, or module.	Operation source information	Communication speed and communication mode	—
00904			Socket communication send error	Sending a message over socket communication failed.		—	
00906			Alive check error	The alive status of an external device could not be checked within the period specified by the response monitoring timer.			
00907			Divided message receive timeout error	• All the data could not be received within the period specified by the response monitoring timer. • Data of the total data length could not be received. • The remaining part of the message divided into the TCP/IP level could not be received within the period specified by the response monitoring timer.			
00908			IP composition timeout error	An IP composition timeout error occurred. (The specified period of time expired without receiving all of the divided data.)			
00909			TCP-specified port number error	A port number being used for opened connection was set (for TCP/IP).			
0090A			UDP-specified port number error	A port number being used for opened connection was set (for UDP/IP).			
00A00			Error detection in the other system	An error was detected in the other system.	—	Error description (the other system)	
00C02			Abnormal response from/to the other system	• An abnormal response was sent to the other system. • An abnormal response was sent from the other system.		Target station information	
00C27			Tracking communication stop	Tracking communication has stopped as a result of any of the following: • The standby system CPU module has been powered off or reset. • Hardware failure of the CPU module has occurred. • An error has occurred in the redundant function module. • A WDT error has occurred. • Tracking cables has been pulled out or disconnected.		Cause of tracking communication stop	
00C28			Retry	• A tracking communication retry was performed due to the change in communication route caused by loopback. • A tracking communication retry was performed due to a line status error, such as faulty cable and incorrect connector connection.		—	
00F00			System switching (by the system)	The systems were switched due to the cause on the redundant system side.	System switching information		

Event code	Event type	Event category	Detected event	Description	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
00F01	System	Warning	Auto memory copy (control system)	The control system automatically copied its memory to the standby system.	—	—	—
00F02			Auto memory copy (standby system)	The memory of the control system was automatically copied to the standby system.			
1000 and after		Error	When a self-diagnostic error occurs, the error is stored as an event.				
10100	Security	Info	Security key registration/deletion	A security key was registered or deleted.	Operation source information	Security key operation information	—
10200			Remote password lock	The remote password was set.		Remote password information	
10201			Remote password unlock	The remote password unlock processing was successfully completed.			
10202			Remote password unlock failed	The remote password unlock processing failed.			
10300			Access from an IP address blocked by the IP filter setting	An access from an IP address blocked by the IP filer setting was accepted.		Blocked IP address information	
10400			File password registration/change/deletion	A file password was successfully registered, changed, or deleted.	Operation source information	File password information	
10401			File password registration/change/deletion failed	Registration, change, or deletion of a file password failed.			
10402			File password unlock	A file password was successfully unlocked.			
10403			File password unlock failed	Unlock of a file password failed.			
10500			Forced invalidation setting	Forced invalidation was set.	—	—	
10501			Forced invalidation cancel	Forced invalidation was canceled.			
10600			User authentication enabled/disabled	The user authentication function was enabled or disabled.	Operation source information	User authentication information	
10601			Login succeeded	The login processing of the user authentication function completed successfully.		Login/logout information	
10602			Login failed	The login processing of the user authentication function failed.			
10605			Auto logout	The system automatically logged out the user by executing the user authentication function.			
10606			User management information update	The user management information was updated. (The information was added, changed, or deleted.)		—	
10607			User management information registration (Administrators)	An operator (Administrators) was registered to the user management information.			
10700			Access from unauthorized access level	A file was accessed from an access level not allowed by the user authentication function (file access control).		File access control information	
18200		Warning	Initialization of all programmable controller information	All information in the programmable controller was initialized.		—	

Event code	Event type	Event category	Detected event	Description	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
20100	Operation	Info	Error clear	The error was cleared.	Operation source information	Operation target information	—
20200			Event history clear	The event history was cleared.		—	
20300			SD memory card enabled	The SD memory card was enabled.	—		
20301			SD memory card forcibly disabled	The SD memory card forced disable function was executed and the SD memory card became ready to be removed.			
20400			Firmware update succeeded (CPU)	The firmware of the CPU module was successfully updated.	Information on CPU module firmware update		
20401			Firmware update failed (CPU)	The firmware update of the CPU module failed.			
20500			CPU module data backup succeeded	Data in the CPU module were successfully backed up.	CPU module data backup/ restoration information	Target folder information of the CPU module data backup/ restoration function	
20501			CPU module data backup failed	Backup of data in the CPU module failed.			
20502			CPU module data restoration succeeded	Data were successfully restored to the CPU module.			
20503			CPU module data restoration failed	Restoration of data to the CPU module failed.			
20510			CPU module data backup setting disabled	Backup function cannot be set	CPU module data backup setting information	—	
20600			iQ Sensor Solution data backup succeeded	Data in the target device supporting iQ Sensor Solution were successfully backed up.	Information on the execution status of the data backup/ restoration function of iQ Sensor Solution	Target device information of the data backup/ restoration function of iQ Sensor Solution	
20601			iQ Sensor Solution data backup failed	Backup of data in the target device supporting iQ Sensor Solution failed.			
20602			iQ Sensor Solution data restoration succeeded	Data were successfully restored to the target device supporting iQ Sensor Solution.			
20603			iQ Sensor Solution data restoration failed	Restoration of data to the target device supporting iQ Sensor Solution failed.			
20610			iQ Sensor Solution data backup start/ cancel/end	The data backup processing of the target device supporting iQ Sensor Solution started, canceled, or ended.	Data backup/ restoration information of iQ Sensor Solution	—	
20611			iQ Sensor Solution data restoration start/cancel/end	The data restoration processing of the target device supporting iQ Sensor Solution started, canceled, or ended.			
20620			Right-to-use acquisition/release	The right to use the data backup/ restoration function of iQ Sensor Solution was acquired or released.	Information on the right to use the data backup/ restoration function of iQ Sensor Solution		

Event code	Event type	Event category	Detected event	Description	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
24000	Operation	Info	Clock setting	The clock data was set.	Operation source information	Clock information (before change)	Clock information (after change)
24001			Remote operation request accepted	A remote request (RUN, STOP, or PAUSE) was accepted.		Remote operation type information	—
24100			Operating status change (RUN)	The operating status of the CPU module was changed to RUN.	—	—	
24101			Operating status change (STOP)	The operating status of the CPU module was changed to STOP.			
24102			Operating status change (PAUSE)	The operating status of the CPU module was changed to PAUSE.			
24200			Creation of new folders, writes to files/folders[*1]	• A new folder was created. • A new file was created or data was written to a file.	Operation source information	Drive/file information	
24201			File copy[*1]	A file was copied.		Copy source drive/file information	Copy destination drive/file information
24202			Folder/file rename[*1]	A folder name or file name was changed.			
24300			Safety operation mode change (SAFETY MODE (wait-for-restart))	The safety operation mode was changed to SAFETY MODE (wait-for-restart).		—	—
			Module communication test	A module communication test was executed.			
24301			Safety operation mode change (TEST MODE)	The safety operation mode was changed to TEST MODE.			
25000			Online module change	The online module change processing completed.		System configuration information	
25200			System A/B setting write	The system A/B setting was written to the CPU modules.	Operation source information	System A/B setting information	—
26000			Redundant operation mode change (backup mode)	The operation mode in a redundant system was changed to backup mode.		—	
26001			Redundant operation mode change (separate mode)	The operation mode in a redundant system was changed to separate mode.			

Event code	Event type	Event category	Detected event	Description	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
2A200	Operation	Warning	Memory initialization*1	The memory was initialized.	Operation source information	Drive/file information	—
2A201			Device/label zero clear	Values in a device or label were cleared to zero.		Target device/ label	
2A202			Folder/file deletion*1	A folder or file was deleted.		Drive/file information	
2B000			System switching (by a user)	The systems were switched due to the cause on the user side.		System switching information	
2B001			Memory copy execution by the engineering tool (control system)	The memory copy function was executed by the engineering tool.		—	
2B002			Memory copy execution by the special relay and special register (control system)	The memory copy function was executed by the special relay and special register.	—		
2B003			Memory copy execution by a user (standby system)	The memory copy function was executed by a user.			
2B004			Control system forced start-up	One system was forcibly started as a control system while waiting for a start-up of the other system.	Operation source information	Control system start-up cause information	

*1 As for file-related events such as writing and deleting a file, the following files are targeted:
· Program file
· FB program file
· Parameter file
· Data logging setting file (common setting file, individual setting file)
· Memory dump setting file
When the Safety CPU is used, the following files are targeted as well:
· Safety program file
· Safety FB program file
· Safety parameter file

*2 The engineering tool displays "Start-up information" in detailed information 1 only when the Process CPU with firmware version "06" or later is used.

A

Appendix 3 Troubleshooting by Symptom

If any function of the CPU module does not operate as designed, perform troubleshooting by checking the following items. If the ERROR LED or USER LED is on or flashing, eliminate the error cause using the engineering tool.

When the POWER LED of the power supply module turns off

For troubleshooting, refer to the following:

MELSEC iQ-R Module Configuration Manual

When the READY LED of the CPU module turns off

Check the following:

Check item	Action
Check if the CPU module is mounted on the main base unit properly.	Remove the CPU module from the main base unit, and mount it back on the main base unit.
Check if the READY LED of another module is on.	A major error has occurred in the CPU module. Replace the CPU module.
Check if the READY LED turns on when the power supply module is replaced and the power is restored to the system. (Check the LED status after the power supply module on the extension base unit is also replaced.)	An error has occurred in the power supply module(s) before the replacement. Replace the power supply module.
Check if the READY LED does not turn on even after the power supply module is replaced and the power is restored to the system. (Check the LED status after the power supply module on the extension base unit is also replaced.)	An error has occurred in a module other than the power supply module. Repeatedly supply power to the system, returning the modules back to the system one by one. The last module mounted immediately before the READY LED turned off has been failed. Replace the module.

If the READY LED of the CPU module does not turn on even after the items above are checked and the actions are taken, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.

When an error has occurred in a redundant function module

When the RUN LED turns off

When the RUN LED turns off after the redundant function module is powered on, check the following.

Check item	Action
Check if the redundant function module has been properly mounted.	If not, properly mount the module on the base unit.

If the RUN LED of the redundant function module still does not turn on even after the above action is taken, the possible cause is a hardware failure of the module. Please consult your local Mitsubishi representative.

When the ERR LED turns on or flashes

Check the following:

Check item	Action
Check if an error has occurred by performing the module diagnostics.	Take actions proposed by the module diagnostics.

If the ERR LED still turns on or flashes even after the above action is taken, perform a module communication test and check the module for a hardware failure. (☞ Page 441 Redundant Function Module Communication Test)

When the L ERR LED turns on

Check the following:

Check item	Action
Check if the tracking cables used are proper.	• Check if the tracking cables satisfy the standards. (📖 MELSEC iQ-R CPU Module User's Manual (Startup)) • Check that the length of each tracking cable is within the range of the specifications. (📖 MELSEC iQ-R CPU Module User's Manual (Startup)) • Check if no disconnection is found on each tracking cable.
Check if each tracking cable has been properly connected.	Check the connection status of each tracking cable by performing the module diagnostics. If a tracking cable has not been properly connected, connect it again.

If the L ERR LED still turns on even after the above actions are taken, perform a module communication test and check the redundant function module and tracking cables for failures. (☞ Page 441 Redundant Function Module Communication Test)

When the specific extension base unit cannot be recognized

For troubleshooting, refer to the following:

📖 MELSEC iQ-R Module Configuration Manual

When the specific Q series extension base unit cannot be recognized

For troubleshooting, refer to the following:

📖 MELSEC iQ-R Module Configuration Manual

When data cannot be written to the programmable controller

Check the following:

Check item	Action
Check if password is not registered.	Authenticate a password using the engineering tool.
Check if the SD memory card is write-protected while the SD memory card is being targeted for writing.	Clear the write-protect for the SD memory card.
Check if the SD memory card is formatted while the SD memory card is being targeted for writing.	Carry out formatting for the SD memory card.
Check if the data to be written exceeds the capacity of CPU built-in memory or SD memory card.	Check the CPU built-in memory or SD memory card for their free space.
Check if a password has been set for the programmable controller.	Write user information to the programmable controller from the engineering tool.
Check if the user has logged on to the programmable controller.	Log on to the programmable controller from the engineering tool.

If data cannot be written to the programmable controller even after the items above are checked and the actions are taken, the possible cause is a hardware failure. Please consult your local Mitsubishi representative.

When data cannot be read from the programmable controller

Check the following:

Check item	Action
Check if password is not registered.	Authenticate a password using the engineering tool.
Check if the target memory to be read is correctly specified.	Check the read target memory (CPU built-in memory, SD memory card, or intelligent function module).
Check if a password has been set for the programmable controller.	Write user information to the programmable controller from the engineering tool.
Check if the user has logged on to the programmable controller.	Log on to the programmable controller from the engineering tool.

If data cannot be read from the programmable controller even after the items above are checked and the actions are taken, the possible cause is a hardware failure. Please consult your local Mitsubishi representative.

When the operating status of the CPU module cannot be changed

Check the following:

Check item	Action
Check if an stop error has been occurred.	Check the error cause using an engineering tool and eliminate the error cause.
Check if the online change processing is left suspended.	Execute the online change again.

If the CPU module operating status cannot be changed even after the items above are checked and the actions are taken, the possible cause is a hardware failure. Please consult your local Mitsubishi representative.

When the Ethernet function cannot be used

For troubleshooting, refer to the following:

MELSEC iQ-R Ethernet User's Manual (Application)

When the CC-Link IE Field Network Basic function cannot be used

For troubleshooting, refer to the following:

CC-Link IE Field Network Basic Reference Manual

When the CPU module database access function cannot be used

When the data source name cannot be set

Check the following:

Check item	Action
Check if use prohibited characters are used in "Data Source Name".	Set "Data Source Name" without using prohibited characters.
Check if the number of use prohibited characters in "Data Source Name" exceeds the range of use.	Set "Data Source Name" with the number of characters within the range of use.

When the connection test fails

When the connection test fails at the data source setting, check the following:

Check item	Action
Check if the CPU module supports the CPU module database access function.	Check that the CPU module supports the CPU module database access function. (☞ Page 779 Added and Enhanced Functions).
Check if the IP address of the CPU module set in "Server" is correct.	Set the correct IP address of the CPU module in "Server" (☞ Page 131 Data Source Name Setup).
Check if 61461 is set in "Port".	Set 61461 in "Port" (☞ Page 131 Data Source Name Setup).
Check if the absolute path of the database of the CPU module set in "Database" is correct.	Set the correct absolute path of the database in "Database" (☞ Page 131 Data Source Name Setup).

When the ODBC server of the CPU module cannot be connected

When the ODBC server of the CPU module cannot be connected from the personal computer, check the following:

Check item	Action
Check if "invalid login/password" is displayed in the error message on the personal computer.	Check that the login name and password that are set in the module parameter of the engineering tool match those that are input to connect to the server. (☞ Page 128 Built-in database access setting)
Check if "cannot open connection to server; open init timeout" is displayed in the error message on the personal computer.	Check that the correct data source is selected to connect to the database of the CPU module.
Check if "Communication link failure" is displayed in the error message on the personal computer.	The maximum number of concurrently connectable databases (number of connections) is used (☞ Page 118 Database specifications) End unused applications connected with the database of the CPU module and connect with the database again.

When the connection is disconnected

When the connection of the personal computer and the CPU module to the ODBC server is disconnected, check the following:

Check item	Action
Check if the built-in database access timer setting in the module parameter of the engineering tool is not too short.	Set the longer time for the built-in database access timer setting. (☞ Page 128 Built-in database access setting)

Appendix 4 List of Special Relay Areas

The following table lists items in the list.

Item	Description
No.	Special relay number
Name	Special relay name
Data stored	Data stored in the special relay and its meaning
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (program, engineering tool, GOT, or other testing operations from external device) • U/S: User and system <Set timing> • Every END: Data is set every time END processing is performed. • Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN). • Status change: Data is set when the status is changed. • Error: Data is set when an error occurs. • Instruction execution: Data is set when an instruction is executed. • Request: Data is set when requested by a user (using the special relay). • Writing: Data is set when a user performs a writing operation. • During END: Data is set when END processing is performed. • Power-on to RUN or STOP to RUN: Data is set when the operating status changes from power-on to RUN or from STOP to RUN. • System switching: Data is set when two systems are switched (between the control system and the standby system)

Point

Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

Diagnostic information

The following is the list of special rely areas relating to the diagnostic information.

No.	Name	Data stored	Details	Set by (setting timing)
SM0	Latest self-diagnostic error (including annunciator ON)	Off: No error On: Error	• This relay turns on when the self-diagnostics returns an error (including the case when an error is detected by turning the annunciator ON). • The on state is maintained even after the error has been later cleared.	S (Error)
SM1	Latest self-diagnostic error (not including annunciator ON)	Off: No error On: Error	• This relay turns on when the self-diagnostics returns an error (not including the case when an error is detected by turning the annunciator ON). • The on state is maintained even after the error has been later cleared. • This relay does not turn on when errors that are notified by the PALERT instruction or the PABORT instruction.	S (Error)
SM50	Error reset	Off→On:Error reset request On→Off:Error reset complete	• This relay clears the error state when the mode transfers from off to on. • This relay switches from on to off when the error reset has been completed.	U/S (Status change)
SM51	Battery low latch	Off: Normal On: Battery low	• This relay switches to on when the battery voltage of the CPU module drops below the specified value. • The On state is maintained even after the battery voltage has been later recovered to the normal value. • This relay synchronizes with BAT LED.	S (Error)
SM52	Battery low	Off: Normal On: Battery low	This relay has the same function as SM51 except for switching to off after the battery voltage has been recovered to a normal value.	S (Error)
SM53	AC/DC DOWN	Off: No AC/DC DOWN detection On: AC/DC DOWN is detected	• This relay switches to on when a momentary power failure within 20ms is detected while the AC power supply module is in use. This relay can be reset when power is turned off and on. • This relay switches to on when a momentary power failure within 10ms is detected while the DC power supply module is in use. This relay can be reset when power is turned off and on.	S (Error)
SM56	Instruction execution fault	Off: Normal On: Instruction execution fault state	• This relay switches to on when an error which can be classified as instruction execution fault is returned. • The On state is maintained even after the error has been later cleared.	S (Error)
SM60	Fuse Blown	Off: Normal On: Fuse blown is detected	• This relay is on when at least one output module is in fuse blown state and the On state is maintained even after later recovering to the normal state. • The fuse blown state check is also done for output modules on the remote I/O station.	S (Error)
SM61	I/O module verification error	Off: Normal On: Error	• This relay switches to on when the state of the I/O module is different from one registered during power-on, and the On state is maintained even after later recovering to the normal state. • I/O module verification is also done for modules on the remote I/O station.	S (Error)
SM62	Annunciator	Off: Not detected On: Detected	• This relay switches to on when at least one annunciator is turned on. • This relay returns to off when all the annunciators are turned off.	S (Instruction execution)

No.	Name	Data stored	Details	Set by (setting timing)
SM80	Detailed information 1: Flag in use	Off: Not used On: In use	This relay switches to on if the detailed information n exists when SM0 switched to on.	S (Status change)
SM112	Detailed information 2: Flag in use			
SM150	Power-off/power supply voltage drop detection	Off: Power-on/normal power supply voltage On: Power-off/voltage drop detected/power supply module not mounted	• This relay turns on when one or more of the power supply modules whose power has been shut off or power supply voltage has dropped (not including a momentary power failure), or one or more of empty slots for the power supply module are detected on the redundant power supply base unit. • This relay turns on if causes to turn on any bits of SD150 have been occurred. • This relay turns off if causes to turn on any bits of SD150 have been removed. • In a multiple CPU system, the flags are stored only to the CPU No.1.	S (Status change)
SM151	Power supply module failure detection	Off: Not detected/power-off/ no power supply module On: Detected	• This relay turns on when one or more power supply module failures have been detected on the redundant power supply base unit. • This relay turns on if causes to turn on any bits of SD151 have been occurred. • This relay turns off if causes to turn on any bits of SD151 have been removed. • In a multiple CPU system, the flags are stored only to the CPU No.1.	S (Status change)
SM152	Momentary power failure detection (power supply module 1)	Off: Not detected On: Detected	• This relay turns on when a momentary power failure of the input power supply to the power supply 1 or 2 is detected one or more times. After turning on, this relay remains on even if the power supply recovers from the momentary power failure. • This relay turns off the flags (SM152 and SM153) of the power supply 1 and 2 when the CPU module starts up. • When one of the two power supply modules is powered off, this relay turns off the corresponding flag to one powered off. • In a multiple CPU system, the flags are stored only to the CPU No.1.	S (Status change)
SM153	Momentary power failure detection (power supply module 2)			S (Status change)
SM154	Invalid power supply module	Off: Valid/power-off/no power supply module On: Invalid	• This relay turns on when one or more of invalid power supply modules have been detected on the redundant power supply base unit. • This relay turns on if a factor to turn on any bits of SD154 have been occurred. • This relay turns off if all the factors to turn on any bits of SD154 have been removed. • In a multiple CPU system, the flags are stored only to the CPU No.1.	S (Status change)

System information

The following is the list of special relay areas relating to the system information.

No.	Name	Data stored	Details	Set by (setting timing)
SM203	STOP contact	Off: Other than STOP state On: STOP state	This relay is on in STOP state.	S (Status change)
SM204	PAUSE contact	Off: Other than PAUSE state On: PAUSE state	This relay is on in PAUSE state. Note that this relay is on during the END processing of the scan which the specified PAUSE contact turns on if PAUSE state is generated at the PAUSE contact.	S (Status change)
SM210	Clock data set request	Off→On:setting request is detected On→Off:setting is completed	• Clock data stored in SD210 to SD216 is written into the CPU module when this relay is switched from off to on. • This relay switches from on to off when writing of clock data stored in SD210 to SD216 into the clock element is completed.	U/S (Status change)
SM211	Clock data set error	Off: No error On: Error	This relay switches to on when an error is generated in values from SD210 to SD216, and to off when no error is generated.	S (Request)
SM213	Clock data read request	Off: Non-processing On: Reading request	Clock data is loaded into SD210 to SD216 when this relay is in On state.	U
SM217*1	Daylight saving time status flag	Off: Not during daylight saving time On: During daylight saving time	Turns on if during daylight saving time with daylight saving time function. Turns off if not during daylight saving time.	S (Status change)
SM220	CPU No.1 preparation completed	Off: Not completed On: Completed	• This relay switches to on at the time when access from the CPU module on other CPUs to the CPU module for CPU No. n is enabled during power-on or resetting. • This relay is used as an interlock to access the CPU module for the CPU No. n when the multiple CPU synchronization setting is configured to asynchronous mode.	S (Status change)
SM221	CPU No.2 preparation completed			
SM222	CPU No.3 preparation completed			
SM223	CPU No.4 preparation completed			
SM230	No.1 CPU error flag	Off: CPU No.n normal On: CPU No.n stop error state	• This relay is off when the CPU module for the CPU No.n is normal (including a continuation error period). • This relay is on when the CPU module for the CPU No.n is in stop error state.	S (Status change)
SM231	No.2 CPU error flag			
SM232	No.3 CPU error flag			
SM233	No.4 CPU error flag			

No.	Name	Data stored	Details	Set by (setting timing)
SM240	No.1 CPU reset flag	Off: CPU No.n not being reset On: CPU No.n in reset mode	• This relay switches to off when the CPU module of the CPU No.1 is not being reset. • This relay is on while the CPU module of the CPU No.1 is being reset (including the case when the CPU module is removed from the base unit). Other CPUs also enter into reset mode.	S (Status change)
SM241	No.2 CPU reset flag		• This relay switches to off when the CPU module of the CPU No.2 is not being reset. • This relay is on while the CPU module of the CPU No.2 is being reset (including the case when the CPU module is removed from the base unit). Errors occur in the other CPU modules.	S (Status change)
SM242	No.3 CPU reset flag		• This relay switches to off when the CPU module of the CPU No.3 is not being reset. • This relay is on while the CPU module of the CPU No.3 is being reset (including the case when the CPU module is removed from the base unit). Errors occur in the other CPU modules.	S (Status change)
SM243	No.4 CPU reset flag		• This relay switches to off when the CPU module of the CPU No.4 is not being reset. • This relay is on while the CPU module of the CPU No.4 is being reset (including the case when the CPU module is removed from the base unit). Errors occur in the other CPU modules.	S (Status change)

*1 Programmable controller CPU only

SFC information

The following is a list of special relay areas relating to SFC information.

No.	Name	Data stored	Details	Set by (setting timing)
SM320	Presence/absence of SFC program	Off: No SFC program On: SFC program	• This relay switches to on if an SFC program has been registered, and switches to off if it has not.	S (Initial)
SM321	Start/stop SFC program	Off: SFC program not executed (stop) On: SFC program executed (start)	• The same value as SM320 is set for the default value. (This relay automatically turns on if there is an SFC program.) • SFC program execution is stopped if this relay switches from on to off, and execution is restarted when it switches from off to on. • If this relay switches off before SFC program processing, execution of the SFC program is not started.	S (Initial)/U
SM322	SFC program start status	Off: Initial start On: Resumption	• If the SFC program start mode in the parameters is off for initial start, off is set for the default, and if resumption, on is set.	S (Initial)/U
SM323	Presence/absence of continuous transition for entire block	Off: No continuous transition On: Continuous transition	• Sets whether there is a continuous transition for blocks for which no SFC information device continuous transition bit has been set. • No continuous transition when off. Continuous transition when on. • There is no effect on operation for blocks for which the continuous transition bit has been set.	U
SM324	Continuous transition prevention flag	Off: When transition executed On: When there is no transition	• This relay switches to off during operation in modes with continuous transition, or during continuous transition, and switches to on when it is not a continuous transition. • The relay is always on during operation in modes with no continuous transition.	S (Status change)
SM325	Output mode at block stop	Off: Off On: Hold	• Selects whether to retain coil output for active steps during block stoppages. • The default value when the output mode for block stoppages in the parameters is coil output off is off, and on when coil output is retained. • All coil outputs are turned off when off. Coil outputs are retained when on.	S (Initial)
SM326	SFC device/label clear mode	Off: Device/label clear On: Device/label retain	• Select the device status (all devices and labels (including latch labels) excluding step relay (S)) when the CPU module status changes from STOP → program write → RUN. • This relay is valid only when an SFC program exists after program writing. • This relay is valid not only when an SFC program is written, but also when the program file and the parameter file are written. • When the Process CPU is used, only labels are cleared at the program writing to the programmable controller after data are rebuilt even though this relay is on.	U
SM327	Output mode at execution of the end step	Off: Hold step output off On: Hold step output retained	• When this relay switches to off, coil output is turned off for steps (SC, SE, ST) for which transition is established and that are on hold reach the END step. • When this relay switches to on, coil output is retained (step becomes inactive) for steps (SC, SE, ST) for which transition is established and that are on hold reach the END step. However, coils are turned off following forced termination.	U

No.	Name	Data stored	Details	Set by (setting timing)
SM328	Clear processing mode when the sequence reaches the end step	Off: Clear processing performed On: Clear processing not performed	• Selects whether to perform clear processing if an active step other than those retained in the block exists when the END step is reached. When this relay switches to off, all active steps are forcibly terminated, and the block is terminated. When this relay switches to on, block execution is continued as is. When the END step is reached, if no active steps other than those retained exist, all retained steps are terminated, and the block is terminated.	U
SM329	Online change (inactive SFC block) status flag	Off: Not being executed On: Being executed	Turns on while the online change (inactive SFC block) is being executed.	S (Status change)

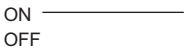
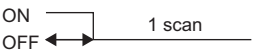
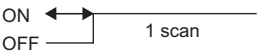
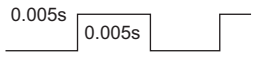
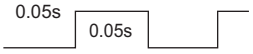
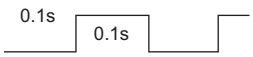
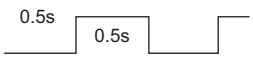
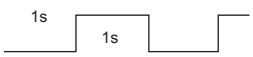
Point

For details on SFC language, refer to the following.

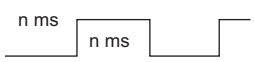
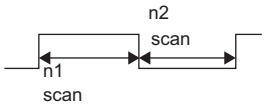
MELSEC iQ-R Programming Manual (Program Design)

System clock

The following is the list of special relay areas relating to the system clock.

No.	Name	Data stored	Details	Set by (setting timing)
SM400	Always On		Always On	Power-on to RUN or STOP to RUN
SM401	Always Off		Always Off	S (Status change)
SM402	After RUN, ON for 1 scan only		• This relay is on during only one scan after RUN mode starts. • This relay is enabled only for the scan execution type program.	S (Status change)
SM403	After RUN, OFF for 1 scan only		• This relay is off during only one scan after RUN mode starts. • This relay is enabled only for the scan execution type program.	S (Status change)
SM409	0.01 second clock		• This relay repeats on/off at 5ms intervals. • The initial state when CPU module is powered on or reset is off. • This relay switches to on or off after a specified time is elaspsed even when the program execution is in progress.	S (Status change)
SM410	0.1 second clock		• This relay repeats on/off at certain intervals. • The initial state when CPU module is powered on or reset is off. • This relay switches to on or off after a specified time is elaspsed even when the program execution is in progress.	S (Status change)
SM411	0.2 second clock		• This relay repeats on/off at certain intervals. • The initial state when CPU module is powered on or reset is off. • This relay switches to on or off after a specified time is elaspsed even when the program execution is in progress.	S (Status change)
SM412	1 second clock		• This relay repeats on/off at certain intervals. • The initial state when CPU module is powered on or reset is off. • This relay switches to on or off after a specified time is elaspsed even when the program execution is in progress.	S (Status change)
SM413	2 second clock		• This relay repeats on/off at certain intervals. • The initial state when CPU module is powered on or reset is off. • This relay switches to on or off after a specified time is elaspsed even when the program execution is in progress.	S (Status change)
SM414	2n second clock		• This relay repeats on/off at regular intervals specified in units of seconds on the SD414. (When the value on SD414 is changed, the elapsed time count starting when on/off state on SM414 was last changed continues, and on/off state will be changed when the new specified time is elapsed.[*1]) • The initial state when CPU module is powered on or reset is off. • This relay switches to on or off after a specified time is elaspsed even when the program execution is in progress.	S (Status change)

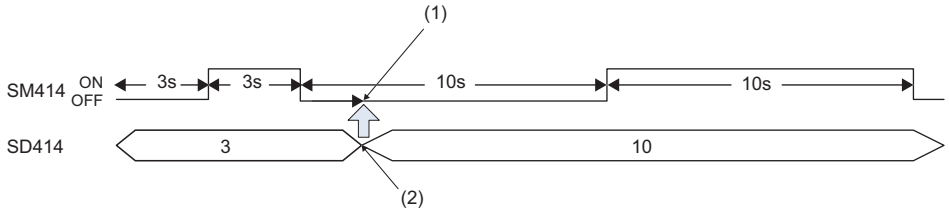
A

No.	Name	Data stored	Details	Set by (setting timing)
SM415	2n millisecond clock		• This relay repeats on/off at regular intervals specified in units of milliseconds on the SD415. (When the value on SD415 is changed, the elapsed time count starting when on/off state on SM415 was last changed continues, and on/off state will be changed when the new specified time is elapsed.*1) • The initial state when CPU module is powered on or reset is off. • This relay switches to on or off after a specified time is elapsed even when the program execution is in progress.	S (Status change)
SM420	User timing clock No.0		• This relay repeats on/off at specified scan intervals. • The initial state when CPU module is powered on or reset is off. • The on/off scan interval is set using the DUTY instruction. (n1: ON scan interval, n2: OFF scan interval)	S (Every END)
SM421	User timing clock No.1			
SM422	User timing clock No.2			
SM423	User timing clock No.3			
SM424	User timing clock No.4			
SM440	On only initial I44 execution after RUN		• This relay is on during the first execution of the inter-module synchronous interrupt program (I44) after RUN mode starts and off during the second execution onwards. • Note that the relay status while the DI instruction is executed is as follows: On during the first execution of the inter-module synchronous interrupt program (I44) after the DI instruction is cleared and Off during the second execution onwards. (no change after the second DI clear). • This contact is enabled only for the inter-module synchronous interrupt program.	S (Power-on to RUN or STOP to RUN)
SM441	On only initial I45 execution after RUN		• This relay is on during the first execution of the multiple CPU synchronous interrupt program (I45) after RUN mode starts and off during the second execution onwards. • Note that the relay status while the DI instruction is executed is as follows: On during the first execution of the multiple CPU synchronous interrupt program (I45) after the DI instruction is cleared and Off during the second execution onwards. (no change after the second DI clear). • This contact is enabled only for the multiple CPU synchronous interrupt program.	S (Power-on to RUN or STOP to RUN)

*1 The following figure shows an example of the process expected when the value for SD414 or SD415 is changed.

Ex.

If the value of SD414 is switched from 3 to 10:

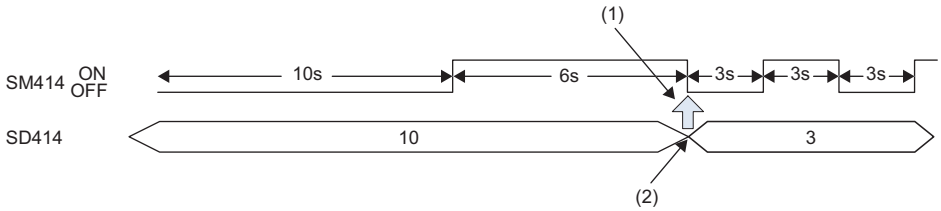


(1) The elapsed time after the ON/OFF state of SM414 changes remains.

(2) Value change

Ex.

If the value of SD414 is changes from 10 to 3:



(1) If the time changed with SM414 has elapsed after the last change of the on/off state of SD414, the on/off state of SM414 switches when the value of SD414 is changed.

(2) Value change

A

Fixed scan function information

The following is the list of special relay areas relating to the fixed scan function information.

No.	Name	Data stored	Details	Set by (setting timing)
SM480	Cycle error flag for inter-module synchronous interrupt program(I44)	Off: No error for the inter-module synchronization program (Normal) On: Error state for the inter-module synchronization program	• This relay switches to on when the inter-module synchronous interrupt program (I44) has not been completed within the specified inter-module synchronization cycle or the program cannot be executed due to various reasons, such as execution of a higher-priority interrupt program and interrupt disabling by the instruction execution. • On state is maintained even after the program is later completed within the specified inter-module synchronization cycle (Clear by turning power off and on or resetting).	S (Status change)
SM481	Cycle error flag for multiple CPU synchronization interrupt program (I45)	Off: No error for the multiple CPU synchronization program (Normal) On: Error state for the multiple CPU synchronization program	• This relay switches to on when the multiple CPU synchronization program (I45) has not been completed within the specified fixed scan communication cycle or the program cannot be executed due to various reasons, such as execution of a higher-priority interrupt program and interrupt disabling by the instruction execution. • On state is maintained even after the program is later completed within the specified fixed scan communication cycle (Clear by turning power off and on or resetting).	S (Status change)
SM484	Execution section excess error flag for multiple CPU synchronization interrupt program	Off: No execution section excess error for the multiple CPU synchronization interrupt program (Normal) On: Execution section excess error state for the multiple CPU synchronization interrupt program	• This relay switches to on when the program is executed exceeding the program execution section within the specified multiple CPU synchronization cycle. • On state is maintained even after the program is later completed within the multiple CPU synchronous interrupt program execution section (cleared by turning power off and on or resetting).	S (Status change)
SM488	Inter-module synchronization error (out of synchronization was detected on the CPU module)	Off: No error (Normal) On: Error	• This relay switches to on when the inter-module synchronization signal cannot be identified within the inter-module synchronization cycle specified in the parameter or more than one signal has been identified within the same inter-module synchronization cycle. • On state is maintained even after the inter-module synchronization signal can be later identified within the specified inter-module synchronization cycle (cleared by turning power off and on or resetting).	S (Status change)

Drive information

The following is the list of special relay areas relating to the drive information.

No.	Name	Data stored	Details	Set by (setting timing)
SM600	Memory card usable flags	Off: Disabled On: Enabled	This relay is on when an SD memory card is enabled (This relay switches to on when a valid SD memory card is attached and prepared for use).	S (Status change)
SM601	Memory card protect flag	Off: Not protected On: Protected	This relay is on when the write protect switch of the SD memory card is set to on.	S (Status change)
SM603	Memory card (drive 2) flag	Off: No SD memory card inserted On: SD memory card inserted	This relay is on when an SD memory card is attached (This relay switches to on when an SD memory card is attached regardless of its state (enabled/disabled) and type).	S (Status change)
SM604	Memory card in-use flag	Off: Not used On: In use	This relay is on when the SD memory card is being used.	S (Status change)
SM605	Memory card remove/insert prohibit flag	Off: Allowed to remove/attach On: Not allowed to remove/attach	This relay can be switched to on when prohibiting the SD memory card from being removed/attached. When this relay is on, if SM607 is set to on, the system switches it to off.	U/S
SM606	SD memory card forced disable instruction	Off: Cancel instruction On: Forced disable instruction	• This relay can be switched to on to issue the instruction that forces the SD memory card to be disabled. However, whenever any function accesses the SD memory card, the disabling process does not start until the access is completed. • This relay can be switched to off to cancel the instruction that forces the SD memory card to be disabled.	U
SM607	SD memory card forced disable status flag	Off: Not disabled On: Disabled	• This relay switches to on when the SD memory card is disabled by switching SM606 to on. • This relay switches to off when the forcibly-disabled SD memory card is enabled by switching SM606 to off.	S (Status change)
SM626	Extended SRAM cassette insertion flag	Off: Not inserted On: Inserted	This relay is on when the extended SRAM cassette is attached.	S (Status change)
SM628	Program memory write error	Off: No write operation/normal On: Write error	This relay switches to on when a write error is detected during write operation to the program memory. This relay switches to off when the write instruction is issued.	S (Writing)
SM629	Program memory write flag	Off: No write operation On: Executing write operation	This relay is on when the write process to the program memory is in progress. The relay is switched to off when the write process is completed.	S (Writing)
SM630	Program memory overwrite count error flag	Off: The number of rewrite operations is lower than 100000 On: The number of rewrite operations reaches 100000	This relay switches to on when the number of program memory rewriting operations reaches 100000 (CPU module must be replaced).	S (Writing)
SM632	Data memory write error	Off: No write operation/normal On: Write error	This relay switches to on when a write error is detected during write operation to the data memory. This relay switches to off when the write instruction is issued.	S (Writing)
SM633	Data memory write flag	Off: No write operation On: Executing write operation	This relay is on when the write process to the data memory is in progress. The relay is switched to off when the write process is completed.	S (Writing)
SM634	Number of rewriting operations error to data memory flag	Off: The number of rewrite operations is lower than 100000 On: The number of rewrite operations reaches 100000	This relay switches to on when the number of data memory rewriting operations reaches 100000 (CPU module must be replaced).	S (Writing)

No.	Name	Data stored	Details	Set by (setting timing)
SM636	System memory write error	Off: No write operation/ normal On: Write error	This relay switches to on when a write error is detected during write operation to the system memory (Flash ROM)*1. This relay switches to off when the write instruction is issued.	S (Writing)
SM637	System memory write flag	Off: No write operation On: Executing write operation	This relay switches to on when the write process to the system memory (Flash ROM)*1 is in progress, and switches to off when the write process is complete.	S (Writing)
SM638	System memory rewrite count error flag	Off: The number of rewrite operations is lower than 100000 On: The number of rewrite operations reaches 100000	This relay switches to on when the number of system memory (Flash ROM)*1 rewrite operations reaches 100000. (CPU module must be replaced).	S (Writing)

*1 This is the memory used by the system when the CPU module is executing functions.

Instruction related

The following is the list of special relay areas relating to the instruction-related items.

No.	Name	Data stored	Details	Set by (setting timing)
SM699	Dedicated instruction skip flag	Off: Instruction being executed or completed On: Instruction not executed	• This relay indicates whether the built-in Ethernet function instruction (the SP.SOCOPEN/SP.SOCCLOSE/SP.SOCRCV/S.SOCRCVS/SP.SOCSND/SP.ECPRTCL/SP.SLMPSND/SP.FTPPUT/SP.FTPGET instruction), intelligent function module dedicated instruction, or multiple CPU dedicated instruction has been skipped. (Checking this flag immediately after the instruction issue allows to determine whether or not the dedicated instruction is non-processing (skipped) in the internal processing.) • This relay saves/returns while the interrupt program is being executed.	S (Status change)
SM700	Carry flag	Off: Carry off On: Carry on	• This relay is a carry flag used while the application instruction is executing. • This relay saves/returns when the interrupt program is being executed.	S (Instruction execution)
SM701	Number of output characters selection	Off: Outputs until reaching NULL code On: Outputs 16 characters	• ASCII codes are output until reaching NULL (00H) code, when SM701 is off. • 16 characters of ASCII codes are output, when SM701 is on. • This relay saves/returns when the interrupt program is being executed.	U
SM702	Search method	Off: Sequential search On: Dichotomizing search	• The search method in the search instruction can be specified by using this relay. • To use the dichotomizing search, data must be sorted. • This relay saves/returns while the interrupt program is being executed.	U
SM703	Sort order	Off: Ascending On: Descending	• This relay can be used to specify how to arrange data in the sort instruction: ascending or descending. • This relay saves/returns while the interrupt program is being executed.	U
SM704	Block comparison	Off: Mismatch is detected On: Completely match	• This relay switches to on when all the data conditions are met in the block data comparison instruction. • This relay saves/returns while the interrupt program is being executed.	S (Instruction execution)
SM705*2	Number of conversion digits selection	Off: Fixed number of digits On: Desired number of digits	• When SM705 is off, the conversion data is specified and output with the fixed number of digits. • When SM705 is on, the conversion data is specified and output with the desired number of digits. • This relay saves/returns while the interrupt program is being executed.	U
SM709	DT/TM instruction improper data detection flag	Off: No improper data On: Improper data is detected	• This relay switches to on when the comparison target data is date data or cannot be recognized as clock data, or the comparison target device (three words) exceeds the specified device range in the DT/TM instruction. • This relay saves/returns while the interrupt program is being executed.	S (Instruction execution)/U

No.	Name	Data stored	Details	Set by (setting timing)
SM752	Dedicated instruction End bit control flag	Off: Automatically controlled On: Not automatically controlled	This relay can be used to set whether the system automatically controls the on/off operation of End bit being used for the dedicated instruction and other (control of End bit which is registered in the End processing of the instruction completion and turned on for only the next one scan (Off: turned on for only one scan after the instruction is completed as well as normal/error End bit, On: turned on when the instruction is completed and the On status is maintained after that)).	U
SM753	File being accessed	Off: Not in progress On: In progress	• This relay is on during a file access by using the SP.FWRITE/SP.FREAD/SP.DEVST instructions. • This relay is on while the SP.FTPPUT or SP.FTPGET instruction is being executed. • This relay is on while a database access instruction is being executed.*2	S (Status change)
SM754	BIN/DBIN instruction error control flag	Off: Executes error detection On: No execute error detection	This relay can be switched to on when the error detection is not desirable in the BIN/DBIN instruction.	U
SM755	Scaling data check settings	Off: Performs data check On: Not perform data check	This relay can be used to enable/disable the check whether the scaling data is sorted in ascending order when the SCL/DSCL/SCL2/DSCL2 instruction is being executed.	U
SM756	Module access completion wait control flag	Off: Not wait the completion On: Waits the completion	This relay can be used to determine whether the system must wait until the access is completed before starting the next instruction when the write access instruction to the buffer memory of another module is being executed.	U
SM772	CCD/CRC instruction conversion mode (16-bit/8-bit) selection	Off: 16-bit conversion mode On: 8-bit conversion mode	This relay can be used to to select 16-bit conversion mode or 8-bit conversion mode with the CCD/CRC instruction.	U
SM773	SMOV instruction BCD conversion prohibit flag	Off: BCD conversion enabled On: BCD conversion disabled	This relay switches to on when disabling conversion from BIN data to BCD data with the SMOV instruction.	U
SM775	Selection of refresh processing during the COM instruction execution	Off: Executes all the refresh processings On: Executes the refresh processing specified in SD775	This relay can be used to select the refresh processing target when the COM instruction is executed from two options: all refresh processing or only refresh processing specified in SD775.	U
SM776	Local device setting at CALL	Off: Disables local devices On: Enables local devices	This relay can be used to determine whether the local devices on the subroutine program called when the CALL instruction is being executed are enabled or not.	U
SM777	Local device setting in interrupt programs	Off: Disables local devices On: Enables local devices	This relay can be used to determine whether the local devices are enabled or not when the interrupt program is being executed.	U
SM792	PID bumpless processing (for the complete differentiation PIDCONT instruction)	Off: Matching On: Not matching	This relay can be used to specify whether to match SV to PV in manual mode.	U
SM794	PID bumpless processing (for the inexact differential S.PIDCONT instruction)	Off: Matching On: Not matching	This relay can be used to specify whether to match SV to PV in manual mode.	U
SM796	Number of used blocks information for the multiple CPU dedicated instruction (for CPU No.1)	Off: The specified number of blocks is reserved On: The number of blocks specified in SD796 is not reserved	• This relay switches to on when the number of remaining blocks in the dedicated instruction transfer area to be used in the multiple CPU dedicated instruction (target machine: CPU No.1) drops below the number of blocks specified in SD796. This relay is also on when the instruction is being executed. • This relay switches to off when there exist free blocks during the END processing.	S (Instruction execution/During END)

No.	Name	Data stored	Details	Set by (setting timing)
SM797	Number of used blocks information for the multiple CPU dedicated instruction (for CPU No.2)	Off: The specified number of blocks is reserved On: The number of blocks specified in SD797 is not reserved	• This relay switches to on when the number of remaining blocks in the dedicated instruction transfer area to be used in the multiple CPU dedicated instruction (target machine: CPU No.2) drops below the number of blocks specified in SD797. This relay is also on when the instruction is being executed. • This relay switches to off when there exist free blocks during the END processing.	S (Instruction execution/ During END)
SM798	Number of used blocks information for the multiple CPU dedicated instruction (for CPU No.3)	Off: The specified number of blocks is reserved On: The number of blocks specified in SD798 is not reserved	• This relay switches to on when the number of remaining blocks in the dedicated instruction transfer area to be used in the multiple CPU dedicated instruction (target machine: CPU No.3) drops below the number of blocks specified in SD798. This relay is also on when the instruction is being executed. • This relay switches to off when there exist free blocks during the END processing.	S (Instruction execution/ During END)
SM799	Number of used blocks information for the multiple CPU dedicated instruction (for CPU No.4)	Off: The specified number of blocks is reserved On: The number of blocks specified in SD799 is not reserved	• This relay switches to on when the number of remaining blocks in the dedicated instruction transfer area to be used in the multiple CPU dedicated instruction (target machine: CPU No.4) drops below the number of blocks specified in SD799. This relay is also on when the instruction is being executed. • This relay switches to off when there exist free blocks during the END processing.	S (Instruction execution/ During END)
SM816*1	Hold mode (S.IN instruction)	Off: Value not held On: Value held	Whether to hold the output value or not is specified when the input value is found to be exceeding the valid range during the range check processing of the S.IN instruction.	U
SM817*1	Hold mode (S.OUT instruction)	Off: Value not held On: Value held	This relay is used to specify whether or not to hold the output values of the S.OUT1, S.OUT2, and S.DUTY instructions if a sensor error occurs.	U

*1 Only the Process CPU supports these special relay areas.

*2 The programmable controller CPU with firmware version "28" or later supports these special relay areas.

Firmware update function

The following is the list of the special relay area relating to the firmware update function. (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SM922	Firmware update completion with/without an error	Off: Update completed without an error (including successful completion) On: Update completed with an error	This relay switches to on when the firmware update function is completed with an error. (switching to on when SD922 is 0100H to 0300H)	S (Initial)

Latch area

The following is the list of special relay areas relating to the latch area.

No.	Name	Data stored	Details	Set by (setting timing)
SM953*1	CPU module data backup error check flag	Off: No error On: Error	• This relay turns on if an error occurs at the execution of backup of the CPU module. • This relay turns off at the start of the CPU module data backup.	S (Status change)
SM959*1	CPU module data restoration error check flag	Off: No error On: Error	• This relay turns on if an error occurs at the execution of restoration of the CPU module. • This relay turns off at the start of restoration of the CPU module.	S (Status change)
SM960*1	Upper limit setting flag for the number of CPU module backup data	Off: Backup continued On: Backup stopped	This relay specifies the operation of backup when the number of backup data of the CPU module reaches the upper limit. (This relay is valid only when bit5 of SD944 is on.) Off: After deleting the oldest date stamp folder, the backup is continued. On: The backup is not continued if the upper limit of the number of backup data is exceeded. (In this case, the backup is completed with an error.)	U
SM961*1	Automatic backup retry failure flag	Off: Retry not executed/Retry being executed On: Retry failed	This relay turns on when the retry of the automatic backup of the CPU module is failed even after the specified number of retries are attempted. This relay turns off at the start of the automatic backup. (This relay does not turn off when SM1351 is on.)	S (Status change)

*1 Programmable controller CPU only

Data logging function

The following is the list of special relay areas relating to the data logging function.

No.	Name	Data stored	Details	Set by (setting timing)
SM1200	Auto logging setting file and registration status	Off: Mismatch On: Matching	• This relay is on when the auto logging setting executed (registered) matches with the content of the configuration file stored in the target memory. This relay is off when the setting does not match with the content.	S (Status change)
SM1201	SD memory card setting file in use flag	Off: Not used On: In use	• This relay switches to on when the data logging configuration file stored in an SD memory card is being used. This relay switches to on when one or more data logging of the settings No.1 to 10 is registered. • On state is maintained even when the data logging later enters into suspend/waiting for start without collection/waiting for RUN without collection mode. However, the relay switches to off when all the data logging stops.	S (Status change)
SM1202	Data memory setting file in use flag	Off: Not used On: In use	• This relay switches to on when the data logging configuration file stored in the data memory is being used. This relay switches to on when one or more data logging of the settings No.1 to 10 is registered. • On state is maintained even when the data logging later enters into suspend/waiting for start without collection/waiting for RUN without collection mode. However, the relay switches to off when all the data logging stops.	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SM1210	Data logging setting No.1 Data logging preparation	Off: Not prepared On: Prepared	• This relay switches to on when the data logging preparation is completed. • On state is maintained even when the data logging later enters into suspend/waiting for start without collection/waiting for RUN without collection mode. This relay switches to off when the data logging stops.	S (Initial)
SM1211	Data logging setting No.1 Data logging start	Off: Suspended/waiting for start On: Start	• This relay switches to on when the data logging starts. The relay turns off when the data logging is in suspend/waiting for start without collection mode. Associated special relay, such as Data logging collection, Data logging end, Data logging trigger, and After data logging trigger, switches to off simultaneously. • This relay also switches to off when the CPU module is moved from RUN to STOP mode and therefore the data collection is halted.	S (Status change)
SM1212	Data logging setting No.1 Data logging collection	Off: Not in progress On: In progress	This relay switches to on when the data logging starts data collection.	S (Status change)
SM1213	Data logging setting No.1 Data logging end	Off: Not completed On: Completed	This relay switches to on when the data logging is completed. For continuous logging, the corresponding bit switches to on when writing reaches the maximum number of storage files and data logging is completed (with "Stop" set for the operation at the time when the number of save files exceeds the limit). For trigger logging, the corresponding bit switches to on when data collection for the specified number of records has been completed followed by writing into the SD memory card after the trigger condition was satisfied. The bit also switches to on when an error is generated (except for data logging error caused by online program change) during the data logging execution.	S (Status change)
SM1214	Data logging setting No.1 Data logging trigger	Off→On:Triggered	The system switches this relay to on when the specified trigger condition is satisfied.	S (Status change)
SM1215	Data logging setting No.1 After data logging trigger	Off: Not post triggering On: Post triggering	• This relay switches to on once data logging triggering occurs. On state is maintained even when the data logging is completed. However, this relay is off when the data logging is in suspend/waiting for start without collection/stop mode • This relay also switches to off when the CPU module is moved from RUN to STOP mode and therefore the data collection is halted.	S (Status change)
SM1216	Data logging setting No.1 Data logging error	Off: No error On: Error	• This relay switches to on when a data logging function error is generated. • This relay switches to off when the setting is registered or by the stop command from CPU Module Logging Configuration Tool.	S (Status change)
SM1217	Data logging setting No.1 Data logging data saving into memory card in progress	Off: Not in progress On: In progress	This relay switches to on when data in the internal buffer are being saved into an SD memory card as a part of the data logging process.	S (Status change)
SM1218	Data logging setting No.1 Logging data storage file switching in progress	Off: Not in progress On: In progress	This relay switches to on when storage file switching is in progress.	S (Status change)
SM1220 to SM1228	Data logging setting No.2	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1230 to SM1238	Data logging setting No.3	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1240 to SM1248	Data logging setting No.4	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1

No.	Name	Data stored	Details	Set by (setting timing)
SM1250 to SM1258	Data logging setting No.5	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1260 to SM1268	Data logging setting No.6	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1270 to SM1278	Data logging setting No.7	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1280 to SM1288	Data logging setting No.8	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1290 to SM1298	Data logging setting No.9	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1300 to SM1308	Data logging setting No.10	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1210 to SM1218).	Same configuration as the setting No.1
SM1312 to SM1321	Data logging setting No.1 to 10 Data logging suspend/resume flag	Off→On:Suspend On→Off:Resume	• When this relay changes from off to on, the data logging function is suspended. In an off state of the data logging start SM, no processing is performed. • When this relay changes from on to off, the data logging function resumes. In an on state of the data logging start SM, no processing is performed.	U

CPU module data backup/restoration function

The following is the list of special relay areas relating to the CPU module data backup/restoration function. (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SM1350	CPU module data backup status flag	Off: Not being executed On: Being executed	This relay turns on during the backup of the CPU module.	S (Status change)
SM1351	CPU module data backup execution request	Off→On:Backup requested On→Off:Backup completed	• At the timing when this relay turns from off to on, the backup of the CPU module is executed. • This relay turns off at the completion of the backup of the CPU module.	S (Status change)/U
SM1353	CPU module data restoration status flag	Off: Not being executed On: Being executed	This relay turns on during the restoration of the CPU module.	S (Status change)
SM1354	CPU module data restoration execution request	Off→On:Restoration requested On→Off:Restoration completed	• At the timing when this relay turns from off to on during STOP, the restoration of the CPU module is executed. • This relay turns off at the completion of the restoration of the CPU module.	S (Status change)/U
SM1356	Retry status flag for CPU module data automatic backup	Off: Not being executed On: Being executed	This relay turns on during the retry of the automatic backup of the CPU module.	S (Status change)

iQ Sensor Solution data backup/restoration function

The following is the list of special relay areas relating to the iQ Sensor Solution data backup/restoration function. (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SM1360	Right-to-use request for iQ Sensor Solution data backup	Off→On:Right to use requested On→Off:Right to use acquired/released	• This relay requests acquiring or releasing the right to use of the data backup/restoration of iQ Sensor Solution. • At the timing when this relay turns from off to on, the relay requests the right to use with the value specified in SD1360. • This relay turns off when the right to use has been acquired or released.	S (Status change)/U
SM1361	iQ Sensor Solution backup request	Off→On:Backup requested On→Off:Backup completed	• This relay requests the data backup of iQ Sensor Solution. • At the timing when this relay turns from off to on, the backup of the set device(s) is executed. • This relay turns off at the completion of the data backup of iQ Sensor Solution. • This relay turns from on to off when the right to use is acquired.	S (Status change)/U
SM1362	iQ Sensor Solution backup normal completion	Off: Other than normal completion On: Completed normally	• This relay turns on when the iQ Sensor Solution data backup is completed normally. • When the iQ Sensor Solution data backup is completed with an error, SM1363 turns on and this relay remains off. • This relay turns from on to off when the right to use is acquired or the backup is requested.	S (Status change)
SM1363	iQ Sensor Solution backup error completion	Off: Other than error completion On: Completed with an error	• This relay turns on when the iQ Sensor Solution data backup is competed with an error. The error code is stored into SD1376 or SD1377. • When the iQ Sensor Solution data backup is competed normally, SM1362 turns on and this relay remains off. • This relay turns from on to off when the right to use is acquired or the backup is requested.	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SM1364	iQ Sensor Solution restoration request	Off→On:Restoration requested On→Off:Restoration completed	• This relay requests the iQ Sensor Solution data restoration. • At the timing when this relay turns from off to on, the restoration of the set device(s) is executed. • This relay turns off at the completion of the iQ Sensor Solution data restoration. • This relay turns from on to off when the right to use is acquired.	S (Status change)/U
SM1365	iQ Sensor Solution restoration normal completion	Off: Other than normal completion On: Completed normally	• This relay turns on when the iQ Sensor Solution data restoration is completed normally. • When the iQ Sensor Solution data restoration is completed with an error, SM1366 turns on and this relay remains off. • This relay turns from on to off when the right to use is acquired or the restoration is requested.	S (Status change)
SM1366	iQ Sensor Solution restoration error completion	Off: Other than error completion On: Completed with an error	• This relay turns on when the iQ Sensor Solution data restoration is completed with an error. The error code is stored into SD1376 or SD1377. • When the iQ Sensor Solution data restoration is completed normally, SM1365 turns on and this relay remains off. • This relay turns from on to off when the right to use is acquired or the restoration is requested.	S (Status change)
SM1367	iQ Sensor Solution backup/ restoration suspend request	Off→On:Suspension requested On→Off:Suspension completed	• This relay requests the suspension of the iQ Sensor Solution data backup/restoration. • At the timing when this relay turns from off to on, the iQ Sensor Solution data backup/ restoration is suspended. • This relay turns off at the suspension completion of the iQ Sensor Solution data backup/restoration. • This relay turns from on to off when the right to use is acquired or the backup/restoration is requested.	S (Status change)/U

File transfer function (FTP client)

The following is the list of special relay areas relating to the file transfer function (FTP client). (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SM1392	FTP client connection status	Off: Not connected (disconnected) On: Connected	This relay turns on when the connection with the FTP server is established. This relay turns off when the connection with the FTP server is cut off.	S (Status change, END processing)

Memory dump function

The following is the list of special relay areas relating to the memory dump function. (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SM1472	Memory dump in progress	Off: Memory dump not executed On: Memory dump in progress	Turns on if memory dump is in progress, and turns off if memory dump is not executed.	S (Status change)
SM1473	Memory dump completed	Off: Not completed On: Completed	Turns on at the time when collection of the data and save to the SD memory card are completed after the establishment of the trigger condition. Thereafter, the on state remains even if memory dump is executed again. If the trigger condition is established again, On changes to Off.	S (Status change)

CPU module database access function

The following is the list of special relay areas relating to the CPU module database access function. (Programmable controller CPU with firmware version "28" or later)

No.	Name	Data stored	Details	Set by (setting timing)
SM1497	Memory card free space flag for CPU module database	Off: 20MB or more free space On: Less than 20MB free space	Turns on when the free space of the SD memory card storing the CPU module database is 20MB or less.	S (Status change)
SM1498	CPU module database start-up flag	Off: Before start-up On: After start-up	Turns on only when the CPU module database access function is enabled by the module parameter setting and the database can be accessed.	S (Status change)
SM1499	CPU module database start-up failure	Off: Not failed On: Failed	Turns on only when the CPU module database access function is enabled by the module parameter setting but the database start-up is failed.	S (Status change)

Ethernet function

The following is the list of special relay areas relating to the Ethernet function.

No.	Name	Data stored	Details	Set by (setting timing)
SM1520	IP address storage area write request	Off→On:Write request exists On→Off:Writing completed	• When this relay is changed from off to on, IP address settings stored in SD1520 to SD1525 are written to the IP address storage area (system memory*1) of the CPU module. • Once write to the IP address storage area (system memory*1) is completed (regardless of whether it was successful or not), this relay is turned off.	S (Status change)/U
SM1521	IP address storage area write error	Off: No error On: Error	This relay is turned to on when write to the IP address storage area (system memory*1) failed, and off when it was successful.	S (Status change)
SM1522	IP address storage area clear request	Off→On:Clear request exists On→Off:Clear completed	When this relay is changed from off to on, the IP address storage area (system memory*1) is cleared. Once clear of the IP address storage area (system memory*1) is completed (regardless of whether it was successful or not), this relay is turned off.	S (Status change)/U
SM1523	IP address storage area clear request	Off: No error On: Error	This relay is turned to on when clear of the IP address storage area (system memory*1) failed, and off when it was successful.	S (Status change)
SM1524	Initial processing successful completion state	Off: Completed with an error On: Completed normally	This relay is turned on when the initial processing for the Ethernet function was completed successfully. *2When the initial processing was completed with an error, SM1525 is turned on and SM1524 stays off.	S (Status change)
SM1525	Initial processing error completion state	Off: Completed normally On: Completed with an error	This relay is turned on when the initial processing for the Ethernet function was completed with an error. *2When the initial processing was completed successfully, SM1524 is turned on and SM1525 stays off.	S (Status change)

*1 This is the memory used by the system when the CPU module is executing functions.

*2 Ethernet function initial processing involves updating parameters set for data communication to Ethernet-equipped modules to enable communication with external devices. Initial processing of Ethernet-equipped modules is performed by setting Ethernet parameters, writing them to the CPU module, and then powering off and on or resetting the CPU module. Note that if Ethernet parameters are not set, initial processing of Ethernet-equipped modules will be performed based on default parameters. (SM1524 (initial processing successful completion status) turns on the moment initial processing is complete, and communication with external devices becomes possible.)

CC-Link IE Field Network Basic function

The following is the list of special relay areas relating to the CC-Link IE Field Network Basic function. (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SM1536	Cyclic transmission status	Off: Not performed On: Being performed	• This relay turns on when the cyclic transmission starts. • This relay is turned off when the cyclic transmission stops.	S (Status change)
SM1540	Data link status	Off: All stations normal On: One or more faulty stations	• This relay turns on when an error exists even in one slave station. • The status of each slave station can be checked in SD1540 (Data link status of each station).	S (Status change)

Online module change function

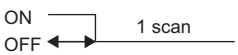
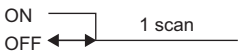
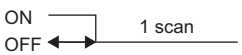
The following is the list of special relay areas relating to the online module change function. (Process CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SM1600	Module selection request flag	Off→On:Requested	This relay is turned on to select an online change target module. When changing a module directly, the system turns on this relay upon removal of the target module. The relay can be turned on only when the value set in SD1617 is 0 (Normal operation). The relay turns off upon completion of the online module change processing. If the selection cancel is requested, the relay turns off after the selection is cancelled.	S (Status change)/U (Request)
SM1601	Module selection completion flag	Off: No module selected On: Selected	This relay turns on when an online change target module has been selected. The relay turns off upon completion of the online module change processing.	S (Status change)
SM1602	Module removal request flag	Off→On:Requested	This relay is turned on to request a removal of the selected module. When changing a module directly, the system turns on this relay upon removal of the target module. The relay can be turned on only when the value set in SD1617 (Online module change progress status) is 2 (Module selected). The relay turns off upon completion of the online module change processing.	S (Status change)/U (Request)
SM1603	Module removal ready flag	Off: Not prepared On: Ready	This relay turns on when the selected module is ready to be removed. The relay turns off upon completion of the online module change processing.	S (Status change)
SM1604	Module removal completion flag	Off: Not completed On: Completed	This relay turns on when the selected module has been removed. The relay turns off upon completion of the online module change processing.	S (Status change)
SM1605	Module mounting completion flag	Off: Not completed On: Completed	This relay turns on when a new module has been mounted. The relay turns off upon completion of the online module change processing.	S (Status change)
SM1606	Module recognition request flag	Off→On:Requested	This relay is turned on to request recognition of the newly-mounted module. When changing a module directly, the system turns on this relay upon mounting of the module. The relay can be turned on only when the value set in SD1617 (Online module change progress status) is 6 (Module mounted). The relay turns off upon completion of the online module change processing.	S (Status change)/U (Request)
SM1607	Module recognition completion flag	Off: Not recognized On: Recognized	This relay turns on when the newly-mounted module is recognized by the system. The relay turns off upon completion of the online module change processing.	S (Status change)
SM1608	Module control resumption request flag	Off→On:Requested	This relay turns on to start control of the replaced module. When changing a module directly, the system turns on this relay upon recognition of the module. The relay can be turned on only when the value set in SD1617 (Online module change progress status) is 8 (Module recognized). The relay turns off upon completion of the online module change processing.	S (Status change)/U (Request)
SM1609	Online module change completion flag	Off: Not completed On: Completed	The relay turns on upon completion of the online module change processing. This relay turns off in the next scan.	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SM1615	Module selection cancellation request flag	Off→On:Module selection cancellation requested	This relay is turned on to cancel a module selection request. The relay can be turned on only when the value set in SD1617 (Online module change progress status) is 2 (Module selected). The relay turns off after the selection is cancelled.	S (Status change)/U (Request)
SM1616	Online module change availability flag	Off: Disabled On: Enabled	This relay turns on when only the Process CPU is used in a single CPU system. In a multiple CPU system, if any of the CPU modules other than the Process CPU does not support the online module change function, the relay turns off. If all the CPU modules support the function, the relay turns on. Note that the setting details in the direct change setting cannot be checked with SM1616 (Online module change availability flag). To check the direct change setting, check the CPU parameters.	S (Initial)
SM1617	Online module change status flag	Off: Function not executed On: Function being executed	This relay turns on when SM1600 (Module selection request flag) is turned on to start the online module change processing. The relay turns off upon completion of the online module change processing.	S (Status change)
SM1618	Online module change error flag	Off: No error On: Error	This relay turns on when an error is detected. This relay turns off when the error cause is eliminated and the online module change related request is executed. An error occurs in selecting a module. Thus, turn off the relay before module selection.	S (Status change)/U (Request)
SM1619	Disable request flag during online module change	Off: No disable request On: Disable request detected	This relay turns on when a disable request is issued during the online module change processing. The relay turns off upon completion of the online module change processing.	S (Status change)

Redundant function

The following is the list of special relay areas relating to the Redundant function. (Process CPU (redundant mode) only)

No.	Name	Data stored	Details	Set by (setting timing)
SM1630	Operation mode identification flag	Off: Redundant system in backup mode, stand-alone system On: Redundant system in separate mode	This relay turns on while a redundant system is operating in separate mode.	S (Every END)
SM1632	System A identification flag	Off: System B, system not determined On: System A	• This relay distinguishes between the system A and the system B. • The flag status does not change even if tracking cables are disconnected.	S (Initial)
SM1633	System B identification flag	Off: System A, system not determined On: System B	• This relay distinguishes between the system A and the system B. • The flag status does not change even if tracking cables are disconnected.	S (Initial)
SM1634	Control system judgment flag	Off: Standby system, system not determined On: Control system	• This relay indicates operating status of the CPU module. The flag is stored in each system when an initial processing is performed (including when the system is determined while waiting for the start-up of the other system) and when a system switching is completed. • The flag status does not change even if tracking cables are disconnected.	S (Initial/Status change)
SM1635	Standby system judgment flag	Off: Control system, system not determined On: Standby system	• This relay indicates operating status of the CPU module. The flag is stored in each system when an initial processing is performed (including when the system is determined while waiting for the start-up of the other system) and when a system switching is completed. • The flag status does not change even if tracking cables are disconnected.	S (Initial/Status change)
SM1636	Previous control system identification flag		When the previous control system is the system B, this relay turns on during one scan in the system A, following the RUN state after both systems were powered on or the CPU module is reset.	S (Every END)
SM1637	System switching detection (standby system to control system)	Off: Not detected On: Detected	This relay turns on after the standby system has been switched to the control system.	S (Status change)
SM1643	ON for only one scan after system switching (standby system to control system)		• This relay turns on during one scan after the standby system was switched to the control system. • This relay can be executed only in a scan execution type program.	S (Every END)
SM1644	ON for only one scan after system switching (control system to standby system)		• This relay turns on during one scan after the control system was switched to the standby system. • This relay can be executed only in a scan execution type program.	S (Every END)
SM1645	System switching request from a network module	Off: System switching request not issued On: System switching request issued	• This relay turns on when a system switching request from a network module is issued. • The module that issued system switching can be checked in SD1645. • This relay turns off when all bits of SD1645 are off.	S (Every END)
SM1646	System switching by a user	Off: Disabled On: Enabled	• This relay stores whether to enable system switching operation by a user using an engineering tool or the SP.CONTSW instruction. • Initial value is off: System switching by a user is prohibited.	U

No.	Name	Data stored	Details	Set by (setting timing)
SM1653	Memory copy start	Off: Not started On: Start	• When this relay is turned from off to on, the memory copy from the control system to the standby system starts. Note that the memory copy does not start as long as the I/O number of the copy destination (standby system CPU module: 03D1H) is not stored when SM1653 is turned from off to on. • Initial value is off: The memory copy has not started.	U
SM1654	Memory copy being executed	Off: Not executed On: Being executed	• This relay is on during the memory copy from the control system to the standby system. • This relay turns off when the memory copy is completed.	S (Status change)
SM1655	Memory copy completion	Off: Not completed On: Completed	• This relay turns on at the completion of the memory copy from the control system to the standby system. • Initial value is off: Memory copy has not completed.	S (Status change)/U
SM1656	Auto memory copy	Off: Disabled On: Enabled	This relay turns on when the automatic memory copy is set to enable.	S (Initial)
SM1673	Tracking transfer completion flag	Off: Transfer not completed On: Transfer completed	• This flag is stored a result of the tracking transfer operated in the preceding END processing. • The flag turns on if any of the tracking transfers in block 1 to 64 has been normally completed, and turns off if the tracking transfers failed due to causes such as tracking communication error.	S (Status change)
SM1679	Error reset (the other system)	Off→On:Standby system error reset requested On→Off:Standby system error reset completed	• This relay is turned from off to on to clear a continuation error occurred in the standby system. • This relay turns from on to off when the error reset has been completed. • Initial value is off.	U/S (Status change)
SM1680	Error of the other system monitoring	Off: No error On: Error	• This relay turns on if an error occurs on communications with the other system when an initial processing (including when the system is determined while waiting for the other system starts up) or an END processing is performed. (This relay turns on when the bit of SD1648 turns on.) • This relay turns off when an error is cleared.	S (Initial/every END/system switching)
SM1681	Latest self-diagnostic error (including annunciator ON) (the other system)	Off: No error On: Error	• This relay turns on if a diagnostic error occurs in the CPU module in the other system. (The relay also turns on if an error is detected by an annunciator.) • The SM0 status for the CPU module in the other system is reflected.	S (Every END)
SM1682	Latest self-diagnostic error (not including annunciator ON) (the other system)	Off: No error On: Error	• This relay turns on if a self-diagnostic error occurred in the CPU module in the other system. (The relay remains off if an error is detected by an annunciator.) • The SM1 status for the CPU module in the other system is reflected.	S (Every END)
SM1683	Detailed information 1: Flag in use (the other system)	Off: Not used On: In use	• This relay turns on when there is detailed information 1 for an error occurred in the CPU module in the other system. • The SM80 status for the CPU module in the other system is reflected.	S (Every END)
SM1684	Detailed information 2: Flag in use (the other system)	Off: Not used On: In use	• This relay turns on when there is detailed information 2 for an error occurred in the CPU module in the other system. • The SM112 status for the CPU module in the other system is reflected.	S (Every END)

No.	Name	Data stored	Details	Set by (setting timing)
SM1754	Waiting for the start-up of the other system	Off: Started up (own system) On: Waiting for the start-up (the other system)	• This relay turns on while waiting for the start-up of the other system after powered on. • This relay turns off under the following conditions. · Own system starts up as the control system or the standby system after succeeding in tracking communications with the other system. · Own system is started up as the control system by a certain operation while waiting for the start-up of the other system.	S (Status change)

Safety information (RnSFCPU)

The following is a list of special relay areas relating to safety information. (Safety CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SM1840	Allowed time over flag for continuous RUN in TEST MODE	Off: Within the specified time On: Specified time exceeded	This relay turns on when the continuous RUN time in TEST MODE has exceeded the allowed time set in the CPU parameter, and turns off in the following cases. • When the Safety CPU is powered on or is reset • When an error is cleared	S (Error)
SM1888	Safety cycle processing time execution cycle error flag	Off: No error (normal) On: Error	• This relay switches to on if a cycle in which safety cycle processing was not executed is detected. • Once the relay turns on, it remains on even if the safety cycle processing is performed each cycle. (To turn off the relay, power off and on the system or reset the CPU module).	S (Status change)
SM1904	CC-Link IE Field Network safety communication setting (1st module)	Off: Not set On: Set	• This relay stores the CC-Link IE Field Network safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1904.	S (Initial)
SM1905	CC-Link IE Field Network safety communication setting (2nd module)	Off: Not set On: Set	• This relay stores the CC-Link IE Field Network safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1905.	S (Initial)
SM1906	CC-Link IE Field Network safety communication setting (3rd module)	Off: Not set On: Set	• This relay stores the CC-Link IE Field Network safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1906.	S (Initial)
SM1907	CC-Link IE Field Network safety communication setting (4th module)	Off: Not set On: Set	• This relay stores the CC-Link IE Field Network safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1907.	S (Initial)
SM1908	CC-Link IE Field Network safety communication setting (5th module)	Off: Not set On: Set	• This relay stores the CC-Link IE Field Network safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1908.	S (Initial)
SM1909	CC-Link IE Field Network safety communication setting (6th module)	Off: Not set On: Set	• This relay stores the CC-Link IE Field Network safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1909.	S (Initial)
SM1910	CC-Link IE Field Network safety communication setting (7th module)	Off: Not set On: Set	• This relay stores the CC-Link IE Field Network safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1910.	S (Initial)
SM1911	CC-Link IE Field Network safety communication setting (8th module)	Off: Not set On: Set	• This relay stores the CC-Link IE Field Network safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1911.	S (Initial)

Appendix 5 List of Special Register Areas

The following table lists items in the list.

Item	Description
No.	Special register number
Name	Special register name
Data stored	Data stored in the special register
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (program, engineering tool, GOT, or other testing operations from external device) • U/S: User and system <Set timing> • Every END: Data is set every time END processing is performed. • Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN). • Status change: Data is set when the status is changed. • Error: Data is set when an error occurs. • Instruction execution: Data is set when an instruction is executed. • Request: Data is set when requested by a user (using the special relay). • Switch change: Data is set when the switch of the CPU module is changed. • Card insertion/removal: Data is set when an SD memory card is inserted or removed. • Writing: Data is set when a user performs a writing operation. • During END: Data is set when END processing is performed. • System switching: Data is set when two systems are switched (between the control system and the standby system)

Point

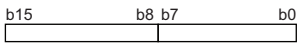
Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

A

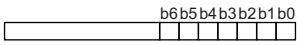
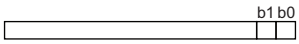
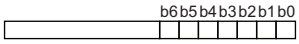
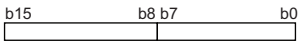
Diagnostic information

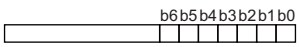
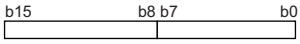
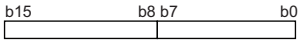
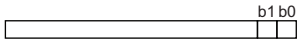
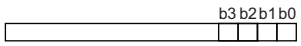
The following is the list of special register areas relating to the diagnostic information.

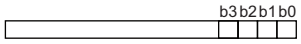
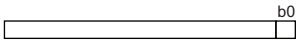
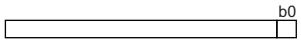
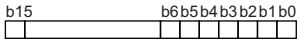
No.	Name	Data stored	Details	Set by (setting timing)
SD0	Latest self diagnostics error code	Latest self diagnostics error code	Error codes are stored in a hexadecimal value when the diagnostics detects an error.	S (Error)
SD1	Latest self-diagnostics error time	Latest self-diagnostics error time	The year value (four digits) of the date/time when SD0 data was updated is stored as a BIN code.	S (Error)
SD2			The month value of the date/time when SD0 data was updated is stored as a BIN code.	
SD3			The day value of the date/time when SD0 data was updated is stored as a BIN code.	
SD4			The hour value of the date/time when SD0 data was updated is stored as a BIN code.	
SD5			The minute value of the date/time when SD0 data was updated is stored as a BIN code.	
SD6			The second value of the date/time when SD0 data was updated is stored as a BIN code.	
SD7			The day of the week value of the date/time when SD0 data was updated is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)	
SD10	Self-diagnostic error number	Self-diagnostic error number 1	The maximum of 16 types of error codes are stored into SD10 onwards when the diagnostics detects an error (The same error code as one already stored in SD10 onwards is not stored). The 17th error code onwards are not stored. Also error codes are not stored when 16 types of error codes have already been stored into SD10 to SD25.	S (Error)
SD11		Self-diagnostic error number 2		
SD12		Self-diagnostic error number 3		
SD13		Self-diagnostic error number 4		
SD14		Self-diagnostic error number 5		
SD15		Self-diagnostic error number 6		
SD16		Self-diagnostic error number 7		
SD17		Self-diagnostic error number 8		
SD18		Self-diagnostic error number 9		
SD19		Self-diagnostic error number 10		
SD20		Self-diagnostic error number 11		
SD21		Self-diagnostic error number 12		
SD22		Self-diagnostic error number 13		
SD23		Self-diagnostic error number 14		
SD24		Self-diagnostic error number 15		
SD25		Self-diagnostic error number 16		
SD53	AC/DC DOWN	The number of AC/DC DOWN detections	The value of this register increments by one and stored as a BIN code, each time input voltage drops to 85% (AC power)/65% (DC power) or less of the nominal value while the CPU module is carrying out an operation. A counting cycle from 0 to 65535 to 0 is repeated.	S (Error)
SD60	Number of module with blown fuse	Number of module with blown fuse	The lowest number of module in which a fuse blew is stored.	S (Error)

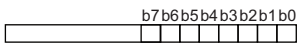
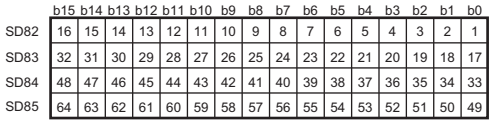
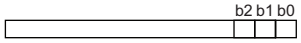
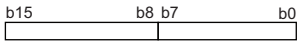
No.	Name	Data stored	Details	Set by (setting timing)
SD61	I/O module verify error module number	I/O module verify error module number	The lowest I/O number of the module which has an error detected by the I/O module verification is stored.	S (Error)
SD62	Annunciator number	Annunciator number	The annunciator number which is detected first is stored.	S (Instruction execution)
SD63	Number of annunciators	Number of annunciators	The number of detected annunciator is stored.	S (Instruction execution)
SD64 to SD79	Table of detected annunciator numbers	Detected annunciator number	• Numbers of activated annunciators are sequentially registered into SD64 to SD79, each time an annunciator (F) is turned on by the SET F instruction. • Number of the annunciator deactivated by the RST F instruction is removed from registers SD64 to SD79, and the numbers of annunciators which were lined up behind the removed one move forward one by one. When the number of detected annunciators has already reached 16, the number of a newly detected annunciator is not stored into SD64 to SD79. For details on the annunciator, refer to the annunciator (F) (☞ Page 315 Annunciator (F))	S (Instruction execution)
SD80	Detailed information 1 information category	Detailed information 1 information category code	• Detailed information 1 information category code is stored.  b0 to b7: Information category code b8 to b15: Unused (fixed to 0) • The following codes are stored into the information category code. 0: N/A 1: program position information 2: drive number and file name 4: Parameter information 5: System configuration information 6: number of times information 7: time information 24: Failure information 27: System switching information 34: Continuous RUN prevention setting in TEST MODE 37: Data type (tracking transfer) information 38: Tracking transfer trigger information 46: CPU module data backup/restoration folder information	S (Error)

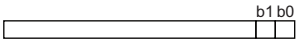
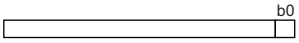
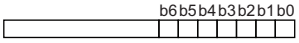
A

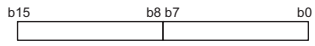
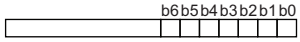
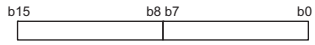
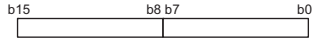
No.	Name	Data stored	Details	Set by (setting timing)
SD81 to SD111	Detailed information 1	Detailed information 1	• Detailed information 1 corresponding to the error code (SD0) is stored. • There are twelve types of information to be stored as shown in the following figures: • The type of the detailed information 1 can be obtained using SD80 (the value of the "Detailed information 1 information category code" stored in SD80 corresponds to the following figures (1), (2), (4) to (7), (24), (27), (34), (37) and (38)). (1) Error location information ■SD81: With or without specification  b0: Argument number b1: SFC block number b2: SFC step number b3: SFC transition number b4: Step number b5: FB number b6: File name ■SD82: Argument number (stored in the range from 1) ■SD83: SFC block number ■SD84 to SD85: SFC step number ■SD86 to SD87: SFC transition number ■SD88 to SD89: Step number ■SD90: FB number ■SD91 to SD98: File name (first eight characters of Unicode string) (2) Drive number and file name ■SD81: With or without specification  b0: Drive number b1: File name ■SD82: Drive number ■SD83 to SD90: File name (first eight characters of Unicode string)	S (Error)
SD81 to SD111	Detailed information 1	Detailed information 1	(4) Parameter information ■SD81: With or without specification  b0: Parameter type b1: Parameter storage location b2: I/O number b3: Parameter number b4: Network number b5: Station number b6: System information ■SD82  • b0 to b7: Parameter type (stored in the following value) 1: System parameter 2: CPU parameter 3: Module parameter 4: Module extension parameter 5: Memory card parameter 6: Safety system parameter 7: Safety CPU parameter 8: Safety module parameter • b8 to b15: Parameter storage location (2: SD memory card, 4: Data memory) ■SD83: I/O number 0xFFFF is stored when no I/O number is assigned. ■SD84: Parameter number ■SD85: Network number ■SD86: Station number 0 to 120 (0 is stored for the master station.) ■SD87 to SD97: System information	S (Error)

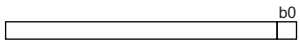
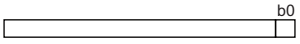
No.	Name	Data stored	Details	Set by (setting timing)
SD81 to SD111	Detailed information 1	Detailed information 1	(5) System configuration information ■SD81: With or without specification  b0: I/O number b1: Slot number b2: Base number b3: Power supply number b4: CPU number b5: Network number b6: Station number ■SD82: I/O number ■SD83  • b0 to b7: Slot number (0 to 11) • b8 to b15: Base number (0: Main base unit, 1 to 7: Extension base unit level 1 to level 7, 8: Higher than level 7) ■SD84  • b0 to b7: Power supply number (1 to 2: Power supply 1 to 2) • b8 to b15: CPU number: (1 to 4: CPU 1 to 4) ■SD85: Network number ■SD86: Station number 0 to 120 (0 is stored for the master station.) (6) Number of times information When there is no specification, 0 is set to each SD. ■SD81: With or without specification  b0: Number of times (set value) b1: Number of times (actual measurement value) ■SD82 to SD83: Number of times (set value) • SD82: Lower word of the number of times (set value) • SD83: Upper word of the number of times (set value) ■SD84 to SD85: Number of times (actual measurement value) • SD84: Lower word of the number of times (actual measurement value) • SD85: Upper word of the number of times (actual measurement value) (7) Time information When there is no specification, 0 is set to each SD. ■SD91: With or without specification  b0: Time (value set) (ms) b1: Time (value set) (μs) b2: Time (actual measurement value) (ms) b3: Time (actual measurement value) (μs)	S (Error)

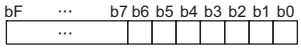
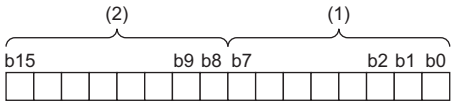
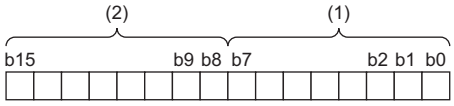
No.	Name	Data stored	Details	Set by (setting timing)
SD81 to SD111	Detailed information 1	Detailed information 1	■SD92: Time (value set) (ms) ■SD93: Time (value set) (μs) ■SD94: Time (actual measurement value) (ms) ■SD95: Time (actual measurement value) (μs) (24) Failure information Failure information is system information. (27): System switching information ■SD81: With or without specification  b0: System switching cause b1: System switching instruction ID number b2: Cause of system switching failure b3: Control system/standby system transition ■SD82: System switching cause 1: Power-off, reset, hardware failure 2: Stop error 3: System switching request from a network module 16: System switching request by using the SP.CONTSW instruction 17: System switching request using an engineering tool ■SD83: System switching instruction ID number ■SD84: Cause of system switching failure 1: Tracking communications disabled 2: Tracking communication timeout 3: Stop error of the standby system 4: Operating status mismatch between both systems 5: Memory copy being executed 6: Online change being executed 7: A failure of a network module detected on the standby system 8: System switching being executed 9: A redundant function module being changed online 10: System switching disabled on the standby system by using the DCONTSW instruction ■SD85: Control system/standby system transition 1: Control system → Standby system 2: Standby system → Control system (34) TEST MODE continuous RUN prevention setting ■SD81: With or without specification  b0: Continuous RUN time value (second) ■SD82 to SD83: Continuous RUN time value (second) • SD82: Lower word of the continuous RUN time value (second) • SD83: Upper word of the continuous RUN time value (second) (37) Data type (tracking transfer) information ■SD81: With or without specification  b0: Data type ■SD82: Data type  b0: Device data b1: Label data b2: Signal flow b3: PID control instruction information b4: SFC information b5: System switching request b6: Operation mode change request b15: System data 0 is stored when each data is not sent, and 1 is stored when each data is being sent in each bit.	S (Error)

No.	Name	Data stored	Details	Set by (setting timing)
SD81 to SD111	Detailed information 1	Detailed information 1	(38) Tracking transfer trigger information ■SD81: With or without specification  b0: Block No. 1 to 8 b1: Block No. 9 to 16 b2: Block No. 17 to 24 b3: Block No. 25 to 32 b4: Block No. 33 to 40 b5: Block No. 41 to 48 b6: Block No. 49 to 56 b7: Block No. 57 to 64 ■SD82 to SD85: Block number  (46) CPU module data backup/restoration folder information ■SD81: With or without specification  b0: Folder specification b1: Date folder b2: Number folder ■SD82: Folder specification 0: Specification allowed 1: Specification not allowed ■SD83 to SD84: Date folder (yyyymmdd) Date of the folder is stored in BCD code. (yyyy: 0 to 9999, mm: 1 to 12, dd: 1 to 31) FFFFFFFFH is stored when a folder cannot be specified. • SD83: Lower word of the date folder (yyyymmdd) • SD84: Upper word of the date folder (yyyymmdd) ■SD85: Number folder 0 to 32767 (FFFFH is stored when a folder cannot be specified.)	S (Error)
SD112	Detailed information 2 information category	Detailed information 2 information category code	Detailed information 2 information category code is stored.  • b0 to b7: Information category code • b8 to b15: Unused (fixed to 0) The following codes are stored into the information category code. • 0: N/A • 2: Drive number and file name • 3: Annunciator number • 4: Parameter information • 5: System configuration information • 25: Process control instruction processing information • 28: Program error information • 29: Other station error information (CC-Link IE Field)	S (Error)

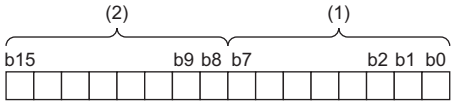
No.	Name	Data stored	Details	Set by (setting timing)
SD113 to SD143	Detailed information 2	Detailed information 2	• Detailed information 2 corresponding to the error code (SD0) is stored. • There are seven types of information to be stored as shown in the following figures: • The type of information can be checked in SD112. (The value of the "Detailed information 2 information category code" stored in SD112 corresponds to the following (2) to (5), (25), (28) and (29).) (2) Drive number and file name ■SD113: With or without specification  b0: Drive number b1: File name ■SD114: Drive number ■SD115 to SD122: File name (first eight characters of Unicode string) (3) Annunciator number ■SD113: With or without specification  b0: Annunciator number ■SD114: Annunciator number (4) Parameter information ■SD113: With or without specification  b0: Parameter type b1: Parameter storage location b2: I/O number b3: Parameter number b4: Network number b5: Station number b6: System information	S (Error)

No.	Name	Data stored	Details	Set by (setting timing)
SD113 to SD143	Detailed information 2	Detailed information 2	■SD114  • b0 to b7: Parameter type (stored in the following value) 1: System parameter 2: CPU parameter 3: Module parameter 4: Module extension parameter 5: Memory card parameter 6: Safety system parameter 7: Safety CPU parameter 8: Safety module parameter • b8 to b15: Parameter storage location (2: SD memory card, 4: Data memory) ■SD115: I/O number FFFFH is stored when no I/O number is assigned. ■SD116: Parameter number ■SD117: Network number ■SD118: Station number 0 to 120 (0 is stored for the master station.) ■SD119 to SD129: System information (5) System configuration information ■SD113: With or without specification  b0: I/O number b1: Slot number b2: Base number b3: Power supply number b4: CPU number b5: Network number b6: Station number ■SD114: I/O number ■SD115  • b0 to b7: Slot number (0 to 11) • b8 to b15: Base number (0: Main base unit, 1 to 7: Extension base unit level 1 to level 7, 8: Higher than level 7) ■SD116  • b0 to b7: Power supply number (1 to 2: Power supply 1 to 2) • b8 to b15: CPU number: (1 to 4: CPU 1 to 4) ■SD117: Network number ■SD118: Station number 0 to 120 (0 is stored for the master station.)	S (Error)

No.	Name	Data stored	Details	Set by (setting timing)
SD113 to SD143	Detailed information 2	Detailed information 2	(25) Process control instruction processing information ■SD113: With or without specification  b0: Process control instruction processing information ■SD114: Processing description The following are processing blocks stored in SD114 (Processing details). 1: — (Instruction with no processing block) 2: Range check 3: Input limiter 4: Engineering value inverse transformation 5: Digital filter 6: Input addition 7: Variation rate & upper/lower limiter 8: Reset windup 9: Output conversion 10: Output ON time conversion 11: Variation rate check 12: Integrated value calculation 13: Control cycle determination 14: SV setting 15: Tracking 16: Gain (Kp) operation 17: PID operation 18: Deviation check 19: PID operation 1 (operation processing of Bn or Cn) 20: PID operation 2 (operation processing of Dn) 21: PID operation 3 (operation processing of ΔMV) 22: PIDP operation 23: Operating time monitor 24: SPI operation 25: IPD operation 26: BPI operation 27: Engineering value transformation 28: Variation rate limiter 29: Ratio calculation 30: Upper/lower limit check 31: Loop stop 32: MV correction 33: Two-position (on/off) control 34: Three-position (on/off) control 35: Operation constant check 36: SV count-up 37: MVPGS operation 38: Output 39: Input check 40: Timeout determination 41: Step manipulated value set 42: Sampling cycle determination 43: Response waveform observation 44: Identification processing 45: PID constants calculation (28) Program error information ■SD113: With or without specification  b0: Program error code	S (Error)

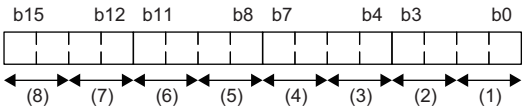
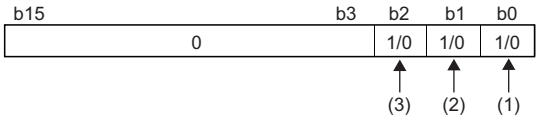
No.	Name	Data stored	Details	Set by (setting timing)
SD113 to SD143	Detailed information 2	Detailed information 2	■SD114: Program error code (29) Error information of other stations (CC-Link IE Field) ■SD81: With or without specification  b0: Error classification b1: Error code b2: Date (yyyymmdd) b3: Time (hhmmss) b4: Day of the week b5: Detailed error information 1 b6: Detailed error information 2 b7 to bF: Detailed error information 3 to detailed error information 10 ■SD82: Error classification ■SD83: Error code ■SD84 to SD85: Date (yyyymmdd) • SD84: Lower word of date (yyyymmdd) • SD85: Upper word of date (yyyymmdd) ■SD86 to SD87: Time (hhmmss) • SD86: Lower word of time (hhmmss) • SD87: Upper word of time (hhmmss) ■SD88: Day of the week ■SD89: Detailed error information 1 ■SD90: Detailed error information 2 ■SD91 to SD98: Detailed error information 3 to 10	S (Error)
SD150	Power-off/power supply voltage drop detection status	Power-off/power supply voltage drop detection status (bit pattern) 0: Power-on/ normal voltage 1: Power-off/ voltage drop detected/no power supply module	• For the redundant power supply base unit, power supply module status (power is shut off, power supply voltage drop (not including a momentary power failure) is detected, or an empty slot for the power-supply module exists) is stored using the following bit pattern. • In a multiple CPU system, the status is stored only to the CPU module No.1.  (1) Powered-off/power supply voltage dropped (power supply module 1) b0:Main base unit b1 to b7: 1st to 7th level of the extension base unit (2) Powered-off/power supply voltage dropped (power supply module 2) b8: Main base unit b9 to b15: 1st to 7th level of the extension base unit	S (Status change)
SD151	Power supply failure detection status	Power supply failure detection status (bit pattern) 0: Not detected/ power-off/no power supply module 1: Detected	• For the redundant power supply base unit, failure detection status of the power supply module is stored using the following bit pattern. • Bits corresponding to power-off or empty slots for the power supply module turn off. • In a multiple CPU system, the status is stored only to the CPU module No.1.  (1) Failure detection status of the power supply module 1 b0:Main base unit b1 to b7: 1st to 7th level of the extension base unit (2) Failure detection status of the power supply module 2 b8: Main base unit b9 to b15: 1st to 7th level of the extension base unit	S (Status change)
SD152	Momentary power failure detection count (power supply module 1)	Momentary power failure detection count for power supply module 1	• This register counts the number of momentary power failures. • This register monitors the status of the power supply module mounted on the main base unit and counts the number of momentary power failures. • When the CPU module starts up, the counters of the both power supplies are cleared to 0. • When one of the two power supply is powered off, the corresponding counter to the one powered off is cleared to 0. • The counter is incremented by one upon one momentary power failure on each power supply. A counting cycle from 0 to 65535 to 0 is repeated. • In a multiple CPU system, the status is stored only to the CPU module No.1.	S (Status change)
SD153	Momentary power failure detection count (power supply module 2)	Momentary power failure detection count for power supply module 2		S (Status change)

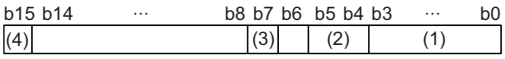
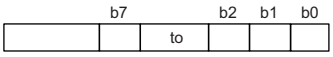
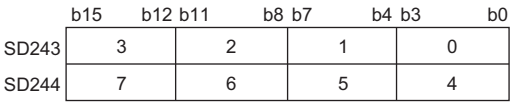
A

No.	Name	Data stored	Details	Set by (setting timing)
SD154	Details of the invalid power supply module	Details of the invalid power supply module (bit pattern) 0: Valid/power-off/no power supply module 1: Invalid	• For the redundant power supply base unit, when an invalid power supply module exists, the status is stored using the following bit pattern. • Bits corresponding to power-off or empty slots for the power supply module turn off. • In a multiple CPU system, the status is stored only to the CPU module No.1.  (1) Details of the invalid power supply module 1 b0:Main base unit b1 to b7: 1st to 7th level of the extension base unit (2) Details of the invalid power supply module 2 b8: Main base unit b9 to b15: 1st to 7th level of the extension base unit	S (Status change)

System information

The following is the list of special register areas relating to the system information.

No.	Name	Data stored	Details	Set by (setting timing)
SD160	Firmware version	Firmware version	The firmware version is stored.	S (Initial)
SD200	Status of switch	Status of CPU switch	The switch status of the CPU module is stored as follows: 0: RUN, 1: STOP	S (Switch change)
SD201	LED status	CPU-LED state	This register stores the information that indicates LED status (0: off, 1: on, 2: flashing) of the CPU module in the following bit patterns.  (1) READY (2) ERROR (3) PROGRAM RUN (4) USER (5) BATTERY (6) CARD READY (7) CARD ACCESS (8) FUNCTION	S (Status change)
SD203	Operating status of CPU	Operating status of CPU	The operating status of the CPU module is stored as follows: 0: RUN, 2: STOP, 3: PAUSE	S (Every END)
SD205*1	Safety operation mode	Safety operation mode	The safety operation mode is stored. (The applicable bit turns on.)  (1) TEST MODE (2) SAFETY MODE (3) SAFETY MODE (wait-for-restart)	S (Status change)
SD206*1	Pair version	Pair version	The Safety CPU pair version is stored in decimal notation.	S (Initial)
SD210	Clock data	Clock data (year)	The year value (four digits) of the clock data is stored as a BIN code.	S/U (Request)
SD211		Clock data (month)	The month value of the clock data is stored as a BIN code.	S/U (Request)
SD212		Clock data (day)	The day value of the clock data is stored as a BIN code.	S/U (Request)
SD213		Clock data (hour)	The hour value of the clock data is stored as a BIN code.	S/U (Request)
SD214		Clock data (minute)	The minute value of the clock data is stored as a BIN code.	S/U (Request)
SD215		Clock data (second)	The second value of the clock data is stored as a BIN code.	S/U (Request)
SD216		Clock data (day of the week)	The day of the week value of the clock data is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)	S/U (Request)
SD218	Time zone setting value	Time zone (in minutes)	The time zone setting value specified in the parameter is stored in increments of minutes. Example: when the setting value of the time zone is "UTC+9," the result of the following equation is stored into SD218: 9 × 60 (minutes) = 540.	S (Initial)

No.	Name	Data stored	Details	Set by (setting timing)
SD228	Multiple CPU system information	Number of CPU modules	The number of CPU modules which constitute a multiple CPU system is stored (one to four, including those reserved).	S (Initial)
SD229		CPU module number in multiple CPU system	The number of this CPU No. is stored when a multiple CPU system is configured.	S (Initial)
SD230		CPU No.1 operating status	The operation information for each CPU No. is stored (the amount of stored information depending on the number of CPU modules indicated in SD228 is stored).  (1) The operating status is stored in b0 to b3. • 0: RUN • 2: STOP • 3: PAUSE • 4: INITIALIZE • FH: RESET (2) The classification is stored in b4 and b5. However, minor or moderate errors will be those set in the CPU parameter RAS settings, system parameter I/O assignment settings, and multiple CPU settings. • 0: Normal • 1: Minor error (error code: 1000H to 1FFFH) • 2: Moderate error (error code: 2000H to 3BFFH) • 3: Major error (error code: 3C00H to 3FFFH) (3) The stop error flag is stored in b7. • 0: No stop error • 1: Stop error (4) The CPU module mounting status is stored in b15. • 0: Not mounted • 1: Mounted	S (During END/ Error)
SD231		CPU No.2 operating status		
SD232		CPU No.3 operating status		
SD233		CPU No.4 operating status		
SD241	Extension stage number	0: base unit only 1 to 7: number of extension base units	The maximum number of implemented extension base units is stored.	S (Initial)
SD242	Identification for whether or not Q series module can be mounted	Identification of the base type 0: Cannot be mounted (There is no base unit on which the Q series module can be mounted.) 1: Can be mounted (There is a base unit on which the Q series module can be mounted.)	Identify whether or not Q series module can be mounted. When no SD memory card is inserted, the value is fixed to 0.  • b0: Main base unit (fixed to 0) • b1: Extension base unit 1 (Q series modules can be mounted.) • b2: Extension base unit 2 (Q series modules can be mounted.) • b3 to b7: Extension base unit 3 to 7 (Q series modules can be mounted.) • b8 to b15: Fixed to 0	S (Initial)
SD243	Number of base slots	Number of base slots	The number of slots of the base unit, which is specified in the Base / power supply / extension cable setting of system parameters, is stored. When the number of slots of the base unit is not specified by the system parameter, that of the mounted base unit is stored.  0: Main 1 to 7: Extension 1 to 7	S (Initial)
SD244				
SD250	Loaded maximum I/O	Loaded maximum I/O number	The value obtained by dividing the last I/O number for an implemented unit plus one by 16 is stored. Example 1: Last I/O number 010FH • SD250 = 0011H Example 2: Last I/O number 0FFFH • SD250 = 0100H	S (Initial)

No.	Name	Data stored	Details	Set by (setting timing)
SD260	Number of points assigned to bit devices	X (L)	The number of points assigned to the X device is stored in 32 bits.	S (Initial)
SD261		X (H)		
SD262		Y (L)	The number of points assigned to the Y device is stored in 32 bits.	S (Initial)
SD263		Y (H)		
SD264		M (L)	• The number of points assigned to the M device is stored in 32 bits. • The number of points is stored even when the number of points assigned to M is 32K or less.	S (Initial)
SD265		M (H)		
SD266		B (L)	• The number of points assigned to the B device is stored in 32 bits. • The number of points is stored even when the number of points assigned to B is 32K or less.	S (Initial)
SD267		B (H)		
SD268		SB (L)	• The number of points assigned to the SB device is stored in 32 bits. • The number of points is stored even when the number of points assigned to SB is 32K or less.	S (Initial)
SD269		SB (H)		
SD270		F (L)	The number of points assigned to the F device is stored in 32 bits.	S (Initial)
SD271		F (H)		
SD272		V (L)	The number of points assigned to the V device is stored in 32 bits.	S (Initial)
SD273		V (H)		
SD274		L (L)	The number of points assigned to the L device is stored in 32 bits.	S (Initial)
SD275		L (H)		
SD276		S (L)	The number of points assigned to the S device is stored in 32 bits.	S (Initial)
SD277		S (H)		
SD280	Number of points assigned to word devices	D (L)	• The number of points assigned to the D device is stored in 32 bits. • The number of points is stored even when the number of points assigned to D is 32K or less.	S (Initial)
SD281		D (H)		
SD282		W (L)	• The number of points assigned to the W device is stored in 32 bits. • The number of points is stored even when the number of points assigned to W is 32K or less.	S (Initial)
SD283		W (H)		
SD284		SW (L)	• The number of points assigned to the SW device is stored in 32 bits. • The number of points is stored even when the number of points assigned to SW is 32K or less.	S (Initial)
SD285		SW (H)		
SD288	Number of points assigned to timer/ counter devices	T (L)	• The number of points assigned to the T device is stored in 32 bits. • The number of points is stored even when the number of points assigned to T is 32K or less.	S (Initial)
SD289		T (H)		
SD290		ST (L)	• The number of points assigned to the ST device is stored in 32 bits. • The number of points is stored even when the number of points assigned to ST is 32K or less.	S (Initial)
SD291		ST (H)		
SD292		C (L)	• The number of points assigned to the C device is stored in 32 bits. • The number of points is stored even when the number of points assigned to C is 32K or less.	S (Initial)
SD293		C (H)		
SD294		LT (L)	• The number of points assigned to the LT device is stored in 32 bits. • The number of points is stored even when the number of points assigned to LT is 32K or less.	S (Initial)
SD295		LT (H)		
SD296		LST (L)	• The number of points assigned to the LST device is stored in 32 bits. • The number of points is stored even when the number of points assigned to LST is 32K or less.	S (Initial)
SD297		LST (H)		
SD298		LC (L)	• The number of points assigned to the LC device is stored in 32 bits. • The number of points is stored even when the number of points assigned to LC is 32K or less.	S (Initial)
SD299		LC (H)		
SD300	Number of points assigned to the index register	Z	The number of points assigned to the Z device is stored.	S (Initial)
SD302	Number of points assigned to the long index register	LZ	The number of points assigned to the LZ device is stored.	S (Initial)
SD306	Number of points assigned to the file register	ZR (L)	• The number of points assigned to the ZR device is stored in 32 bits. • The number of points is stored even when the number of points assigned to ZR is 32K or less.	S (Initial)
SD307		ZR (H)		

No.	Name	Data stored	Details	Set by (setting timing)
SD308	Number of points assigned to refresh devices	RD (L)	• The number of points assigned to the RD device is stored in 32 bits. • The number of points is stored even when the number of points assigned to RD is 32K or less.	S (Initial)
SD309		RD (H)		
SD312	File register block number	File register block number	The block number of the file register currently selected is stored.	S (Status change)
SD329	Online change (inactive SFC block) target block number	SFC block number	• A target SFC block number is stored while the online change (inactive SFC block) is being executed (SM329=ON). • FFFFH is stored when the online change (inactive SFC block) is not executed.	S (Status change)

*1 Safety CPU only

System clock

The following is the list of special register areas relating to the system clock.

No.	Name	Data stored	Details	Set by (setting timing)
SD412	One second counter	The number of counts that is counted once per second.	• The value in this register increments by one for each second after the CPU module enters in RUN mode. • A counting cycle from 0 to 65535 to 0 is repeated.	S (Status change)
SD414	2n second clock setting	Unit setting for 2n second clock	• n value of the 2n second clock is stored (Default: 30). • Configurable range is -32768 to 32767 (0 to FFFFH).	U
SD415	2n ms clock setting	Unit setting for 2n ms clock	• The n value for the 2n ms clock is stored. (Default: 30). • Configurable range is -32768 to 32767 (0 to FFFFH).	U
SD420	Scan counter	The number of counts that is counted once for each scan.	• The value in this register increments by one for each scan after the CPU module enters in RUN mode (however, the count is skipped for scans by the initial execution type program). • A counting cycle from 0 to 65535 to 0 is repeated.	S (Every END)

Fixed scan function information

The following is the list of special register areas relating to the fixed scan function information.

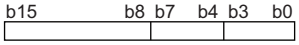
No.	Name	Data stored	Details	Set by (setting timing)
SD480	Number of cycle overrun events for inter-module synchronization cycle program (I44)	0: No cycle overrun event 1 to 65535: Accumulated number of cycle overrun events	The number of events in which the inter-module synchronous interrupt program (I44) has not been completed within the inter-module synchronization cycle or the program cannot be executed due to various reasons, such as execution of a higher-priority interrupt program and interrupt disabling by the instruction execution is stored. When the count exceeds 65535, it returns to 0 and starts a new cycle. The number of cycle overrun events is counted regardless of the setting content for the error check setting of the RAS setting (execution check of the inter-module synchronous interrupt (I44)).	S (Status change)
SD481	Number of cycle overrun events for multiple CPU synchronization program (I45)	0: No cycle overrun event 1 to 65535: Accumulated number of cycle overrun events	The number of events in which the multiple CPU synchronization program (I45) has not been completed within the fixed scan communication cycle or the program cannot be executed due to various reasons, such as execution of a higher-priority interrupt program and interrupt disabling by the instruction execution is stored. When the count exceeds 65535, it returns to 0 and starts a new cycle. The number of cycle overrun events is counted regardless of the setting content for the error check setting of the RAS setting (execution check of the multiple CPU synchronization program (I45)).	S (Status change)
SD484	Number of execution section excess errors for multiple CPU synchronization interrupt program	0: No error (Normal) 1 to 65535: Accumulated number of errors	The number of events in which the program is executed exceeding the program execution section within the specified multiple CPU synchronization cycle is stored. When the count exceeds 65535, it returns to 0 and starts a new cycle. Note that the number of error occurrences is counted regardless of the CPU module operation setting for error detections within the RAS setting of the CPU parameter.	S (Status change)
SD500	Execution program number	Execution program number	The program number which is currently executed is stored as a BIN value.	S (Status change)
SD518	Initial scan time	Initial scan time (unit: ms)	• The initial scan time is stored into SD518 and SD519 (The time is measured in microseconds (μs).) SD518: stores a value in the ms place (storage range: 0 to 65535) SD519: stores a value in the μs place (storage range: 0 to 999) • This register is cleared to 0 when the mode transfers from STOP to RUN mode.	S (Every END)
SD519		Initial scan time (unit: μs)		

A

No.	Name	Data stored	Details	Set by (setting timing)
SD520	Current scan time	Current scan time (unit: ms)	• The current scan time is stored into SD520 and SD521 (it is measured in increments of μs). SD520: stores a value in the ms place (storage range: 0 to 65535) SD521: stores a value in the μs place (storage range: 0 to 999) Example: If the current scan time is 23.6ms, the following values are stored: SD520 = 23 SD521 = 600 • This register is cleared to 0 when the mode transfers from STOP to RUN mode. • In a redundant system, when the operating status of the CPU module is RUN state, the counter is cleared to 0 at the system switching.	S (Every END)
SD521		Current scan time (unit: μs)		
SD522	Minimum scan time	Minimum scan time (unit: ms)	• The minimum value of the scan time other than one for the initial execution program is stored into SD522 and SD523 (it is measured in increments of μs). SD522: stores a value in the ms place (storage range: 0 to 65535) SD523: stores a value in the μs place (storage range: 0 to 999) • This register is cleared to 0 when the mode transfers from STOP to RUN mode. • In a redundant system, when the operating status of the CPU module is RUN state, the counter is cleared to 0 at the system switching.	S (Every END)
SD523		Minimum scan time (unit: μs)		
SD524	Maximum scan time	Maximum scan time (unit: ms)	• The maximum value of the scan time other than one for the initial execution program is stored into SD524 and SD525 (it is measured in increments of μs). SD524: stores a value in the ms place (storage range: 0 to 65535) SD525: stores a value in the μs place (storage range: 0 to 999) • This register is cleared to 0 when the mode transfers from STOP to RUN mode. • In a redundant system, when the operating status of the CPU module is RUN state, the counter is cleared to 0 at the system switching.	S (Every END)
SD525		Maximum scan time (unit: μs)		
SD526	END processing time	END processing time (unit: ms)	• The time period from completion of a scan program until start of the next scan is stored into SD526 to SD527 (it is measured in increments of μs). SD526: stores a value in the ms place (storage range: 0 to 65535) SD527: stores a value in the μs place (storage range: 0 to 999) • This register is cleared to 0 when the mode transfers from STOP to RUN mode. • In a redundant system, when the operating status of the CPU module is RUN state, the counter is cleared to 0 at the system switching.	S (Every END)
SD527		END processing time (unit: μs)		
SD528	Constant scan wait time	Constant scan wait time (unit: ms)	• The waiting time specified in the constant scan setting process is stored into SD528 and SD529 (it is measured in increments of μs). SD528: stores a value in the ms place (storage range: 0 to 65535) SD529: stores a value in the μs place (storage range: 0 to 999) • This register is cleared to 0 when the mode transfers from STOP to RUN mode. • In a redundant system, when the operating status of the CPU module is RUN state, the counter is cleared to 0 at the system switching.	S (Every END)
SD529		Constant scan wait time (unit: μs)		
SD530	Scan program execution time	Scan program execution time (unit: ms)	• The execution time of the scan program for one scan is stored into SD530 and SD531 (it is measured in increments of μs). SD530: stores a value in the ms place (storage range: 0 to 65535) SD531: stores a value in the μs place (storage range: 0 to 999) • This register is cleared to 0 when the mode transfers from STOP to RUN mode. • In a redundant system, when the operating status of the CPU module is RUN state, the counter is cleared to 0 at the system switching.	S (Every END)
SD531		Scan program execution time (unit: μs)		

Drive information

The following is the list of special register areas relating to the drive information.

No.	Name	Data stored	Details	Set by (setting timing)
SD600	Memory card mounting status	SD memory card type	This register indicates the type of mounted SD memory cards.  • b0 to b3: Fixed to 0 • b4 to b7 0: Does not exist 4: SD memory card b8 to b15: Fixed to 0	S (Initial, card insertion/ removal)
SD604	SD memory card (drive 2) usage status	SD memory card (drive 2) usage status	Usage status of the SD memory card is stored using the following bit pattern. (On indicates being used.) b0: Event history b1: Module extension parameter*1 b2: Label communication data b3 to b15: Not used	S (Status change)
SD606	SD memory card (drive 2) capacity	SD memory card (drive 2) capacity: the lower digits (unit: K bytes)	The capacity of the SD memory card is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial, card insertion/ removal)
SD607		SD memory card (drive 2) capacity: the higher digits (unit: K bytes)	The capacity of the SD memory card is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial, card insertion/ removal)
SD610	SD memory card (drive 2) free space	SD memory card (drive 2) free space: the lower digits (unit: K bytes)	The amount of free space of the SD memory card is stored in increments of 1K byte.	S (at change)
SD611		SD memory card (drive 2) free space: the higher digits (unit: K bytes)	The amount of free space of the SD memory card is stored in increments of 1K byte.	S (at change)
SD614	Device/label memory (drive 3) usage status	Device/label memory (drive 3) usage status	Usage status of the device/label memory is stored using the following bit pattern. (On indicates being used.) b0: File register b1 to b15: Not used	S (Status change)
SD616	Device/label memory (drive 3) capacity	Device/label memory (drive 3) capacity: the lower digits (unit: K bytes)	The capacity of the device/label memory is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)
SD617		Device/label memory (drive 3) capacity: the higher digits (unit: K bytes)	The capacity of the device/label memory is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)
SD618	Device/label memory (file storage area) capacity	Device/label memory (file storage area) (drive 3) capacity: the lower digits (unit: K bytes)	The capacity of the device/label memory (file storage area) is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)
SD619		Device/label memory (file storage area) (drive 3) capacity: the higher digits (unit: K bytes)	The capacity of the device/label memory (file storage area) is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)
SD620	Data memory (drive 4) usage status	Data memory (drive 4) usage status	Usage status of the data memory is stored using the following bit pattern. (On indicates being used.) b0: Event history b1: Module extension parameter*1 b2: Label communication data b3 to b15: Not used	S (Status change)
SD622	Data memory (drive 4) capacity	Data memory (drive 4) capacity: the lower digits (unit: K bytes)	The capacity of the data memory is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)
SD623		Data memory (drive 4) capacity: the higher digits (unit: K bytes)	The capacity of the data memory is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)

No.	Name	Data stored	Details	Set by (setting timing)
SD626	Extended SRAM cassette capacity identification information	Capacity identification information of the Extended SRAM cassette	Capacity identification information of the Extended SRAM cassette is stored. Unmounted: 0, 1M: 1, 2M: 2, 4M: 3, 8M: 4, 16M: 5	S (Initial)
SD629	Program memory write (transfer) status	Write (transfer) status display (percent)	This register displays write (transfer) status to the program memory in percentage (0 to 100%). The initial value is "0". Upon completion of writing, this register is set to "100". It is set to "0" at the time when the write command is issued.	S (Writing)
SD630 SD631	Program memory write count index	Index of the number of write operations up to now	• This register indicates the index value for the number of write operations to the program memory up to now (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value is continued to be counted even when it exceeds 100000). If the index value exceeds 100000, the CPU module must be replaced.	S (Writing)
SD633	Data memory write (transfer) status	Write (transfer) status display (percent)	This register displays write (transfer) status to the data memory in percentage. (0 to 100%). The initial value is "0". Upon completion of writing, this register is set to "100". It is set to "0" at the time when the write command is issued.*2	S (Writing)
SD634 SD635	Index for the number of data memory write operations	Index of the number of write operations up to now	• This register indicates the index value for the number of write operations to the data memory up to now (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value is continued to be counted even when it exceeds 100000). If the index value exceeds 100000, the CPU module must be replaced.	S (Writing)
SD638 SD639	Index for the number of system memory write operations	Index of the number of write operations up to now	• This register indicates the index value for the number of write operations to the system memory (Flash ROM)*3 up to now. (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value is continued to be counted even when it exceeds 100000). If the index value exceeds 100000, the CPU module must be replaced.	S (Writing)
SD640	Internal buffer empty area usage status	Internal buffer empty area usage status (usage status of the internal buffer for functions where the internal buffer capacity is not set in the internal buffer capacity setting of the CPU parameters)	This register stores the following in a bit pattern: usage status of the internal buffer for functions where the internal buffer capacity is not set in the internal buffer capacity setting of the CPU parameters. (On indicates being used.) b0: Used for realtime monitor b1 to b15: Not used	S (Status change)
SD642	Internal buffer capacity	Internal buffer capacity Low-order (in K bytes)	The capacity of the internal buffer is stored in K bytes.	S (Initial)
SD643		Internal buffer capacity High-order (in K bytes)	The capacity of the internal buffer is stored in K bytes.	S (Initial)
SD644	Internal buffer free area space	Internal buffer free area space Low-order (in K bytes)	This register stores the following in K bytes: internal buffer free space that is not set in the internal buffer capacity setting of the CPU parameters.	S (Status change)
SD655		Internal buffer free area space High-order (in K bytes)	This register stores the following in K bytes: internal buffer free space that is not set in the internal buffer capacity setting of the CPU parameters.	S (Status change)

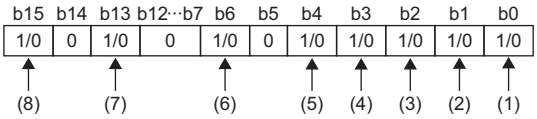
*1 It is stored when used for the Ethernet function of CPU module.

*2 When the save destination for the event history is the data memory: the event history will be stored at the timing of powering off and on and resetting the CPU module. Accordingly, since the data will be written into the data memory, "100" is stored. (When the save destination for the event history is the SD memory, the initial value remains "0".)

*3 This is the memory used by the system when the CPU module is executing functions.

Instruction related

The following is the list of special register areas relating to the instruction-related items.

No.	Name	Data stored	Details	Set by (setting timing)
SD757	Current interrupt priority	Current interrupt priority	The priority for the interrupt of the interrupt program currently executed is stored. 1 to 8:Priority for the interrupt pointer of the interrupt program currently being executed 0: No interrupt operation (default)	S (Status change)
SD758	Interrupt disabling for each priority setting value	Interrupt disabling for each priority setting value	The interrupt priority of the interrupt program that the interrupt is disabled by using the DI instructions (Disabling interrupt programs/Disabling interrupt programs with specified priority or lower) and the EI instruction (Enabling interrupt programs) is stored in this register. Note that safety programs are unaffected by SD758, and are executed based on the safety cycle time. 1: Interrupt programs with priority 1 or lower (all priority levels) disabled (default) 2: Interrupt programs with priority 2 or lower disabled 3: Interrupt programs with priority 3 or lower disabled 4: Interrupt programs with priority 4 or lower disabled 5: Interrupt programs with priority 5 or lower disabled 6: Interrupt programs with priority 6 or lower disabled 7: Interrupt programs with priority 7 or lower disabled 8: Interrupt programs with priority 8 or lower disabled 0: No interrupt program disabled (interrupt programs with any priority level enabled)	S (Status change)
SD760	Unicode text file faulty area in the DBIMPORT instruction	Row number of the Unicode text file (lower)	If an error in the Unicode text file is detected when the DBIMPORT(P) instruction is executed, the row number of the Unicode text file is stored. The value is stored when the error completion signal of the DBIMPORT(P) instruction is turned on and is cleared to 0 when the DBIMPORT(P) instruction is executed.	S (Status change)
SD761		Row number of the Unicode text file (upper)		S (Status change)
SD771	Specification of the number of write instruction executions to data memory	Specification of the number of write instruction executions to data memory	In this register, the maximum number of the data memory writing instruction (SP.DEVST) executions per day is specified. When the number of data memory writing instruction executions exceeds the value specified in this register, an error is generated. Setting range is from 1 to 32767. If a value is set outside the range, an error is generated during the data memory writing instruction execution.	U
SD775	Selection of refresh processing during the COM instruction execution	b0 to b4, b6, b13 0: No refresh execution 1: Refresh execution b15 0: Device/Label access service processing execution 1: No device/Label access service processing execution	In this register, whether or not each processing is executed during the COM instruction execution is selected (Default: 0). The specification on the SD775 is effective when SM775 is turned on.  (1) I/O refresh, I/O fetch from a group out of the multiple CPU system (2) Link refresh of the CC-Link module (3) Link refresh of the CC-Link IE Controller Network module and MELSECNET/H module (4) Intelligent function module refresh (5) Refresh with the CPU buffer memory of the multiple CPU system (END) (6) Link refresh of the CC-Link IE Field Network module (7) Link refresh of CC-Link IE Field Network Basic (8) Device/label access service processing (communications with an engineering tool, GOT, or other external devices)	U
SD792 and SD793	PID limit setting (for complete derivative)	0: Limit restriction applied 1: No limit restriction	The limit restriction for each PID loop is specified as follows: (for the PIDCONT instruction) b15 ... b1 b0 SD792 16 to 2 1 SD793 32 to 18 17 1 to 32: Loop 1 to 32	U

A

No.	Name	Data stored	Details	Set by (setting timing)
SD794 and SD795	PID limit setting (for incomplete derivative)	0: Limit restriction applied 1: No limit restriction	The limit restriction for each PID loop is specified as follows: (for the S.PIDCONT instruction) b15 ... b1 b0 SD794: 16 to 2 1 SD795: 32 to 18 17 1 to 32: Loop 1 to 32	U
SD796	Maximum number of blocks used for the multiple CPU dedicated instruction (for CPU No.1)	The maximum number of blocks to be used for the dedicated instruction Depending on the number of CPU modules which constitute a multiple CPU system, the range is as follows.*1 When constituting two modules: 2 to 599 When constituting three modules: 2 to 299 When constituting four modules: 2 to 199 (Default: 2).	• The maximum number of blocks used for the multiple CPU dedicated instruction is specified (for CPU No.1). • When executing the multiple CPU dedicated instruction on CPU No. 1, if the number of free blocks in the dedicated instruction transfer area is less than the setting value on this register, SM796 is turned on. • This value is used as interlock signal for the continuous executions of the multiple CPU dedicated instruction.	U
SD797	Maximum number of blocks setting used for the multiple CPU dedicated instruction (for CPU No.2)		• The maximum number of blocks used for the multiple CPU dedicated instruction is specified (for CPU No.2). • When executing the multiple CPU dedicated instruction on CPU No. 2, if the number of free blocks in the dedicated instruction transfer area is less than the setting value on this register, SM797 is turned on. • This value is used as interlock signal for the continuous executions of the multiple CPU dedicated instruction.	U
SD798	Maximum number of blocks setting used for the multiple CPU dedicated instruction (for CPU No.3)		• The maximum number of blocks used for the multiple CPU dedicated instruction is specified (for CPU No.3). • When executing the multiple CPU dedicated instruction on CPU No. 3, if the number of free blocks in the dedicated instruction transfer area is less than the setting value on this register, SM798 is turned on. • This value is used as interlock signal for the continuous executions of the multiple CPU dedicated instruction.	U
SD799	Maximum number of blocks setting used for the multiple CPU dedicated instruction (for CPU No.4)		• The maximum number of blocks used for the multiple CPU dedicated instruction is specified (for CPU No.4). • When executing the multiple CPU dedicated instruction on CPU No. 4, if the number of free blocks in the dedicated instruction transfer area is less than the setting value on this register, SM799 is turned on. • This value is used as interlock signal for the continuous executions of the multiple CPU dedicated instruction.	U
SD816*2 SD817*2	Basic period	Execution cycle	An execution cycle (unit: second) of process control instructions is set in real number.	U
SD818*2	Bumpless function availability setting for the S.PIDP instruction	0: Enabled 1: Disabled	The availability of the bumpless function for the S.PIDP instruction is set.	U
SD820*2 SD821*2	Dummy device	Dummy device	A dummy device used in process control instructions is set.	U

*1 When the value out of the range is specified, operation runs while its value is being regarded as max value of each range of multiple CPU system configuration.

*2 Only the Process CPU supports these special register areas.

Firmware update function

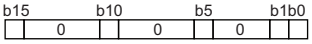
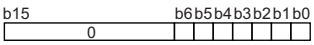
The following is the list of special register areas relating to the firmware update function. (Programmable controller CPU only)

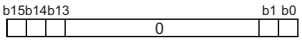
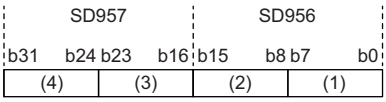
No.	Name	Data stored		Details	Set by (setting timing)
SD912	Latest firmware update information	History information	Execution time (year)	The year value (four digits) of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
SD913			Execution time (month)	The month value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
SD914			Execution time (day)	The day value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
SD915			Execution time (hour)	The hour value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
SD916			Execution time (minute)	The minute value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
SD917			Execution time (second)	The second value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
SD918			Execution time (day of the week)	The day of the week value of the date/time when the firmware update was executed is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)	S (Initial)
SD919			Version after the update	The firmware version after the update execution is stored. When the update is completed with an error, 0 is stored.	S (Initial)
SD920			Version before the update	The firmware version before the update execution is stored.	S (Initial)
SD921	Latest firmware update result	Target		The start I/O number of the module where the firmware update was executed is stored. • CPU module: 3FFH	S (Initial)
SD922		Execution result		The execution result of the firmware update is stored. • 0001H: Completed successfully • 0100H: Flash ROM error • 0200H: Model mismatched • 0201H: File invalid • 0203H: Firmware update prohibited state • 0300H: Firmware data error	S (Initial)

No.	Name	Data stored		Details	Set by (setting timing)
SD923	Previous firmware update information	History information	Execution time (year)	The year value (four digits) of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
SD924			Execution time (month)	The month value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
SD925			Execution time (day)	The day value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
SD926			Execution time (hour)	The hour value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
SD927			Execution time (minute)	The minute value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
SD928			Execution time (second)	The second value of the date/time when the firmware update was executed is stored as a BIN code.	S (Initial)
SD929			Execution time (day of the week)	The day of the week value of the date/time when the firmware update was executed is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)	S (Initial)
SD930			Version after the update	The firmware version after the update execution is stored. When the update is completed with an error, 0 is stored.	S (Initial)
SD931			Version before the update	The firmware version before the update execution is stored.	S (Initial)
SD932	Previous firmware update result	Target		The start I/O number of the module where the firmware update was executed is stored. • CPU module: 3FFH	S (Initial)
SD933		Execution result		The execution result of the firmware update is stored. • 0001H: Completed successfully • 0100H: Flash ROM error • 0200H: Model mismatched • 0201H: File invalid • 0203H: Firmware update prohibited state • 0300H: Firmware data error	S (Initial)

Latch area

The following is the list of special register areas relating to the latch area.

No.	Name	Data stored	Details	Set by (setting timing)
SD944*1	Backup function of the CPU module data backup/restoration function	Backup function setting	Set the backup function using the following bit pattern. (Off: Disabled, On: Enabled)  b0: Automatic backup setting by specification of day and time b1: Automatic backup setting by specification of time and day of the week b5: Upper limit value status for the number of the CPU module backup data b10: Retry execution for the automatic backup b15: Automatic backup setting at occurrence of the CPU stop error	U
SD947*1		Day and time setting for automatic backup (day)	Store the day when the automatic backup on the specified day and time is executed using the BIN code. • Day (1 to 31)	U
SD948*1		Day and time setting for automatic backup (hour)	Store the hour when the automatic backup on the specified day and time is executed using the BIN code. • Hour (0 to 23)	U
SD949*1		Day and time setting for automatic backup (minute)	Store the minute when the automatic backup on the specified day and time is executed using the BIN code. • Minute (0 to 59)	U
SD950*1		Time and day of the week setting for automatic backup (hour)	Store the hour when the automatic backup on the specified day of the week and time is executed using the BIN code. • Hour (0 to 23)	U
SD951*1		Time and day of the week setting for automatic backup (minute)	Store the minute when the automatic backup on the specified day of the week and time is executed using the BIN code. • Minute (0 to 59)	U
SD952*1		Time and day of the week setting for automatic backup (day of the week)	Set the day of the week when the automatic backup is executed using the following bit pattern. (Off: Disabled, On: Enabled)  b0: Sunday, b1: Monday, b2: Tuesday, b3: Wednesday, b4: Thursday, b5: Friday, b6: Saturday	U
SD953*1		Backup error cause	• The cause of an error that occurred during the CPU module data backup is stored. 0H: No error Other than 0H:For details on the values stored when an error occurs, refer to the list of error codes. (☞ Page 512 List of error codes) • "0" is set at the start of the CPU module data backup.	S (Error)

No.	Name	Data stored	Details	Set by (setting timing)
SD954*1	Restoration function of the CPU module data backup/restoration function	Restoration target data setting	Set the target data to be restored with the CPU module data restoration function. 0: All the target data 1: Device/label data only 2: All the target data except for the device/label data	U
SD955*1		Restoration function setting	Set the CPU module data restoration function using the following bit pattern. (Off: Disabled, On: Enabled)  b0: Automatic restoration setting b1: Initialization setting at the automatic restoration b13: Restoration setting for the latest data b14: Restoration setting for the special relay and special register b15: Setting for continuous operation with the status at backup	U
SD956*1		Restoration target date folder setting	Store the target folder (date folder) of the CPU module data restoration using BCD code.  (1) Day (1 to 31) (2) Month (1 to 12) (3) Year (last two digits) (0 to 99) (4) Year (first two digits) (0 to 99) [Example] To specify the date folder of June 15 2015, store "H20150615".	U
SD957*1				U
SD958*1		Restoration target number folder setting	Specify the target folder of the CPU module data restoration. 1 to 32767: Serial number of the backup folder (*****) in a date folder (00001 to 32767)	U
SD959*1		Restoration error cause	• The cause of an error that occurred during the CPU module data restoration is stored. 0H: No error Other than 0: For details on the values stored when an error occurs, refer to the list of error codes. (☞ Page 512 List of error codes) • "0" is set at the start of the CPU module data backup.	S (Error)
SD960*1	Backup function of the CPU module data backup/restoration function	Upper limit status for the number of CPU module backup data	This register indicates the set value of the upper limit for the number of backup data in accordance with bit5 of SD944. Bit5 of SD944 is off: 0 Bit5 of SD944 is on: 1 to 100	S (Status change)
SD988*2	Memory copy completion status (latch)	Memory copy completion status (latch)	This register stores a value indicating the completion status of the memory copy from the control system to the standby system. • The value same as the SD1654 value is stored at the completion or abend of the memory copy from the control system to the standby system. • Since data have been backed up in case of power failure, this register holds the value indicating the latest memory copy completion status from the control system to the standby system. • This register is cleared to 0 by latch clear.	S (Status change)

*1 Programmable controller CPU only

*2 Process CPU (redundant mode) only

Data logging function

The following is the list of special register areas relating to the data logging function.

No.	Name	Data stored	Details	Set by (setting timing)
SD1210 SD1211	Data logging setting No.1 Latest storage file number	Latest storage file number	• The latest storage file number • This register is cleared to 0 by the stop command from CPU Module Logging Configuration Tool.	S (Status change)
SD1212 SD1213	Data logging setting No.1 Oldest storage file number	Oldest storage file number	• The oldest storage file number • This register is cleared to 0 by the stop command from CPU Module Logging Configuration Tool.	S (Status change)
SD1214	Data logging setting No.1 Internal buffer free space	Free space size of the internal buffer (K bytes)	• The amount of free space of the internal buffer is stored in K bytes. The smaller the value, the higher the generating ratio of processing overflow. • For the trigger logging, it refers to the total capacity of internal buffer until the data for the number of records is collected after a trigger occurred. • This register is cleared to 0 by the stop command from CPU Module Logging Configuration Tool.	S (Error)
SD1215	Data logging setting No.1 Number of processing overflow occurrences	Number of processing overflow occurrences	• The number of data logging processing overflow occurrences • When overflow occurs, data is lost. • When the count exceeds 65535, it returns to 0 and starts a new cycle. • With "Stop" set for the operation at the time when the number of save files exceeds the limit, a processing overflow may occur until the stop operation is completed after the collection of data corresponding to the specified number of storage files was completed. • This register is cleared to 0 when the setting is registered or by the stop command from CPU Module Logging Configuration Tool.	S (Error)
SD1216	Data logging setting No.1 data logging error cause	Data logging error cause	Cause of the error generated during data logging operations is stored. 0: No error Other than 0:For details on the values stored when an error occurs, refer to the list of error codes. (☞ Page 512 List of error codes)	S (Error)
SD1220 to SD1226	Data logging setting No.2	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1230 to SD1236	Data logging setting No.3	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1245 to SD1246	Data logging setting No.4	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1250 to SD1256	Data logging setting No.5	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1260 to SD1266	Data logging setting No.6	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1270 to SD1276	Data logging setting No.7	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1280 to SD1286	Data logging setting No.8	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1290 to SD1296	Data logging setting No.9	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1300 to SD1306	Data logging setting No.10	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1

CPU module data backup/restoration function

The following is the list of special register areas relating to the CPU module data backup/restoration function. (Programmable controller CPU only)

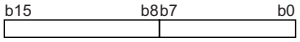
No.	Name	Data stored	Details	Set by (setting timing)
SD1350	Number of uncompleted folders/files of CPU module data backup/ restoration	Number of uncompleted folders/files of CPU module data backup/ restoration	This register indicates the number of folders/files where the backup/ restoration of the CPU module is not completed. When the backup/ restoration processing is started, the total number of folders and files to be backed up or restored is stored. The number is reduced one each time one folder/file is backed up or restored, and 0 is stored when all the data is backed up or restored.	S (Status change)
SD1351	Progression status of CPU module data backup/restoration	Progression status of CPU module data backup/restoration	• This register indicates the progression status of the backup or restoration is stored in percentage.*1 Range of the value: 0 to 100 (%) • "0" is set at the start of the CPU module data backup.	S (Status change)
SD1353	Upper limit value setting for the number of CPU module backup data	Setting of the upper limit value for the number of CPU module backup data	Set the upper limit value for the number of the backup data for the CPU module data backup. (1 to 100)	U

*1 When program files are restored, the progress in SD1351 stops while data is being written (transferred) to the program memory in the restoration processing because the data is transferred from the program cache memory to the program memory. The progress of data transfer to the program memory can be checked in SD629.

Data backup/restoration function of iQ Sensor Solution

The following is the list of special register areas relating to the data backup/restoration function of iQ Sensor Solution. (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SD1360	Right-to-use request number for iQ Sensor Solution data backup/ restoration	Right-to-use request number for data backup/restoration	Set the request number (desired 4-digit number) to acquire/release the right to use. Use the request number as follows depending on the request source. 0000H: Right-to-use release request 1000H to 1FFFH: Request from the program	U
SD1361	Right-to-use acquisition number for iQ Sensor Solution data backup/ restoration	Right-to-use acquisition number for data backup/restoration	This register stores the request number of when the right to use for the iQ Sensor Solution data backup/ restoration has been acquired. (0000H is stored when the right to use is not acquired or has been released)	S (Status change)
SD1362	Target module/ execution unit setting for iQ Sensor Solution data backup/restoration	Target module/ execution unit setting for data backup/ restoration	Set the target module and execution unit of the iQ Sensor Solution data backup/restoration. • Lower 8 bits (target module) 1H: AnyWireASLINK 3H: Ethernet 4H: CC-Link IE Field Network • Upper 8 bits (execution unit) <AnyWireASLINK> 1H: In module units 2H: In ID units <CC-Link IE Field Network> 1H: In module units 2H: In station units 3H: In station sub ID units <Ethernet> 1H: In module units 2H: In IP address units	U
SD1363	Target folder number setting for iQ Sensor Solution data backup/ restoration	Target folder number setting for data backup/ restoration	Set a number of folder in which the backup data is to be stored or a folder in which the data to be restored is stored. 00 to 99: Target folder specification FFFEH: Automatic specification (Folder deletion supported) FFFFH (default): Automatic specification	U

No.	Name	Data stored	Details	Set by (setting timing)
SD1364	Target setting for iQ Sensor Solution data backup/restoration (target module)	Target setting for data backup/restoration (target module)	Set the target module of the iQ Sensor Solution data backup/restoration. I/O number.[*1]: Module 3FFH: Built-in Ethernet[*2]	U
SD1365	Target setting for iQ Sensor Solution data backup/restoration (target device 1)	Target setting for data backup/restoration (target device 1)	Set the target device of the iQ Sensor Solution data backup/restoration. <AnyWireASLINK> ID number <CC-Link IE Field Network> Station number <Ethernet> IP address (lower 16 bits) Example: When the IP address is 192.168.3.40, 3 = 3H and 40 = 28H. Therefore, the value of lower 16 bits (0328H) is 808.	U
SD1366	Target setting for iQ Sensor Solution data backup/restoration (target device 2)	Target setting for data backup/restoration (target device 2)	Set the target device of the iQ Sensor Solution data backup/restoration. <AnyWireASLINK> 0 (Not used) <CC-Link IE Field Network> Station sub-ID number <Ethernet> IP address (upper 16 bits) Example: When the IP address is 192.168.3.40, 192 = C0H and 168 = A8H. Therefore, the value of upper 16 bits (C0A8H) is 49320.	U
SD1367	Operation setting for iQ Sensor Solution data backup/restoration	Operation setting for data backup/ restoration.	Set the operation of the iQ Sensor Solution data backup/restoration.  b0 to b7: Operation setting at error occurrence (0: Continue, 1: Stop) b8 to b15: Fixed to 0	U
SD1371	Execution status of iQ Sensor Solution data backup/restoration (total number of target devices)	Execution status of data backup/restoration (total number of target devices)	• This register stores the total number of target devices of the data backup/restoration of iQ Sensor Solution for each execution unit. • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change)
SD1372	Execution status of iQ Sensor Solution data backup/restoration (normal completion count)	Execution status of data backup/restoration (normal completion count)	• This register indicates the execution status (the number of devices where the processing has been completed normally) of the iQ Sensor Solution data backup/restoration. • Among the devices where the operation has been completed, the number of devices where the operation has been completed normally is stored. • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change)
SD1373	Execution status of iQ Sensor Solution data backup/restoration (error completion count)	Execution status of data backup/restoration (error completion count)	• This register indicates the execution status (the number of devices where the processing has been completed with an error) of the iQ Sensor Solution data backup/restoration. • Among the devices where the operation has been completed, the number of devices where the operation has been completed with an error is stored. • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change)
SD1374	Execution status of iQ Sensor Solution data backup/restoration (progress per device)	Execution status of data backup/restoration (progress per device)	• This register indicates the progression status of the iQ Sensor Solution data backup/restoration of the device in progress in percentage (0 to 100%). Range of the value: 0 to 100 (%) • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change)
SD1375	Folder number of iQ Sensor Solution data backup	Backup target folder number	• This register stores a number of the target folder where the backup data of the device supporting iQSS is stored. 0 to 99: Folder number FFFFH: Backup data not saved • This register stores FFFFH when the right to use is acquired or the backup/restoration is requested.	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SD1376	Module error cause of iQ Sensor Solution data backup/restoration	Module error cause of data backup/restoration	• This register stores the error cause that is detected in the module during the iQ Sensor Solution data backup/restoration. When errors are detected in multiple devices, the error detected first is stored. (For details on the error cause stored, refer to the manual for the CPU module used or the target device.) • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)
SD1377	Target device error cause of iQ Sensor Solution data backup/ restoration	Target device error cause of data backup/ restoration	• This register stores the error cause that is detected in the target device during the iQ Sensor Solution data backup/restoration. When errors are detected in multiple devices, the error detected first is stored. (For details on the error cause stored, refer to the manual for the target device.) • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)
SD1378	Module information/ execution unit information of iQ Sensor Solution data backup/restoration error	Module information/ execution unit information of an error	• At an occurrence of the iQ Sensor Solution data backup/restoration error, this register stores the information of the module where the error has been occurred and the execution unit information. When errors are detected in multiple devices, the information of a device where the error detected first is stored. • Lower 8 bits (target module) 1H: AnyWireASLINK 3H: Ethernet 4H: CC-Link IE Field Network • Upper 8 bits (execution unit) <AnyWireASLINK> 1H: In module units 2H: In ID units <CC-Link IE Field Network> 1H: In module units 2H: In station units 3H: In station sub ID units <Ethernet> 1H: In module units 2H: In IP address units • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)
SD1379	Folder number information of iQ Sensor Solution data backup/restoration error	Folder number information of an error	• At an occurrence of the iQ Sensor Solution data backup/restoration error, this register stores the folder number (0 to 99) of the backup/ restoration target. If the folder number cannot be specified or a backup error is detected before creating a backup folder (I/O number_station number), FFFFH is stored. • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)
SD1380	Details on iQ Sensor Solution data backup/ restoration error (module)	Information of the module where an error has occurred	• This register stores the information of a module where an error of the iQ Sensor Solution data backup/restoration has been occurred. When errors are detected in multiple modules, the information of a module where the error detected first is stored. I/O number.[*1]: Module 3FFH: Built-in Ethernet[*2] • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)
SD1381	Details on iQ Sensor Solution data backup/ restoration error (device 1)	Information of the device (device 1) where an error has occurred	• This register stores the information of a device (device 1) where an error of the iQ Sensor Solution data backup/restoration has been occurred. When errors are detected in multiple devices, the information of a device (device 1) where the error detected first is stored. <AnyWireASLINK> ID number <CC-Link IE Field Network> Station number <Ethernet> IP address (lower 16 bits) Example: When the IP address is 192.168.3.40, 3 = 3H and 40 = 28H. Therefore, the value of lower 16 bits (0328H) is 808. • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)

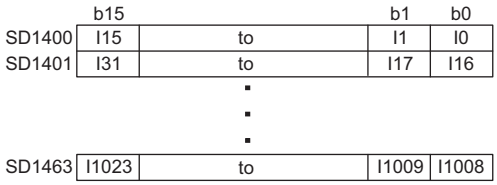
No.	Name	Data stored	Details	Set by (setting timing)
SD1382	Details on iQ Sensor Solution data backup/ restoration error (device 2)	Information of the device (device 2) where an error has occurred	• This register stores the information of a device (device 2) where an error of the iQ Sensor Solution data backup/restoration has been occurred. When errors are detected in multiple devices, the information of a device (device 2) where the error detected first is stored. <AnyWireASLINK> 0 (Not used) <CC-Link IE Field Network> Station sub-ID number <Ethernet> IP address (upper 16 bits) Example: When the IP address is 192.168.3.40, 192 = C0H and 168 = A8H. Therefore, the value of upper 16 bits (C0A8H) is 49320. • This register is cleared to 0 when the right to use is acquired or the backup/restoration is requested.	S (Status change/Error)

*1 The start I/O number (first three digits in four-digit hexadecimal representation) of each module

*2 The built-in Ethernet port for RnENCPU

Interrupt pointer mask pattern

The following is the list of special register areas relating to the mask pattern for interrupt pointers.

No.	Name	Data stored	Details	Set by (setting timing)
SD1400 to SD1463	Interrupt pointer mask pattern	Mask pattern	The mask pattern for interrupt pointers is stored as shown in the following figure: 	S (at execution)

User authentication function

The following is the list of special register area relating to the user authentication function. (Safety CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SD1468	Number of login users	Number of login users	The number of users logged in to the CPU module is stored. The value is incremented by one every time the login operation to the programmable controller or files is performed. The value is decremented by one every time the logout operation is performed.	S (Status change)

Memory dump function

The following is the special register area relating to the memory dump function. (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SD1472	Memory dump error cause	Memory dump error cause	The cause of the error occurred during the memory dump function is stored. 0: No error Other than 0:For details on the values stored when an error occurs, refer to the list of error codes. (☞ Page 512 List of error codes)	S (upon error)

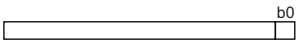
Real-time monitor function

The following is the special register area relating to the real-time monitor function. (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SD1484	Real-time monitor internal buffer free space	Free space size of the internal buffer (K bytes)	The amount of free space of the internal buffer is stored in K bytes. The smaller the value, the higher the generating ratio of processing overflow.	S (Status change)

Debug function

The following is the special register area relating to the debug function. (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SD1488	Debug function usage status	Debug function usage status	The usage status of the debug function is stored using the following bit pattern.  b0: External input/output forced on/off b1 to b15: Empty (fixed to 0)	S (Status change)

A

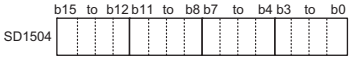
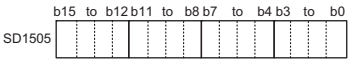
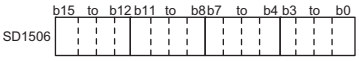
CPU module database access function

The following is the special register area relating to the CPU module database access function. (Programmable controller CPU with firmware version "28" or later)

No.	Name	Data stored	Details	Set by (setting timing)
SD1498	Start-up status of CPU module database	Start-up status of CPU module database	When the CPU module database access function is enabled with the module parameter, the CPU module database start-up status is stored. 0:No error FFFFH: Starting-up Other than above: Start-up error For the stored value at the start-up error, refer to the list of error codes that occur when a database access instruction is executed. (MELSEC iQ-R Programming Manual (Instructions, Standard Functions/ Function Blocks))	S (Status change)

Ethernet function

The following is the list of special register areas relating to the Ethernet function.

No.	Name	Data stored	Details	Set by (setting timing)
SD1504	Open completion signal	In this register, open completion status is stored.	Open statuses of connection No.1 to 16 are stored. (0: Close/Open not completed, 1: Open completed).  b0: Connection 1 b1: Connection 2 b2 to b15: Connection 3 to 16 For details on the on/off timing, refer to the following. (📖 MELSEC iQ-R Ethernet User's Manual (Application))	S (Status change)
SD1505	Open request signal	In this register, open request status is stored.	Open processing statuses of connection No.1 to 16 are stored. (0: No open request, 1: Open request exists).  b0: Connection 1 b1: Connection 2 b2 to b15: Connection 3 to 16 For details on the on/off timing, refer to the following. (📖 MELSEC iQ-R Ethernet User's Manual (Application))	S (Status change)
SD1506	Socket communications receive status signal	In this register, receive status is stored.	Receiving statuses of connection No.1 to 16 are stored. (0: No data received, 1: Data receiving completed)  b0: Connection 1 b1: Connection 2 b2 to b15: Connection 3 to 16 For details on the on/off timing, refer to the following. (📖 MELSEC iQ-R Ethernet User's Manual (Application))	S (Status change)

A

<table>
<tr><th>No.</th><th>Name</th><th>Data stored</th><th>Details</th><th>Set by (setting timing)</th></tr>
<tr><td>SD1520</td><td rowspan="4">IP address setting</td><td>IP Address (lower)</td><td rowspan="2">• Specify the IP address to be stored into the IP address storage area (system memory*1). Range: 00000001H to DFFFFFFEH (0.0.0.1 to 223.255.255.254)
• Upon completion of writing or clearing the IP address to/from the IP address storage area (system memory*1), the value of IP address stored in the IP address storage area (system memory*1) is stored.
<table><tr><th></th><th>b15 to b8</th><th>b7 to b0</th></tr><tr><td>SD1520</td><td>3</td><td>4</td></tr><tr><td>SD1521</td><td>1</td><td>2</td></tr></table>1 to 4: First to fourth byte</td><td rowspan="2">S (Status change)/U</td></tr>
<tr><td>SD1521</td><td>IP Address (upper)</td></tr>
<tr><td>SD1522</td><td>Subnet mask pattern (lower)</td><td rowspan="2">• Specify the Subnet mask pattern to be stored into the IP address storage area (system memory*1). Setting range: C0000000H to FFFFFFFCH (192.0.0.0 to 255.255.255.252), 00000000H (no setting)
• Upon completion of writing or clearing the IP address to/from the IP address storage area (system memory*1), the value of subnet mask pattern stored in the IP address storage area (system memory*1) is stored.
<table><tr><th></th><th>b15 to b8</th><th>b7 to b0</th></tr><tr><td>SD1522</td><td>3</td><td>4</td></tr><tr><td>SD1523</td><td>1</td><td>2</td></tr></table>1 to 4: First to fourth byte</td><td rowspan="2">S (Status change)/U</td></tr>
<tr><td>SD1523</td><td>Subnet mask pattern (upper)</td></tr>
<tr><td>SD1524</td><td>Default Router IP Address (lower)</td><td rowspan="2">• Specify the default router IP address to be stored into the IP address storage area (system memory*1). Setting range: 00000001H to DFFFFFFEH (0.0.0.1 to 223.255.255.254), 00000000H (no setting)
• Upon completion of writing or clearing the IP address to/from the IP address storage area (system memory*1), the value of default router IP address stored in the IP address storage area (system memory*1) is stored.
<table><tr><th></th><th>b15 to b8</th><th>b7 to b0</th></tr><tr><td>SD1524</td><td>3</td><td>4</td></tr><tr><td>SD1525</td><td>1</td><td>2</td></tr></table>1 to 4: First to fourth byte</td><td rowspan="2">S (Status change)/U</td></tr>
<tr><td>SD1525</td><td>Default Router IP Address (upper)</td></tr>
<tr><td>SD1526</td><td>IP address storage area writing error cause</td><td>The error cause for the case when failing in writing into IP address storage area is stored.</td><td>The error cause generated when writing into IP address storage area (system memory*1) will be stored. (Linked with SM1521)
0H: No error
100H:SD1520 to SD1525 are out of the specified range.
200H:Write error
400H:Impossible to execute the write processing because clear processing is in progress.</td><td>S (Status change)</td></tr>
<tr><td>SD1527</td><td>IP address storage area clearing error cause</td><td>The cause of the error for the case when failing in clearing IP address storage area is stored.</td><td>The cause of the error generated during writing into IP address storage area (system memory*1) is stored. (Linked with SM1523)
0H: No error
200H:Clear error
400H:Impossible to execute the clear processing because write processing is in progress.</td><td>S (Status change)</td></tr>
</table>

*1 This is the memory used by the system when the CPU module is executing functions.

CC-Link IE Field Network Basic function

The following is the list of special register areas relating to the CC-Link IE Field Network Basic function. (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SD1536 to SD1539	Cyclic transmission status of each station	Cyclic transmission status of each station	The cyclic transmission status of each station is stored using the following bit pattern. (Off: Not performed, On: Being performed) b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SD1536: 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 SD1537: 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 SD1538: 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 SD1539: 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 The numbers in the figure indicate station numbers. (Condition) • Only the bit of the start station number turns on. • The status is not stored for the reserved stations and the station numbers after the maximum station number. Use this register as an interlock for cyclic transmission. For details on the interlock program, refer to the following. CC-Link IE Field Network Basic Reference Manual	S (Status change)
SD1540 to SD1543	Data link status of each station	Data link status of each station	The data link status of each station is stored using the following bit pattern. (Off: Normally operating station*1, On: Faulty station) b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SD1540: 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 SD1541: 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 SD1542: 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 SD1543: 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 The numbers in the figure indicate station numbers. (Condition) • Only the bit of the start station number turns on. • The status is not stored for the reserved stations and the station numbers after the maximum station number. This register can be used to monitor errors in slave stations, connected cables, and a connected hub.	S (Status change)

*1 This status includes the case where a slave station has not responded to the first request from the master station due to a power-off of the slave station. (The slave station is not judged as a faulty station because the data link status is not determined.)

A

Online module change function

The following is the list of special register areas relating to the online module change function. (Process CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SD1600	Module selection (base unit No.)	Base unit No. where the online change target module is mounted	The base unit number where the online change target module is mounted is specified. 0: Main base unit 1: Extension base unit 1 ⋮ 7: Extension base unit 7 FFFFH: Not specified (Default) The stored value returns to its default (FFFFH) upon completion of the online module change processing. When changing a module directly, the system stores the corresponding base unit number upon removal of the target module.	S (Status change)/U (Request)
SD1601	Module selection (slot No.)	Slot No. where the online change target module is mounted	The slot number where the online change target module is mounted is specified. 0: Slot No.0 1: Slot No.1 ⋮ 11: Slot No.11 FFFFH: Not specified (Default) The specified value returns to its default (FFFFH) upon completion of the online module change processing. When changing a module directly, the system stores the corresponding slot number upon removal of the target module.	S (Status change)/U (Request)
SD1602	I/O No. of the module being changed online	I/O No. of the module being changed online	The value of the I/O number of the module being changed online divided by 16 is stored. Other than FFFFH: I/O No. ÷ 16 FFFFH: Not specified (Default) The stored value returns to its default (FFFFH) upon completion of the online module change processing.	S (Status change)
SD1617	Online module change progress status	Online module change progress status	The online module change progress status is stored. 0: Normal operation 1: Module being selected 2: Module selected 3: Module removal requested 4: Module removal ready 5: Module removed 6: Module mounted 7: Module being recognized 8: Module recognized 9: Module control resumed The value becomes 0 upon completion of the online module change processing.	S (Status change)
SD1618	Online module change error code	0: Normal operation Other than 0: Error code	The corresponding error code is stored when an error is detected during online module change. For the value stored, refer to the list of error codes. (☞ Page 592 Error codes related to the online module change function) This register is cleared to 0 when the error cause is eliminated and the online module change related request is executed. However, the error code needs to be cleared to 0 before module selection since this register is not cleared to 0 if an error occurs in selecting a module.	S (Status change)/U (Request)
SD1619	Disable request error code during online module change	0: Normal operation Other than 0: Error code	The corresponding error code is stored when a disable request is executed during online module change. The error code is cleared to 0 when the error cause is eliminated and the online module change related request is executed. For the value stored, refer to the list of error codes. (☞ Page 592 Error codes related to the online module change function)	S (Status change)

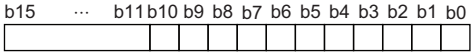
System information (Process CPU)

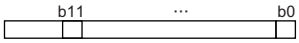
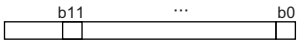
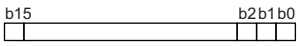
The following is the list of special register areas relating to the system information.

No.	Name	Data stored	Details	Set by (setting timing)
SD1622	Process CPU operation mode	Process CPU operation mode	This register stores the operation mode of the Process CPU 80H: Process mode 81H: Redundant mode	S (Initial)

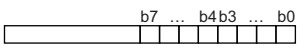
Redundant function

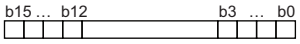
The following is the list of special register areas relating to the redundant function. (Process CPU (redundant mode) only)

No.	Name	Data stored	Details	Set by (setting timing)
SD1642	BACKUP/SEPARATE LED flashing cause	BACKUP/SEPARATE LED flashing cause	Flashing cause of BACKUP/SEPARATE LED is stored using the following bit pattern.  Corresponding bits to each cause turns on. b1: Tracking communications disabled b3: Stop error of the standby system b4: Operating status mismatch between both systems b5: Memory copy being executed b6: Online change being executed b7: A failure of a network module detected on the standby system b8: System switching being executed b9: A redundant function module being changed online b10: System switching disabled on the standby system by using the DCONTSW instruction • 0 is stored as initial value. • When the multiple causes occurred at the same time, bits corresponding to each cause turn on simultaneously. If the tracking communications are disabled or the redundant function module is being changed online, the other causes turn off. • When the redundant function module in the other system is being changed online, "Tracking communications disabled" is stored in own system. • Even if a cause of the system switching failure has occurred, the system is switched when the system switching causes which have higher priority, such as power-off or reset of the CPU module, have occurred. (☞ Page 398 Execution availability of system switching)	S (Status change)
SD1643	System switching cause	System switching cause occurred in own system (normal/abnormal systems witching)	Stores the system switching cause occurred in own system. The system switching cause is stored in this register even if the systems cannot be switched by a cause of system switching failure. This register is initialized with 0 when the CPU module is powered off and on or is reset. 0: Initial value (the control system never been switched) 1: Power-off, reset, hardware failure 2: Stop error 3: System switching request from a network module 16: System switching request by using the SP.CONTSW instruction 17: System switching request using an engineering tool When the system is switched upon the power-off or reset of the CPU module in the control system, 1 is not stored in SD1643 of the new standby system.	S (At system switching)

No.	Name	Data stored	Details	Set by (setting timing)
SD1644	Cause of system switching failure	Cause number of system switching failure	• If a system switching is failed because a cause of system switching failure has occurred, either of the following values is stored in this register. 0: System switching completed normally (default) 1: Tracking communications disabled 2: Tracking communication timeout 3: Stop error of the standby system 4: Operating status mismatch between both systems 5: Memory copy being executed 6: Online change being executed 7: A failure of a network module detected on the standby system 8: System switching being executed 9: A redundant function module being changed online 10: System switching disabled on the standby system by using the DCONTSW instruction • This register is initialized with 0 when own system is powered on. • Zero is stored in this register upon completion of system switching.	S (At system switching)
SD1645	System switching request status from a network module of own system	System switching request status from a network module of own system	• The system switching request status from a network module of the own system is stored using the following bit pattern. (The bits of the positions where no modules are mounted turn off.)  b0 to b11: Slot No.0 to slot No.11 • The system turns off the bit after the error in the corresponding module is removed by a user. • For the system switching request status from a network module of the other system, refer to SD1646.	S (Error/Status change)
SD1646	System switching request status from a network module of the other system	System switching request status from a network module of the other system	• The system switching request status from a network module of the other system is stored using the following bit pattern. (The bits of the positions where no modules are mounted turn off.)  b0 to b11: Slot No.0 to slot No.11 • The system turns off the bit after the error in the corresponding module is removed by a user. • For the system switching request status from a network module of the own system, refer to SD1645.	S (Every END)
SD1648	Cause of the other system monitoring error	Cause of the other system monitoring error	• When an error has occurred on the communications with the other system during an initial processing (including when the system is determined while waiting for the other system starts up) or an END processing, either of the following bits turns on. Once the error is cleared, the bit turns off.  b0: Communications with the other system disabled (other than a redundant function module hardware failure in own system) b1: Power-off, reset, or hardware failure (the CPU module or the redundant function module) in the other system b2: Stop error in the other system b15: Communications with the other system disabled (a redundant function module hardware failure in own system) • If either of b0, b1, b2 or b15 is on, the other bits are off. • When the communications with the other system is disabled at an initial processing (including when the other system is powered off), b0 turns on. (When power-off or reset of the CPU module in the other system is recognized, b1 turns on.) • The tracking cable status of when one of the above mentioned bits of this register is on can be checked in SD1755.	S (Initial/every END)/system switching

No.	Name	Data stored	Details	Set by (setting timing)
SD1649	System switching cause (when the systems are successfully switched)	System switching cause (when the systems are successfully switched)	System switching cause is stored. • System switching cause is stored in SD1649 of both systems when the system is switched. • This register is initialized with 0 when the CPU module is powered off and on or is reset. • Either of the following is stored in this register. 0: Initial value (the control system never been switched) 1: Power-off, reset, hardware failure 2: Stop error 3: System switching request from a network module 16: System switching request by using the SP.CONTSW instruction 17: System switching request using an engineering tool Note that 1 is not stored in SD1649 of the new standby system when the system is switched by the power-off or reset of the CPU module in the control system.	S (At system switching)
SD1650	System switching instruction ID number	System switching instruction ID number	• This register stores the argument (system switching instruction ID number) of the executed SP.CONTSW instruction when the system is switched by using the SP.CONTSW instruction. (The argument is stored in SD1650 of both systems at the system switching.) • This register is valid only while 16 (System switching request by using the SP.CONTSW instruction) is stored in SD1649. • This register is updated only when the system is switched by the SP.CONTSW instruction. • The initial value is 0.	S (At system switching)
SD1653	Memory copy destination I/O number	Memory copy destination I/O number	• Before SM1653 is turned off and on, the I/O number of the memory copy destination (CPU module in the standby system: 03D1H) is stored. • The initial value is 0.	U
SD1654	Memory copy completion status	Memory copy completion status	This register stores the execution status of the memory copy. When the value other than 0 is stored, the memory copy is completed with an error or cannot be executed. 0H: Memory copy completed normally 4025H: Memory copy execution during file access 402CH: Memory format being executed 41FEH: SD memory card not inserted or SD memory card disabled 41CFH: Capacity excess of the SD memory card used 41D8H: Files of the CPU module in the standby system being accessed 41DFH: Write protection enabled on the SD memory card 4202H: Online module change in progress 4246H: Operation mode being changed or the system being switched 4248H: The value other than 03D1H stored in SD1653 424CH: Programs being changed online 425FH: Different CPU module models used in the control system and standby system 6F30H: Communication error with the other system	S (Status change)
SD1662	Tracking transfer data receive completion wait time	Value to be added to the tracking transfer data receive completion wait time	• Specify the value to be added to waiting time for completion of tracking data reception for the CPU module in the control system. • Specify the value in units of ms. (Range is 0 to 2100ms.) If the value out of the range is specified, the system recognized the value as 2100ms. • The initial value is 0ms.	U
SD1664	Tracking transfer error count	Tracking transfer error count	• For the tracking transfer of the device/label data at each scan, the number is added (+1) when the tracking transfer cannot be executed because of disconnection of the tracking cables, power-off, reset, stop error of the CPU module in the standby system, or error in the redundant function module. • A counting cycle from 0 to 65535 to 0 is repeated.	S (upon error)

No.	Name	Data stored	Details	Set by (setting timing)
SD1667 to SD1670	Tracking transfer trigger	Off: No trigger On: Trigger	• When data is transferred based on the tracking transfer setting of redundant settings, specify the target block as trigger. ■SD1667 b0 to b15: Block 1 to block 16 ■SD1668 b0 to b15: Block 17 to block 32 ■SD1669 b0 to b15: Block 33 to block 48 ■SD1670 b0 to b15: Block 49 to block 64 • When "Transfer Automatically" is selected for the tracking block No.1 auto transfer setting, b0 of SD1667 is turned on by the system at power-on or when the CPU module is switched from STOP to RUN. In other cases, turn on bit0 of SD1667 to b15 of SD1670 by the user. • Initial values for b0 of SD1667 to b15 of SD1670 are off (no trigger)	S (Initial)/U
SD1673 to SD1676	Tracking transfer completion status	Off: Transfer not completed On: Transfer completed	• This flag is stored a result of the tracking transfer operated in the preceding END processing. • The flag turns on if any of the tracking transfers in the corresponding blocks has been normally completed, and turns off if the tracking transfers failed due to causes such as tracking communication error. ■SD1673 b0 to b15: Block 1 to block 16 ■SD1674 b0 to b15: Block 17 to block 32 ■SD1675 b0 to b15: Block 33 to block 48 ■SD1676 b0 to b15: Block 49 to block 64	S (Status change)
SD1680	CPU module operation information (the other system)	CPU module operation information (the other system)	This register stores the operating status of the CPU module in the other system in the following bit pattern. When communications with the other system are disabled, 00FFH is stored.  b0 to b3: 0H: RUN, 2H: STOP, 3H: PAUSE, FH: Communications with the other system disabled b4 to b7: 0H: No error, 1H: Continuation error, 2H: Stop error, FH: Communications with other systems disabled Communications with the other system are disabled in the following states. • The CPU module in the other system is powered off or is being reset. • A hardware failure has occurred in own or the other system. • Tracking cables are not connected or disconnected.	S (Initial/every END)/system switching
SD1681	Latest self-diagnostic error code (the other system)	Latest error code (the other system)	This register stores an error code for the error occurred in the other system with hexadecimal. SD0 of the CPU module in the other system is reflected.	S (Every END)
SD1682 to SD1688	Latest self-diagnostic error time (the other system)	Latest self-diagnostic error time (the other system)	• This register stores the error time in the other system. • Data configuration is the same as SD1 to SD7. • The values in SD1 to SD7 of the CPU module in the other system are reflected.	S (Every END)
SD1689	Detailed information 1 information category (the other system)	Information category code of the detailed information 1 (the other system)	• This register stores the category code of the detailed information 1 for the error that has occurred in the other system. • Data configuration is the same as SD80. • The value in SD80 of the CPU module in the other system is reflected.	S (Every END)
SD1690 to SD1720	Detailed information 1 (the other system)	Detailed information 1 (the other system)	• This register stores the detailed information 1 corresponding to the error that has occurred in the other system. • Data configuration is the same as SD81 to SD111. • The values in SD81 to SD111 of the CPU module in the other system are reflected.	S (Every END)
SD1721	Detailed information 2 information category (the other system)	Information category code of the detailed information 2 (the other system)	• This register stores the category code of the detailed information 2 for the error that has occurred in the other system. • Data configuration is the same as SD112. • The value in SD112 of the CPU module in the other system is reflected.	S (Every END)

No.	Name	Data stored	Details	Set by (setting timing)
SD1722 to SD1752	Detailed information 2 (the other system)	Detailed information 2 (the other system)	• This register stores the detailed information 2 corresponding to the error that has occurred in the other system. • Data configuration is the same as SD113 to SD143. • The values in SD113 to SD143 of the CPU module in the other system are reflected.	S (Every END)
SD1754	Cause of control system start-up	Cause of start-up as the control system	This register stores the cause that one of the redundant system has been stated up as the control system. • This register is initialized with 0 when the CPU module is powered off and on or is reset.  b0 to b3: Cause of start-up by specific operations to start up the system as the control system 0: No specific operation, 1: Switch operation (RUN→STOP→RUN), 2: Online operation, 4: Contact input (X) b12 to b15: Information when the system is started up as the control system 0: Before start-up, 1: Start-up as the control system by starting up the other system, 2: Start-up as the control system by the specific operation	S (Status change)
SD1755	Tracking cable connection status	Tracking cable connection status	• This register stores the tracking cable connection status when an initial processing (including when the system is determined while waiting for the other system starts up) or an END processing is performed. 0H: Normal 12H: Cable disconnection of the OUT side 13H: Cable insertion error of the OUT side 14H: Communications on the OUT side being established 21H: Cable disconnection of the IN side 22H: Cable disconnection of the IN or OUT side 23H: Cable disconnection of the IN side or cable insertion error of the OUT side 24H: Cable disconnection of the IN side or communications on the OUT side being established 31H: Cable insertion error of the IN side 32H: Cable insertion error of the IN side or cable disconnection of the OUT side 33H: Cable insertion error of the IN or OUT side 34H: Cable insertion error of the IN side or communications on the OUT side being established 41H: Communications on the IN side being established 42H: Communications on the IN side being established or cable disconnection of the OUT side 43H: Communications on the IN side being established or cable insertion error of the OUT side 44H: Communications on the IN or OUT side being established • The tracking cable status can be checked with this register when bit of SD1648 turns on.	S (Initial/every END)

Safety information (RnSFCPU)

The following is a list of special registers relating to safety information. (Safety CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SD1840	Continuous RUN time in TEST MODE	Continuous RUN time in TEST MODE (L)	Continuous RUN time in TEST MODE is stored. (This register stores measurement values within the range from 1 to 2147483647 (measured in increments of 1 second).) However, measurement starts when the operating status of the CPU module is switched from STOP to RUN. (The time is not measured while the CPU module in the STOP state.) Note that measurement continues even if the time limit is exceeded. After stopping measurement, the register is cleared to 0 when the measurement value is cleared. (☞ Page 492 Measuring the continuous RUN time in TEST MODE)	S (Every END)
SD1841		Continuous RUN time in TEST MODE (H)		
SD1844	Number of points assigned to safety bit devices	SA\X (L)	The number of points assigned to the SA\X device is stored in 32 bits.	S (Initial)
SD1845		SA\X (H)		
SD1846		SA\Y (L)	The number of points assigned to the SA\Y device is stored in 32 bits.	S (Initial)
SD1847		SA\Y (H)		
SD1848		SA\M (L)	The number of points assigned to the SA\M device is stored in 32 bits.	S (Initial)
SD1849		SA\M (H)		
SD1850		SA\B (L)	The number of points assigned to the SA\B device is stored in 32 bits.	S (Initial)
SD1851		SA\B (H)		
SD1864	Number of points assigned to safety word devices	SA\D (L)	The number of points assigned to the SA\D device is stored in 32 bits.	S (Initial)
SD1865		SA\D (H)		
SD1866		SA\W (L)	The number of points assigned to the SA\W device is stored in 32 bits.	S (Initial)
SD1867		SA\W (H)		
SD1872	Number of points assigned to safety timer/counter devices	SA\T (L)	The number of points assigned to the SA\T device is stored in 32 bits.	S (Initial)
SD1873		SA\T (H)		
SD1874		SA\ST (L)	The number of points assigned to the SA\ST device is stored in 32 bits.	S (Initial)
SD1875		SA\ST (H)		
SD1876		SA\C (L)	The number of points assigned to the SA\C device is stored in 32 bits.	S (Initial)
SD1877		SA\C (H)		
SD1888	Safety cycle processing time execution cycle error count	0: No safety cycle processing time execution cycle error occurred (normal) 1 to 65535: Cumulative number of execution cycle errors during safety cycle processing time	The number of times that cycles for which safety cycle processing was not executed were detected is stored.	S (Status change)
SD1890	Current safety cycle processing time	Current safety cycle processing time (unit: ms)	• The current safety cycle processing time is stored. (The time is measured in microseconds (μs).) SD1890: The ms part is stored. (Range: 0 to 65535) SD1891: The μs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.	S (Status change)
SD1891		Current safety cycle processing time (unit: μs)		
SD1892	Minimum safety cycle processing time	Minimum safety cycle processing time (unit: ms)	• The minimum safety cycle processing time is stored. (The time is measured in microseconds (μs).) SD1892: The ms part is stored. (Range: 0 to 65535) SD1893: The μs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.	S (Status change)
SD1893		Minimum safety cycle processing time (unit: μs)		
SD1894	Maximum safety cycle processing time	Maximum safety cycle processing time (unit: ms)	• The maximum safety cycle processing time is stored. (The time is measured in microseconds (μs).) SD1894: The ms part is stored. (Range: 0 to 65535) SD1895: The μs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.	S (Status change)
SD1895		Maximum safety cycle processing time (unit: μs)		

No.	Name	Data stored	Details	Set by (setting timing)
SD1903	Total standard/safety shared label usage capacity	Total standard/safety shared label usage capacity (unit: words)	The total usage capacity for labels defined in the standard/safety shared label setting file is stored in 4-word units.	S (Status change)
SD1904	Start I/O number of the CC-Link IE Field Network safety communication target station (1st module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for CC-Link IE Field Network safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)
SD1905	Start I/O number of the CC-Link IE Field Network safety communication target station (2nd module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for CC-Link IE Field Network safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)
SD1906	Start I/O number of the CC-Link IE Field Network safety communication target station (3rd module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for CC-Link IE Field Network safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)
SD1907	Start I/O number of the CC-Link IE Field Network safety communication target station (4th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for CC-Link IE Field Network safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)
SD1908	Start I/O number of the CC-Link IE Field Network safety communication target station (5th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for CC-Link IE Field Network safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)
SD1909	Start I/O number of the CC-Link IE Field Network safety communication target station (6th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for CC-Link IE Field Network safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)
SD1910	Start I/O number of the CC-Link IE Field Network safety communication target station (7th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for CC-Link IE Field Network safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)
SD1911	Start I/O number of the CC-Link IE Field Network safety communication target station (8th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for CC-Link IE Field Network safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)

Appendix 6 List of Safety Special Relay Areas

The following table lists items in the list.

Item	Description
No.	Safety special relay number
Name	Safety special relay name
Description	Data stored in the safety special relay and its meaning
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (program, engineering tool, GOT, or other testing operations from external device) • U/S: User and system <Set timing> • Every END: Data is set every time END processing is performed. • Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN). • Status change: Data is set when the status is changed. • Error: Data is set when an error occurs. • Instruction execution: Data is set when an instruction is executed. • Request: Data is set when requested by a user (using the safety special relay). • Writing: Data is set when a user performs a writing operation. • During END: Data is set when END processing is performed. • Power-on to RUN or STOP to RUN: Data is set when the operating status changes from power-on to RUN or from STOP to RUN.

Point

Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

System clock

The following is a list of safety special relays relating to the system clock. (Safety CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SA\SM400	Always on	ON OFF	This relay is always on.	S (Power-on to RUN or STOP to RUN)
SA\SM401	Always off	ON OFF	This relay is always off.	S (Power-on to RUN or STOP to RUN)
SA\SM444	On at the first safety program execution	ON OFF First time only	This relay turns on at the first safety program execution.	S (Power-on to RUN or STOP to RUN)

Safety information (RnSFCPU)

The following is a list of safety special relays relating to safety information. (Safety CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SA\SM1008	Safety refresh communication status of each module (1st module)	Off: Normal On: Communication error	The safety refresh communication status of the 1st CC-Link IE Field Network module (the one whose start I/O number is stored in SD1904) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1008 to SA\SD1015.)	S (Status change)
SA\SM1016	Safety refresh communication status of each module (2nd module)	Off: Normal On: Communication error	The safety refresh communication status of the 2nd CC-Link IE Field Network module (the one whose start I/O number is stored in SD1905) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1016 to SA\SD1023.)	S (Status change)
SA\SM1024	Safety refresh communication status of each module (3rd module)	Off: Normal On: Communication error	The safety refresh communication status of the 3rd CC-Link IE Field Network module (the one whose start I/O number is stored in SD1906) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1024 to SA\SD1031.)	S (Status change)
SA\SM1032	Safety refresh communication status of each module (4th module)	Off: Normal On: Communication error	The safety refresh communication status of the 4th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1907) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1032 to SA\SD1039.)	S (Status change)
SA\SM1040	Safety refresh communication status of each module (5th module)	Off: Normal On: Communication error	The safety refresh communication status of the 5th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1908) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1040 to SA\SD1047.)	S (Status change)
SA\SM1048	Safety refresh communication status of each module (6th module)	Off: Normal On: Communication error	The safety refresh communication status of the 6th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1909) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1048 to SA\SD1055.)	S (Status change)
SA\SM1056	Safety refresh communication status of each module (7th module)	Off: Normal On: Communication error	The safety refresh communication status of the 7th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1910) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1056 to SA\SD1063.)	S (Status change)
SA\SM1064	Safety refresh communication status of each module (8th module)	Off: Normal On: Communication error	The safety refresh communication status of the 8th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1911) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1064 to SA\SD1071.)	S (Status change)
SA\SM1088	CC-Link IE Field safety communications information storage area module specification switch request	Off: Module specification completed On: Module specification requested	The SA\SD1088 content is updated when this relay changes from off to on. This relay turns off when SA\SD1090 to 1097 and SA\SD1104 to 1223 data update is complete.	U (Request)/S (Status change)

Appendix 7 List of Safety Special Register Areas

The following table lists items in the list.

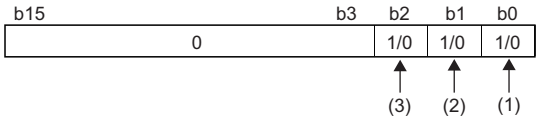
Item	Description
No.	Safety special register number
Name	Safety special register name
Description	Data stored in the safety special register
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (program, engineering tool, GOT, or other testing operations from external device) • U/S: User and system <Set timing> • Every END: Data is set every time END processing is performed. • Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN). • Status change: Data is set when the status is changed. • Error: Data is set when an error occurs. • Instruction execution: Data is set when an instruction is executed. • Request: Data is set when requested by a user (using the safety special relay). • Switch change: Data is set when the switch of the CPU module is changed. • Card insertion/removal: Data is set when an SD memory card is inserted or removed. • Writing: Data is set when a user performs a writing operation. • During END: Data is set when END processing is performed.

Point

Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

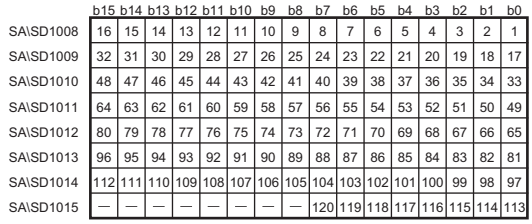
System information

The following is a list of safety special registers relating to system information. (Safety CPU only)

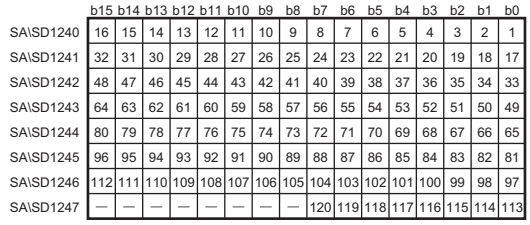
No.	Name	Data stored	Details	Set by (setting timing)
SA\SD205	Safety operation mode	Safety operation mode	The safety operation mode is stored. (The applicable bit turns on.)  (1) TEST MODE (2) SAFETY MODE (3) SAFETY MODE (wait-for-restart)	S (Status change)

Safety information (RnSFCPU)

The following is a list of safety special registers relating to safety information. (Safety CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SA\SD1008 to SA\SD1015	Safety refresh communication status of each safety connection (1st module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the 1st CC-Link IE Field Network module (the one whose start I/O number is stored in SD1904) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223.  1 to 120: Safety connection number —: Fixed to 0	S (Status change)
SA\SD1016 to SA\SD1023	Safety refresh status of each safety connection (2nd module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the 2nd CC-Link IE Field Network module (the one whose start I/O number is stored in SD1905) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1024 to SA\SD1031	Safety refresh status of each safety connection (3rd module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the 3rd CC-Link IE Field Network module (the one whose start I/O number is stored in SD1906) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1032 to SA\SD1039	Safety refresh status of each safety connection (4th module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the 4th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1907) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1040 to SA\SD1047	Safety refresh status of each safety connection (5th module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the 5th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1908) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)

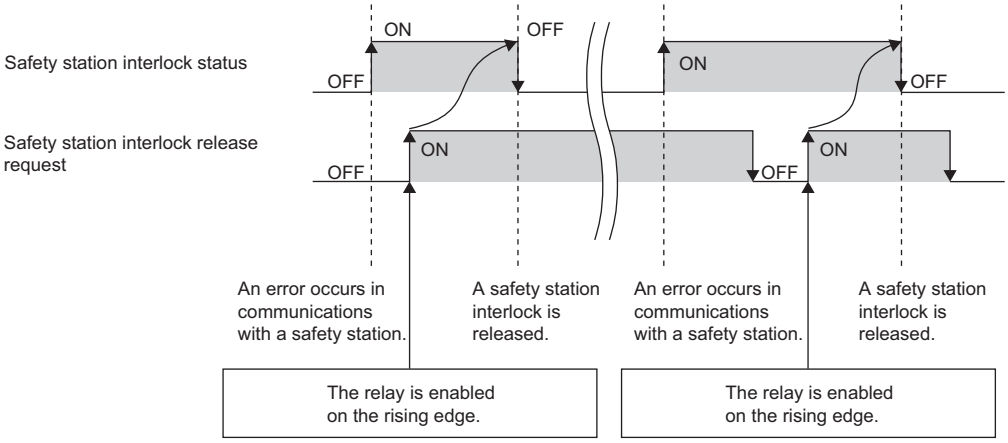
No.	Name	Data stored	Details	Set by (setting timing)
SA\SD1048 to SA\SD1055	Safety refresh status of each safety connection (6th module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the 6th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1909) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1056 to SA\SD1063	Safety refresh status of each safety connection (7th module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the 7th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1910) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1064 to SA\SD1071	Safety refresh communication status of each safety connection (8th module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the 8th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1911) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1088	CC-Link IE Field safety communications information storage area module specification	1 to 8:Target module	• The CC-Link IE Field Network master/local module targeted for safety communication status check is specified. • To reflect the value set in this register, turn on SA\SM1088. • If an unmounted module is specified, 0 is stored in SA\SD1090 to SA\SD1097 and SA\SD1104 to SA\SD1223.	U (Request)
SA\SD1089	CC-Link IE Field safety communications information storage area module	1 to 8:Target module	• The module number specified in SA\SD1088 is stored. • Data for the module displayed in this register is stored in SA\SD1090 to SA\SD1097 and SA\SD1104 to SA\SD1223.	S (Status change)
SA\SD1090 to SA\SD1097	Safety communication setting of each safety connection (1st to 8th module)	0: Not set 1: Set	The safety communication setting status of safety connections of the module stored in SA\SD1089 is stored. b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SA\SD1090: 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 SA\SD1091: 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 SA\SD1092: 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 SA\SD1093: 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 SA\SD1094: 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 SA\SD1095: 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 SA\SD1096: 112 111 110 109 108 107 106 105 104 103 102 101 100 99 98 97 SA\SD1097: — — — — — — — — 120 119 118 117 116 115 114 113 1 to 120: Safety connection number —: Fixed to 0	S (Initial)
SA\SD1104 to SA\SD1223	Safety communication status of each safety connection (1st to 8th module) Safety connection No. 1 to 120	Safety communication status of safety connection numbers 1 to 120	• The safety communication status of safety connections of the module stored in SA\SD1089 is stored. • 0 is stored if not used in SA\SD1104 to SA\SD1223, or if own station. 0H: Performing safety refresh communications 10 to 15H: Performing safety initial communications 1A40H, 1A50H, 1A51H, 1A52H, 1A60H, 1A61H, 1A62H, 1A63H, 1A64H, 1A65H, 1A66H, 1A70H, 1A71H, 1A72H, 1A73H: ☞ Page 512 List of error codes	S (Status change)
SA\SD1232 to SA\SD1239	Interlock status of each safety connection (1st module)	0: Not interlocked 1: Interlocked	After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the error connection is set to 1. b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SA\SD1232: 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 SA\SD1233: 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 SA\SD1234: 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 SA\SD1235: 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 SA\SD1236: 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 SA\SD1237: 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 SA\SD1238: 112 111 110 109 108 107 106 105 104 103 102 101 100 99 98 97 SA\SD1239: — — — — — — — — 120 119 118 117 116 115 114 113 1 to 120: Safety connection number —: Fixed to 0	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SA\SD1240 to SA\SD1247	Interlock release request for each safety connection (1st module)	0: Do not release the interlock. 1: Release the interlock.	Change the number for releasing each safety connection interlock status from 0 to 1. (The number does not automatically change to 0 after execution is complete.)*1  1 to 120: Safety connection number —: Fixed to 0	U (Request)
SA\SD1248 to SA\SD1255	Interlock status of each safety connection (2nd module)	0: Not interlocked 1: Interlocked	After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the error connection is set to 1. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1256 to SA\SD1263	Interlock release request for each safety connection (2nd module)	0: Do not release the interlock. 1: Release the interlock.	Change the number for releasing each safety connection interlock status from 0 to 1. (The number does not automatically change to 0 after execution is complete.)*1 The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	U (Request)
SA\SD1264 to SA\SD1271	Interlock status of each safety connection (3rd module)	0: Not interlocked 1: Interlocked	After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the error connection is set to 1. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1272 to SA\SD1279	Interlock release request for each safety connection (3rd module)	0: Do not release the interlock. 1: Release the interlock.	Change the number for releasing each safety connection interlock status from 0 to 1. (The number does not automatically change to 0 after execution is complete.)*1 The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	U (Request)
SA\SD1280 to SA\SD1287	Interlock status of each safety connection (4th module)	0: Not interlocked 1: Interlocked	After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the error connection is set to 1. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1288 to SA\SD1295	Interlock release request for each safety connection (4th module)	0: Do not release the interlock. 1: Release the interlock.	Change the number for releasing each safety connection interlock status from 0 to 1. (The number does not automatically change to 0 after execution is complete.)*1 The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	U (Request)
SA\SD1296 to SA\SD1303	Interlock status of each safety connection (5th module)	0: Not interlocked 1: Interlocked	After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the error connection is set to 1. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1304 to SA\SD1311	Interlock release request for each safety connection (5th module)	0: Do not release the interlock. 1: Release the interlock.	Change the number for releasing each safety connection interlock status from 0 to 1. (The number does not automatically change to 0 after execution is complete.)*1 The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	U (Request)
SA\SD1312 to SA\SD1319	Interlock status of each safety connection (6th module)	0: Not interlocked 1: Interlocked	After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the error connection is set to 1. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1320 to SA\SD1327	Interlock release request for each safety connection (6th module)	0: Do not release the interlock. 1: Release the interlock.	Change the number for releasing each safety connection interlock status from 0 to 1. (The number does not automatically change to 0 after execution is complete.)*1 The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	U (Request)
SA\SD1328 to SA\SD1335	Interlock status of each safety connection (7th module)	0: Not interlocked 1: Interlocked	After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the error connection is set to 1. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SA\SD1240	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SA\SD1241	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
SA\SD1242	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
SA\SD1243	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
SA\SD1244	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65
SA\SD1245	96	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81
SA\SD1246	112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97
SA\SD1247	—	—	—	—	—	—	—	—	120	119	118	117	116	115	114	113

No.	Name	Data stored	Details	Set by (setting timing)
SA\SD1336 to SA\SD1343	Interlock release request for each safety connection (7th module)	0: Do not release the interlock. 1: Release the interlock.	Change the number for releasing each safety connection interlock status from 0 to 1. (The number does not automatically change to 0 after execution is complete.)*1 The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	U (Request)
SA\SD1344 to SA\SD1351	Interlock status of each safety connection (8th module)	0: Not interlocked 1: Interlocked	After safety communication error is detected and the safety connection is interlocked, the bit corresponding to the error connection is set to 1. The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1352 to SA\SD1359	Interlock release request for each safety connection (8th module)	0: Do not release the interlock. 1: Release the interlock.	Change the number for releasing each safety connection interlock status from 0 to 1. (The number does not automatically change to 0 after execution is complete.)*1 The bit arrangement of the safety connection No. is same as that of the 1st module. (Note that the SA\SD numbers differ.)	U (Request)

*1 The safety station interlock release request is executed when the bit rises, and the interlock status is then released. To release the interlock again, it is necessary to turn the safety station interlock release request off and then on again. Turn the safety station interlock release request on with the Safety CPU after eliminating the cause of any communication errors. If performing safety communications between Safety CPUs, it is necessary to do so with both Safety CPUs.



Appendix 8 Buffer Memory

The buffer memory is memory used with the following applications.

Module	Application
CPU module	Stores values such as Ethernet function setting values.
Safety function module	Stores the safety function module status (such as diagnostics information and system information). (☞ Page 635 List of Special Register Areas)

The buffer memory content returns to its default status (initial value) when the CPU module is powered off or is reset.

List of buffer memory addresses

CPU module

The following table lists the CPU module buffer memory addresses.

■Ethernet function

For a list of buffer memory addresses as well as details relating to the Ethernet function, refer to the following.
(📖 MELSEC iQ-R Ethernet User's Manual (Application))

■CC-Link IE Field Network Basic function

The following table lists buffer memory addresses relating to the CC-Link IE Field Network Basic function. (Programmable controller CPU only) (☞ Page 690 Details on buffer memory addresses)

Address	Name	Data stored
Un\G1024	Total number of connected stations	The total number of connected stations set in parameter is stored.
Un\G1025	Reserved station specification status	The reserved station specification status of the slave station specified in parameter is stored.
Un\G1026 to Un\G1029	Reserved station specification status of each station	The reserved station specification status is stored.
Un\G1030 to Un\G1032, Un\G1034 to Un\G1036, Un\G1038 to Un\G1040, Un\G1042 to Un\G1044	Link scan information	The link scan time during cyclic transmission is stored.
Un\G1050	Diagnostic information display request	When a station number (1 to 16) to be diagnosed is specified in 'Diagnostic request information' (Un\G1051) and the bit 0 of 'Diagnostic information display request' (Un\G1050) is turned off and on, the diagnostic information of the specified slave station is stored in 'Diagnostic information 1' (Un\G1053 to Un\G1064) and 'Diagnostic information 2' (Un\G1068 to Un\G1077).
Un\G1051	Diagnostic request information	
Un\G1052	Diagnostic information status flag	
Un\G1053 to Un\G1064	Diagnostic information 1	
Un\G1068 to Un\G1077	Diagnostic information 2	

■Safety information (RnSFCPU)

The following table lists a buffer memory address relating to the Safety information. (Safety CPU only)

Address	Name	Data stored
Un\G6	Pair version	The Safety CPU pair version is stored in decimal notation.

Safety function module

The following table lists buffer memory addresses of the safety function module.

Address	Name	Data stored
Un\G0	Latest self-diagnostics error code	Safety function module error codes are stored in a hexadecimal value when the diagnostics detects an error.
Un\G1	Latest self-diagnostics error time	The year value (four digits) for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G2		The month value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G3		The day value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G4		The hour value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G5		The minute value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G6		The second value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G7		The day value for the date/time when Un\G0 data was updated is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)
Un\G10 to Un\G25	Self-diagnostics error code 1 to 16	A maximum of 16 types of safety function module error codes are stored in Un\G10 onwards when the diagnostics detects an error. (The same error code as one already stored in Un\G10 onwards is not stored). The 17th error code onwards are not stored. Also, error codes are not stored when 16 types of error codes have already been stored in Un\G10 to Un\G25.
Un\G50	Error clear	Set 1 in Un\G50 to clear the error. After the error is cleared, 0 is automatically set in Un\G50 by the system.
Un\G52	LED status	This register stores the information that indicates the LED status (0: off, 1: on, 2: flashing) of the safety function module in the following bit patterns. • b0 to b1: READY • b2 to b3: ERROR • b4 to b5: PROGRAM RUN • b6 to b7: SAFETY COM RUN • b8 to b9: SAFETY COM ERR • b10 to b11: TEST
Un\G54	Safety cycle processing time execution cycle error count	The number of times that cycles for which safety cycle processing was not executed at the safety function module was detected is stored.
Un\G60	Pair version	The safety function module pair version is stored in decimal notation.
Un\G62	Current safety cycle time (ms unit)	• The current safety function module safety cycle processing time is stored. (The time is measured in 1 μs units.) Un\G62: The ms part is stored. (Range: 0 to 65535) Un\G63: The .s part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.
Un\G63	Current safety cycle time (μs unit)	
Un\G64	Minimum safety cycle time (ms unit)	• The minimum safety function module safety cycle processing time value is stored. (The time is measured in 1 μs units.) Un\G64: The ms part is stored. (Range: 0 to 65535) Un\G65: The .s part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.
Un\G65	Minimum safety cycle time (μs unit)	
Un\G66	Maximum safety cycle time (ms unit)	• The maximum safety function module safety cycle processing time value is stored. (The time is measured in 1 μs units.) Un\G66: The ms part is stored. (Range: 0 to 65535) Un\G67: The .s part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.
Un\G67	Maximum safety cycle time (μs unit)	
Un\G80	Detailed information 1 information category	The detailed information 1 information category code is stored.
Un\G81 to Un\G111	Detailed information 1	• Detailed information 1 corresponding to the safety function module error code (Un\G0) is stored. • The stored information contains the following eight types. 0: N/A 1: Program position information 2: Drive number and file name 4: Parameter information 5: System configuration information 6: Number of times information 7: Time information 24: Failure information 34: Continuous RUN prevention setting in TEST MODE • The type of detailed information 1 can be obtained using Un\G80. (The bit patterns for values in the "Detailed information 1 information category code" stored in Un\G80 correspond to those in SD81 to SD111.) (☞ Page 635 List of Special Register Areas)

Address	Name	Data stored
Un\G112	Detailed information 2 information category	The detailed information 2 information category code is stored.
Un\G113 to Un\G143	Detailed information 2	• Detailed information 2 corresponding to the safety function module error code (Un\G0) is stored. • The stored information contains the following six types. 0: N/A 2: Drive number and file name 3: Annunciator number 4: Parameter information 5: System configuration information 25: Process control instruction processing information 29: Other station error information (CC-Link IE Field) • The type of detailed information 1 can be obtained using Un\G112. (The bit patterns for values in the "Detailed information 2 information category code" stored in Un\G112 correspond to those in SD113 to SD143.) (☞ Page 635 List of Special Register Areas)
Un\G150	Program memory writing	• b0: This relay switches to on when a write error is detected during write operation to the program memory. This relay switches to off when the write instruction is issued. • b1: This relay is on when the write process to the program memory is in progress. The relay is switched to off when the write process is completed. • b2: This relay switches to on when the number of program memory rewriting operations reaches 100000 (The safety function module must be replaced).
Un\G152	Program memory write (transfer) status	This register displays the write (transfer) status to the program memory with a percentage (0 to 100%). The initial value is "0". Upon completion of writing, this register is set to "100". It is set to "0" at the time when the write command is issued.
Un\G153 to Un\G154	Program memory write count index	• This register indicates the index value for the number of write operations to the program memory up to now (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value continues to be counted even when it exceeds 100000). If the index value exceeds 100000, the safety function module must be replaced.
Un\G155	Data memory writing	• b0: This relay switches to on when a write error is detected during write operation to the data memory. This relay switches to off when the write instruction is issued. • b1: This relay is on when the write process to the data memory is in progress. The relay is switched to off when the write process is completed. • b2: This relay switches to on when the number of data memory rewriting operations reaches 100000 (The safety function module must be replaced).
Un\G157	Data memory write (transfer) status	This register displays write (transfer) status to the data memory with a percentage. (0 to 100%). The initial value is "0". Upon completion of writing, this register is set to "100". It is set to "0" at the time when the write command is issued.
Un\G158 to Un\G159	Index for the number of data memory write operations	• This register indicates the index value for the number of write operations to the data memory up to now (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value continues to be counted even when it exceeds 100000). If the index value exceeds 100000, the safety function module must be replaced.
Un\G160	System memory writing	• b0: This relay switches to on when a write error is detected during write operation to the system memory (Flash ROM)*1. This relay switches to off when the write instruction is issued. • b1: This relay switches to on when the write process to the system memory (Flash ROM)*1 is in progress, and switches to off when the write process is complete. • b2: This relay switches to on when the number of system memory (Flash ROM)*1 rewrite operations reaches 100000. (The safety function module must be replaced).
Un\G163 to Un\G164	Index for the number of system memory write operations	• This register indicates the index value for the number of write operations to the system memory (Flash ROM)*1 up to now. (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value continues to be counted even when it exceeds 100000). If the index value exceeds 100000, the safety function module must be replaced.

*1 This is the memory used by the system when the safety function module is executing functions.

Details on buffer memory addresses

The following table lists items in the list.

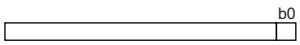
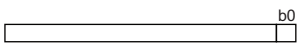
Item	Description
Address	Buffer memory address of the CPU module
Name	Buffer memory name of the CPU module
Data stored	Data stored in the CPU module buffer memory and its meaning
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (program, engineering tool, GOT, or other testing operations from external device) • U/S: User and system <Set timing> • Every END: Data is set every time END processing is performed. • Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN). • Status change: Data is set when the status is changed. • Error: Data is set when an error occurs. • Instruction execution: Data is set when an instruction is executed. • Request: Data is set when requested by a user (using the special relay). • Switch change: Data is set when the switch of the CPU module is changed. • Card insertion/removal: Data is set when an SD memory card is inserted or removed. • Writing: Data is set when a user performs a writing operation. • During END: Data is set when END processing is performed.

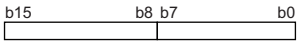
Point

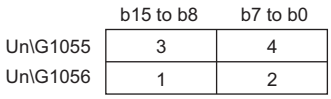
Do not change the data set by the system in a program or by a device test. Doing so may cause an unintended operation.

CC-Link IE Field Network Basic function

The following table lists buffer memory areas of the CPU module relating to the CC-Link IE Field Network Basic function. (Programmable controller CPU only)

Address	Name	Data stored	Details	Set by (setting timing)
Un\G1024	Total number of connected stations	Total number of connected stations	The total number of connected stations set in parameter is stored. Range: 1 to 16	S (Initial)
Un\G1025	Reserved station specification status	Reserved station specification status	The reserved station specification status of the slave station specified in parameter is stored. (0: Not specified, 1: Specified)  b0: Reserved station specification status b1 to b15: Empty (fixed to 0) The station number that is specified as a reserved station can be checked in 'Reserved station specification status of each station' (Un\G1026).	S (Initial)
Un\G1026 to Un\G1029	Reserved station specification status of each station	Reserved station specification status of each station	The reserved station specification status is stored using the following bit pattern. (Off: Other than the reserved station, On: Reserved station) b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 Un\G1026: 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 Un\G1027: 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 Un\G1028: 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 Un\G1029: 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 The numbers in the figure indicate station numbers. (Condition) • Only the bit of the start station number turns on. • The status is not stored for the station numbers after the maximum station number.	S (Initial)
Un\G1030	Link scan information	Group No.1 maximum link scan	The maximum link scan time value during cyclic transmission is stored. (Unit: ms)	S (Status change)
Un\G1031		Group No.1 minimum link scan	The minimum link scan time value during cyclic transmission is stored. (Unit: ms)	S (Status change)
Un\G1032		Group No.1 current link scan	The current link scan time value during cyclic transmission is stored. (Unit: ms)	S (Status change)
Un\G1034		Group No.2 maximum link scan	The maximum link scan time value during cyclic transmission is stored. (Unit: ms)	S (Status change)
Un\G1035		Group No.2 minimum link scan	The minimum link scan time value during cyclic transmission is stored. (Unit: ms)	S (Status change)
Un\G1036		Group No.2 current link scan	The current link scan time value during cyclic transmission is stored. (Unit: ms)	S (Status change)
Un\G1038		Group No.3 maximum link scan	The maximum link scan time value during cyclic transmission is stored. (Unit: ms)	S (Status change)
Un\G1039		Group No.3 minimum link scan	The minimum link scan time value during cyclic transmission is stored. (Unit: ms)	S (Status change)
Un\G1040		Group No.3 current link scan	The current link scan time value during cyclic transmission is stored. (Unit: ms)	S (Status change)
Un\G1042		Group No.4 maximum link scan	The maximum link scan time value during cyclic transmission is stored. (Unit: ms)	S (Status change)
Un\G1043		Group No.4 minimum link scan	The minimum link scan time value during cyclic transmission is stored. (Unit: ms)	S (Status change)
Un\G1044		Group No.4 current link scan	The current link scan time value during cyclic transmission is stored. (Unit: ms)	S (Status change)
Un\G1050	Diagnostic information display request	Diagnostic information display request	After the END instruction of the scan where the bit 0 is turned off and on is executed, the diagnostic information of a slave station specified in 'Diagnostic request information' (Un\G1051) is read to Un\G1052 to Un\G1077. When reading of the diagnostic information has completed at END processing, 0 is stored.  b0: Diagnostic information display request b1 to b15: Empty (fixed to 0)	S (Status change)/*1U

Address	Name	Data stored	Details	Set by (setting timing)
Un\G1051	Diagnostic request information	Diagnostic request information	Specify a slave station number whose diagnostic information is to be displayed. Range: 1 to 64 • For the programmable controller CPU with firmware version earlier than "28", the range is from 1 to 16.	U
Un\G1052	Diagnostic information status flag	Diagnostic information status flag	After the END instruction of the scan where the bit 0 of 'Diagnostic information display request' (Un\G1050) is turned off and on is executed, the status (valid or invalid) of diagnostic information (Diagnostic information 1, Diagnostic information 2) of the slave station specified in 'Diagnostic request information' (Un\G1051) is stored. (Valid: 1, Invalid: 0)  b0 to b7: Diagnostic information 1 b8 to b15: Diagnostic information 2 • If the station number of the slave station that is specified in 'Diagnostic request information' (Un\G1051) is the start station number of the occupied stations and the cyclic transmission is performed for the slave station, 1 is stored in b0 to b7 and b8 to b15. (If the specified slave station is a reserved station, 0 is stored in b8 to b15.) • If the station number of the slave station that is specified in 'Diagnostic request information' (Un\G1051) is other than the start station number of the occupied stations or the cyclic transmission is not performed for the slave station, 0 is stored in b0 to b7 and b8 to b15. • When b0 to b7 are valid, the number of occupied stations, group number, IP address, the accumulated number of timeouts, and the accumulated number of disconnection detection are stored in 'Diagnostic information 1' (Un\G1053 to Un\G1064). When invalid, 0 is stored in 'Diagnostic information 1' (Un\G1053 to Un\G1064). • When b8 to b15 are valid, the Manufacturer code, model code, device version, module information, error code, and detailed module information are stored in 'Diagnostic information 2' (Un\G1068 to Un\G1077). When invalid, 0 is stored in 'Diagnostic information 2' (Un\G1068 to Un\G1077).	S (Status change)

Address	Name	Data stored	Details	Set by (setting timing)
Un\G1053 to Un\G1064	Diagnostic information 1*3	Diagnostic information 1	When 1 (valid) is stored in b0 to b7 of Un\G1052, the number of occupied stations, group number, IP address, the accumulated number of timeouts, and the accumulated number of disconnection detection are stored. When 0 (invalid) is stored in b0 to b7 of Un\G1052, 0 is stored. ■Un\G1053: Number of occupied stations ■Un\G1054: Group number ■Un\G1055: IP address (lower) ■Un\G1056: IP address (upper)  1 to 4: First octet to fourth octet When the IP address has not been set in the parameter, 0 is stored. ■Un\G1063: Accumulated number of timeouts After the END instruction of the scan where the bit 0 of 'Diagnostic information display request' (Un\G1050) is turned off and on is executed, the accumulated number of timeouts that occurred in a slave station specified in 'Diagnostic request information' (Un\G1051) is stored. • 0: No timeouts • 1 to 65535: Number of timeouts (accumulated number)*2 ■Un\G1064: Accumulated number of disconnection detection After the END instruction of the scan where the bit 0 of 'Diagnostic information display request' (Un\G1050) is turned off and on is executed, the accumulated number of disconnections that detected in a slave station specified in 'Diagnostic request information' (Un\G1051) is stored. • 0: No disconnections • 1 to 65535: Number of disconnection detection (accumulated number)*2	S (Status change)
Un\G1068 to Un\G1077	Diagnostic information 2*3	Diagnostic information 2	When Diagnostic information 2 is valid (1 is stored in b8 to b15 of Un\G1052), the manufacturer code, model code, device version, module information, error code, and detailed module information are stored. When Diagnostic information 2 is invalid (0 is stored in b8 to b15 of Un\G1052), 0 is stored. ■Un\G1068: Manufacturer code ■Un\G1070: Model code (lower) ■Un\G1071: Model code (upper) ■Un\G1072: Device version ■Un\G1074: Module information ■Un\G1075: Error code ■Un\G1076: Detailed module information (lower) ■Un\G1077: Detailed module information (upper)	S (Status change)

*1 Data is stored at END processing where the value is changed.
*2 When the count exceeds 65535, counting is continued from 1 again.
*3 A slave station specified is being disconnected, the information immediately before the disconnection is stored.

A

Appendix 9 Processing Time

The scan time of the CPU module is the sum of the instruction execution time, the program execution time, and the END processing time. Each of the processing time that constitutes the scan time is as follows.

Instruction execution time

Total of each instruction processing time used for the program to be executed by CPU module For the processing time of each instruction, refer to the following.

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

Program execution time

Shows the processing time when multiple programs are executed. When the interrupt program is executed, the overhead time during the interrupt program being executed is also included.

Processing time when multiple programs are executed (program switching time)

When multiple programs are executed, the scan time becomes longer.

■When multiple programs run

The processing time [μs] with multiple programs run is given by: the number of program files × 1.2.

Overhead time at executing interrupt program and fixed scan execution type program

Shows the overhead time when executing the interrupt program and fixed scan execution type program.

Point

The processing time for the interrupt program and fixed scan execution type program is as shown below.
Instruction processing time for each instruction + Overhead time + Each refresh processing time (when setting)

■Overhead time when executing the interrupt program

The overhead time when executing the interrupt program includes the one before executing the interrupt program and the other when the interrupt program finished.

• Overhead time before executing the interrupt program

Interrupt factor	Condition	Overhead time
Internal timer interrupt (I28 to I31)	Saves/stores the file register (R) block number.	19.5μs
	Not save/store the file register (R) block number.	12.9μs
Inter-module synchronous interrupt (I44)	Saves/stores the file register (R) block number.	25.4μs
	Not save/store the file register (R) block number.	17.0μs
Multiple CPU synchronous interrupt (I45)	Saves/stores the file register (R) block number.	24.2μs
	Not save/store the file register (R) block number.	16.1μs
High-speed internal timer interrupt 2 (I48), high-speed internal timer interrupt 1 (I49)	Saves/stores the file register (R) block number.	25.8μs
	Not save/store the file register (R) block number.	18.1μs
Interrupt from module (I0 to I15, I50 to I1023)	Saves/stores the file register (R) block number.	19.1μs
	Not save/store the file register (R) block number.	12.8μs

• Overhead time when finishing the interrupt program

Interrupt factor	Condition	Overhead time
Internal timer interrupt (I28 to I31)	Saves/stores the file register (R) block number.	16.0μs
	Not save/store the file register (R) block number.	8.5μs
Inter-module synchronous interrupt (I44)	Saves/stores the file register (R) block number.	19.0μs
	Not save/store the file register (R) block number.	11.4μs
Multiple CPU synchronous interrupt (I45)	Saves/stores the file register (R) block number.	18.7μs
	Not save/store the file register (R) block number.	10.5μs
High-speed internal timer interrupt 2 (I48), high-speed internal timer interrupt 1 (I49)	Saves/stores the file register (R) block number.	19.4μs
	Not save/store the file register (R) block number.	10.3μs
Interrupt from module (I0 to I15, I50 to I1023)	Saves/stores the file register (R) block number.	15.7μs
	Not save/store the file register (R) block number.	8.5μs

■Overhead time when executing the fixed scan execution type program

Shows the Overhead time when executing the fixed scan execution type program.

Condition	Overhead time
Saves/stores the file register (R) block number.	37.9μs
Not save/store the file register (R) block number.	20.6μs

Overhead time when executing safety cycle processing

The safety cycle processing execution time indicates the overhead time when executing safety cycle processing.

■Overhead time before starting safety cycle processing

Overhead time before starting safety cycle processing [μs] = Total standard/safety shared label usage capacity*1 × KM1 + KM2

*1 The capacity indicates the total usage capacity for labels defined in the standard/safety shared label setting file. This can be verified in SD1903. (☞ Page 635 List of Special Register Areas)
This is not the capacity set in the Safety CPU parameter Standard/Safety Shared Label Area Capacity.

Constant	Constant value
KM1	0.16
KM2	100

■Overhead time when ending safety cycle processing

The overhead time [μs] when ending safety cycle processing is 40 μs.

Processing time of safety communications

This section describes the processing time of safety communications.

■Input processing time of safety communications

Input processing time of safety communications [μs] = (Number of connections set in the safety communication setting × KM1) + (Number of mountable CC-Link IE Field Network modules set for safety communications × KM2) + KM3

Constant	Constant value
KM1	5
KM2	15
KM3	70

■Output processing time of safety communications

Output processing time of safety communications [μs] = (Number of connections set in the safety communication setting × KM1) + (Number of mountable CC-Link IE Field Network modules set for safety communications × KM2) + KM3

Constant	Constant value
KM1	12
KM2	10
KM3	85

■Processing time when multiple programs are executed (safety program switching time)

The processing time [μs] with multiple programs run is given by: the number of program files × 1.0.

END processing time

The END processing time is as shown below.

- Common processing time
- I/O refresh processing time
- Link refresh processing time for the network module
- Link refresh processing time for CC-Link IE Field Network Basic
- Intelligent function module refresh processing time
- Multiple CPU refresh processing time
- Prolonged time of END processing when executing each function
- Device/label access service processing time
- Safety diagnostics processing time
- Safety operation mode switching time
- Safety data identify check processing time
- Tracking transfer time (☞ Page 714 Increase in the scan time due to tracking transfer)

Common processing time

Shows the common processing time of CPU module to be treated by the system.

Condition	Common processing time			
	Programmable controller CPU	Process CPU (process mode)	Process CPU (redundant mode)	Safety CPU
One unit of CPU module (module is not mounted)	85μs	85μs	175μs	0.38 to 4.00ms

I/O refresh processing time

The I/O refresh processing time for module mounted on main base unit and extension base unit can be calculated by the following formula.

I/O refresh processing time [μs] = ((Number of input refresh points[*1] × KM1) + (Number of unit having number of input points × KM2) + KM3[*3]) + ((Number of output refresh points[*2] × KM4) + (Number of unit having number of output points × KM5) + KM6)[*3]

*1 Shows the numeric value that is obtained through the division of number of input points by 16.
*2 Shows the numeric value that is obtained through dividing the number of output points by 16.
*3 When the number of I/O points is 0, this value is handled as 0.

Condition				Constant value
MELSEC iQ-R series module	Main base unit	Input	KM1	0.04
			KM2	0.56
			KM3	11.2
		Output	KM4	0.02
			KM5	0.94
			KM6	6.30
	Extension base unit	Input	KM1	0.04
			KM2	0.56
			KM3	11.2
		Output	KM4	0.02
			KM5	0.94
			KM6	6.30
Q series module	RQ extension base unit	Input	KM1	1.15
			KM2	1.33
			KM3	24.0
		Output	KM4	0.03
			KM5	1.20
			KM6	6.21
	Q series extension base unit	Input	KM1	1.85
			KM2	1.98
			KM3	39.4
		Output	KM4	0.02
			KM5	1.29
			KM6	6.01

A

Link refresh processing time for the network module

This section describes the link refresh processing time for the network module.

■Link refresh processing time for the CC-Link IE Controller Network module

The link refresh processing time between the CC-Link IE Controller Network module and the modules mounted on the main base unit and on the extension base unit can be calculated by the following formula.

$\alpha T, \alpha R$ [ms] = KM1 + KM2 × ((LB + LX + LY + SB) ÷ 16 + LW + SW) + αU

αU [ms] = KM3 × (SBU ÷ 16 + SWU)

- αT: Link refresh time (sending side)
- αR: Link refresh time (receiving side)
- αU: Unit label (SB/SW) refresh time
- LB: Total number of points of link relay (LB) that is refreshed by the station*1
- LW: Total number of points of link register (LW) that is refreshed by the station*1
- LX: Total number of points of link input (LX) that is refreshed by the station*1
- LY: Total number of points of link output (LY) that is refreshed by the station*1
- SB: Number of points of link special relay (SB)*2
- SW: Number of points of link special register (SW)*2
- SBU: Number of points of link special relay (SB)*3
- SWU: Number of points of link special register (SW)*3

Condition		Constant value
KM1 ($\times 10^{-3}$)		27
KM2 ($\times 10^{-3}$)	Main base unit*4	0.01
	Extension base unit*4	0.12
KM3 ($\times 10^{-3}$)	Main base unit*4	0.03
	Extension base unit*4	0.15

*1 Shows the total number of points of link device that has been set by the refresh parameter setting and the network range assignment setting. Note that the number of points that has been assigned in the reserved station is excluded.

*2 Shows the number of points when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.

*3 Shows the number of points when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.

*4 Shows the base unit type provided with network module targeted for refresh.

■Link refresh processing time for the CC-Link IE Field Network module

The link refresh processing time between the CC-Link IE Field Network module and the modules mounted on the main base unit and on the extension base unit can be calculated by the following formula.

$\alpha T, \alpha R$ [ms] = KM1 + KM2 × ((RX + RY + SB) ÷ 16 + RW_r + RW_w + SW) + αU

αU [ms] = KM3 × (SBU ÷ 16 + SWU)

- αT: Link refresh time (sending side)
- αR: Link refresh time (receiving side)
- αU: Unit label (SB/SW) refresh time
- RX: Total number of points of remote input (RX) refreshed by the master station/local station*1
- RY: Total number of points of remote output (RY) refreshed by the master station/local station*1
- RWw: Total number of points of remote register (RW_w) refreshed by the master station/local station*1
- RWr: Total number of points of remote register (RW_r) refreshed by the master station/local station*1
- SB: Number of points of link special relay (SB)*2
- SW: Number of points of link special register (SW)*2
- SBU: Number of points of link special relay (SB)*3
- SWU: Number of points of link special register (SW)*3

Condition		Constant value
KM1 ($\times 10^{-3}$)		27
KM2 ($\times 10^{-3}$)	Main base unit*4	0.01
	Extension base unit*4	0.12
KM3 ($\times 10^{-3}$)	Main base unit*4	0.03
	Extension base unit*4	0.15

*1 Shows the range specified in the refresh parameter and the total number of points of link device that has been set by the network configuration setting. Note that the number of points that has been assigned in the reserved station is excluded.

*2 Shows the number of points when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.

*3 Shows the number of points when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.

*4 Shows the base unit type provided with network module targeted for refresh.

A

■Link refresh processing time for the CC-Link module

The link refresh processing time between the CC-Link module and the modules mounted on the main base unit and on the extension base unit can be calculated by the following formula. (Remote net Ver.1 mode, Remote net Ver.2 mode)

$\alpha T, \alpha R$ [ms] = KM1 + KM2 × ((RX + RY + SB) ÷ 16 + RW_r + RW_w + SW) + αU

αU [ms] = KM3 × (SBU ÷ 16 + SWU)

- αT: Link refresh time (sending side)
- αR: Link refresh time (receiving side)
- αU: Unit label (SB/SW) refresh time
- RX: Total number of points of remote input (RX) refreshed by the master station/local station*1
- RY: Total number of points of remote output (RY) refreshed by the master station/local station*1
- RWw: Total number of points of remote register (RW_w) refreshed by the master station/local station*1
- RWr: Total number of points of remote register (RW_r) refreshed by the master station/local station*1
- SB: Number of points of link special relay (SB)*2
- SW: Number of points of link special register (SW)*2
- SBU: Number of points of link special relay (SB)*3
- SWU: Number of points of link special register (SW)*3

Condition		Constant value
KM1 ($\times 10^{-3}$)		27
KM2 ($\times 10^{-3}$)	Main base unit*4	0.01
	Extension base unit*4	0.12
KM3 ($\times 10^{-3}$)	Main base unit*4	0.03
	Extension base unit*4	0.15

*1 Shows the range specified in the refresh parameter and the total number of points of link device that has been set by the network configuration setting. Note that the number of points that has been assigned in the reserved station is excluded.

*2 Shows the number of points when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.

*3 Shows the number of points when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.

*4 Shows the base unit type provided with network module targeted for refresh.

■Link refresh processing time for the MELSECNET/H module

The link refresh processing time between the MELSECNET/H module and the modules mounted on the RQ extension base unit and on the Q series extension base unit can be calculated by the following formula.

$\alpha T, \alpha R$ [ms] = KM1 + KM2 × ((LB + LX + LY + SB) ÷ 16 + LW + SW)

- αT: Link refresh time (sending side)
- αR: Link refresh time (receiving side)
- LB: Total number of points of link relay (LB) that is refreshed by the station*1
- LW: Total number of points of link register (LW) that is refreshed by the station*1
- LX: Total number of points of link input (LX) that is refreshed by the station*1
- LY: Total number of points of link output (LY) that is refreshed by the station*1
- SB: Number of points of link special relay (SB)
- SW: Number of points of link special register (SW)

Condition		Constant value
KM1 ($\times 10^{-3}$)		65
KM2 ($\times 10^{-3}$)	RQ extension base unit (RQ6□B)*2	0.41
	Q series extension base unit (Q5□B/Q6□B)*2	0.92

*1 Shows the total number of points of link device that has been set by the refresh parameter setting and the network range assignment setting. Note that the number of points that has been assigned in the reserved station is excluded.

*2 Shows the base unit type provided with network module targeted for refresh.

Link refresh processing time for CC-Link IE Field Network Basic

Link refresh processing time for CC-Link IE Field Network Basic (the increase in END processing time for the CPU module) can be calculated by the following formula.

αT [μs] = KM1 + KM2 × (((RX + RY) ÷ 16) + RWw + RWr) + αE

αE [μs] = KM3 + KM4 × ((RX + RY) ÷ 16) + RWw + RWr)

- αT: Link refresh time
- αE: Link refresh time when the file register (R, ZR) is used*1
- RX: Total number of points of remote input (RX) refreshed by the master station*2
- RY: Total number of points of remote output (RY) refreshed by the master station*2
- RWw: Total number of points of remote register (RWw) refreshed by the master station*2
- RWr: Total number of points of remote register (RWr) refreshed by the master station*2

Condition		Constant value
KM1		12.5
KM2		0.02
KM3		14.5
KM4	Extended SRAM cassette not used	0.04
	Extended SRAM cassette used	0.11

*1 This is added when file register (R, ZR) is used.

*2 This value is determined according to the number of slave stations to be connected and the number of occupied stations.

A

Intelligent function module refresh processing time

The following calculation formula shows the refresh processing time for the intelligent function module mounted on the main base unit and the extension base unit.

Point

The refresh processing time described in this manual is for the case when the "Target" is set to "Device" in the refresh setting. For the refresh processing time of when the refreshing target is set to the module label or refresh data register (RD), refer to the manual for the module used.

Refresh processing time of one module [μs] = Read refresh time*1 + Write refresh time*1

*1 When the number of settings of read refresh (Module → CPU module) or write refresh (CPU module → Module) is 0, each processing time is 0.

■Read refresh time [μs]

Number of read refresh settings × KM1 + Refresh time for the first item (A) + Refresh time for the second item (A) + ··· + Refresh time for the nth item (A) + KM2

■Write refresh time [μs]

Number of write refresh settings × KM4 + Refresh time for the first item (B) + Refresh time for the second item (B) + ··· + Refresh time for the nth item (B) + KM5

- A: KM3 × Number of refresh transfer (word) [μs]
- B: KM6 × Number of refresh transfer (word) [μs]
- n: Number of blocks for refresh parameter*1
- KM1 to KM6: Constant value as shown below

Condition			Constant value
MELSEC iQ-R series module	KM1	Module on the main base unit → CPU module	0.98
		Module on the extension base unit → CPU module	0.98
	KM2	Module on the main base unit → CPU module	11.6
		Module on the extension base unit → CPU module	11.6
	KM3	Module on the main base unit → CPU module	0.05
		Module on the extension base unit → CPU module	0.05
	KM4	CPU module → Main base unit	0.58
		CPU module → Extension base unit	0.58
	KM5	CPU module → Main base unit	9.10
		CPU module → Extension base unit	9.10
	KM6	CPU module → Main base unit	0.01
		CPU module → Extension base unit	0.01
Q series module	KM1	Module on the RQ extension base unit → CPU module	1.47
		Module on the Q series extension base unit → CPU module	2.92
	KM2	Module on the RQ extension base unit → CPU module	21.2
		Module on the Q series extension base unit → CPU module	20.2
	KM3	Module on the RQ extension base unit → CPU module	0.38
		Module on the Q series extension base unit → CPU module	0.91
	KM4	CPU module → Module on the RQ extension base unit	0.83
		CPU module → Module on the Q series extension base unit	1.20
	KM5	CPU module → Module on the RQ extension base unit	15.8
		CPU module → Module on the Q series extension base unit	15.0
	KM6	CPU module → Module on the RQ extension base unit	0.43
		CPU module → Module on the Q series extension base unit	0.97

*1 The number of blocks for refresh parameter can be checked in "Auto Refresh Setting Total Counts" of "Module Parameter List" window. For details, refer to the manual for the module used.

Multiple CPU refresh processing time

Shows the Multiple CPU refresh processing time.

Refresh processing time [μs] = Send refresh time + Receive refresh time

Send refresh time [μs] = KM1 + KM2 × Number of points of send word

Receive refresh time [μs] = KM3 + KM4 × Number of other CPU modules + KM5 × Number of points of receive word

■Refresh (when END)

Shows the constant value when refresh is executed in END processing at CPU buffer memory area.

Constant	Constant value
KM1	6
KM2	0.03
KM3	43
KM4	21
KM5	0.04

■Refresh (when I45 is executed)

Shows the constant value when refresh is executed in Multiple CPU synchronous interrupt (I45) at fixed scan communication area.

Constant	Constant value
KM1	20
KM2	0.007
KM3	20
KM4	4
KM5	0.007

Prolonged time of END processing when executing each function

This section describes the prolonged time of END processing when executing each function.

■Latch processing time

If the latch function is used and the range for the latch time setting is effective, the scan time becomes longer. Also when the time setting is configured for the latch time setting, the scan time may be prolonged in the next END processing after the specified time has passed. The increase in scan time when the latch range is set can be calculated by the following formula.

Increase in scan time[μs] = (KM1 × Number of settings of latch range setting[*1]) + (KM2 × (Number of points of bit device with latch specified ÷ 16 + Number of points of word device with latch specified + Number of points of double word device with latch specified × 2)) + KM3

*1 The latch range (1) and the latch range (2) are counted as different device types.

Condition		Constant value
When set to "each scan"	KM1	1.00
	KM2	0.09
	KM3	1.20
When set to "hour"	KM1	1.0
	KM2	0.004
	KM3	17.5

■Data logging function

When the data logging function is executed, the scan time becomes longer. The increase in scan time can be calculated by the following formula.

Increase in scan time [μs] = KM1 + (KM2 × Number of data logging settings) + (KM3 × Number of device points of internal device[*1])

*1 Shows the total number of points of data logging setting No.1 to No.10.

Condition		Constant value (NZ1MEM-□GBSD)			
		Programmable controller CPU	Process CPU (process mode)	Process CPU (redundant mode)	Safety CPU
When all the columns are outputted under the condition of file format of Unicode text file	KM1	34	35		34
	KM2	34	37		34
	KM3	0.08	0.35		0.08

■Processing time when file register is used

When "Use File Register of Each Program" has been set through setting the file register, the scan time becomes longer. The increase in the scan time [ms] is given by: the number of program files × 0.016.

Point

When "Use Common File Register in All Programs" has been set, the scan time may not become longer.

■Memory dump function

The execution of the memory dump function results in an increase in the scan time. The increase in scan time can be calculated by the following formula.

Increase in scan time [μs] = (KM1 × Number of points[*1]) + KM2

Constant	Setting of the internal buffer capacity	Constant value
KM1	32K bytes	0.70
	256K bytes	4.70
	3040K bytes	40.00
KM2	32K bytes	320.00
	256K bytes	2290.00
	3040K bytes	16400.00

*1 Indicates the total number of points (total number of words) of devices that are set in the device area, file storage area, and refresh data register area. (unit: K words).

■Real-time monitor function

The execution of the real-time monitor function results in an increase in the scan time. The increase in scan time can be calculated by the following formula.

Increase in scan time [μs] = KM1 + (KM2 × Number of points of word device) + (KM3 × Number of points of bit device)

• In monitoring internal user devices

Constant	Constant value
KM1	149.00
KM2	3.20
KM3	0.80

• In monitoring file registers

Constant	Constant value
KM1	149.00
KM2	3.50
KM3	1.10

• In monitoring module access devices/link direct devices

Constant	Constant value
KM1	156.00
KM2	11.60
KM3	8.90

■Sequence scan synchronization sampling function

When the sequence scan synchronization sampling function is executed, the scan time becomes longer. The increase in scan time can be calculated by the following formula.

Increase in scan time [μs] = (KM1 × Number of processing times[*1]) + (KM2 × Total number of points of device that has been set[*2]) + (KM3 × Number of settings[*2]) + (KM4 × Number of modules[*3]) + KM5

*1 The number of processing times indicates the number of times to perform the processing for 64 points of the device that can be processed at a time. The number of times is calculated by the total number of points of device that has been set n ÷ 64. (The decimal point is rounded up.)

*2 For the device types and number of settings that can be set, refer to the manual for the module used.

*3 The number of modules indicates the total number of modules which has executed this function.

Constant	Constant value
KM1	8.00
KM2	0.42
KM3	0.20
KM4	24.00
KM5	85.00

Device/label access service processing time

Indicates the Device/Label access service processing time when "Processing time = One time" in the setting of Device/Label access service processing.

Condition			Device/label access service processing time
Connected via USB	Ladder block change during RUN (online program change)	100 steps are inserted into the head of program of 40K steps.	1.0ms maximum*1
	Monitor data registration	Data register (D) (Number of device points = 32 points) is registered to monitor.	0.03ms maximum
When connecting the Ethernet port (TCP)	Ladder block change during RUN (online program change)	100 steps are inserted into the head of program of 40K steps.	1.0ms maximum*1
	Monitor data registration	Data register (D) (Number of device points = 32 points) is registered to monitor.	0.07ms maximum
When connecting the Ethernet port (UDP)	Ladder block change during RUN (online program change)	100 steps are inserted into the head of program of 40K steps.	1.0ms maximum*1
	Monitor data registration	Data register (D) (Number of device points = 32 points) is registered to monitor.	0.07ms maximum

*1 When data are written to the program using a pointer (P) during RUN, the processing time is extended depending on the pointer number used. For example, when the data are written to the program using P8191 during RUN, the processing time is extended up to 3.0ms.

Safety data identify check processing time

The extended scan time when executing the safety data identify check function is 800 μs under the following conditions.

- Safety programs: 20K steps
- Create safety global label setting file (32 word definition)
- Create standard/safety shared label setting file (32 word definition)
- Safety cycle time: 10ms
- Perform safety communications for 3 connections
- Scan time: 1.5ms (when safety data identify check not executed)

Data logging function processing time

Shows the processing time taken to store the data when executing the data logging function. Shows the minimum specified time value that can collect without losing the data when executing the continuous logging.

When the file format is Unicode text file

Shows the collection interval at which data can be collected under the following conditions.

- Scan time = 1.5ms (up to 3ms)*1
- Internal buffer capacity setting = 128K bytes per setting (default setting)
- Collection setting = Time specification (data collection at time interval)
- Data setting = Data register (D) (Data-type: word-signed (decimal number type))
- Output setting = Day/time column (output format is default), Index column are outputted.
- Save setting = file switching timing: 10000 records, operation when the number of saved files exceeded: overwriting
- SD memory card: NZ1MEM-2/4/8/16GBSD used

*1 3ms (to 4.5ms) for the Process CPU (redundant mode)

Number of points		Collection interval where data can be collected	
		When the NZ1MEM-2GBSD is used	When the NZ1MEM-4/8/16GBSD is used
8 points	(8 points × 1 setting)	1.0ms	1.0ms
16 points	(16 points × 1 setting)	1.0ms	2.0ms
64 points	(64 points × 1 setting)	2.0ms	5.0ms
128 points	(128 points × 1 setting)	4.0ms	7.0ms
256 points	(128 points × 2 setting)	7.0ms	13.0ms
1280 points	(128 points × 10 setting)	24.0ms	26.0ms

When the file format is binary file

Shows the Collection interval at which data can be collected under the following conditions.

- Scan time = 1.5ms (up to 3ms)*1
- Internal buffer capacity setting = 128K bytes per setting (default setting)
- Collection setting = Time specification (data collection at time interval)
- Data setting = Data register (D) (Data-type: word-signed (decimal number type))
- Output setting = Day/time column (output format is default), Index column are outputted.
- Save setting = file switching timing: 10000 records, operation when the number of saved files exceeded: overwriting
- SD memory card: NZ1MEM-2/4/8/16GBSD used

*1 3ms (to 4.5ms) for the Process CPU (redundant mode)

Number of points		Collection interval where data can be collected	
		When the NZ1MEM-2GBSD is used	When the NZ1MEM-4/8/16GBSD is used
8 points	(8 points × 1 setting)	1.0ms	1.0ms
16 points	(16 points × 1 setting)	1.0ms	2.0ms
64 points	(64 points × 1 setting)	2.0ms	5.0ms
128 points	(128 points × 1 setting)	4.0ms	7.0ms
256 points	(128 points × 2 setting)	7.0ms	13.0ms
1280 points	(128 points × 10 setting)	24.0ms	26.0ms

Memory dump function processing time

This section describes the time (unit: seconds) taken to save all of the data when the memory dump function is in execution. The following table shows the time taken for data save under the following conditions:

- Scan time = 1.5ms (up to 3ms)
- Internal buffer capacity setting = 256K bytes (by default)
- SD memory card: NZ1MEM-2/4/8/16GBSD used

Number of points*1	Time taken for data save when the memory dump function is in execution	
	When the NZ1MEM-2GBSD is used	When the NZ1MEM-4/8/16GBSD is used
550.4K points*2	17.00 seconds	25.00 seconds
1224K points*3	35.00 seconds	56.00 seconds
5320K points*4	145.00 seconds	240.00 seconds
9416K points*5	244.00 seconds	404.00 seconds

*1 Indicates the total number of points (total number of words) of devices that are set in the device area, file storage area, and refresh data register area.

*2 Indicates the number of device points (user device: 38.4K points (words), refresh data register: 512K points (words)) with the CPU parameters set to default values.

*3 Assumes the following conditions without an extended SRAM cassette:
· User device: 40K points (words), file register: 160K points (words), and refresh data register: 1024K points (words)
· With " Use Common File Register in All Programs" selected in the file register setting.

*4 Assumes the following conditions with an 8M extended SRAM cassette:
· Total number of points (total number of words) of user devices, local devices, and file registers: 4296K points (words), and refresh data register: 1024K points (words)
· With " Use File Register of Each Program" selected in the file register setting.
· The number of program files is 10.

*5 Assumes the following conditions with an 16M extended SRAM cassette:
· Total number of points (total number of words) of user devices, local devices, and file registers: 8392K points (words), and refresh data register: 1024K points (words)
· With " Use File Register of Each Program" selected in the file register setting.
· The number of program files is 10.

Real-time monitor function processing time

This section shows the minimum value (unit: ms) of monitor interval that allows monitoring under the following conditions without data loss during execution of the real-time monitor function.

- Scan time = 1.5ms (up to 3ms)*1
- Data = agrees with the "Number of points of word device" column in the table below
- Timing = Condition specification (step No. specification)*2
- Internal buffer capacity = 32K bytes (default)
- Fixed scan interrupt timing = generating a fixed scan interrupt on the head step number of the program where step numbers are specified
- Performance of the system of personal computer = CPU: Intel® Core™ i3-2100 CPU (3.10GHz), Memory: 2GB, OS: Windows® 7 Professional SP1 32 bits

*1 The scan time during non-execution of the real-time monitor function shall be an average of 1.0ms.

*2 Program name: Arbitrary, Step No.: Head step of the program, Execution condition: Set to Always

Condition	Number of points of word device	Value of monitor interval
Connected via USB	1 point	0.15ms
	8 points	0.25ms
	16 points	0.45ms
When connecting the Ethernet port (TCP)	1 point	0.55ms
	8 points	1.25ms
	16 points	1.35ms
When connecting the Ethernet port (UDP)	1 point	0.15ms
	8 points	0.30ms
	16 points	0.50ms

Database function processing time

Database operation by database access instruction

The table below lists the processing time under the following conditions:

Condition 1	Condition 2
• Number of records: 10000 • Number of fields: 16 The data type of the field is WORD (6 fields), LREAL (6 fields), WSTRING (124 characters) (3 fields), and BOOL (1 field). • Main key: Provided • The functions for file access to the SD memory card other than the database function are not active.	• Number of records: 10000 • Number of fields: 64 The data type of the field is WORD (20 fields), LREAL (20 fields), WSTRING (124 characters) (6 fields), and BOOL (18 fields). • Main key: Provided • The functions for file access to the SD memory card other than the database function are not active.

This section shows the time taken for the completion signal of the completion device to turn on from the execution of each instruction of the database access instructions.

Instruction			Time taken to complete the execution of instruction	
			Condition 1	Condition 2
Database import	DBIMPORT		155s	255s
Database export	DBEXPORT		75s	195s
Database connection	DBOPEN	First time*1	150ms	150ms
		Second time or later	105ms	105ms
Database disconnection	DBCLOSE		30ms	30ms
Database record search	DBSELECT		700ms	750ms
Database record addition	DBINSERT		700ms	800ms
Database record update	DBUPDATE		550ms	600ms
Database record deletion	DBDELETE		700ms	1000ms

*1 After power-on, initial database access involves the diagnostics to check whether the database is corrupted, and thus it takes time to execute the DBOPEN instruction.

■For the programmable controller CPU with firmware version earlier than "28"

The table below lists the processing time under the following conditions:

Condition
• Number of tables: 1 • Number of records: 10000 • Number of fields: 4 The data type of the field is specified as WORD, DWORD, WSTRING (16 characters), and WSTRING (32 characters). • Index: specified • Transaction: Provided (The DBINSERT/DBUPDATE/DBSELECT/DBDELETE instructions only are subject to this condition.) • The function that performs the file access to the SD memory card other than the database instruction is not operating.

This section shows the time taken for the completion signal of the completion device to turn on from the execution of each instruction of the database access instructions.

Instruction			Time taken to complete the execution of instruction
Database import	DBIMPORT		320s
Database export	DBEXPORT		220s
Database connection	DBOPEN	First time*1	590ms
		Second time or later	140ms
Database disconnection	DBCLOSE		3ms
Database record search	DBSELECT		180ms
Database record addition	DBINSERT		210ms
Database record update	DBUPDATE		260ms
Database record deletion	DBDELETE		300ms

*1 After power-on, initial database access involves the diagnostics to check whether the database is corrupted, and thus it takes time to execute the DBOPEN instruction.

Processing time of the CPU module database access (from external device) function

The table below lists the processing time under the following conditions:

Condition 1	Condition 2
• Number of records: 10000 • Number of fields: 16 The data type of the field is specified as WORD (6 fields), DOUBLE PRECISION (6 fields), NLSCHAR(UNICODE) [124 characters] (3 fields), and BOOLEAN (1 field). • Main key: Provided • The functions other than the CPU module database access function for the file access to the SD memory card are not active.	• Number of records: 10000 • Number of fields: 64 The data type of the field is specified as WORD (20 fields), DOUBLE PRECISION (20 fields), NLSCHAR(UNICODE) [124 characters] (6 fields), and BOOLEAN (18 fields). • Main key: Provided • The functions other than the CPU module database access function for the file access to the SD memory card are not active.

The following table lists the processing time until the database operation is completed with the CPU module database access function.

Item	Time until the operation is completed	
	Condition 1	Condition 2
Connect to the database of the CPU module from the application (Microsoft Access).	5s	
Select a record in the database of the CPU module from the application (Excel). (SELECT)	3s	3s
Insert 1000 records in the database of the CPU module from the application (Microsoft Access). (INSERT)	230s	330s
Update a record in the database of the CPU module from the application (Microsoft Access). (UPDATE)	1s	1s
Delete a record in the database of the CPU module from the application (Microsoft Access). (DELETE)	1s	1s

Process control function processing time

This is the time required to process the control loop when the process control function is executed. The following table lists the processing time of some control loop examples.

Loop type	Configuration	Processing time
Two-degree-of-freedom PID control (S2PID)	S.IN, S.PHPL, S.2PID, and S.OUT1 instructions	160μs
PID control (SPID)	S.IN, S.PHPL, S.PID, and S.OUT1 instructions	151μs
Monitor (SMON)	S.IN and S.PHPL instructions	73μs

Point

For the processing time of each process control instruction, refer to the following.

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

SFC program processing time

Shows the time required for SFC program processing.

Point

For details on the SFC program, refer to the following.

MELSEC iQ-R Programming Manual (Program Design)

SFC program processing performance

The SFC program execution time can be calculated with the following formula.

- SFC program execution time = (A) + (B) + (C)

Item		Description
(A)	SFC processing time	This is the total time shown in the following table.
(B)	Operation output processing time for all steps	This is the total processing time for each instruction used for operation output for all steps in the active status.
(C)	Processing time for all transition conditions	This is the total processing time for each instruction used for transition conditions associated with each step in the active status.

*1 For the processing time for each instruction, refer to the following.

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

Shows the SFC processing time (A).

- SFC processing time (A) = (a) + (b) + (c) + (d) + (e) + (f) + (g) + (h)

Item		Processing time calculation (unit: μs)	Description
(a)	Active block processing time	Active block processing time coefficient × Number of active blocks	This is the system processing time required to execute active blocks.
(b)	Inactive block processing time	Inactive block processing time coefficient × Number of inactive blocks	This is the processing time required to execute inactive blocks.
(c)	Nonexistent block processing time	Nonexistent block processing time coefficient × Number of nonexistent blocks	This is the system processing time required to execute blocks that have not been created.
(d)	Active step processing time	Active step processing time coefficient × Number of active steps	This is the time required to execute active steps.
(e)	Active transition processing time	Active transition processing time coefficient × Number of active transitions	This is the system processing time required to execute active transitions.
(f)	Transition establishment step processing time	Transition establishment step processing time coefficient × Number of transitions	This is the time required to turn off active steps when transitions are established.
(g)	SFC end processing time	SFC end processing time = SFC end processing time	This is the system processing time required for SFC end processing.
(h)	Operation output processing time	Action processing time coefficient × Number of actions	This is the system processing time required to process operation outputs.

Shows the coefficient value for each processing time.

Item		Coefficient value
Active block processing time coefficient		2.8
Inactive block processing time coefficient		2.6
Nonexistent block processing time coefficient		0.4
Active step processing time coefficient		7.5
Active transition processing time coefficient		0.1
Transition establishment step processing time coefficient	Hold step	24.3
	Normal step	34.5
SFC end processing time		58.7
Operation output processing time coefficient		1.0

SFC program switching

Shows the processing time required to switch the SFC program from the standby status to the scan execution type.

- Switching time [μs] = (Number of blocks created × Km) + (Number of steps created × Kn) + (SFC program capacity × Kp) + Kq

Constant	Constant value	
	R04CPU, R04ENCPU, R08CPU, R08ENCPU, R08PCPU, R16CPU, R16ENCPU, R16PCPU, R32CPU, R32ENCPU, R32PCPU	**R120CPU, R120ENCPU, R120PCPU**
Km	3.97	3.97
Kn	0.41	0.41
Kp	0.39	0.32
Kq	2500	2500

When the SFC program capacity is the following conditions, the processing time required to switch the SFC program from the standby status to the scan execution type will be fixed.

Model	Condition	Processing time (constant value)
R04CPU, R04ENCPU, R08CPU, R08ENCPU, R08PCPU, R16CPU, R16ENCPU, R16PCPU, R32CPU, R32ENCPU, R32PCPU	28K steps or less	14ms
R120CPU, R120ENCPU, R120PCPU	48K steps or less	18ms

When the SFC information device is set, the processing time required to switch the SFC program from the standby status to the scan execution type will become longer.

Processing time by the redundant function

The following describes the processing time by the redundant function with the Process CPU (redundant mode).

Increase in the scan time due to tracking transfer

The following describes the increase in the scan time of the CPU module due to tracking transfer.
The increase in the scan time determined by the following calculation formula is as a rough standard for a system start-up. Check the increase in the scan time with the actual systems.

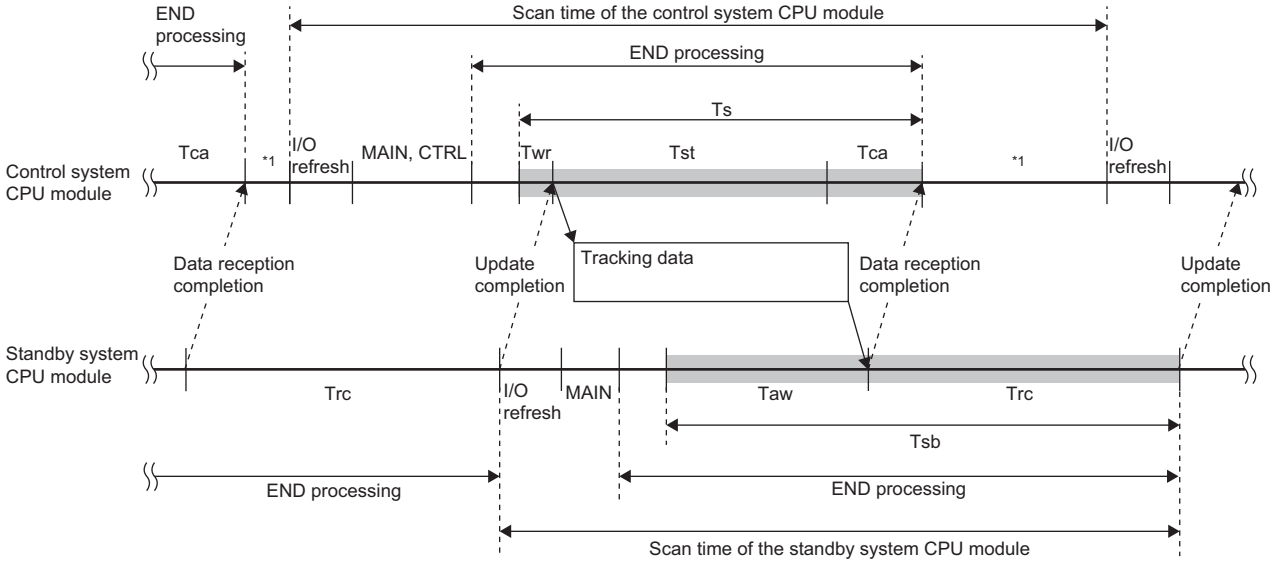
Point

The scan time can be checked in SD520 and SD521 (Current scan time). (☞ Page 651 Fixed scan function information)

Item	Increase in the scan time	Reference
Increase in the scan time of the CPU module in the control system (Ts)	Ts = Twr + Tst + Tca + α [ms]	Twr: Page 715 Waiting time for completion of the previous tracking data reflection Tst: Page 715 Tracking data send time Tca: Page 716 Waiting time for completion of tracking data reception α: Page 716 Other extended times
Increase in the scan time of the CPU module in the standby system (Tsb)	Tsb = Taw + Trc [ms]	Taw: Page 717 Waiting time for tracking data reception Trc: Page 717 Tracking data reflection time

Ex.

When CTRL is a control system execution program and MAIN is a program executed in both systems



*1 If the constant scan is used, waiting time for the constant scan generated.

■Waiting time for completion of the previous tracking data reflection

The following describes the waiting time (Twr) for completion of the previous tracking data reflection in the CPU module of the control system.

Twr = 1 + Tdrm - Toth [ms]

- Tdrm: Maximum time taken for reflection of tracking data in the CPU module of the standby system
- Toth: Scan time excluding Ts (Increase in the scan time) in the CPU module of the control system

Tdrm is determined as follows.

- When no extended SRAM cassette is inserted: 1 + (No. of tracking blocks × 0.3) + (Total size of tracking data [word] × 45.0) × 10^{-6}
- When an extended SRAM cassette is inserted: 1 + (No. of tracking blocks × 0.3) + (Total size of tracking data [word] × 106.0) × 10^{-6}

If Tdrm - Toth is smaller than 0, Twr is handled as 1.

Twr is not generated in the first scan in which the tracking transfer is started.

■Tracking data send time

The following describes the tracking data send time (Tst) in the CPU module of the control system. The calculation method differs depending on whether an extended SRAM cassette is inserted or not.

When no extended SRAM cassette is inserted: Tst = 0.5 + (26.7 × 10^{-6}) × D1 + (43.5 × 10^{-6}) × D2 + (1.5 × 10^{-3}) × D3 [ms]

When an extended SRAM cassette is inserted: Tst = 0.5 + (26.7 × 10^{-6}) × D1 + (113.5 × 10^{-6}) × D2 + (1.5 × 10^{-3}) × D3 [ms]

- D1: Size [word] of tracking data in the system data, signal flow memory, and refresh data register (RD)
- D2: Size [word] of tracking data of global devices, local devices, global labels, and local labels
- D3: Number of tracking transfer settings of global devices

D1 is determined as follows.

Item	Size
System data	■Fixed data 7318 ■PID control instruction information 782 (0 (when not used)) ■Data that varies depending on the tracking device/label setting 16 × (No. of tracking settings of T, ST, C, and LC) + 8 × (No. of global device tracking settings of other devices)
Signal flow memory	(Total number of steps of control system execution programs, or total number of steps of rising/falling instruction in an FB)/16 (The digit after the decimal point is round up.)
Refresh data register (RD)	Follows the tracking transfer settings. (☞ Page 411 Tracking transfer setting)

D2 is determined as follows.

Item	Size*1
Global device (other than the refresh data register (RD))	Follows the tracking transfer settings. (☞ Page 411 Tracking transfer setting)
Local device	Total size of local devices of the device/label memory area detailed setting × the number of programs on the program setting*3
Global label*2	Follows the global label settings.
Local label*2	Follow the local label settings.

*1 The size of the devices/labels to be transferred (tracked) can be checked by clicking the [Size Calculation] button in "Detailed Setting" of "Device/Label Detailed Setting". (☞ Page 413 Detailed setting)

*2 Even though global labels/local labels are deleted by online change, the size of the tracking data of the global labels/local labels are the same as the one before the online change. When the program and global label setting file to which all data is rebuilt (reassigned) are written to the programmable controller and the status of the CPU module is switched from STOP to RUN, the tracking data of the global labels/local labels is updated.

*3 This indicates the number of programs where local devices are set to use in the "Setting of Device/File Use Or Not" window of the program setting of the CPU parameter.

■Waiting time for completion of tracking data reception

The following describes the waiting time for completion of tracking data reception (Tca) in the CPU module of the control system.

Tca = 2[ms]

■Other extended times

The following describes other delay time (α) in the CPU module of the control system.

α = 0.6 + α1[ms]

When an error occurs during a tracking transfer, the scan time may be extended (α1).α 1 is determined as follows.

Item	Time
Disconnection, pulling out, inserting of the other tracking cable	6[ms]
Momentary power failure or power-off in the standby system	16 to 56 [ms]
Hardware failure in the standby system	50 to Total size of tracking data [word] × 26.7 × 10^{-6} + 10^{*1} [ms]

*1 Maximum extended time for the redundant function module

Point

When using the constant scan, take one of the following measures.

- The setting time of the constant scan is set by adding the above extended time (α1) when an error occurs. (☞ Page 115 Setting Constant Scan)
- When one of the above errors has occurred and a continuation error occurs due to the excess of constant scan time, clear the error. (☞ Page 233 Error Clear)

■Waiting time for tracking data reception

The following describes the waiting time for tracking data reception (Taw) in the CPU module of the standby system.

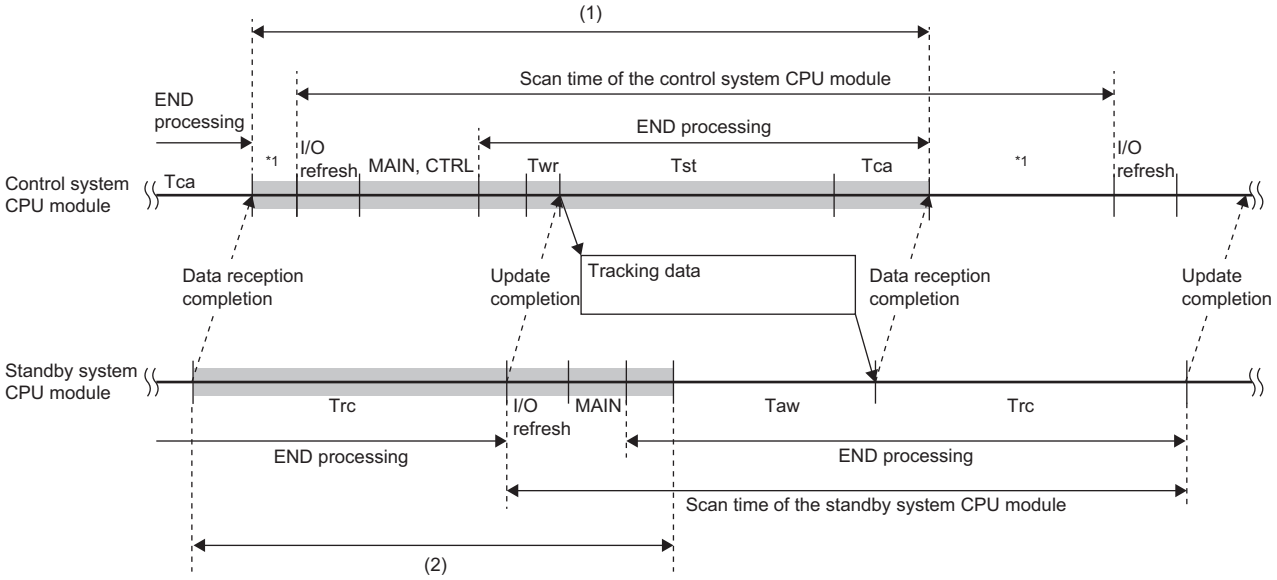
The waiting time for tracking data reception is determined by two types of period; (1) the period from when the control system CPU module receives a receive completion notification until next time it receives the notification and (2) the period when the standby system CPU module sends a receive completion notification until it starts receiving tracking data.

When (1) ≤ (2): 0 [ms]

When (1) > (2): (1) - (2) [ms]

Ex.

When CTRL is a control system execution program and MAIN is a program executed in both systems



*1 If the constant scan is used, waiting time for the constant scan generated.

- For the I/O refresh processing time for (1) and (2) and END processing time for others, refer to the following.

☞ Page 696 END processing time

- If the values of each item for (1) and (2) vary, the waiting time for tracking data reception also varies.

■Tracking data reflection time

The following describes the tracking data reflection time (Trc) in the CPU module of the standby system. The calculation method differs depending on whether an extended SRAM cassette is inserted or not.

When no extended SRAM cassette is inserted: $Trc = 1 + (20.0 \times 10^{-6}) \times D1 + (37.2 \times 10^{-6}) \times D2 + (300.0 \times 10^{-6}) \times F1$ [ms]

When an extended SRAM cassette is inserted: $Trc = 1 + (20.0 \times 10^{-6}) \times D1 + (106.0 \times 10^{-6}) \times D2 + (300.0 \times 10^{-6}) \times F1$ [ms]

- D1: Size [word] of tracking data in the system data, signal flow memory, and refresh data register (RD)
- D2: Size [word] of tracking data of global devices, local devices, global labels, and local labels
- F1: Number of transferred blocks

For the calculation formula for the size of tracking data, refer to the following.

☞ Page 715 Tracking data send time

Extended time required for detecting a system switching cause

The following describes the extended time required from occurrence of a system switching cause for the CPU module or redundant function module in the control system to the detection of a system switching request, or the extended time required for the CPU module to detect a system switching request sent from the network module.

System switching cause		Extended time required for detecting a system switching cause
Power-off		19.26 to 38.83ms
CPU module hardware failure, CPU module stop error	When the tracking communication cannot be made	10ms
	When the tracking communication can be made	0ms
System switching request from a network module		1 scan
System switching request by using the SP.CONTSW instruction		0ms
System switching request from the engineering tool		0ms

Time required for system switching

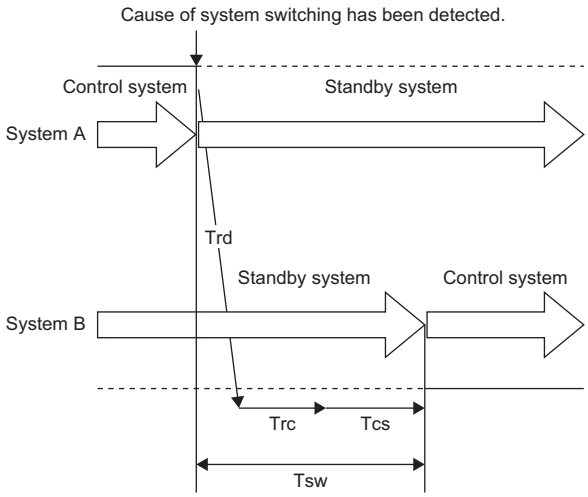
The following describes the time required for the CPU module in the new control system to start operating as the new control system after the detection of a system switching cause in the control system.

The time required for system switching determined by the following calculation formula is as a rough standard for a system start-up. Check the time with the actual systems.

Tsw = Trd + Trc + Tcs [ms]

- Tsw: Time required for system switching*1
- Trd: Delay until the CPU module in the standby system receives the system switching request sent from the control system
- Trc: Time required for reflection of tracking data in the CPU module of the standby system
- Tcs: Setup time required for the CPU module in the standby system to start operating as the new control system

*1 This is the maximum system switching time.



System switching cause	Processing time per each system switching cause			
	Trd	Trc	Tcs	
• Power supply module failure*2, power-off • Reset	■When the standby system scan time is longer than 19.26ms*1 0.517 + Standby system scan time - 19.26 [ms] ■When the standby system scan time is 19.26ms or shorter*1 0.517ms	0ms	1ms	The following time is added according to the parameter setting. When the signal flow memory is tracked: 0ms When the signal flow memory is not tracked: 2ms
• CPU module failure • Base unit failure*3 • CPU module stop error (major) • Redundant function module failure*3	■When the standby system scan time is longer than 10ms*1 0.517 + Standby system scan time - 10 [ms] ■When the standby system scan time is 10ms or shorter*1 0.517ms			
CPU module stop error (moderate)	Increase in the scan time of the CPU module in the control system (Ts) by tracking transfer*4 - Waiting time for completion of the previous tracking data reflection (Twr)	☞ Page 720 Tracking data reflection time*4	5ms	
System switching request from a network module	Increase in the scan time of the CPU module in the control system (Ts) by tracking transfer - Waiting time for completion of the previous tracking data reflection (Twr)	☞ Page 720 Tracking data reflection time		
System switching request by using the SP.CONTSW instruction				
System switching request from the engineering tool				

*1 The standby system scan time is the total of "Execution time of the standby system program after a system switching cause is received + Tracking data reflection time".

*2 When the CC-Link IE Field Network module has not been mounted, the systems may not be switched depending on the state of the power supply module failure. To switch the systems regardless of the state of the power supply module failure, mount the CC-Link IE Field Network module. Note, in this case, that Trd is determined by the following:
Trd = 400ms (maximum) + Standby system scan time

*3 When the CC-Link IE Field Network module has not been mounted and a communication error has been detected in the communications between the redundant function module and the CPU module, the systems may not be switched. To switch the systems even though the redundant function module has been removed from the base unit or a failure has occurred in the base unit, mount the CC-Link IE Field Network module. Note, in this case, that Trd is determined by the following:
Trd = Standby system scan time

*4 When a system switching occurs due to a stop error of the CPU module, only the system data is transferred and the other data (such as the signal flow memory, devices and labels) are not transferred.

■Tracking data reflection time

The following describes the time required for the CPU module of the standby system to reflect tracking data. The calculation method differs depending on whether an extended SRAM cassette is inserted or not.

When no extended SRAM cassette is inserted: $Trc = 1 + (20.0 \times 10^{-6}) \times D1 + (37.2 \times 10^{-6}) \times D2 + (300 \times 10^{-6}) \times F1$ [ms]

When an extended SRAM cassette is inserted: $Trc = 1 + (20.0 \times 10^{-6}) \times D1 + (106.0 \times 10^{-6}) \times D2 + (300 \times 10^{-6}) \times F1$ [ms]

- D1: Size [word] of tracking data in the system data, signal flow memory, and refresh data register (RD)
- D2: Size [word] of tracking data of global devices, local devices, global labels, and local labels
- F1: Number of transferred blocks

For the calculation formula for D1 and D2, refer to the following.

☞ Page 714 Increase in the scan time due to tracking transfer

Extended time from completion of a system switching to the first output

The following describes the time required for the CPU module that has started operating as the new control system to output values to the network module for the first time.

Tjo = Sc - Twc - Ts - Toref [ms]

- Tjo: Extended time from completion of a system switching to the first output
- Sc: Scan time of the CPU module in the control system
- Twc: Constant scan waiting time
- Ts: Increase in the scan time of the CPU module in the control system
- Toref: Output refresh of the intelligent function module (CPU module → Intelligent function module)

Memory copy time

The following table shows the time taken for the memory copy under the following conditions:

- CPU module: R120PCPU
- Sequence scan time: 5ms
- The SD memory card of the standby system CPU module has been formatted.
- Data in the program memory, data memory, and SD memory card are mismatched between the control system and standby system.

Memory copy target memory and size of data to be transferred	Memory copy time
Size of data transferred from the data memory and program memory: 144K bytes Size of data transferred from the SD memory card: 128K bytes	30s
Size of data transferred from the data memory and program memory: 282K bytes Size of data transferred from the SD memory card: 512K bytes	40s

Memory copy time depends on the data to be copied and the use of the SD memory card. Use the time above as a rough standard for memory copy.

Appendix 10 Parameter List

This section lists parameters.

System parameters

The following is the list of system parameters.

Item			Parameter No.
I/O Assignment	Base/Power/Extension Cable Setting	Slot	0201H
		Base, Power Supply Module, Extension Cable	0203H
	I/O Assignment Setting	Points, Start XY, Module Status Setting	0200H
		Control PLC Settings	0202H
		Module Name	0203H
	Points Occupied by Empty Slot Batch Setting		0100H
	Redundant Module Group Setting		0400H
Multiple CPU Setting	Communication Setting between CPUs	CPU Buffer Memory Setting (at refresh END)	0304H
		CPU Buffer Memory Setting (at execution of refresh I45)	0308H
		PLC Unit Data	0309H
		Fixed Scan Communication Function	—
		Fixed Scan Communication Area Setting	0307H
	Fixed Scan Communication Setting	Fixed Scan Interval Setting of Fixed Scan Communication	0306H
		Fixed Scan Communication Function and Inter-module Synchronization Function	0306H
	Operation Mode Setting	Stop Setting	0302H
		Synchronous Startup Setting	030AH
	Other PLC Control Module Setting	I/O Setting Outside Group	0305H
Synchronization Setting within the Modules	Use Inter-module Synchronization Function in System		—
	Select Synchronous Target Unit between Unit		0101H
	Fixed Scan Interval Setting of Inter-module Synchronization		0101H
	Inter-module Synchronous Master Setting		0102H

A

CPU Parameters

The following is the list of CPU parameters.

Item			Parameter No.
Name Setting	Title Setting		3100H
	Comment Setting		3101H
Operation Related Setting	Timer Limit Setting		3200H
	RUN-PAUSE Contact Setting		3201H
	Remote Reset Setting		3202H
	Output Mode Setting of STOP to RUN		3203H
	Module Synchronous Setting		3207H
	Clock Related Setting		3209H
Interrupt Settings	Fixed Scan Interval Setting		3A00H
	Fixed Scan Execution Mode Setting		3A00H
	Interrupt Enable Setting in Executing Instruction		3A00H
	Block No. Save/Recovery Setting		3A00H
	Interrupt Priority Setting from Module		3A01H
Service Processing Setting	Device/Label Access Service Processing Setting		3B00H
File Setting	File Register Setting		3300H
	Initial Value Setting		3301H
	Label Initial Value Reflection Setting		3302H
	File Setting for Device Data Storage		3303H
Memory/Device Setting	Device/Label Memory Area Setting	Extended SRAM Cassette	3404H
		Device/Label Memory Area Capacity Setting	3400H
		Device Points	3401H
		Local Device	3405H
		Latch Range Setting	3407H
		Latch Type Setting of Latch Type Label	3408H
	Index Register Setting		3402H
	Refresh Memory Setting		3403H
	Device Latch Interval Setting		3406H
	Pointer Setting		340BH
	Internal Buffer Capacity Setting		340AH
RAS Setting	Scan Time Monitoring Time (WDT) Setting		3500H
	Constant Scan Setting		3503H
	Error Detections Setting		3501H
	CPU Module Operation Setting at Error Detected		3501H
	LED Display Setting		3502H
	Event History Setting		3504H
	Online module change function setting	Direct change setting	3505H
	RnSFCPU Operation Mode Setting		3506H
Program Setting	Program Setting	Program Name	3700H
		Execution Type	3700H
		Type (Fixed Scan)	3700H
		Type (Event)	3701H
		Detail Setting Information	—
		Refresh Group Setting	3700H
		Device/File Use or not	3700H
		Both Systems Program Executions Setting	3700H
	FB/FUN File Setting		3702H
Refresh Setting between Multiple CPU	Refresh Setting (At the END)		3901H
	Refresh Setting (At I45 Exe.)		3902H
Routing Setting	Routing Setting		3800H

Item		Parameter No.
SFC Setting	SFC Program Start Mode Setting	3C00H
	Start Condition Setting	3C00H
	Output Mode at Block Stop Setting	3C00H
Redundant System Settings	Redundant Behavior Setting	5000H
	Tracking Setting	5001H

Module parameters

The following are the lists of module parameters.

Ethernet function

For the module parameters relating to the Ethernet function, refer to the following.

MELSEC iQ-R Ethernet User's Manual (Application)

CC-Link IE Field Network Basic function

The following is the list of module parameters relating to the CC-Link IE Field Network Basic function.

Item		Parameter No.
CC-Link IEF Basic Setting	Network Configuration Settings	7A00H
	Refresh Settings	7420H

CPU module database access function

The following is the list of the module parameter related to the CPU module database access function.

Item		Parameter No.
Application settings	Built-in Database Access Setting	A0E0H

Safety CPU parameters

The following is the list of safety CPU parameters.

Item			Parameter No.
Memory/Device Setting	Device/Label Memory Area Setting	Device/Label Memory Area Capacity Setting	5A00H
		Safety Device Point Quantity Setting	5A01H
		Safety Local Device Setting	5A05H
Program Setting	Program Setting	Execute Order	5A13H
		Program Name	5A10H, 5A13H
		Execution Type	5A10H
		Refresh Group Setting	5A10H
		Device/File Use or not	5A10H
	FB/FUN File Setting		5A12H
Safety Function Setting	Safety Cycle Time		5A20H

Memory card parameters

The following is the list of memory card parameters.

Item	Parameter No.
Boot Setting	2000H
Setting of File/Data Use or Not in Memory Card	2010H

Appendix 11 List of the Targets of Device/Label Access Service Processing Setting

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

Applicable function	Description
Functions to perform read/write to files accessed by programs	When read/write is performed to the relevant files during execution of a program, file inconsistency may occur. Therefore, such operation is performed during the END processing.
Functions to perform write to devices/labels	When write is performed to devices/labels during execution of a program, the result of operation processing may become inconsistent. Therefore, such operation is performed during the END processing.

Communication functions via the MC protocol

Among communication functions using the MC protocol, those to which the device/label access service processing setting can be applied are as follows.

Function			Command
Device memory	Multiple blocks batch read		0406 (00□0)
	Multiple blocks batch write		1406 (00□0)
	Batch read	In units of bits	0401 (00□1)
		In units of words	0401 (00□0)
	Batch write	In units of bits	1401 (00□1)
		In units of words	1401 (00□0)
	Random read		0403 (00□0)
	Test (Random write)	In units of bits	1402 (00□1)
		In units of words	1402 (00□0)
	Monitor		0802 (00□0)
Files	New file creation*1		1820 (0000)
	File copy*1	File password function incompatible	1824 (0000)
		File password function compatible	1824 (0004)
	File open*1	File password function incompatible	1827 (0000)
		File password function compatible	1827 (0004)
	File read*1		1828 (0000)
	File write*1		1829 (0000)
	File close*1		182A (0000)
Programmable controller CPU monitoring	Registration*2		0630 (0000)
	Clear*2		0631 (0000)

*1 Only the file register file can support it. (Other file access is always executed asynchronously from the program.)

*2 When registration/clear command is executed, it does not affect the scan time. However, since the serial communication module will periodically access into the device memory of the CPU module when Function is enabled, it becomes the target for Device/Label access service processing setting.

Communication function using an engineering tool

Of the communication functions using an engineering tool, the functions targeted for the setting of the Device/Label access service processing are indicated.

Function	
Writing data to the programmable controller	File register file
	Device data (Local device also included)
	Global label and local label data
Reading data from the programmable controller	File register file
	Device data (Local device also included)
	Global label and local label data
	Device data storage file
Monitor function	Circuit monitor
	Device/buffer memory batch monitor
	Label batch monitor (local label included)
	Buffer memory monitor
Ethernet function	File transfer (FTP server)*1
	File transfer (FTP client)*1

*1 Only the file register file can support it. (Other file access is always executed asynchronously from the program.)

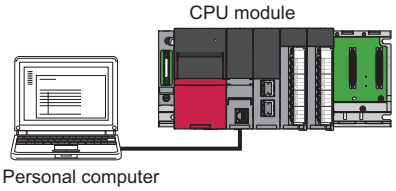
Appendix 12 How to Use CPU Module Logging Configuration Tool

This section describes the system configuration and specifications, and operation/configuration procedures with CPU Module Logging Configuration Tool when using the data logging function.

System configuration

This section describes the system configuration when using the data logging function.

Overall system configuration



Software

This section describes the software used for the data logging function.

Name	Description
CPU Module Logging Configuration Tool	This software is used for configuration and maintenance of the data logging.
GX LogViewer	This software graphically displays data collected by the data logging function.

Point

GX LogViewer is not an essential tool for using the data logging function. Use this tool for various objectives, such as graphically displaying data logged by this function. GX LogViewer is beyond the scope of this manual unless it specifically relates to the topic of this manual. For details on GX LogViewer, refer to the following:

GX LogViewer Version 1 Operating Manual

■Supported versions of CPU Module Logging Configuration Tool

Each CPU module can be used with CPU Module Logging Configuration Tool of the supported version (listed in the table below) or any later version.

Version of CPU Module Logging Configuration Tool	Compatible CPU module
"1.35M" or later	RnCPU
"1.43V" or later	Process CPU (process mode)
"1.49B" or later	RnENCPU, Safety CPU
"1.58L" or later	Process CPU (redundant mode)

■Compatible OS

The following table lists OSs compatible with the CPU Module Logging Configuration Tool.

Version of CPU Module Logging Configuration Tool	Compatible OS
"1.05F" or later	32-bit Windows®7
"1.18U" or later	64-bit Windows®7
"1.30G" or later	32-bit/64-bit Windows®8
"1.35M" or later	32-bit/64-bit Windows®8.1
"1.58L" or later	32-bit/64-bit Windows®10

■Operating environment of CPU Module Logging Configuration Tool

For details on the operating environment for CPU Module Logging Configuration Tool, refer to following manual which is stored in the installer.

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

■Display language change

The CPU Module Logging Configuration Tool supports multiple languages, and can be used by changing the display language for menus and so on at the same computer.* 1

*1 CPU Module Logging Configuration Tool with version "1.46Y" or later supports this function.

Operating procedure

[View] ⇨ [Switch Display Language]

Precautions

Text may be cut off if the OS and set display language differ.

Communication route

To connect the CPU module to a personal computer, use the following methods: (☞ Page 759 Transfer setup)

■Connection through a USB port

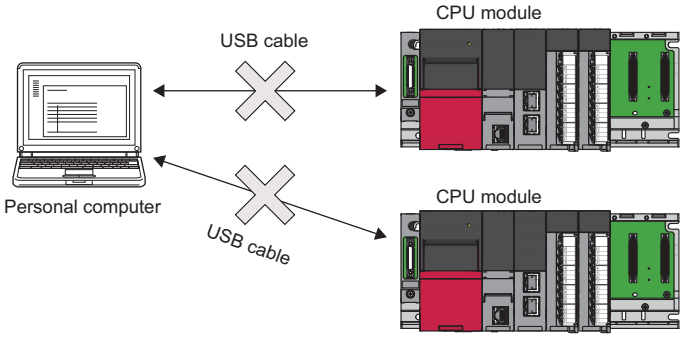
Use a USB cable. The following types of USB cables can be used:

- USB cable (USB A type—USB miniB type)
- USB cable (USB B type—USB miniB type)

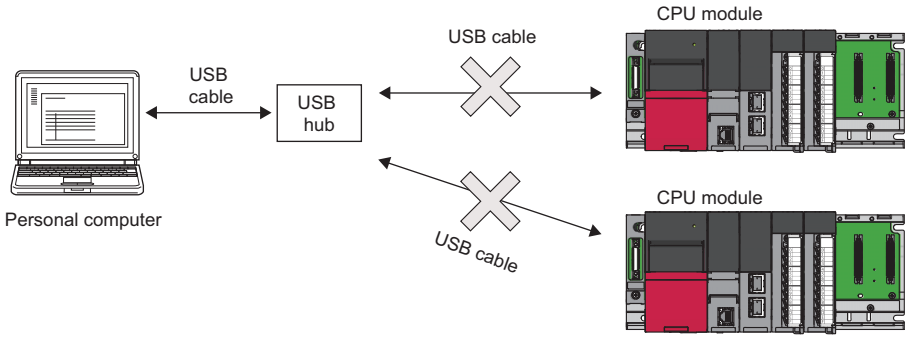
Restriction

Only one CPU module can be connected to the same personal computer simultaneously. The following connection configurations are prohibited.

- Connecting a personal computer with multiple USB ports to more than one CPU module



- Connecting to more than one CPU module via a USB hub



A

■Connection through an Ethernet port

• Connection via a hub

Connect the CPU module via a hub to a personal computer on the same local network.*1 Note that IP address of the CPU module must be specified. Also the personal computer should have the same network address as the CPU module.

*1 If the CPU module is RnENCPU, use the RnENCPU (CPU part) Ethernet port to connect. Connection is not possible from the RnENCPU (network part) Ethernet port.

Restriction

Only local area network can be used for connections. Connections via the Internet are not allowed.

• Direct connection

One-to-one direct connection with an Ethernet crossing cable is possible. This method requires no hub.*2 Note that IP address of the CPU module is not required to be specified with this method. (broadcast address is used for this communication).

*2 If the CPU module is RnENCPU, use the RnENCPU (CPU part) Ethernet port to connect. Connection is not possible from the RnENCPU (network part) Ethernet port.

■Precautions about Ethernet connection

- Do not directly connect to a personal computer via LAN line. Load imposed on the LAN line adversely affects communications by other devices.
- Do not configure the direct connection setting when using one-to-one connection via a hub between the CPU module and a personal computer.
- If the following conditions are met, the direct connection communication may be disabled. If the communication is disabled, review the settings of the CPU module and personal computer.

Ex.

When all the bits of the CPU module-side IP address that correspond to 0 part of the personal computer-side subnet mask are on or off:

CPU module-side IP address: 64.64.255.255

Personal computer-side IP address: 64.64.1.1

Personal computer-side subnet mask: 255.255.0.0

Ex.

When all the bits of the CPU module-side IP address that correspond to the host address of each class of the personal computer-side IP address are on or off*1*2:

CPU module-side IP address: 64.64.255.255

Personal computer-side IP address: 192.168.0.1

Personal computer-side subnet mask: 255.255.0.0

*1 The IP address for each class is as follows:
·Class A: 0.x.x.x to 127.x.x.x
·Class B: 128.x.x.x to 191.x.x.x
·Class C: 192.x.x.x to 223.x.x.x

*2 The host address for each class is 0 part in the following addresses:
·Class A: 255.0.0.0
·Class B: 255.255.0.0
·Class C: 255.255.255.0

Ex.

When the CPU module-side IP address is obtained through DHCP:

- Disable Windows firewall setting, if enabled.
- Do not set direct connection method in a configuration in which more than one IP address is enabled simultaneously as described in the following list.
  - An IP address is assigned to each Ethernet port (network device) of the personal computer with more than one port.
  - Wireless LAN setting is enabled as well as the Ethernet port on the personal computer.
  - More than one IP address is assigned to a single Ethernet port of the personal computer.

■Connection to the Process CPU (redundant mode)

When the Process CPU (redundant mode) is connected, select "Connect to RnPCPU (Redundant Mode)" and select a target system for "Transfer Setup System".

Specifications

This section describes the specifications of the data logging function.

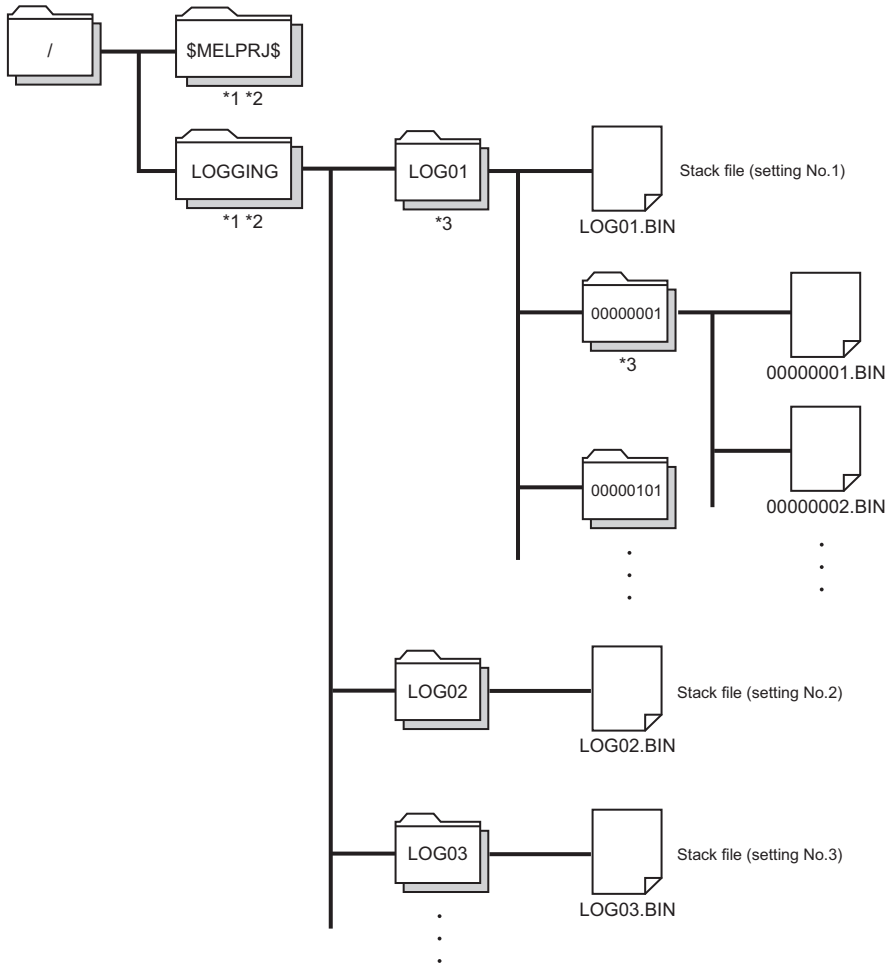
Functional specifications

The following table lists functional specifications.

Item			Specifications
Number of data logging settings			10
Data storage location			• Data memory (only data logging configuration file) • SD memory card
Logging type			• Continuous logging • Trigger logging
Data collection	Collection interval		• Every scan operation • Time specification • Interrupt occurrence • Condition specification (device specification, step No. specification)
	Number of points for collection		Maximum of 1280 (128 per setting)
	AND conjunction		AND conjunction of the device specification and step No. specification is enabled.
Data processing	Trigger logging	Trigger condition	• Condition specification (device change specification, step No. specification) • When trigger instruction executed
		AND conjunction	AND conjunction of the device change specification and step No. specification is enabled.
		Trigger logging range	Number of records specified before and after the trigger establishment
		Number of trigger establishments (number of events that can be handled as trigger)	one
		Number of records	Maximum of 1000000
File output	File name		Additional information plus file number
	File storage format		• Unicode text file • Binary file
	Data type		• Bit • Word (signed) • Double word (signed) • Word (unsigned) • Double word (unsigned) • Single-precision real number • Double-precision real number • String • Numeric string • Time
	Data output format	Unicode text file	• Decimal format • Hexadecimal format • Decimal fraction format • Exponential format
		Binary file	• Word (signed) • Double word (signed) • Word (unsigned) • Double word (unsigned) • Single-precision real number • Double-precision real number
Output file handling	Storage file switching	File switching timing	• Number of records • File size
		Maximum number of storage files	1 to 65535
Other functions	Operation settings when entering into RUN mode		This function sets data logging operations when entering into RUN mode after the data logging setting is registered.
	Auto logging		When inserting an SD memory card, which holds data logging setting, into the CPU module, the data logging automatically starts based on the data logging setting information on the SD memory card.
	SD memory card replacement		SD memory cards can be replaced using the SD memory card forced disable function even when the data logging is in progress.

Folder configuration

The following figure shows the folder configuration of the SD memory card attaching to a CPU module.



*1 Folder names cannot be modified.

*2 Do not create folders/files under the $MELPRJ$ and LOGGING folders using a personal computer and other.

*3 To remove unnecessary folders, use the following methods:

· Use a personal computer.

· Logging file operation (☞ Page 764 Logging file operation).

Data output type

This section describes each of file output types.

■Unicode text file output type

This section describes the format specifications of the Unicode text file output type and output content of each data.

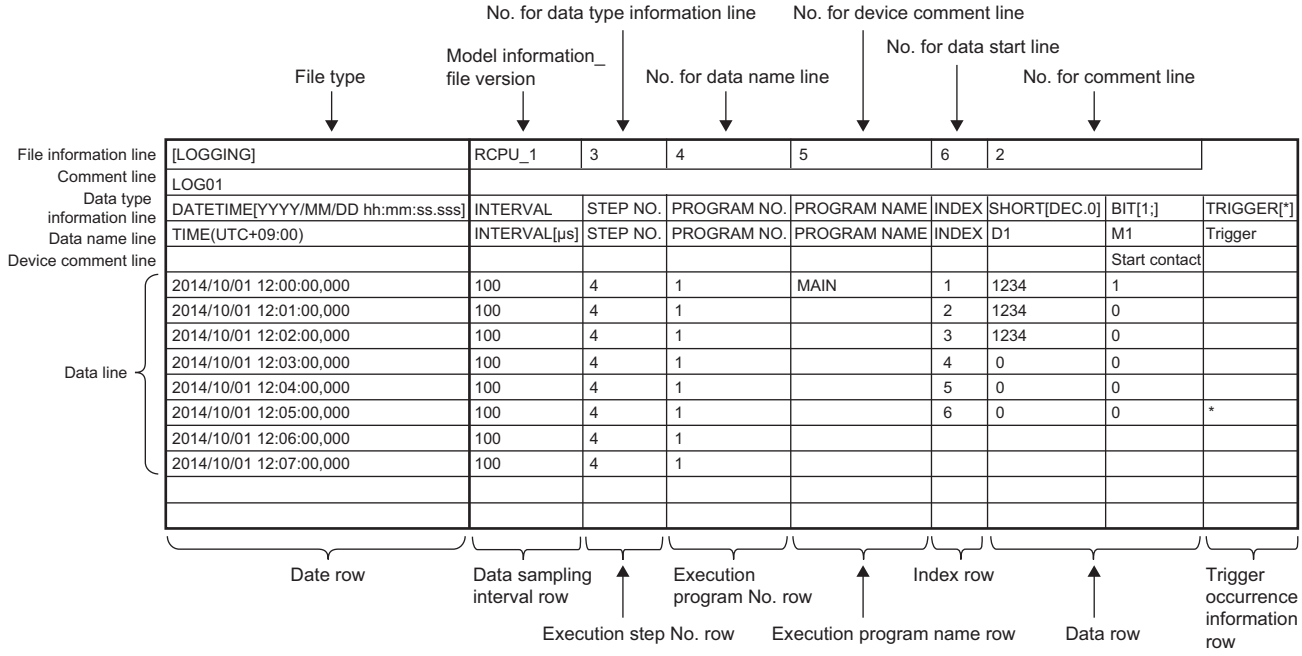
• Format specifications

Item	Description
Delimiter	Tab
Return code	CRLF (0x0D, 0x0A)
Character code	Unicode
Character encoding schema	UTF-16 (Little-Endian)
Field data	Not enclosed by double quotation marks (" ") Double quotation marks (" ") and commas (,) cannot be used in each data.*1

*1 When "Output device comment" is selected in "Device comment output" of the output settings and double quotation marks (" ") and/or commas (,) are included in a device comment, the following processing occurs:
· Whole data is enclosed by double quotation marks (" ").
· When the data includes double quotation marks (" "), another set of double quotation marks will be appended to each of them.

• File format example

Output items can be specified (☞ Page 752 Output)



• Output content for each data

· File information row

File-related information is displayed.

Item	Description	Size
File type	[LOGGING] is output.	14 bytes
Model information_file version	File version that indicates model information (fixed to "RCPU_1") is displayed.	12 bytes
No. for data type information line	Numerical value indicating the position of the data type information row from the top of the file is placed.	2 bytes
No. for data name line	Numerical value indicating the position of the data name row from the top of the file is placed.	2 bytes
No. for device comment line	Numerical value indicating the position of the device comment row from the top of the file is placed.	2 bytes
No. for data start line	Numerical value indicating the starting position of the data row from the top of the file is placed.	2 bytes
No. for comment line	Numerical value indicating the position of the comment row from the top of the file is placed. When the comment row is not output, this field is blank.	0 to 2 bytes

Ex.

The total size of the file information row can be obtained by the following equation: (when comment is output)
14(file type) + 12(model information_file version) + 2(data type information row number) + 2(data name row number) + 2(device comment row number) + 2(data start row number) + 2(comment row number) + 10(number of tabs) + 4(CR + LF) = 50 bytes

• Comment row file information row

Comments are displayed.

Item	Description	Size
Comment	Comment specified in CPU Module Logging Configuration Tool is output (the comment can contain up to 256 characters. *2When no comment is set, a blank row is output).	0 to 512 bytes

*2 Double quotation marks (" "), commas (,), and semicolons (;) cannot be used.

Ex.

The total size of the comment row can be obtained by the following equation:
Character size of the specified comment (depending on the specified character string) + 4(CR + LF)

• Data type information row

The data type for each column is displayed. This information is output in the following format: (data type)[(additional information)].

Item	"Data type" output content	Size	"Additional information" output content	Size
Date/time column	DATETIME	16 bytes	Format is output. [YYYY/MM/DD hh:mm:ss.sss]	6 to 68 bytes
Data collection interval column	INTERVAL	16 bytes	No additional information	0 byte
Execution step No. column	STEP NO.	16 bytes		
Execution program number column	PROGRAM NO.	22 bytes		
Execution program name column	PROGRAM NAME	24 bytes		
Index column	INDEX	10 bytes		
Data column	Bit type: BIT	6 bytes	Bit type: [1;0]	10 bytes
	16-bit integer (unsigned): USHORT	12 bytes	For decimal format: [DEC.0]	14 bytes
	16-bit integer (signed): SHORT	10 bytes		
	32-bit integer (unsigned): ULONG	10 bytes	For hexadecimal format: [HEX]	10 bytes
	32-bit integer (signed): LONG	8 bytes		
	Single-precision floating point (32-bit): FLOAT	10 bytes	For exponent expression: [EXP. (number of digits of decimal part)]	14 to 16 bytes
	Double-precision floating point (64-bit): DOUBLE	12 bytes		
	Character string type: STRING	12 bytes	Character string type, numeric string type: the specified data length value (unit: bytes) is output.	6 to 10 bytes
	Numeric string type: RAW	6 bytes		
	Time: TIME	8 bytes	No additional information	0 byte
Trigger occurrence information column	TRIGGER	14 bytes	[(string occurred)] is output (semicolons (;), double quotation marks (" "), and commas (,) cannot be used).	6 to 512 bytes

Ex.

The size of the data type information row can be obtained by the following equation, when performing the data logging of 128 16-bit integer (signed) data items in decimal format (when selecting the option which outputs date/time column with the format YYYY/MM/DD hh:mm:ss.sss, data collection interval column, execution step No. column, execution program number column, execution program name column, and index column):

(16 + 50)(date/time column) + 16(data collection interval column) + 16(execution step No. column) + 22(execution program number column) + 24(execution program name column) + 10(index column) + (10 + 14) × 128(data column) + 264(number of tabs) + 4(CR + LF) = 3494 bytes

• Data name row

The data name for each column is displayed.

Item	Description	Size
Date/time column	TIME (time zone) is output.	28 bytes
Data collection interval column	INTERVAL[us] is output.	24 bytes
Execution step No. column	STEP NO. is output.	16 bytes
Execution program number column	PROGRAM NO. is output.	22 bytes
Execution program name column	PROGRAM NAME is output.	24 bytes
Index column	INDEX is output.	10 bytes
Data column	The specified data name is output.	512 bytes
Trigger occurrence information column	Trigger is output.	14 bytes

Ex.

The size of the data name row can be obtained by the following equation, when performing the data logging of 128 data items from D100 to D227 (when selecting the option which outputs date/time column, data collection interval column, execution step No. column, execution program number column, execution program name column, and index column):

28(date/time column) + 24(data collection interval column) + 16(execution step No. column) + 22(execution program number column) + 24(execution program name column) + 10(index column) + (6 × 128)(data column) + 264(number of tabs) + 4(CR + LF) = 1138 bytes

• Device comment row

The device comment row for each column is displayed.

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

*3 When "Output device comment" is selected in "Device comment output" of the output settings and double quotation marks (" ") and/or commas (,) are included in a device comment, the size of data in use becomes larger than the size of the character string of the device comments.

Ex.

When data logging is performed on data of 128 points of D100 to D227, the following calculation formula determines the size of a device comment row. (The date and time format is YYYY/MM/DD hh:mm:ss.sss, data collection interval column, execution step No. column, execution program No. column, execution program name column, and index column are output, and "Output device comment" is specified only within the range of D200 to D227.)

0(date/time column) + 0(data collection interval column) + 0(execution step number column) + 0(execution program number column) + 0(execution program name column) + 0(index column) + (0 × 100 + 14 × 28)(data column) + 264(number of tabs) + 2(CR + LF) = 660 bytes

• Data row

The collected data value is displayed. All the data items collected during a single collection is displayed in a single row.

Item	Description	Size
Date/time column	Information is output according to the data row output character string specified in the format.	2 to 64 bytes
Data collection interval column	The time interval from the previous collection time to the current collection time is output. If the maximum display range is exceeded, the count returns to 1 and starts again to output a new time interval (unit: μs, display range: 1 to 100000000000).	2 to 24 bytes
Execution step No. column	Execution step No. on CPU Module Logging Configuration Tool at the time interval and timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the step No. at the time of execution of END instruction in the last executed program on the scan. With the collection condition "Interrupt occurrence" specified, the resulting output is the step No. at the time of execution of IRET instruction in the interrupt program. If the system operation (such as system interrupt) is running during collection interval and collection timing or the FB program is in running, "0" is output.	2 to 12 bytes
Execution program number column	Execution program number (FB file number) on CPU Module Logging Configuration Tool at the time interval and timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the program number of the last executed program in the scan. If two or more system operations, such as collection timing and system interrupt, occur simultaneously, — is output.	2 to 6 bytes
Execution program name column	Execution program name (FB file name) on CPU Module Logging Configuration Tool at the time interval and timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the program name of the last executed program in the scan. If a program name with the same program number has already been in the file, a null value is output. If two or more system operations, such as collection timing and system interrupt, occur simultaneously, "* SYSTEM" is output.	0 to 200 bytes
Index column	A value which increments in ascending order from 1 is output. When it exceeds the upper limit, it returns to 1 and increments again (range: 1 to 4294967295).	2 to 20 bytes
Data column	When bits are specified: bit On = 1 and bit Off = 0 are output.	2 bytes
	When unsigned/signed word type is specified: data value is output according to the specified output type.	• Decimal format: 2 to 22 bytes • Hexadecimal format: 2 to 16 bytes • Decimal fraction format: 2 to 42 bytes • Exponential format: 10 to 42 bytes
	When unsigned/signed double word type, single-precision real number, or double-precision real number is specified: data value is output according to the specified output type.	• Decimal format: 2 to 22 bytes*4 • Hexadecimal format: 2 to 16 bytes • Decimal fraction format: 2 to 52 bytes • Exponential format: 10 to 44 bytes
	When character string is specified: the specified character string is output.	2 to 256 bytes
	When numeric string is specified: the character string which represented by the hexadecimal in increments of a byte is output without clearance.	4 to 1024 bytes
	When time is specified: T#-24d20h31m23s648ms to T#24d20h31m23s647ms is displayed.	26 to 40 bytes
Trigger occurrence information column	The specified character string is output when the trigger occurs. In other cases, no character string is output.	0 to 1024 bytes

*4 When single-precision real number or double-precision real number is specified, if the numerical value to be output does not fall into the range -2147483648.0 to 4294967295.0, it is displayed in an equivalent format to "exponential format and the number of decimal part digits is nine."

Ex.

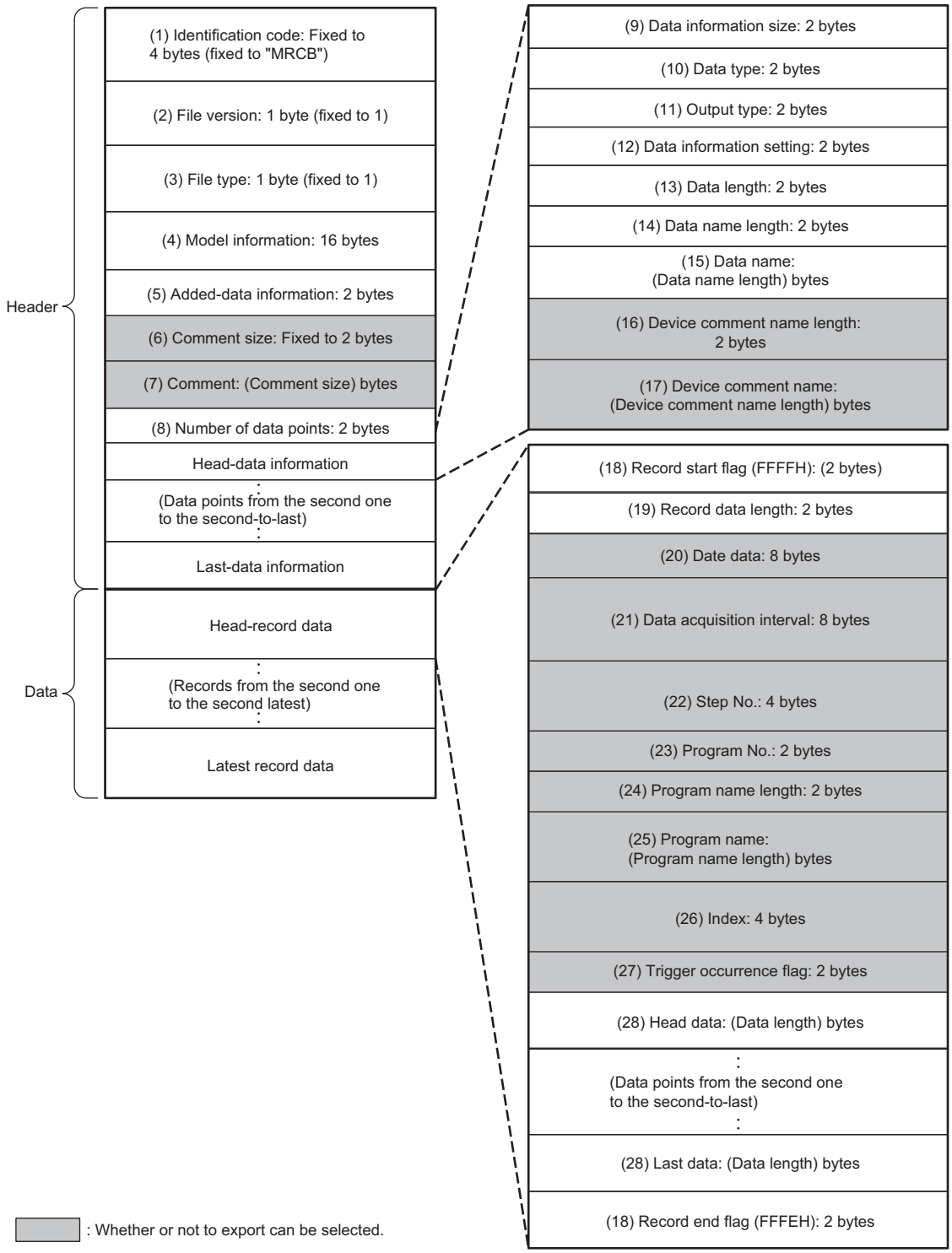
When data logging is performed on data of 128 points of D100 to D227 in the unsigned word type format (decimal), the following calculation formula determines the size of a data row. (The date and time format is YYYY/MM/DD hh:mm:ss.sss, data collection interval column, execution step No. column, execution program No. column, execution program name column, and index column are output.)

46 (date and time) + 24 (data collection interval column) + 12 (execution step No. column) + 6 (execution program No. column) + 200 (execution program name column) + 20 (index column) + (12 × 128) (data column) + 264 (number of tabs) + 4 (CR + LF) = 2112 bytes

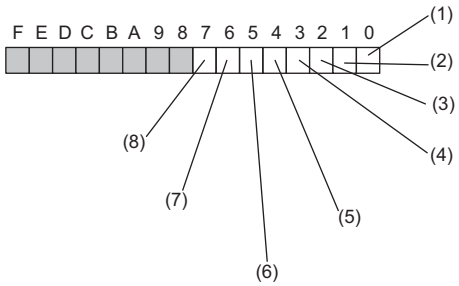
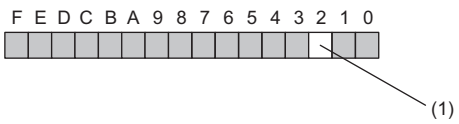
■Binary file output format

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

• Configuration figure of binary format



• Details of each data

No.	Item	Description	Size (byte)
(1)	Identification code	The MRCB is always output to identify the file.	4
(2)	File version	File version 1 is displayed.	1
(3)	File type	The file type is output. (fixed to 1: Continuous/trigger logging)	1
(4)	Model name information	The module model name that outputted binary file is output. R□CPU is output to the first eight bytes and 00H is output to the last eight bytes.*1	16
(5)	Added data information	The output selection setting for the data that can be output is output.  (1) 1: Output date and time data. 0: Do not output date and time data. (2) 1: Output a data sampling interval. 0: Do not output a data sampling interval. (3) 1: Output an execution step No. 0: Do not output an execution step No. (4) 1: Output an execution program name and No. 0: Do not output an execution program name or No. (5) 1: Output a trigger flag. 0: Do not output a trigger flag. (6) 1: Output index. 0: Do not output index. (7) 1: Output device comments. 0: Do not output device comments. (8) 1: Output comments. 0: Do not output comments.	2
(6)	Comment size	The comment length of (7) Comment is output.	2
(7)	Comment	The comment specified in the setting is output in Unicode.	2 to 512
(8)	Number of data	The number of data points of the data information ((9) to (17)) for data logging is output.	2
(9)	Data information size	The total size of the data information ((9) to (17)) for data logging is output.	2
(10)	Data type	The numeric value shown below is output depending on the data type. 0000H: Bit 0001H: Word (signed) 0002H: Double word (unsigned) 0003H: Word (signed) 0004H: Double word (unsigned) 0005H: Single-precision real number 0006H: Double-precision real number 0007H: String 0008H: Numeric string 0009H: Time	2
(11)	Output format	The same numerical value as the value in (10) Data type is output. However, when the data type is bit, character string, numeric string, or time, FFFFH is output because the output type cannot be specified.	2
(12)	Data information setting	The data-related information is output.  (1) 1: No device comment specified 0: Device comment specified	2
(13)	Data length	The data length of data is output. When the data type is the bit type, it will be output as two bytes.	2
(14)	Data name length	The length of the data name specified in the setting is output.	2
(15)	Data name	The data name specified in the setting is output in Unicode.	2 to 512
(16)	Device comment length	The length of the device comment name specified in the setting is output.	2

No.	Item	Description	Size (byte)
(17)	Device comment name	The device comment name specified in the setting is output in Unicode.	0 to 2048
(18)	Record Start Flag, Record End Flag	The flags for identifying the start and end of the record are output. The FFFFH is output for record start while the FFFEH is output for record end as the fixed flag.	2
(19)	Record data length	The total size of (20) Day and time data to (28) Last data is output.	2
(20)	Day and time data	The Day and time data is output. b15 to b0 Year / Month: Year: Last 2 digits of the year, Month: 1 to 12 Day / Time: Day: 1 to 31, Time: 0 to 23 Minute / Second: Minute: 0 to 59, Second: 0 to 59 Millisecond: Millisecond: 0 to 999	8
(21)	Data sampling interval	The time interval from the previous collection time to the current collection time is output. (Unit: μs, Display range: 1 to 100000000000 (When it exceeds the max value, it returns to "1" and incrementing runs again.)) After logging collection is started, 0 is stored at the first collection.	8
(22)	Step No.	The step No. executed with the engineering tool at the timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the step No. at the time of execution of END instruction in the last executed program on the scan. With the collection condition "Interrupt occurrence" specified, the resulting output is the step No. at the time of execution of IRET instruction in the interrupt program. If the system operation (such as system interrupt) is running during collection interval and collection timing or the FB program is in running, "0" is output.	4
(23)	Program No.	The program No. (FB File No) executed on the engineering tool at the timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the program number of the last executed program in the scan. If system operation (such as system interrupt) is running at collection interval and collection timing, "0" is output.	2
(24)	Program name length	The name length of a program that is executed at the time interval and timing in which data was collected is output. If the same program number or program name has already been in the file, "0" is output.	2
(25)	Program name	The program name that is executed at the time interval and timing in which data was collected is output in Unicode. With the collection condition "Each scanning cycle" specified, the resulting output is the program name of the last executed program in the scan. If system operation (such as system interrupt) is running at collection interval and collection timing, "* SYSTEM" is output.	0 to 200
(26)	Index	The index number ranging from 1 to 4294967295 of data, which was collected by the data logging function, is output. When it exceeds the max value, it returns to "1" and incrementing runs again. If missing occurs in processing data, index will be reassigned from 1 again.	4
(27)	Trigger occurrence flag	The trigger occurrence information is output. b15 to b1 b0 (1) (1) 1: A trigger has occurred. 0: No trigger has occurred.	2
(28)	Data	Data collected by the data logging function is output corresponding to (13) Data length and (10) Data type. • When bits are specified: bit On = 1 and bit Off = 0 are output. • When word type (signed/unsigned) or double-word type (signed/unsigned) is specified: the data values are output in the specified unit. • When single-precision real number or double-precision real number is specified: data value is output in the specified unit. (☞ Page 739 Numerical value range for each output type) • When character string type is specified: the character string with the specified size is output. If the character string terminator "0" exists in the middle of data, NULL is generated on from said point onward until the terminator of the specified size. • When numeric string type is specified: the data value with the specified size is output. • When time is specified: T#-24d20h31m23s648ms to T#24d20h31m23s647ms is output.	• Bit: 2 • Word (signed/unsigned): 2 • Double word (signed/unsigned): 4 • Single-precision real number: 4 • Double-precision real number: 8 • String/numeric string: 1 to 256 • Time: 4

*1 The following capacity values are output to □based on the model.
Example: R08CPU, R08ENCPU, R08PCPU, R08SFCPU: □ = 08

Numerical value range for each output type

This section describes the numerical value ranges that can be output for each output type.

■Integer type

The following table lists the numerical value ranges that can be expressed for each integer type.

Output format	Lower limit	Upper limit
Word (unsigned)	0	65535
Word (signed)	-32768	32767
Double word (unsigned)	0	4294967295
Double word (signed)	-2147483648	2147483647

■Real number type

The following table lists the numerical value ranges that can be expressed for each real number type.

Output format	Negative value		Positive value	
	Lower limit	Upper limit	Lower limit	Upper limit
Single-precision real number	-3.4028235E+38	-1.401298E-45	1.401298E-45	3.4028235E+38
Double-precision real number	-1.79769313486231570E+308	-4.94065645841246544E-324	4.94065645841246544E-324	1.79769313486231570E+308

When the value of the data exceeds the numerical value range, the following rule is applied to the output.

- When the value exceeds the upper limit of the positive value, +Inf is output.
- When the value falls below the lower limit of the negative value, -Inf is output.
- When the value is between the upper limit of the negative value and the lower limit of the positive value, 0 is output.

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

A

Data logging procedure

This section describes the data logging procedure.

1. Install CPU Module Logging Configuration Tool. (📖 CPU Module Logging Configuration Tool/GX LogViewer Installation Instructions (BCN-P5999-0506))
2. Launch CPU Module Logging Configuration Tool. (☞ Page 740 Launching CPU Module Logging Configuration Tool)
3. Configure the data logging setting. (☞ Page 745 Setting data logging)
4. If auto logging is used, configure the common settings (auto logging common settings). (☞ Page 758 Auto logging common setting)
5. Attach an SD memory card into the CPU module, and power up the module.
6. Connect the CPU module to a personal computer. (☞ Page 726 System configuration, ☞ Page 759 Transfer setup).
7. Write the settings to the CPU module or the SD memory card. (☞ Page 761 Writing the logging setting)
8. Switch the CPU module to RUN state to start the data logging. (☞ Page 762 Logging status and operation)
9. Stop the data logging and read the data logging file. (☞ Page 764 Logging file operation)
10. Check the file into which data has been read.

Point

For details on installation and uninstallation procedures of CPU Module Logging Configuration Tool, refer to the following.
📖 CPU Module Logging Configuration Tool/GX LogViewer Installation Instructions (BCN-P5999-0506)

Launching CPU Module Logging Configuration Tool

The following describes three methods to launch CPU Module Logging Configuration Tool.

■Launch from the start menu

After CPU Module Logging Configuration Tool is installed, it can be launched by the following operation.

[Start] ⇨ [All Programs] ⇨ [MELSOFT] ⇨ [Logging Function] ⇨ [CPU Module Logging Configuration Tool]

■Launch from GX Works3

After GX Works3 is launched, CPU Module Logging Configuration Tool can be launched by the following operation.

[Tool] ⇨ [Logging Configuration Tool]

It is launched with the information (connection target device, connection target specification/connection target setting, and display language) of the GX Works3 project.

■Launch from GX LogViewer

Refer to the following manual.

📖 GX LogViewer Version 1 Operating Manual

Menu items and setting methods

This section describes the operation method and/or setting windows of CPU Module Logging Configuration Tool.

Menu structure

The following table describes the menu structure of CPU Module Logging Configuration Tool.

Menu item		Description
Project	New	Create a new project.
	Open	Open a stored project file.
	Save	Overwrite an edited project to the file and saves it.
	Save As	Save an edited project with a new file name.
	Read Logging Setting from Memory Card (SD)	Read the data logging setting written in the SD memory card attached to the personal computer.
	Write Logging Setting into Memory Card (SD)	Write the settings being edited in a format with which the CPU module can operate. Once the settings is directly written into an SD memory card attached to the personal computer and then the SD memory card is mounted onto the CPU module, the data logging starts.
	Recent Files	Open a recently used file.
	Exit	Exit CPU Module Logging Configuration Tool.
Edit	Delete Data Logging Setting	Remove the data logging setting selected in the Edit item tree.
	Copy and Add Data Logging Setting	Copy and add the data logging setting selected in the Edit item tree.
	Batch Data Insertion	Configure the multiple setting items at once.
	Cut Setting Item	Cut the setting with a table format.
	Copy Setting Item	Copy the setting with a table format.
	Paste Setting Item	Paste the setting with a table format.
	Paste device copied in other application software	Paste a device which is copied in other application into a line selected. (Existing data in the line are overwritten and erased.)
	Insert Copied Setting Item	Paste the copied setting with a table format.
	Insert device copied in other application software	Paste a device which is copied in other application into a line selected and shifts down existing data in the selected line and below.
	Insert Cut Setting Item	Paste the cut setting with a table format.
	Delete Setting Item	Remove the setting with a table format.
	Move Setting Item Upward	Move up a setting item.
	Move Setting Item Downward	Move down a setting item.
	Device Batch Replacement	Replace devices for all the settings.
View	Switch Display Language	Change the display language for menus and so on.
Online	Transfer Setup	Configure the communication setting used for connection to the CPU module.
	Read Logging Setting	Read the setting from the CPU module.
	Write Logging Setting	Write the setting to the CPU module.
	Delete Logging Setting	Remove the setting data from the CPU module.
	Logging Status and Operation	Check the data logging status.
	Logging File Operation	Connect to the CPU module and reads or removes the files on the attached SD memory card.
Tool	Start GX LogViewer	Launch GX LogViewer.
Help	Open Manual	e-Manual Viewer opens and its manual is displayed.
	Connection to MITSUBISHI ELECTRIC FA Global Website	The Mitsubishi Electric Corporation FA website is displayed.
	About Configuration tool	The product information is displayed.

Project management

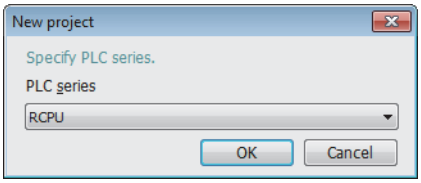
This function creates and saves the project, and reads/writes it from/to an SD memory card.

■Creating new project

Create a new project.

[Project] ⇨ [New]

Window



Displayed items

Item	Description
PLC series	Select "RCPU".

■Opening project

Open a stored project file.

[Project] ⇨ [Open]

■Saving project

• Overwriting

Overwrite an edited project to the file and saves it.

[Project] ⇨ [Save]

• Saving with a new file name

Save an edited project with a new file name.

[Project] ⇨ [Save As]

■Reading the logging setting from SD memory card

The following procedure is to read the data logging setting written in an SD memory card attached to the personal computer.

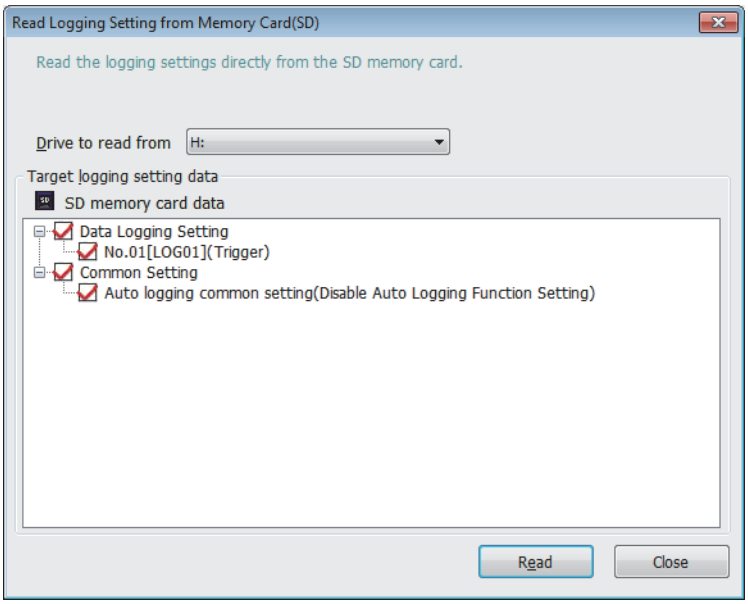
Operating procedure

1. Attach an SD memory card to the personal computer.
2. Open the following window.

[Project] ⇨ [Read Logging Setting from Memory Card (SD)]

3. Select the drive from which data is read and data to be read.
4. Click the [Read] button.

Window



Displayed items

Item	Description
Drive to read from	Select the drive where the data to be read is stored.
Target logging setting data	Select the data item to be read.

Point

- Any existing data (data logging setting with the same setting number or common setting) on the target is overwritten.
- The PLC series of the setting data in the SD memory card must be "RCPU".

■Writing the logging setting to SD memory card

The following procedure is to write the settings being edited in a format with which the CPU module can operate. Once writing the settings directly into an SD memory card attached to the personal computer and attaching the card to the CPU module, the data logging starts.

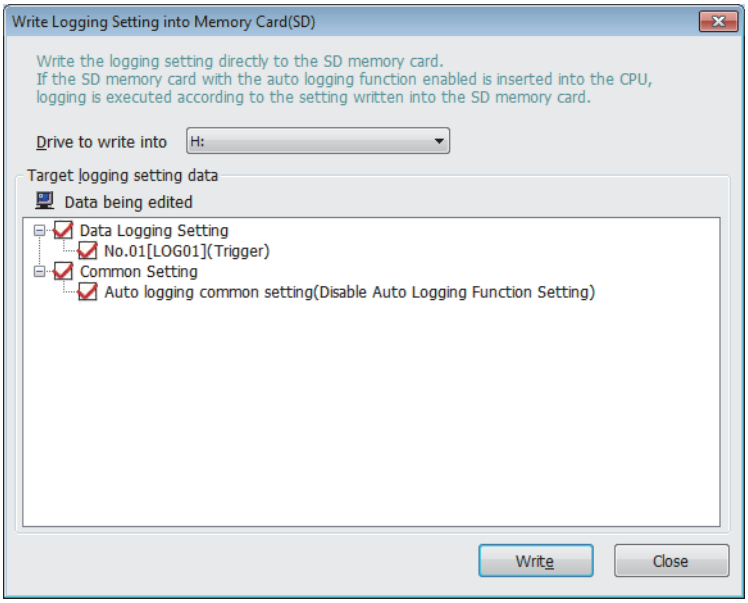
Operating procedure

1. Attach an SD memory card to the personal computer.
2. Open the following window.

[Project] ⇨ [Write Logging Setting into Memory Card (SD)]

3. Select the drive to which data is written and data to be written.
4. Click the [Write] button.

Window



Displayed items

Item	Description
Drive to write into	Select the drive where the data to be written is stored.
Target logging setting data	Select the data to be written.

Point

Any existing data (data logging setting with the same setting number or common setting) on the target is overwritten.

Setting data logging

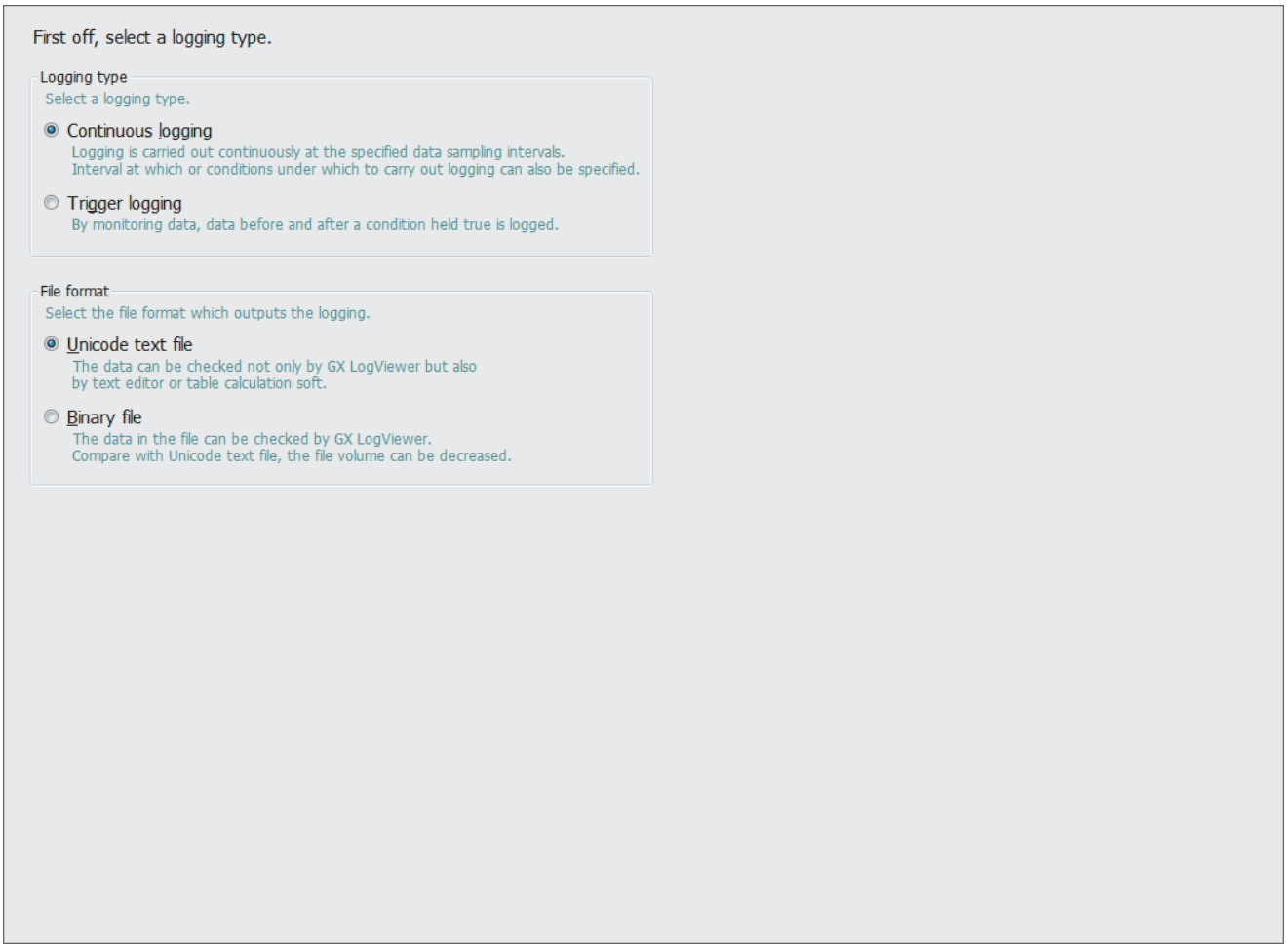
This menu item launches a wizard that helps users to configure the required settings for using the data logging function.

Edit item tree ⇨ [RCPU] ⇨ [Data Logging Setting] ⇨ [Edit] button

■Logging type

The following window configures the data logging type and file format (☞ Page 194 Logging Type, ☞ Page 731 Data output type).

Window



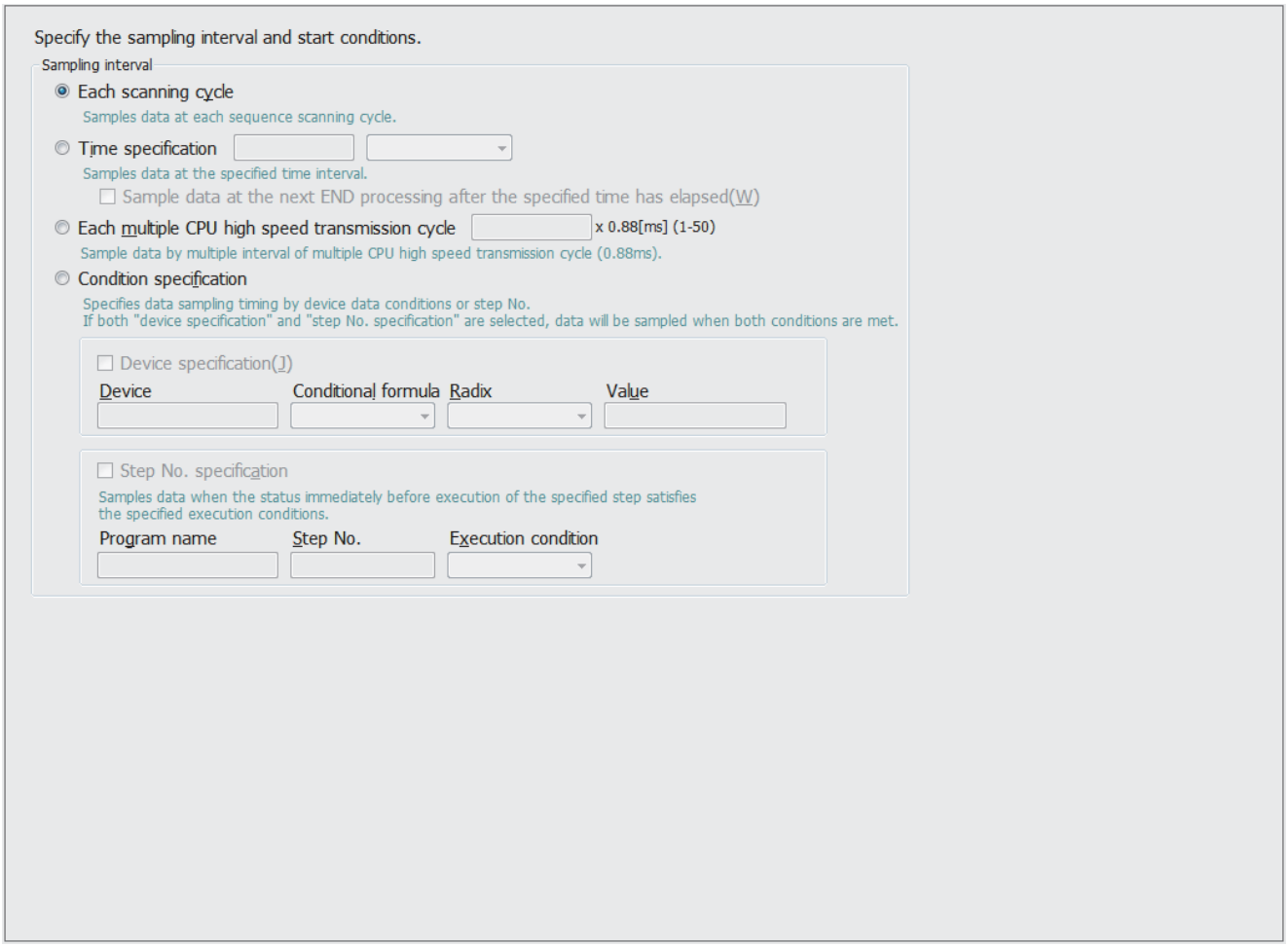
Displayed items

Item	Description	Setting range	Default
Logging type	Select the logging type.	• Continuous logging • Trigger logging	Continuous logging
File format	Select the output file format. (☞ Page 203 Storage format of data logging files)	• Unicode text file • Binary file	Unicode text file

■Collection

The following window configures the collection interval and/or collection start conditions (☞ Page 189 Data Collection Conditions)

Window



Displayed items

Item		Description	Setting range	Default
Each scan		Select this item to collect scan data obtained for each scan operation.	—	—
Time specification	Collection time value	Specify the collection interval value.	• ms: 1 to 32767 • second: 1 to 86400	1ms
	Sample data at the first END processing after the specified time has elapsed	Select this checkbox to collect data at a timing when the first END processing is done after the specified time interval is elapsed.	—	Checked
Interrupt occurrence	Interrupt pointer	Select the interrupt cycle.	• I28: interrupt by internal timer • I29: interrupt by internal timer • I30: interrupt by internal timer • I31: interrupt by internal timer • I44: inter-module synchronous interrupt • I45: multiple CPU synchronous interrupt • I48: high-speed internal timer interrupt 2 • I49: high-speed internal timer interrupt 1	I28: interrupt by internal timer
	Multiple	Specify a multiple applied to the interval value.	1 to 50	1
Condition specification	Device specification	Specify the data collection timing according to the device data conditions.	☞ Page 191 Device specification	Checked
	Step No. specification	Specify the data collection timing based on the step No.	☞ Page 192 Step No. specification	—

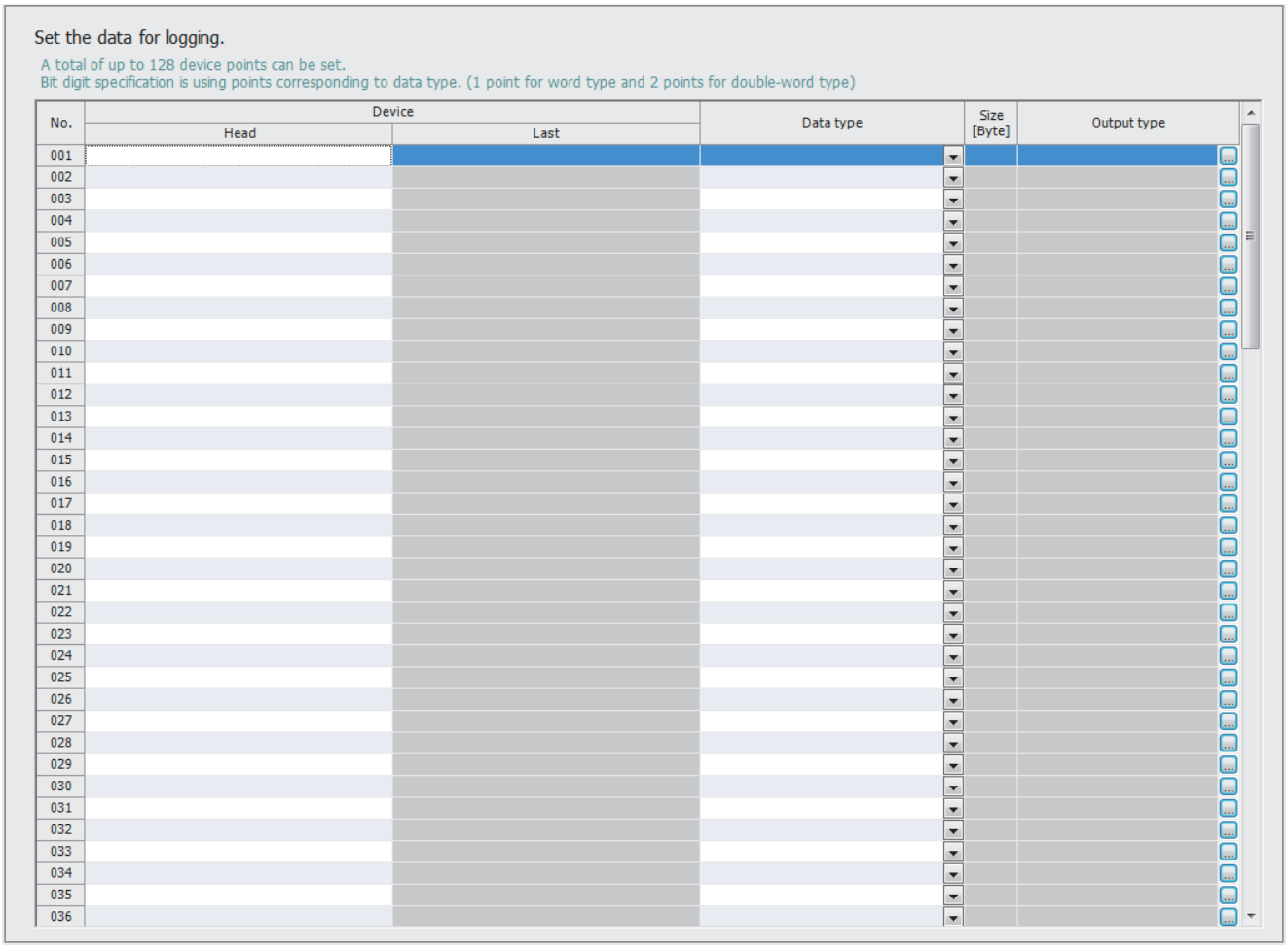
■Data

The following window configures the various items such as data format of the target collection device.

Point

- To paste a device which is copied in other application (such as Watch window of GX Works3 or GX Works2, a spreadsheet, or a text editor), use "Paste device copied in other application software" or "Insert device copied in other application software". (☞ Page 748 Paste/Insert device copied in other application software).
- To insert a block of data into the data list all at once, use "Batch Data Insertion". (☞ Page 749 Batch insertion of data)

Window



Displayed items

Item		Description	Setting range	Default
No.		In this column, the data setting numbers from 001 to 128 are displayed.	—	—
Device	Head	Specify the start device number.		
	Last	In this column, the end device number calculated based on the data type and size is displayed.		
Data type		Select the type of data to be collected.	☞ Page 188 Data type	
Size [Byte]		Specify the data size when the data type is set to "String" or "Raw".	1 to 256 bytes	
Output Format		Clicking the [...] button at the rightmost part of each row displays the "Output Format (integer·float)" list. Select the format to be used when data is output to the file.	☞ Page 731 Data output type	

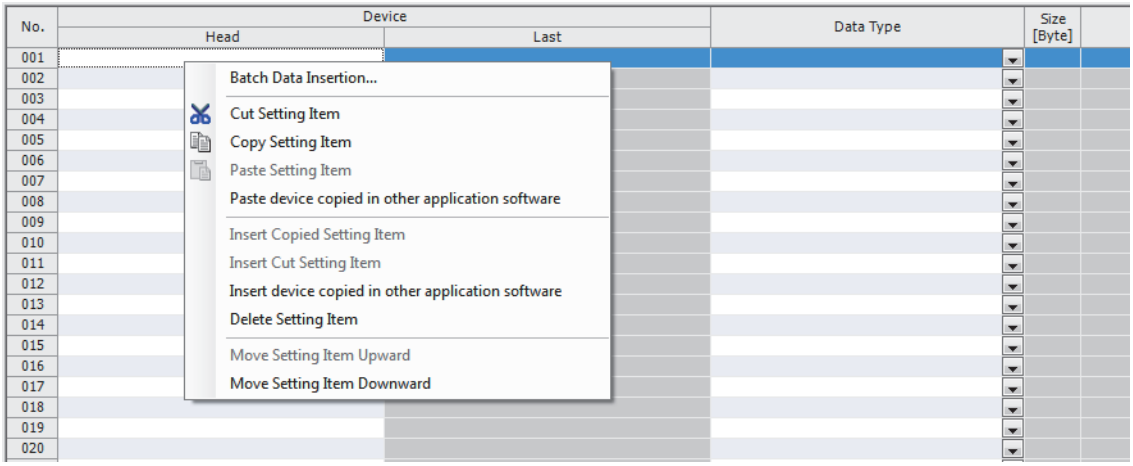
■Paste/Insert device copied in other application software

A device which is copied in other application (such as Watch window of GX Works3 or GX Works2, a spreadsheet, or a text editor) can be paste to any line in the setting window of "Data".*1

*1 CPU Module Logging Configuration Tool with version "1.70Y" or later supports this function.

Operating procedure

1. On the setting window of "Data", select a line where copied data is to be pasted and select the item in the right-click menu.



Point

The keyboard shortcut for "Paste device copied in other application software" is "Ctrl + Shift + V". Note that this keyboard shortcut is valid only when the start device column is selected.

Processing details

The following table summarizes the operation details of the functions.

Item	Operation
Paste device copied in other application software	Pastes a device which is copied in other application into a line selected. (Existing data in the line are overwritten and erased.)*1*2 Settings other than start devices are entered as default settings correspond to each device.
Insert device copied in other application software	Pastes a device which is copied in other application into a line selected and shifts down existing data in the selected line and below.*1*2 Settings other than start devices are entered as default settings correspond to each device.

*1 From the start to the newline or the first tab of strings in each line on the clipboard are pasted or inserted.

*2 If a copied string is invalid as a device, it cannot be pasted or inserted.

■Batch insertion of data

The following window is to insert data items into the data list at once. Data is inserted into blank rows in the list of the "Data" setting window in order from the top (when a setting already exists in the target insertion row, the row is skipped without overwriting it).

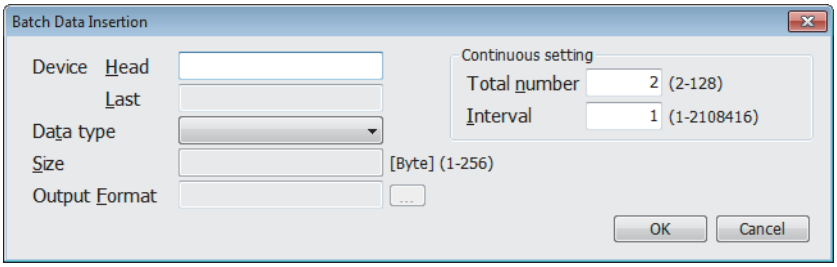
Operating procedure

1. Open the following window.

[Edit] ⇨ [Batch Data Insertion]

2. Configure the setting items and continuous settings, and click the [OK] button.

Window



Displayed items

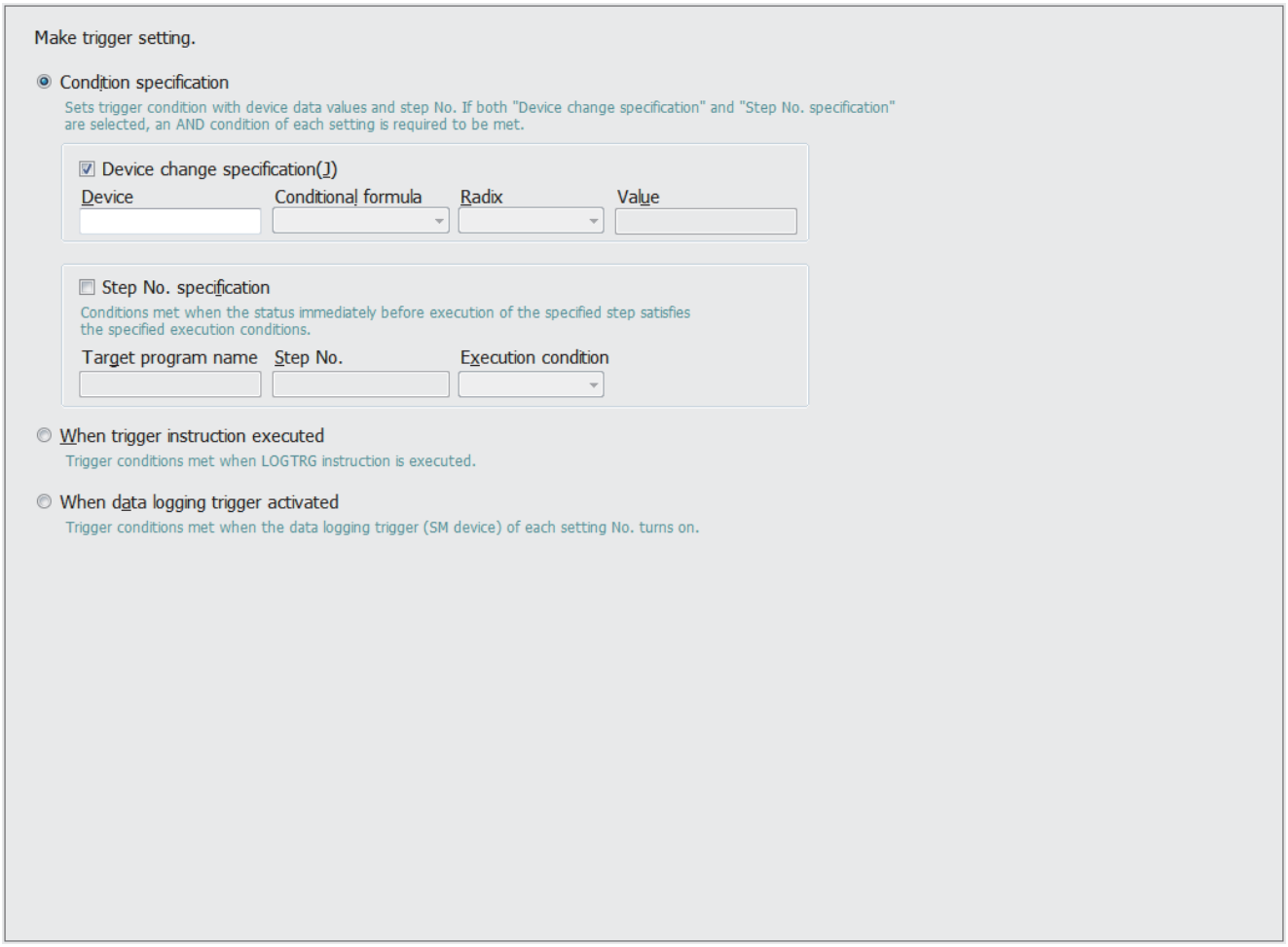
Item		Description	Setting range	Default
Device	Head	Same as the data setting (☞ Page 747 Data)	—	—
	Last			
Data type				
Size				
Output Format				
Continuous setting	Total number	Specify the total number of data items to be inserted at once.	2 to 128	
	Interval	Specify the device interval of data to be inserted at once.	1 to 268435455	

A

■Trigger

The following window specifies the trigger condition when the trigger logging is selected (☞ Page 196 Trigger Condition)

Window



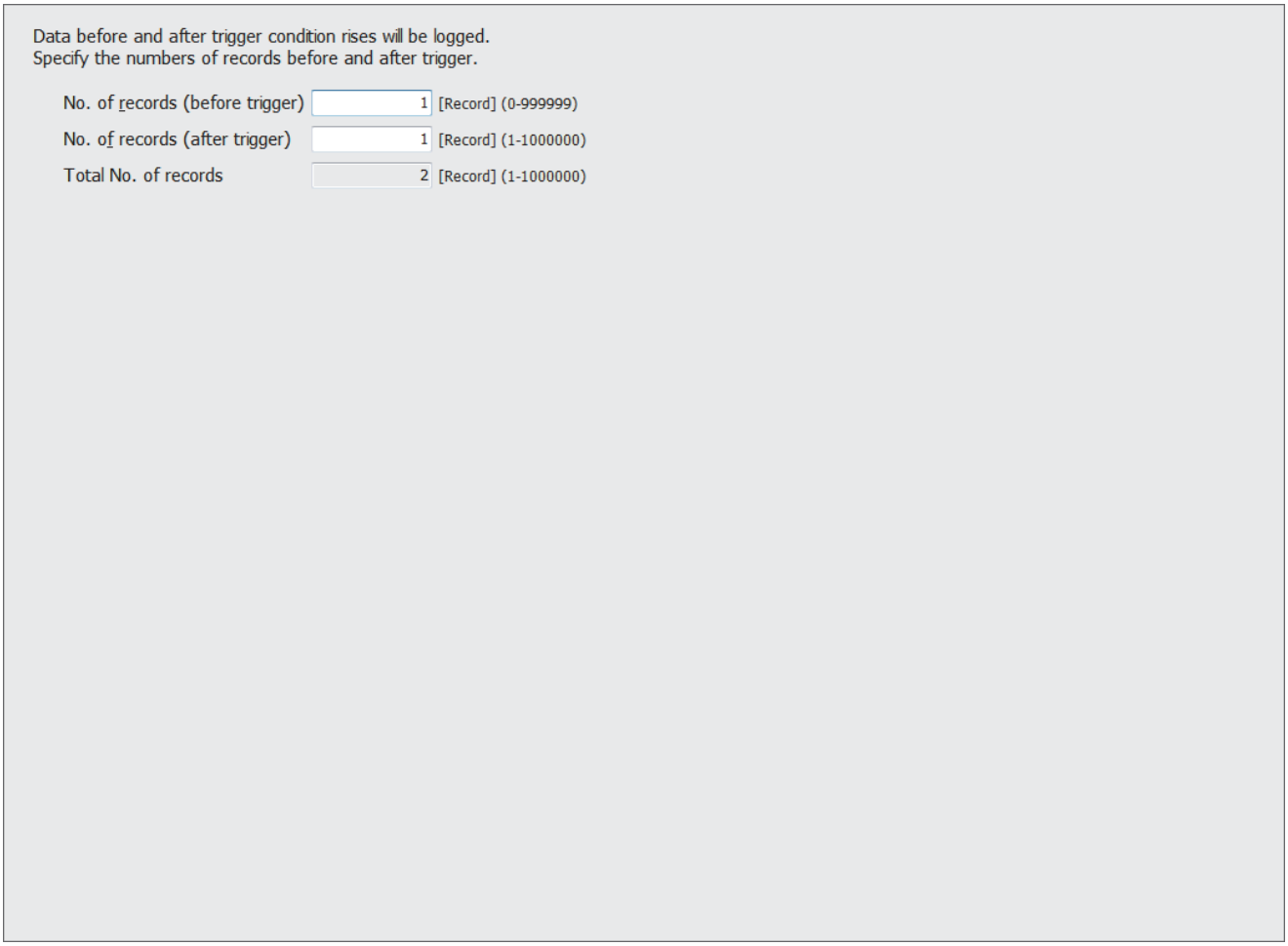
Displayed items

Item		Description	Setting range	Default
Condition specification	Device change specification	Configure the trigger condition based on the device data condition.	☞ Page 196 Device change specification	Checked
	Step No. specification	Configure the trigger condition based on the step No.	☞ Page 192 Step No. specification	—
When trigger instruction executed		Trigger condition is established when the LOGTRG instruction is executed.	—	—

■Number of records

The following window specifies the number of records to be output before and after trigger occurrences when the trigger logging is selected (☞ Page 195 Number of records)

Window



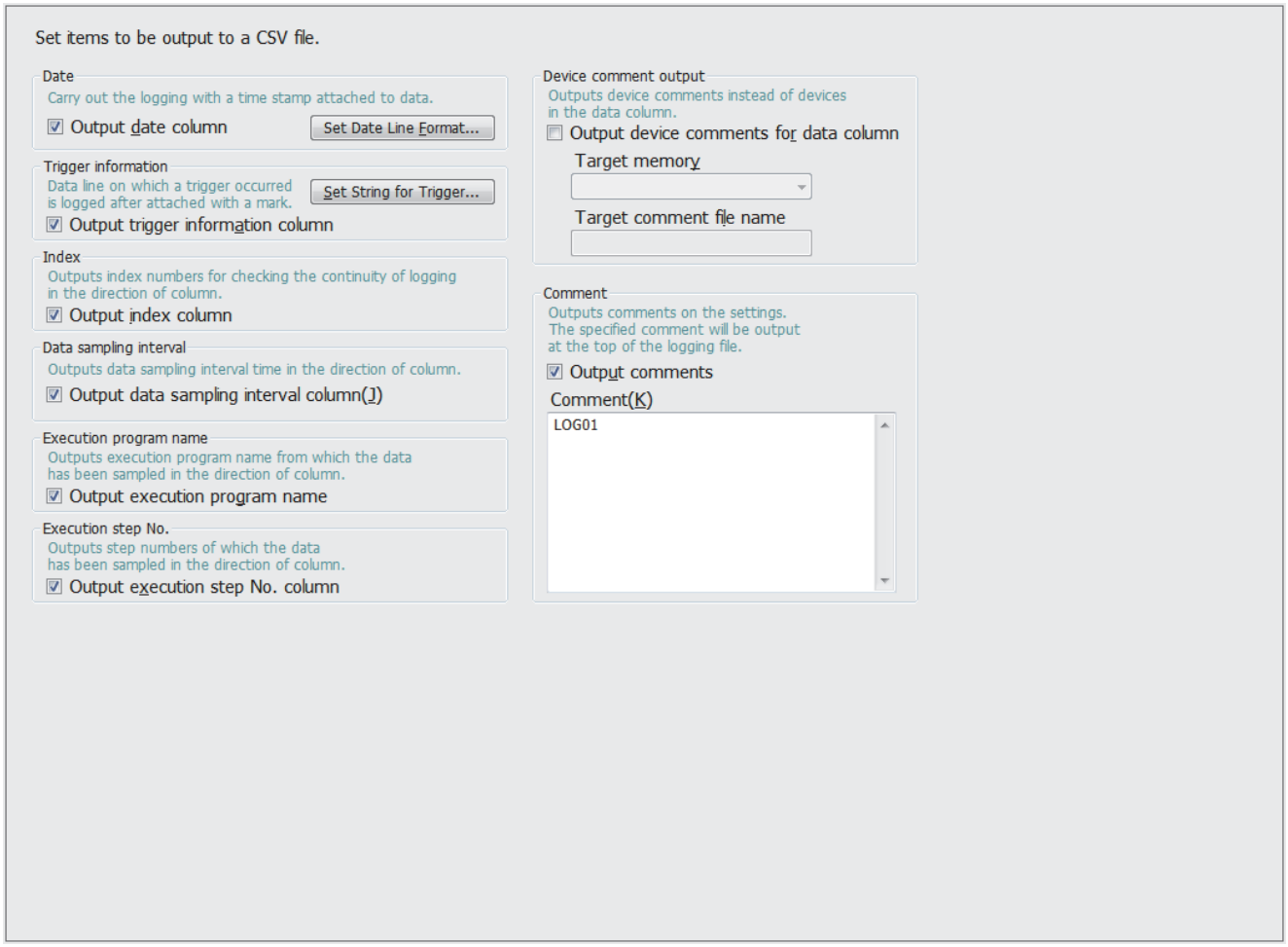
Displayed items

Item	Description	Setting range	Default
No. of records (before trigger)	Specify the number of records to be output as pre-trigger record.	0 to 999999	1
No. of records (after trigger)	Specify the number of records to be logged during and after a trigger occurrence.	1 to 1000000	1
Total No. of records	View the total number of pre-trigger and post-trigger records.	—	2

■Output

The following window specifies the items to be output into the file. (☞ Page 731 Data output type)

Window



Displayed items

Item		Description	Setting range	Default
Date	Output date	Add a time stamp to data for the data logging. The date/time format to be output can be configured by clicking the [Date Line Output Format] button.	*1*2	Checked
Trigger information	Output trigger information	Add a mark to data items that are associated with a trigger occurrence for the data logging. A character string to be added to the trigger occurrence data can be configured by clicking the [String for Indicating Trigger Occurrence] button	256 characters or less	
Index	Output index	Output the index number used for checking the logging continuity.	—	
Data sampling interval	Output data sampling interval	Output the data collection interval.		
Execution program name or execution program No.	Output execution program name or execution program No.	Output the execution program name and execution program number used for the data collection.		
Execution step No.	Output execution step No.	Output the step No. used for the data collection.		
Device comment output	Output device comment	If the collection data was specified with the device, the device comment is output in addition to the device.		Not checked
	Target memory	Select the storage memory for the comment file used when the comment is displayed.	• Memory card (SD) • Data memory	—
	Target comment file name	Set the comment file name used when the comment is displayed.	60 characters or less	—
	Target comment No.*3	Specify the comment number to be displayed.	1 to 16	—
Comment	Output comments	Output the comment at the top of the file.	—	Checked
	Comment*4	Input the comment in this box.	256 characters or less (No line feed can be used.)	LOG01

*1 Data output format can be created by combining the following formats.
· Year: YYYY for four-digit expression; YY for two-digit expression
· Month: MM
· Day: DD
· Hour: hh
· Minute: mm
· Second: ss
· Millisecond: ms (three-digit expression), or s, ss, sss, ssss, sssss, ssssss, or sssssss (second unit after the decimal point, maximum of seven digits)
Example: YY/MM/DD hh:mm:ss.sss → 10/10/13 09:44:35.241

*2 When either of "Year," "Month," "Day," "Hour," "Minute," or "Second" is omitted, if opening the data logging file by GX LogViewer, the index expression is used rather than the time expression.
GX LogViewer Version 1 Operating Manual

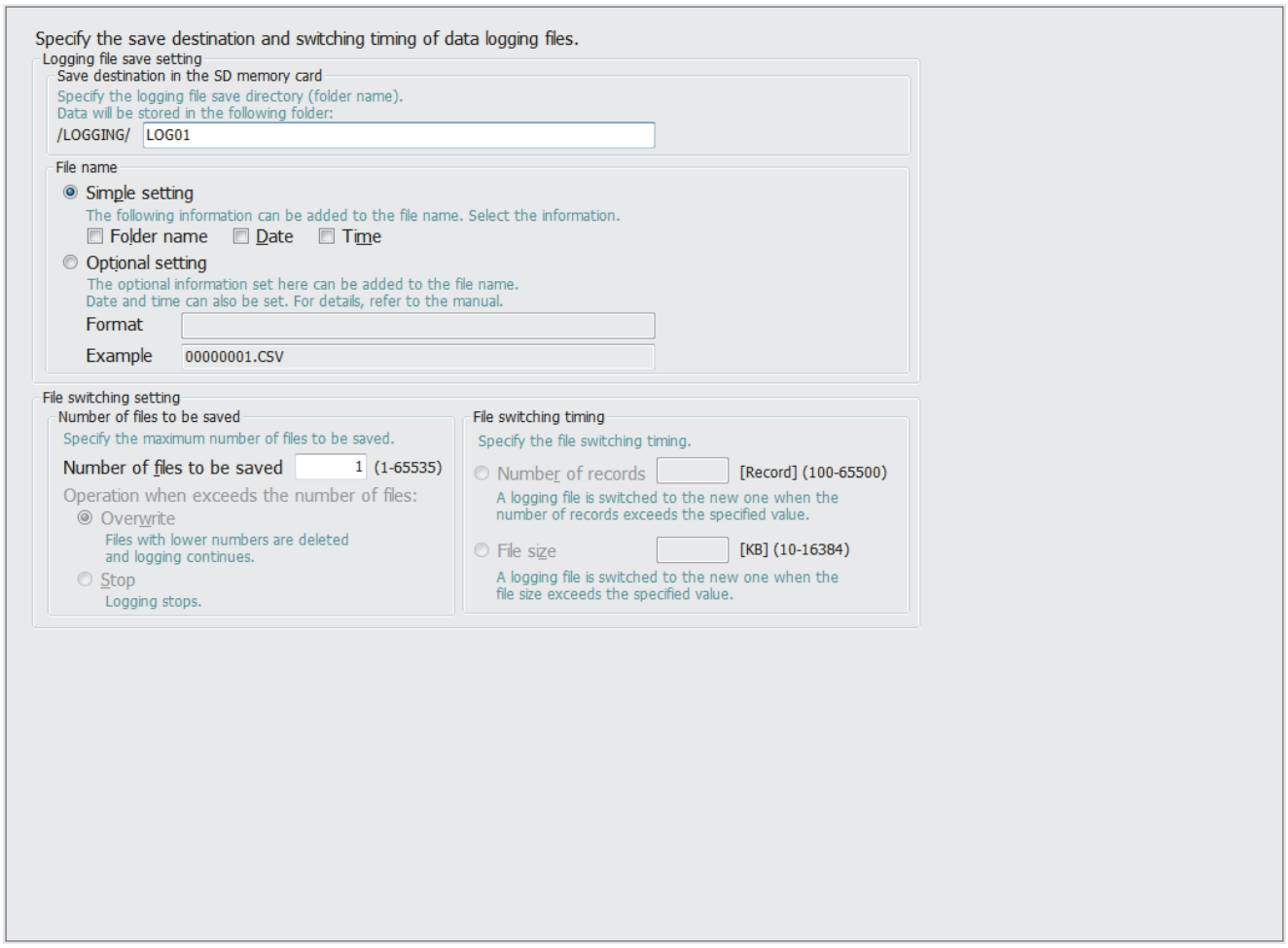
*3 The target comment No refers to the numbers, which correspond to the comment numbers in the [View] ⇨ [Multiple Comment Display Setting] of the Engineering Tool.

*4 When the following user action is detected, character entry will be disabled
· Entered a character which cannot be handled with the OS language character code.
· Entered a character whose language code is different from the one for characters already input in the same data logging setting.

■Save

The following window configures the target storage for data logging file and switching timing of storage files (☞ Page 200 Switching to a storage file)

Window



Displayed items

Item			Description	Setting range	Default
Logging file save setting	Save destination in the SD memory card		Specify the storage folder for the data logging file.	60 characters or less (double-byte character not allowed)	LOG01
	Folder to store file to be saved		Select information to be added to the name of the folder which stores the storage file.	—	Not checked
	File name	Simple setting	Select information to be added to the name of the storage file.	—	Checked
		Optional setting	Specify information format to be added to the name of the storage file.	*1*2	—
		Add date type	Add date/time when the file switching condition is satisfied and when the file is created if the optional setting is selected.	—	—
File switching setting	Number of files to be saved	Number of files to be saved	Specify the maximum number of storage files.	1 to 65535	1
		Operation when the number of files exceeds the set value	Select the operation when the number of files exceeds the maximum number of files to be saved. (☞ Page 201 Action to take when the maximum number of storage files is exceeded)	• Overwrite • Stop	Overwrite
	File switch timing*3		Select the timing at which the file is replaced with new one (☞ Page 201 File switching condition)	• Number of records: 1 to 65500 • File size: 10 to 16384K bytes	—

*1 Date and/or time can be added in any format by using the following character strings.
· Year: YYYY for four-digit expression; YY for two-digit expression
· Month: MM
· Day: DD
· Day of the week: ddd (Sunday: Sun, Monday: Mon, Tuesday: Tue, Wednesday: Wed, Thursday: Thu, Friday: Fri, Saturday: Sat)
· Hour: hh
· Minute: mm
· Second: ss
Example: for June 18, 2014 (Wednesday), 09:30:15, YYYYMMDDdddhhmmss → 20140618Wed093015_00000001.bin
Also when using the additional information simply as a character string rather than the above format, any character string can be added by enclosing it with double-quotation marks (" ").
Example: when adding the character string "address" to the file name, "address" → address_00000001.bin can be used.

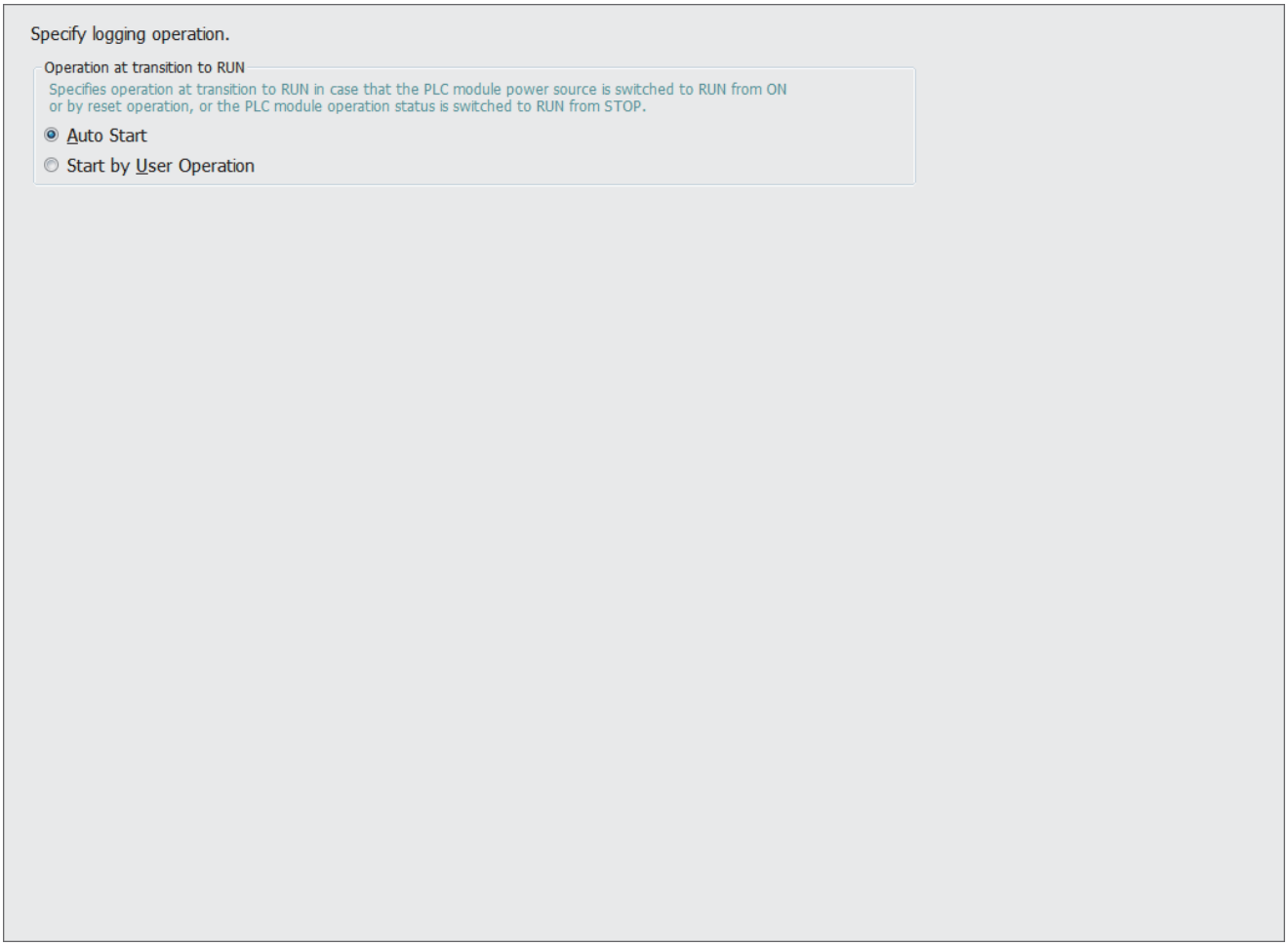
*2 Maximum of 64 characters (including underscore (_), serial number (eight digits), period, and extension) can be used.
However, when a character string that contains double quotation marks (" ") is specified, the maximum number reduces by the number of the double quotation marks.
In the Process CPU (redundant mode), since three characters of an identifier are added to a file name at system switching, the save file name should be within 61 one-byte characters (including underscores (_), serial number (eight digits), period, and extension). (☞ Page 450 Switching to a storage file)

*3 Reducing the setting value results in frequent file switching, so that it is possible that the scan time and/or the device/label access service processing time can be extended.

■Logging operation

The following window specifies the data logging operation when the mode transfers to RUN mode (☞ Page 205 Setting the behavior at the time of a transition to RUN)

Window



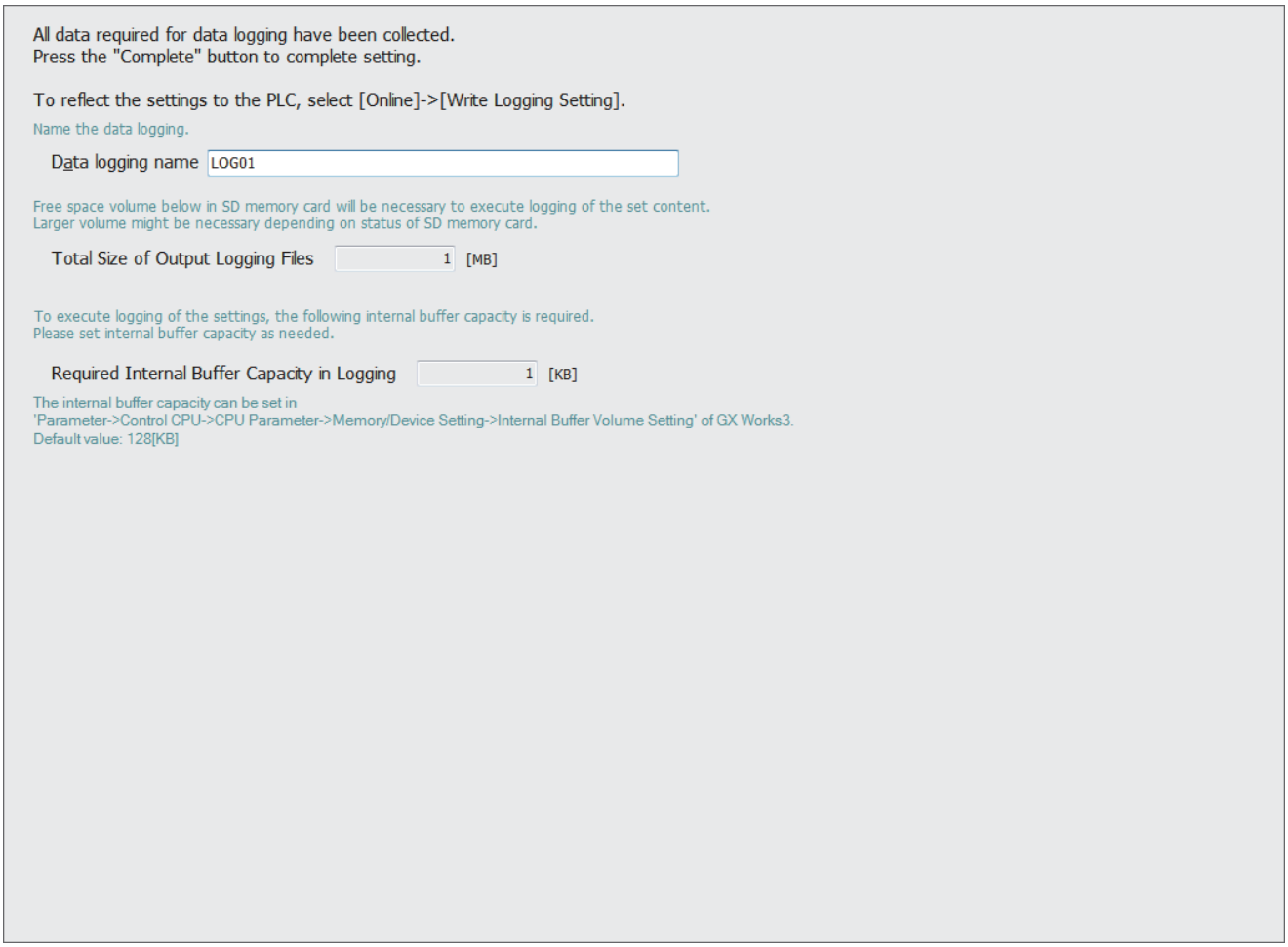
Displayed items

Item	Description	Setting range	Default
Operation at transition to RUN	Select the operation when the mode transfers to RUN mode.	• Auto Start • Start by User Operation	Auto Start

■Finish

The following window is to give the data logging setting a name.

Window



Displayed items

Item	Description	Setting range	Default
Data logging name*1	Give the data logging setting being configured a name.	32 characters or less	LOG01
Total Size of Output Logging Files	View the total capacity of the data logging file which is output based on the specified settings. The total capacity can be increased/decreased by adding/removing the items to be output to the file.	—	1
Capacity of internal buffer necessary for logging	View the internal buffer capacity required to execute the data logging based on the specified settings. This value can be specified with the internal buffer capacity setting of engineering tool (☞ Page 203 Internal buffer capacity setting)	—	1

*1 When the following user action is detected, character entry will be disabled

· Entered a character which cannot be handled with the OS language character code.

· Entered a character whose language code is different from the one for characters already input in the same data logging setting.

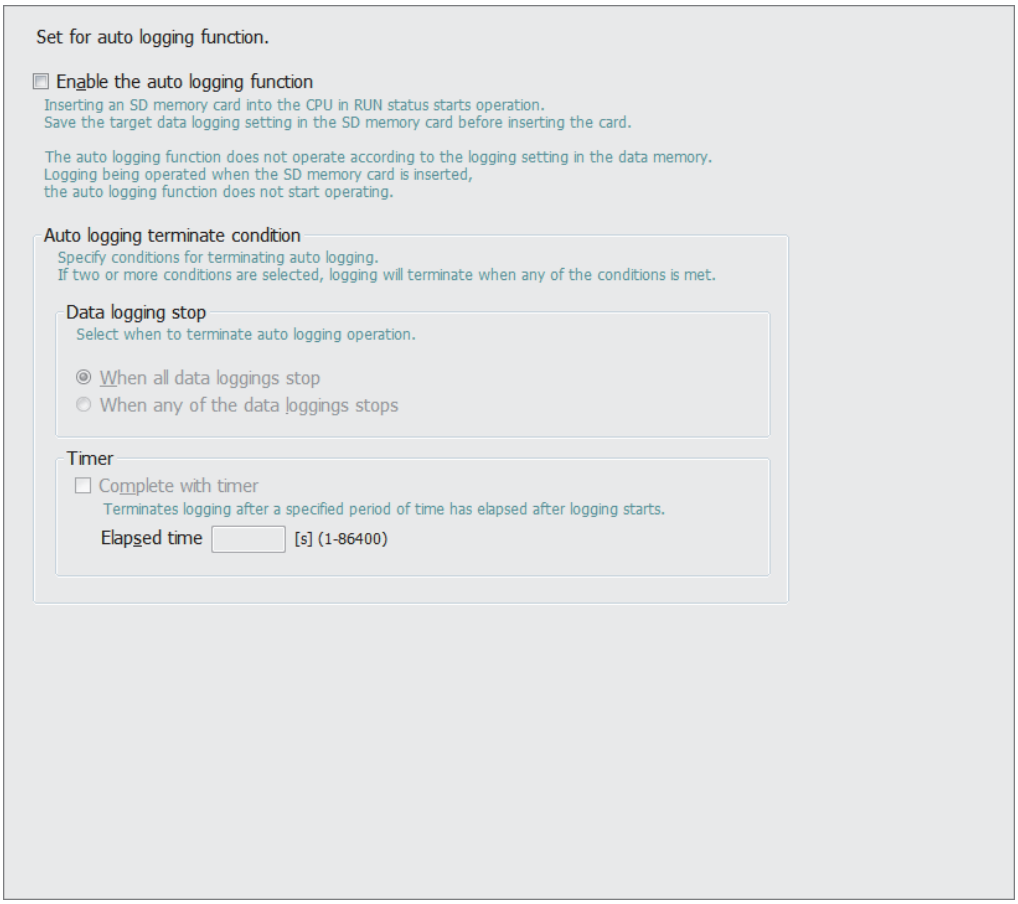
Common setting

The common settings include the auto logging common setting.

■Auto logging common setting

The following window configures the required settings for using the auto logging function. (☞ Page 206 Auto logging)

Window



Displayed items

Item			Description	Setting range	Default
Enable the auto logging function			Select this item to use auto logging.	—	Not checked
Auto logging terminate condition	Data logging stop		Select the condition to complete the auto logging operation. (☞ Page 206 Conditions for auto logging completion)	• When all data loggings stop • When any of the data loggings stops	When all data loggings stop
	Timer	Complete with timer	Select this checkbox to stop the auto logging operation by a timer.	—	—
		Elapsed time	Specify the time interval from the start of the data logging until stopping it.	1 to 86400 seconds	

Online operation

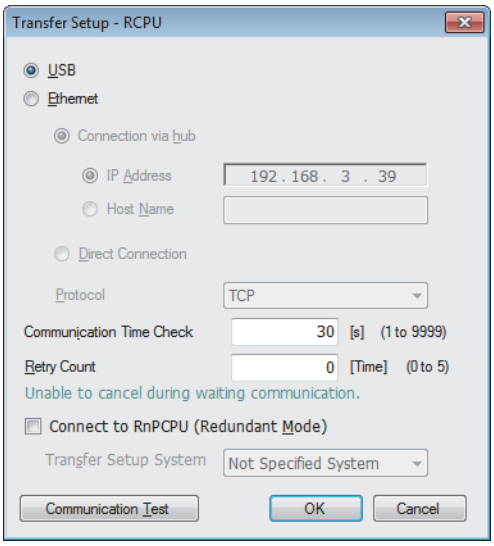
The online operation enables users to read/write/remove the data logging settings, view the data logging status, and operate the data logging file.

■Transfer setup

The following window specifies the communication route between the CPU module and a personal computer.

[Online] ⇨ [Transfer Setup]

Window



Displayed items

Item			Description
USB			Select this item when connecting with a USB cable.
Ethernet	Connection via HUB	IP Address	Configure the IP address and host name used for connection via a hub with an Ethernet cable.
		Host Name	
	Direct Connection		Select this item for direct connection with an Ethernet cable.
	Protocol		Select the protocol used when connecting via a hub.
Communication Time Check			Specify the communication time.
Retry Count			Specify the number of retries.
Connect to RnPCPU (Redundant Mode)			Select this item when the Process CPU (redundant mode) is connected, and select a connection target system.*1 Note that the connection target system will not change even though a system switching is performed. To change the connection target system, perform the following operations. • Opening the windows for reading/deleting logging settings, displaying the data logging status and operating logging, and operating the data logging file again • Changing the selection of the "Online operation" check button on the window for reading/deleting logging settings • Writing logging settings again (when systems are switched after the write button is pressed) • Restarting monitoring the window for displaying the data logging status and operating logging (when monitoring has stopped due to system switching)
[Communication Test] button			This button checks the communication status.

*1 If "Not Specified System" is selected, the personal computer is connected with the host station.

A

■Reading the logging setting

The following procedure reads the data logging setting from the target memory.

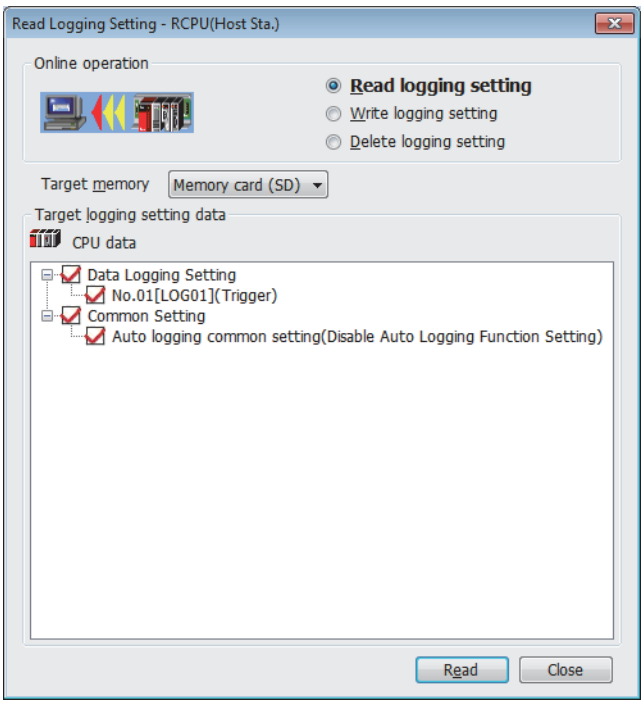
Operating procedure

1. Open the "Read Logging Setting" window.

[Online] ⇨ [Read Logging Setting]

2. Select the memory where the data to be read is stored from the "Target Memory" list.
3. Select the checkbox corresponding to the data item to be read in the "Target logging setting data" list, and click the [Read] button.

Window



Displayed items

Item	Description
Target Memory	Select the memory where the data to be read is stored.
Target logging setting data	Select the data item to be read.

Point

Any existing data (data logging setting with the same setting number or common setting) on the target is overwritten.

■Writing the logging setting

The following procedure is to write the data logging setting to the target memory.

Operating procedure

1. Open the "Write Logging Setting" window.

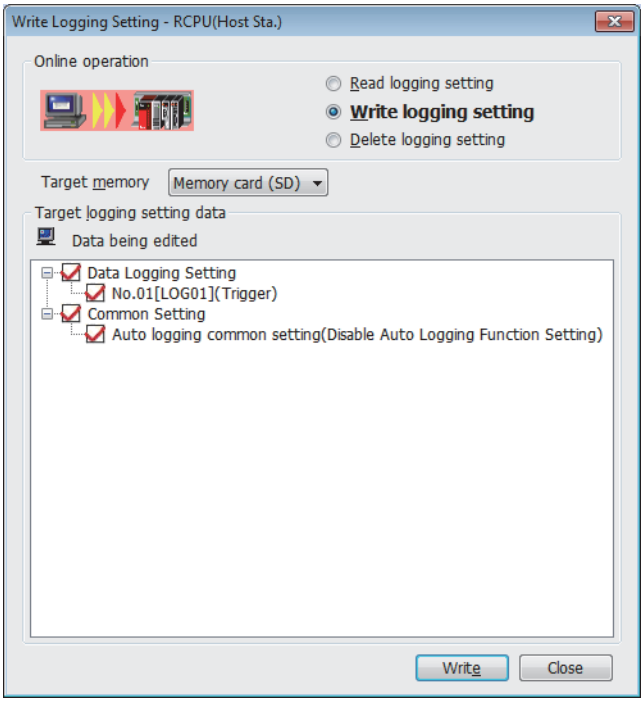
[Online] ⇨ [Write Logging Setting]

2. Select the memory where the data to be written is stored from "Target Memory" list.

3. Select the checkbox corresponding to the data item to be written in the "Target logging setting data" list, and click the [Write] button.*1

*1 In backup mode of the Process CPU (redundant mode), data can be written to both systems or one of the systems. Click the [Write] button and a window appears. Select whether or not to write data to both systems.

Window



Displayed items

Item	Description
Target Memory	Select the memory where the data to be written is stored.
Target logging setting data	Select the data to be written.

Point

Any existing data (data logging setting with the same setting number or common setting) on the target is overwritten.

■Deleting the logging setting

The following procedure removes the data logging setting on the target memory.

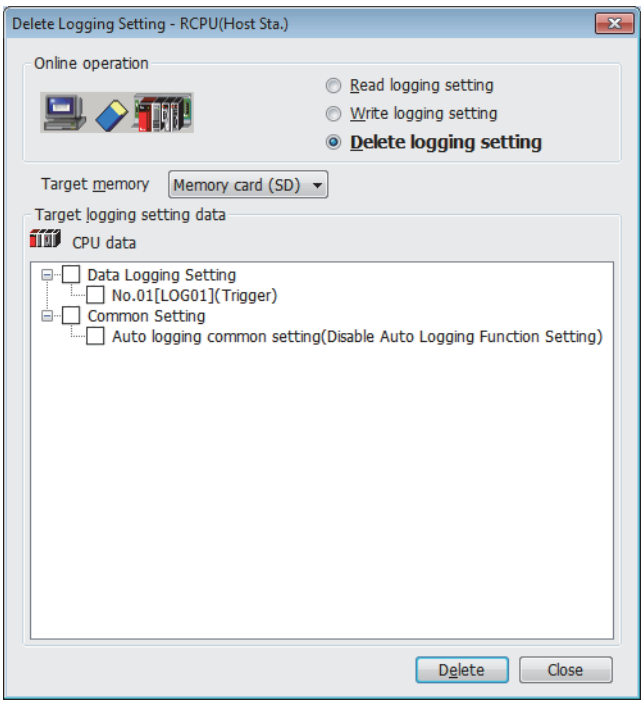
Operating procedure

1. Open the "Delete Logging Setting" window.

[Online] ⇨ [Delete Logging Setting]

2. Select the memory where the data to be removed is stored from the "Target Memory" list.
3. Select the checkbox corresponding to the data item to be removed in the "Target logging setting data" list, and click the [Delete] button.

Window



Displayed items

Item	Description
Target Memory	Select the memory where the data to be removed is stored.
Target logging setting data	Select the data to be removed.

■Logging status and operation

The following procedure is to execute or stop the data logging. Also the data logging status can be checked through this procedure.

Operating procedure

1. Open the "Logging Status and Operation" window.

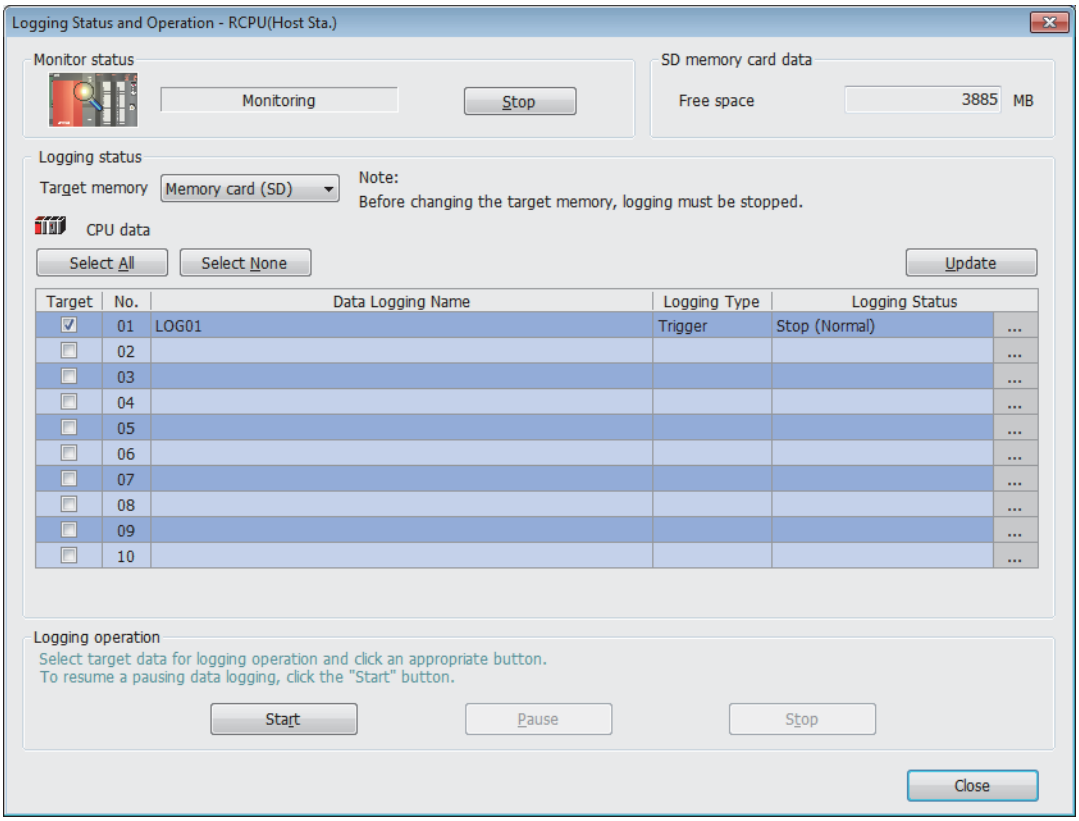
[Online] ⇨ [Logging Status and Operation]

2. Specify the target memory (either data memory or SD memory card) where the effective setting data is stored.
3. Select the checkbox corresponding to the setting number to be executed (One or more numbers can be selected).
4. Start the data logging by clicking the [Start] button (when multiple items are selected, they are executed simultaneously).
5. To suspend the data logging, click the [Pause] button, and to stop the data logging, click the [Stop] button (when multiple items are selected, they are executed simultaneously).

Point

- The data logging cannot be started even when writing the setting and powering off and on or resetting the CPU module. Be sure to click the [Start] button to start the data logging (except for the auto logging).
- With regards to the trigger logging, the data logging setting registration attempt fails if the trigger condition is satisfied.
- It takes a certain time to stop or suspend the data logging after either of these commands is issued by CPU Module Logging Configuration Tool (because the data logging is not stopped or suspended unless the data stored in the internal buffer data has been transferred into the SD memory card in response to these commands).
- There may be a case where a time-out error occurs and the data logging is suspended after CPU Module Logging Configuration Tool starts the logging.

Window



Displayed items

Item		Description
Monitor Status	[Start (Stop)] button	Start or stop monitoring.
SD memory card data	Free space	View the amount of free space of the SD memory card.
Logging status	Target Memory	Select the memory used for this operation.*1
	[Select All] button	Select all the checkboxes in the setting data list.
	[Deselect All] button	Clear all the checkboxes in the setting data list.
	[Update] button	Update monitoring status.
	Target	Select the target setting data for this operation (one or more numbers can be selected).
	[...] button	Clicking this button when an error occurs displays the error details window.
Logging operation	[Start] button	Execute the logging of the selected setting data.
	[Pause] button	Suspend the logging of the selected setting data.
	[Stop] button	Stop the logging of the selected setting data.

*1 This menu item can be selected only when all the data logging statuses are "Stop."

■Logging file operation

The following procedure is to save or remove data logging files on an SD memory card from/to the personal computer.

Operating procedure

1. Open the "Logging File Operation" window.

[Online] ⇨ [Logging File Operation]

2. Specify the directory and select the targeted file.

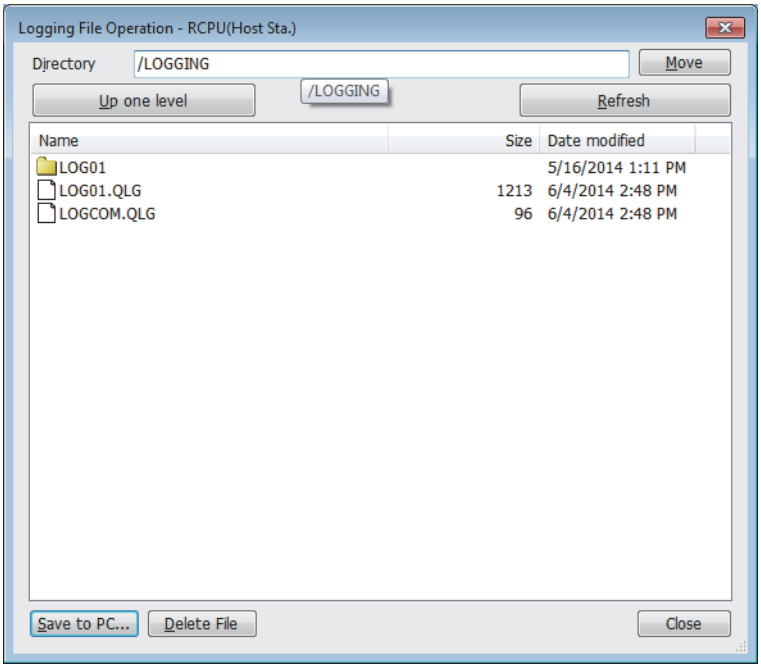
3. To save click the [Save to PC] button, and to remove click the [Delete] button.

Point

Attempting the following operations may result in delay of other monitor update because a certain time period is required for saving data logging files.

- When saving data logging files during the data logging execution.
- When saving a large data logging file.

Window



Displayed items

Item	Description
Directory	View the path to the displayed folder. To change the folder, specify the target folder path.
[Move] button	Move to the specified folder.
[Up one level] button	Moves up to a higher level in the folder hierarchy.
[Refresh] button	Update the displayed content.
[Save to PC] button	Display the "Save As" window and save the selected file to the personal computer.
[Delete] button	Remove the selected file or folder.

Help operation

The following procedures allow to view or use the help function of CPU Module Logging Configuration Tool.

■Opening user's manual

e-Manual Viewer opens and its manual is displayed.

Operating procedure

[Help] ⇨ [Open Manual]

■Connection to MITSUBISHI ELECTRIC FA Global Website

Access Mitsubishi Electric Corporation FA site home page.

Operating procedure

[Help] ⇨ [Connection to MITSUBISHI ELECTRIC FA Global Website]

■Checking version information

Check the version of CPU Module Logging Configuration Tool.

Operating procedure

[Help] ⇨ [Version Information]

A

Supported characters

This section describes the supported characters.

■Supported characters for CPU Module Logging Configuration Tool

Any characters that can be expressed by Unicode are supported. However, the supported characters vary for each position as shown in the following table. Note that if attempting to input an unsupported character, the entry is rejected or a message window appears in response to the improper entry.

Place where character is used		Support status of the target character																	
		(SP)*1	"	'	*	+	,	/	:	;	<	>	?	[	\	]	|	.	Two-byte characters
Data logging setting	• Data logging name • Trigger occurrence information column • Comment	○	×	○	○	○	×	○	○	×	○	○	○	○	○	○	○	○	○
	Date/time column of data row output format	○	×	○	○	○	×	○	○	×	○	○	○	○	○	○	○	○	×
Logging file operation	Directory	×	×	○	×	○	×	○	×	×	×	×	×	○	×	○	×	×	×

*1 (SP) means a space.

■Supported characters for file and/or folder (directory) name

Characters in the shaded area can be used.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NULL		(SP)	0	@	P	`	p				-	タ	ミ		
1			!	1	A	Q	a	q			。	ア	チ	ム		
2			"	2	B	R	b	r			「	イ	ツ	メ		
3			#	3	C	S	c	s			」	ウ	テ	モ		
4			$	4	D	T	d	t				エ	ト	ヤ		
5			%	5	E	U	e	u			、	オ	ナ	ユ		
6			&	6	F	V	f	v			・	カ	ニ	ヨ		
7			'	7	G	W	g	w			ヲ	キ	ヌ	ラ		
8			(	8	H	X	h	x			ァ	ク	ネ	リ		
9			)	9	I	Y	i	y			ィ	ケ	ノ	ル		
A			*	:	J	Z	j	z			ゥ	コ	ハ	レ		
B			+	;	K	[	k	{			ェ	サ	ヒ	ロ		
C			,	<	L	¥	l	|			ャ	シ	フ	ワ		
D			-	=	M	]	m	}			ュ	ス	ヘ	ン		
E			.	>	N	^	n	~			ョ	セ	ホ	゛		
F			/	?	O	_	o				ッ	ソ	マ	°		

Installing a USB driver

To communicate with the CPU module via USB, a USB driver needs to be installed. This section describes the installation procedure of a USB driver. If multiple MELSOFT products are installed, refer to their installed location.

■Windows® XP

Operating procedure

1. Connect a personal computer to the CPU module with a USB cable, and power on the programmable controller.
2. Select "Install from a list or specific location (Advanced)" on the "Found New Hardware Wizard" window.
3. On the next window, select "Search for the best driver in these locations". Check the "Include this location in the search" checkbox, and specify the "Easysocket\USBDrivers" folder where the CPU Module Logging Configuration Tool has been installed.

Precautions

If the driver cannot be installed, check the following setting on Windows®.
Select [Control Panel] ⇨ [System] ⇨ [Hardware], and click the [Driver Signing] button. If "Block - Never install unsigned driver software" is selected, the USB driver may not be installed. Select "Ignore - Install the software anyway and don't ask for my approval" or "Warn - Prompt me each time to choose an action", and then install the USB driver.

■Windows Vista®

Operating procedure

1. Connect a personal computer to the CPU module with a USB cable, and power on the programmable controller.
2. Select "Locate and install driver software (recommended)" on the "Found New Hardware" window.
3. On the next window, select "Browse my computer for driver software (advanced)".
4. On the next window, select "Search for the best driver in these locations". Check the "Include subfolders" checkbox, and specify the "Easysocket\USBDrivers" folder where the CPU Module Logging Configuration Tool has been installed.

Precautions

If "Windows can't verify the publisher of this driver software" appears on the "Windows Security" window, select "Install this driver software anyway".

■Windows® 7 and later

Operating procedure

1. Connect a personal computer to the CPU module with a USB cable, and power on the programmable controller.
2. Select [Start] ⇨ [Control Panel] ⇨ [System and Security] ⇨ [Administrative Tools] ⇨ [Computer Management] ⇨ [Device Manager]. Right-click "Unknown device", and click "Update Driver Software".
3. On the "Update Driver Software" window, select "Browse my computer for driver software" and specify the "Easysocket\USBDrivers" folder where the CPU Module Logging Configuration Tool has been installed on the next window.

Appendix 13 List of Available SQL Commands for CPU Module Database Access Function

This section describes the available SQL commands for the CPU module database access function.

Data Definition

CREATE TABLE

This command creates a table in the database.

Syntax
CREATE TABLE [table name] ([field name] [data type] [option], [field name] [data type] [option], ···, [field name] [data type] [option]);

■Option

Item	Description	Syntax
Primary key constraint	Maintains the uniqueness of a value stored in the field. (NULL is not available.) Only one primary key constraint can be set per table. Key names must be up to 16 single-byte alphabetical characters and are case sensitive.	PRIMARY KEY [key name]
Foreign key constraint	Refers to the value of a field of another table. Only the field with the name where the primary key constraint of another table is set can be set as the reference field. (Always set the same field name to the reference field and own field.) Key names must be up to 16 single-byte alphabetical characters and are case sensitive.	FOREIGN KEY [key name] REFERENCES [reference table name] ([reference field name])
NOT NULL constraint	Disables storing NULL values. (Always store a value other than NULL.)	NOT NULL

■Application example

- To create table1 (fld1 (integer type, primary key (key name: pk1)), fld2 (integer type), fld3 (character string type (120 characters, NOT NULL)))

```
CREATE TABLE "table1" ("fld1" INT PRIMARY KEY "pk1", "fld2" INT, "fld3" NLSCHAR (120) NOT NULL);
```

- To create table2 (fld1 (integer type, foreign key (key name: fk1, reference: table1 fld1)), fld2 (BOOLEAN type), fld3 (single-precision real number type))

```
CREATE TABLE "table2" ("fld1" INT FOREIGN KEY "fk1" REFERENCES "table1" ("fld1"), "fld2" BOOLEAN, "fld3" REAL);
```

■Precautions

- To define the NLSCHAR field, check the data size of the field.
- Define the NLSCHAR type with the size specified in NLSCHAR = (n+1) × 3[*1] since it needs to be defined with the data size. For example, the definition is 39 characters for "fld3" NLSCHAR (120) of table1 in the application example.

*1 n indicates the number of characters.

DROP TABLE

This command deletes the specified table in the database.

Syntax
DROP TABLE [table name];

■Application example

- To delete table 1

```
DROP TABLE "table1";
```

CREATE INDEX

This command creates an index in the specified field of the specified table in the database. Index names must be up to 16 single-byte alphabetical characters and are case sensitive.

Syntax
CREATE INDEX [index name] ON [table name] (field name);

■Application example

• To set the index "idx1" to fld2 of table1

```
CREATE INDEX "idx1" ON "table1" ("fld2");
```

DROP INDEX

This command deletes the specified index in the database.

Syntax
DROP INDEX [index name] ON [table name];

■Application example

• To delete the "idx1" index in table1

```
DROP INDEX "idx1" ON "table1";
```

EMPTY TABLE

This command deletes all records of the specified table in the database.

Syntax
EMPTY [table name];

■Application example

• To delete all records in table1

```
EMPTY "table1";
```

A

Data Operation

INSERT

This command adds a record to the specified table in the database.

Syntax[*1*2]
• To add records to all fields INSERT INTO [table name] VALUES (field 1 value, field 2 value, ···, field X value); • To add records to the specified fields INSERT INTO [table name] (field specification 1, field specification 2, ···, field specification X) VALUES (field 1 value, field 2 value, ···, field X value);

*1 Input null not to store the field value.

*2 When the field value is a character string, enclose it in single quotation marks (').

■Application example

• To add records to all fields

INSERT INTO "table1" VALUES (11, null, 'abc');

• To add records to the specified fields

INSERT INTO "table1" ("fld1", "fld3") VALUES (12, 'efg');

SELECT

This command outputs a value of the specified record from the specified table in the database.

Syntax
• To output all values SELECT [output target field or numeric expression] FROM [table name] [option]; • To output a value excluding overlapped values SELECT DISTINCT [output target field or numeric expression] FROM [table name] [option 1] [option 2];

■Option 1

Item	Description	Syntax
INNER JOIN	Performs inner join. (Records that exist in the specified both tables are output.)	SELECT [output target field or numeric expression] FROM [table name 1] INNER JOIN [table name 2] ON [combining condition];
LEFT JOIN	Performs outer join. (All records that exist in [table name 1] are output.)	SELECT [output target field or numeric expression] FROM [table name 1] LEFT JOIN [table name 2] ON [combining condition];
RIGHT JOIN	Performs outer join. (All records that exist in [table name 2] are output.)	SELECT [output target field or numeric expression] FROM [table name 1] RIGHT JOIN [table name 2] ON [combining condition];
FULL JOIN	Performs outer join. (All records that exist in both of specified tables are output.)	SELECT [output target field or numeric expression] FROM [table name 1] FULL JOIN [table name 2] ON [combining condition];

■Option 2[*1]

Item	Description
WHERE	Specifies the condition to obtain a specific record. (☞ Page 772 WHERE).
GROUP BY	Specifies the grouping condition of the record to be obtained. (☞ Page 772 GROUP BY).
HAVING	Specifies the condition to narrow down a search using the result of the aggregate function. (☞ Page 772 HAVING).
ORDER BY	Specifies the sorting condition of the record to be obtained. (☞ Page 773 ORDER BY).

*1 Multiple options can be used.

■Application example

• To output all record values in table1 (all fields of table1)

SELECT * FROM table1;

• To output all record values in table1 (fld1 and fld2 in table1)

SELECT "fld1", "fld2" FROM table1;

• To output all records excluding the overlapped records in fld1 of table1

SELECT DISTINCT "fld1" FROM table1;

UPDATE

This command updates the value of the specified record of the specified table in the database.

Syntax*1
UPDATE [table name] SET [field name = assigned value], [field name = assigned value], ···, [field name = assigned value] [option];

*1 When the field value is a character string, enclose it in single quotation marks (').

■Option

Item	Description	Syntax
WHERE	Specifies the condition to obtain (process) a specific record (☞ Page 772 WHERE).	WHERE

■Application example

• To store a value in "fld1" and "fld3" of the record of "fld2" field value = 935 of table1 (storing fld1 = 20, fld3 = abc)

UPDATE table1 SET "fld1" = 20, "fld3" = 'abc' WHERE "fld2" = 935;

DELETE

This command deletes a record of the specified table in the database.

Syntax*1
DELETE [table name] [option];

*1 When the field value is a character string, enclose it in single quotation marks (').

■Option

Item	Description	Syntax
WHERE	Specifies the condition to obtain (process) a specific record (☞ Page 772 WHERE).	WHERE

■Application example

• To delete all records in table1

DELETE "table1";

• To delete the record of the fld3 value = abc in table1

DELETE "table1" WHERE "fld3" = 'abc';

Clause

WHERE

This command conducts search with conditions.

Syntax
• To specify a condition WHERE [condition] • To specify an exception WHERE NOT [condition]; • To specify two conditions with AND WHERE [condition 1] AND [condition 2]; • To specify two conditions with OR WHERE [condition 1] OR [condition 2];

■Operators that can be used in [condition]

Item	Description	Syntax
=	Equals	[numeric expression 1] = [numeric expression 2]
!=	Not equal to	[numeric expression 1] != [numeric expression 2]
<>	Not equal to	[numeric expression 1] <> [numeric expression 2]
>	Greater than	[numeric expression 1] > [numeric expression 2]
>=	Greater than or equal to	[numeric expression 1] >= [numeric expression 2]
<	Less than	[numeric expression 1] < [numeric expression 2]
<=	Less than or equal to	[numeric expression 1] <= [numeric expression 2]
IN	Brings the conditions to be set together.	IN ([numeric expression 1], [numeric expression 2], ···, [numeric expression 3])
BETWEEN	Specifies the range to be output.	[numeric expression 1] BETWEEN [numeric expression 2] AND [numeric expression 3]

■Application example

• To output a record with the fld1 value of 1 to 100 of table1

```
SELECT * FROM "table1" WHERE "fld1" BETWEEN 1 AND 100;
```

• To output fld1 and fld3 of the record with the fld3 value other than abc of table1

```
SELECT "fld1", "fld3" FROM "table1" WHERE NOT "fld3" = 'abc';
```

GROUP BY

This command groups elements.

Syntax
GROUP BY [numeric expression]

■Application example

• To group Table1 according to the value of fld1 and aggregate the number of records

```
SELECT COUNT(*) FROM "table1" GROUP BY "fld1";
```

HAVING

This command narrows down a search using the result of the aggregate function.

Syntax
HAVING [numeric expression]

■Application example

• To output only the groups whose total value is 50 or less

```
SELECT SUM ("fld2") FROM "table1" GROUP BY "fld1" HAVING SUM ("fld2") < 50;
```

ORDER BY

This command sorts output.

Syntax
• Ascending ORDER BY [numeric expression] ASC • Descending ORDER BY [numeric expression] DESC

■Application example

• To sort and output fld1 and fld2 of all records in ascending order of the fld1 value from table1

```
SELECT "fld1", "fld2" FROM "table1" ORDER BY "fld1" ASC;
```

A

Aggregate Function

AVG

This command obtains a mean value.

Syntax
AVG ([numeric expression])

■Application example

• To obtain the mean value of fld1 from table1 (including overlapped values)

SELECT AVG ("fld1") FROM "table1";

• To obtain the mean value of fld1 from table1 (excluding overlapped values)

SELECT AVG (DISTINCT "fld1") FROM "table1";

COUNT

This command aggregates the number of records.

Syntax
COUNT [target]

■Application example

• To obtain the number of records of fld1 from table1 (including overlapped values)

SELECT COUNT ("fld1") FROM "table1";

• To obtain the number of records of fld1 from table1 (excluding overlapped values)

SELECT COUNT (DISTINCT "fld1") FROM "table1";

MAX

This command obtains a maximum value.

Syntax
MAX ([numeric expression])

■Application example

• To obtain the maximum value of fld1 from table1

SELECT MAX ("fld1") FROM "table1";

MIN

This command obtains a minimum value.

Syntax
MIN ([numeric expression])

■Application example

• To obtain the minimum value of fld1 from table1

SELECT MIN ("fld1") FROM "table1";

SUM

This command obtains the total.

Syntax
SUM ([numeric expression])

■Application example

• To obtain the total of fld1 from table1 (including overlapped values)

SELECT SUM ("fld1") FROM "table1";

• To obtain the total of fld1 from table1 (excluding overlapped values)

SELECT SUM (DISTINCT "fld1") FROM "table1";

Arithmetic Function

ABS

This command obtains an absolute value.

Syntax
ABS ([numeric expression])

■Application example

• To obtain the absolute value of fld1 from table1

```
SELECT ABS ("fld1") FROM "table1";
```

ACOS

This command obtains an arc cosine.

Syntax
ACOS ([numeric expression])

■Application example

• To obtain the arc cosine of fld1 from table1

```
SELECT ACOS ("fld1") FROM "table1";
```

ASIN

This command obtains an arc sine.

Syntax
ASIN ([numeric expression])

■Application example

• To obtain the arc sine of fld1 from table1

```
SELECT ASIN ("fld1") FROM "table1";
```

ATAN

This command obtains an arc tangent.

Syntax
ATAN ([numeric expression])

■Application example

• To obtain the arc tangent of fld1 from table1

```
SELECT ATAN ("fld1") FROM "table1";
```

ATAN2

This command obtains an arc tangent.

Syntax
ATAN2 ([numeric expression 1], [numeric expression 2])*1

*1 This syntax is ATAN2 (y, x).

■Application example

• To obtain an arc tangent of fld1/fld2 from table1

```
SELECT ATAN2 ("fld1", "fld2") FROM "table1";
```

CEILING

This command obtains the smallest integer greater than or equal to the specified numeric expression.

Syntax
CEILING ([numeric expression])

■Application example

• To obtain the smallest integer greater than or equal to the fld1 value from table1

SELECT CEILING ("fld1") FROM "table1";

COS

This command obtains a cosine.

Syntax
COS ([numeric expression])

■Application example

• To obtain the cosine of fld1 from table1

SELECT COS ("fld1") FROM "table1";

COT

This command obtains a cotangent.

Syntax
COT ([numeric expression])

■Application example

• To obtain the cotangent of fld1 from table1

SELECT COT ("fld1") FROM "table1";

DEGREES

This command obtains degrees from radians.

Syntax
DEGREES ([numeric expression])

■Application example

• To obtain degrees of fld1 from table1

SELECT DEGREES ("fld1") FROM "table1";

EXP

This command obtains an exponent value. (It calculates e raised to the power of [numeric expression].)

Syntax
EXP ([numeric expression])

■Application example

• To obtain e raised to the power of fld1 from table1

SELECT EXP ("fld1") FROM "table1";

FLOOR

This command obtains the largest integer less than or equal to the specified numeric expression.

Syntax
FLOOR ([numeric expression])

■Application example

• To obtain the largest integer less than or equal to the fld1 value from table1

```
SELECT FLOOR ("fld1") FROM "table1";
```

LOG

This command obtains a natural logarithm.

Syntax
• To obtain a natural logarithm LOG ([numeric expression]) • To obtain a decadic logarithm LOG10 ([numeric expression])

■Application example

• To obtain the natural logarithm of fld1 from table1

```
SELECT LOG ("fld1") FROM "table1";
```

• To obtain the decadic logarithm of fld1 from table1

```
SELECT LOG10 ("fld1") FROM "table1";
```

POW

This command raises a value. (It raises [numeric expression 1] to the power of [numeric expression 2].)

Syntax
POW ([numeric expression 1], [numeric expression 2]) When [numeric expression 1] = 0, [numeric expression 2] must be greater than 0. When [numeric expression 1] is less than 0, [numeric expression 2] must be an integer.

■Application example

• To obtain fld1 raised to the power of fld2 from table1

```
SELECT POW ("fld1", "fld2") FROM "table1";
```

RADIANS

This command obtains radians from degrees.

Syntax
RADIANS ([numeric expression])

■Application example

• To obtain radians of fld1 from table1

```
SELECT RADIANS ("fld1") FROM "table1";
```

ROUND

This command rounds a value. (The value of [numeric expression 1] is rounded off to the nth decimal place.)*1

Syntax
ROUND ([numeric expression 1], [numeric expression 2])

*1 Add one to the value specified in n = [numeric expression 2].

■Application example

• To obtain the value of fld1 rounded off to the third decimal place from table1.

```
SELECT ROUND ("fld1", 2) FROM "table1";
```

SIGN

This command obtains a sign.

Syntax
SIGN ([numeric expression])

■Application example

• To obtain the sign of fld1 from table1

SELECT SIGN ("fld1") FROM "table1";

SIN

This command obtains a sine.

Syntax
SIN ([numeric expression])

■Application example

• To obtain the sine of fld1 from table1

SELECT SIN ("fld1") FROM "table1";

SQRT

This command obtains a square root.

Syntax
SQRT ([numeric expression])

■Application example

• To obtain the square root of fld1 from table1

SELECT SQRT ("fld1") FROM "table1";

TAN

This command obtains a tangent.

Syntax
TAN ([numeric expression])

■Application example

• To obtain the tangent of fld1 from table1

SELECT TAN ("fld1") FROM "table1";

Appendix 14 Added and Enhanced Functions

This section describes added and enhanced functions of the CPU module and the engineering tool, as well as the corresponding firmware versions of the CPU module and software versions of the engineering tool.

—: No restrictions on the firmware version of the CPU module and the software version of the engineering tool

Programmable controller CPU

Added or enhanced function	CPU module firmware version	Engineering tool software version	Reference
LED specifications (enhanced use of the FUNCTION LED)	"06" or later	"1.005F" or later	Page 232 LED display setting
Memory dump function			Page 216 MEMORY DUMP FUNCTION
Real-time monitor function		"1.40S" or later[*1]	• Page 227 REAL-TIME MONITOR FUNCTION • GX LogViewer Version1 Operating Manual
CC-Link IE Controller Network function of the RJ71EN71		"1.005F" or later	• MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup) • MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)
Mounting of MELSEC iQ-R series modules occupying two slots	"08" or later	—	For the applicable modules, refer to the following. MELSEC iQ-R Module Configuration Manual
MELSECNET/H network module	"10" or later	"1.010L" or later	For the applicable modules, refer to the following. MELSEC iQ-R Module Configuration Manual
SFC support[*2]	"12" or later	"1.015R" or later	• MELSEC iQ-R CPU Module User's Manual (Startup) • MELSEC iQ-R Programming Manual (Program Design)
Extended SRAM cassette (NZ2MC-16MBS)			• Page 153 Device/Label Memory Area Setting • MELSEC iQ-R CPU Module User's Manual (Startup)
iQ Sensor Solution functions supporting Ethernet • Automatic detection of connected device • Communication setting reflection • Sensor parameter read/write			iQ Sensor Solution Reference Manual
iQ Sensor Solution functions supporting CC-Link IE Field Network • Automatic detection of connected device • Communication setting reflection • Sensor parameter read/write		—	iQ Sensor Solution Reference Manual
Daylight saving time function	"17" or later	"1.020W" or later	Page 94 Daylight Saving Time Function
Sequence scan synchronization sampling function			Page 304 SEQUENCE SCAN SYNCHRONIZATION SAMPLING FUNCTION
Duplication check for POU	"19" or later	"1.022Y" or later	GX Works3 Operating Manual
Background processing of program transfer (writing the program restoration information) of online change	"20" or later	"1.022Y" or later	GX Works3 Operating Manual

A

Added or enhanced function	CPU module firmware version	Engineering tool software version	Reference
CPU module data backup/ restoration function	"22" or later	—	Page 242 CPU MODULE DATA BACKUP/RESTORATION FUNCTION
iQ Sensor Solution function supporting CC-Link IE Field Network and AnyWireASLINK • Data backup/restoration*3			iQ Sensor Solution Reference Manual
Redundant power supply system			MELSEC iQ-R Module Configuration Manual
Firmware update function (RnCPU)	"23" or later	—	MELSEC iQ-R Module Configuration Manual
External input/output forced on/off function	"25" or later	"1.030G" or later	Page 265 EXTERNAL INPUT/OUTPUT FORCED ON/OFF FUNCTION
iQ Sensor Solution functions supporting AnyWireASLINK • Automatic detection of connected device • Communication setting reflection • Sensor parameter read/write			iQ Sensor Solution Reference Manual
iQ Sensor Solution functions supporting Ethernet • Data backup/restoration*3			
CC-Link IE Field Network Basic			CC-Link IE Field Network Basic Reference Manual
Multiple CPU system (RnENCPU)		"1.032J" or later	MELSEC iQ-R Module Configuration Manual
CPU module database access function	"28" or later	"1.035M" or later	Page 117 CPU module database access (from external device) function
Online change (inactive SFC block)			MELSEC iQ-R Programming Manual (Program Design)
"Use MC/MCR in EN Control" of subroutine-type FB			• GX Works3 Operating Manual • MELSEC iQ-R Programming Manual (Program Design)

*1 Indicates the compatible software version of GX LogViewer.

*2 Step relays (BLn\Sn) can be specified with the CPU Module Logging Configuration Tool or GX LogViewer. Note that the compatible software version for CPU Module Logging Configuration Tool and GX LogViewer is "1.49B" or later.

*3 Data is backed up or restored by using programs (special relay and special register).

Process CPU

Added or enhanced function	CPU module firmware version	Engineering tool software version	Reference
SFC support*1*2	"03" or later	"1.020W" or later	• MELSEC iQ-R CPU Module User's Manual (Startup) • MELSEC iQ-R Programming Manual (Program Design)
Redundant function	"04" or later	"1.025B" or later	Page 387 FUNCTIONS
Redundant power supply system			MELSEC iQ-R Module Configuration Manual
Label initialization function			Page 306 LABEL INITIALIZATION FUNCTION
Duplication check for POU			GX Works3 Operating Manual
Background processing of program transfer (writing the program restoration information) of online change*2	"09" or later	"1.022Y" or later	GX Works3 Operating Manual

*1 A step relay (BLn\Sn) can be specified with the CPU Module Logging Configuration Tool. Note that the compatible software version for CPU Module Logging Configuration Tool is "1.54G" or later.

*2 This function is not supported in redundant mode.

Safety CPU and safety function module

Added or enhanced function	Module firmware version	Engineering tool software version	Reference
Extension of points for safety input device and safety output device	"04" or later	"1.019V" or later	• Page 499 Safety device area setting range • Page 504 List of safety devices • MELSEC iQ-R CPU Module User's Manual (Startup)
Auto logout time change of the user authentication function (5 minutes → 60 minutes)	"04" or later	—	—
Duplication check for POU	"04" or later	"1.022Y" or later	GX Works3 Operating Manual
Background processing of program transfer (writing the program restoration information) of online change			

A

MEMO

INDEX

Symbols

"$MELPRJ$" folder 78

A

Accuracy of Constant Scan 116
Accuracy of timers 322
Annunciator (F) 315
Auto logging 206

B

Backup mode 27
Behavior at the time of a transition to RUN 205
Block number 337
Block switching method 337
Boot operation 176
Buffer memory 27

C

CC-Link IE Field Network-equipped master/local module 27
Character string constant 349
Clearing file registers 339
Clock function 91
Constant scan 115
Constants 347
Continuous logging 194
Continuous RUN allowed time in TEST MODE . . 492
Continuous RUN prevention in TEST MODE . . . 492
Continuous RUN time in TEST MODE 492
Control system 27
Control system execution program 27
Controlled module 273
Counter (C) 325
CPU buffer memory 78
CPU buffer memory access device 334
CPU Module Logging Configuration Tool 726
CPU parameters 722

D

Data inconsistency 69
Data logging 187
Data logging file 203
Data logging states 198,448
Data memory 78
Data register (D) 327
Database 117
Database function 117
Decimal 347
Decimal constant (K) 347
Device 27
Device initial value 158
Device supporting iQSS 27
Device/label access service processing 180
Device/label memory 77
Double-precision real number 348

E

Edge relay (V) 318
END processing 33
Engineering tool 27
Error clear 233,502
Error codes 510
Ethernet interface module with built-in CC-Link IE 27
Ethernet-equipped module 27
Event execution type program 47
Event history 236
Event history file 237

F

File password 302
File register (R/ZR) 337
File register setting 338
File registers common to all programs 338
Files 81
Fixed scan execution mode 45
Fixed scan execution type program 42
Fixed scan interval 43
Function device 329
Function input (FX) 329
Function output (FY) 329
Function register (FD) 329

G

Global device 350
Global label 27
Global pointer 341
GX LogViewer 726

H

Hexadecimal 347
Hexadecimal constant (H) 347
High-speed timer 321
High-speed timer (T/ST) 321

I

I/O No. specification device (U) 346
I/O refresh 32
Index modification 335,355
Index register (Z) 335
Index register setting 336
Indirect specification 358
Initial execution type program 41
Initial processing (when powered on) 32
Initial processing (when switched to RUN) 32
Initial scan time 33
Initial scan time monitoring time 33
Initialization of labels after rebuilt all (reassignment) 306
Input (X) 314
Internal buffer 203
Internal buffer capacity setting 156
Internal relay (M) 315

Interrupt factor 344
Interrupt pointer (I) 344
Interrupt pointer number 344
Interrupt priority 106
Interrupt program 55
IP filter 302

J

Jump instructions 341

L

Label 27
Label initial value 158
Label Initial value reflection setting 308
Latch 168
Latch (1) 168
Latch (2) 168
Latch clear 173
Latch relay (L) 315
Librarization of programs 51
Link direct device 331
Link refresh 332
Link register (W) 327
Link relay (B) 315
Link special register (SW) 328
Link special relay (SB) 318
Local pointer 342
LOGTRG instruction 197
LOGTRGR instruction 212
Long counter (LC) 325
Long index register (LZ) 335
Long retentive timer (LST) 320
Long timer (LT) 319
Low-speed timer 321
Low-speed timer (T/ST) 321

M

Master control instructions 340
Maximum counting speed for counters 327
Memory card parameters 723
Memory configuration 75
Memory copy 420
Memory dump file 221
Memory dump function 216
Memory dump status 222
Missing data 204
Module access device 333
Module communication test 441
Module label 27
Module label assignment area 339
Momentary power failure 73
Multiple CPU synchronous interrupt 300
Multiple CPU system 272
Multiple interrupt 106
Multiple interrupt function 106

N

Nesting (N) 340
Nesting structure 340
Network No. specification device (J) 346
New control system 27
New standby system 27
Non-controlled module 273

O

Online change 98
Online ladder block change 98
Operating environment 726
Operation mode 362,389
Other devices 346
Output (Y) 314
Output mode at operating status change (STOP to RUN) 162
Overflow 348
Own system 27

P

Parameter No. 721
PAUSE state 71
PID control 111
Pointer (P) 341
Pointer setting 343
Priority for interrupt factors 345
Process control function 113
Process control instruction 114
Process CPU 27
Process CPU (process mode) 27
Process CPU (redundant mode) 27
Processing time 694
Program arrangement change 51
Program cache memory 76
Program executed in both systems 28
Program execution type 41
Program list monitor 33
Program memory 76
Programmable controller CPU 28

R

Real constant (E) 348
Real number 348
Real-time monitor function 227
Redundant function module 28
Redundant system 28
Refresh 289
Refresh data register (RD) 339
Refresh memory 78
Refresh memory setting 339
Remote operation 146
Remote password 302
Remote RESET 148
Remote RUN 146
Remote STOP 146
Reserved area for online change 100
Resetting counters 326
Retentive timer (ST) 320
RnCPU 28
RnENCPU 28
RnENCPU (CPU part) 28
RnENCPU (network part) 28
RUN state 71
RUN-PAUSE contacts 148

S

Safety communication function 494

Safety communications 28
Safety control 28
Safety counter 504,507
Safety CPU 28
Safety cycle processing 28
Safety cycle time 28,477
Safety data identify check 494
Safety data register 504,507
Safety devices 28,504
Safety diagnostic function 493
Safety function module 28
Safety global devices 508
Safety input 504,505
Safety internal relay 504,505
Safety link register 504,507
Safety link relay 504,505
Safety local devices 508
SAFETY MODE 488
SAFETY MODE (wait-for-restart) 488
Safety operation mode 488
Safety output 504,505
Safety program 28,475
Safety programmable controller 28
Safety retentive timer 504,506
Safety special register 504
Safety special relay 504
Safety system devices 504,507
Safety timer 504,506
Safety user devices 504,505
Saving/restoring of the file register (R) block number 66
Saving/restoring of the index register (Z, LZ) 67
Scan execution type program 42
Scan monitoring function 89
Scan time 33
Scan time monitoring time 89
Scan time monitoring time setting 89
SD memory card 78
SD memory card forced disable 174
Security function 302
Security key authentication 302
Separate mode 28
Serial number method 337
Setting RUN-PAUSE contacts 148
Setting time zone 93
SFC block device (BL) 346
SFC transition device (TR) 346
Signal flow memory 412,455
Single-precision real number 348
Special register (SD) 330
Special relay (SM) 330
Stack file 200
Standard communications 28
Standard control 28
Standard CPU 29
Standard device 29
Standard program 29
Standard programmable controller 29
Standard/safety shared label 28
Standby system 28
Standby type program 51
Station sub ID number 28
Step relay (S) 319
STOP state 71
Storage file 201
Subroutine call instruction 54
Subroutine program 54
System A 28
System B 28
System clock 97
System consistency check 428
System device 329
System parameters 721
System switching 392

T

TEST MODE 488
The other system 28
Time setting 91
Time zone 93
Timer (T) 319
Timer current value 321
Timer limit setting 321
Timer time limit value 321
Tracking cable 28
Tracking transfer 405
Transaction 124
Trigger condition 196
Trigger instruction 197
Trigger logging 194
Troubleshooting by symptom 602

U

Underflow 348
Unicode text file 121
User authentication function 302
User device 314

W

Watchdog timer 89

I

MEMO

REVISIONS

*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
June 2014	SH(NA)-081264ENG-A	First edition
July 2014	SH(NA)-081264ENG-B	Error correction
October 2014	SH(NA)-081264ENG-C	■Added functions Memory dump function, real-time monitor function, LED specifications (enhanced) ■Added or modified parts SAFETY PRECAUTIONS, RELEVANT MANUALS, Section 1.4, 1.5, 4.4, 6.2, 8.1, 12.1, 12.2, 12.3, 12.4, Chapter 14, Section 17.1, 17.2, 20.2, Chapter 22, Section 22.1, 22.2, 22.4, 22.5, 22.12, 22.13, 23.1, 23.2, 23.3, 23.4, 23.5, 23.6, 23.7, 23.8, Chapter 24, Section 25.1, 25.3, 26.4, 30.2, Appendix 1, 2, 4, 5, 6, 10
November 2014	SH(NA)-081264ENG-D	■Added or modified parts Section 1.4, 23.2, 28.2, Appendix 6
January 2015	SH(NA)-081264ENG-E	■Added models R08PCPU, R16PCPU, R32PCPU, R120PCPU ■Added function Process control function ■Added or modified parts SAFETY PRECAUTIONS, INTRODUCTION, TERMS, Section 1.4, 4.2, 4.4, Chapter 9, 10, Section 14.1, 14.2, 14.4, Chapter 15, 18, Section 23.12, Chapter 24, 25, Section 27.1, 27.2, 29.1, 29.2, 29.11, 30.3, Appendix 1, 2, 4, 5, 6, 7, 10
May 2015	SH(NA)-081264ENG-F	■Added or modified parts TERMS, Section 6.3, 19.2, Appendix 1, 5, 6, 9, 10
August 2015	SH(NA)-081264ENG-G	■Added models R04ENCPU, R08ENCPU, R08SFCPU, R16ENCPU, R16SFCPU, R32ENCPU, R32SFCPU, R120ENCPU, R120SFCPU, R6SFM ■Added functions User authentication function, safety function ■Added or modified parts SAFETY PRECAUTIONS, CONDITIONS OF USE FOR THE PRODUCT, INTRODUCTION, TERMS, Section 14.4, Chapter 28, Section 29.1, 29.2, PART 4, Chapter32, 33, 34, 35, Appendix 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, WARRANTY
August 2015	SH(NA)-081264ENG-H	■Added or modified parts TERMS, Section 1.5, 1.6, 4.3, 7.1, 14.2, 14.3, 14.4, 21.2, 24.1, 26.2, 27.4, Chapter 28, Section 29.1, 29.2, 31.3, 32.2, 32.4, 33.4, Chapter 34, Section 34.2, 34.6, 34.9, 35.1, Appendix 1, 4, 5, 8, 10, 13, WARRANTY
January 2016	SH(NA)-081264ENG-I	■Added functions Daylight saving time function, sequence scan synchronization sampling function ■Added or modified parts Section 6.3, 14.4, 23.1, 23.2, Chapter 29, Section 30.1, 32.3, 35.6, 36.1, Appendix 1, 2, 4, 9, 13
May 2016	SH(NA)-081264ENG-J	■Added model R6RFM ■Added functions CPU module data backup/restoration function, iQ Sensor Solution data backup/restoration function, redundant function, and Label initialization function ■Added or modified parts SAFETY PRECAUTIONS, INTRODUCTION, TERMS, Section 1.1, 2.2, 4.4, 16.3, 20.2, 20.3, 26.2, Chapter 27, Section 27.1, 27.2, Chapter 31, Section 31.1, Chapter 31.2, Chapter 32, PART 4, Chapter 36, 37, 38, 39, 40, Appendix 1, 2, 3, 4, 5, 9, 10, 12, 13
June 2016	SH(NA)-081264ENG-K	■Added or modified parts SAFETY PRECAUTIONS, TERMS, Section 21.1, 21.2, 26.1, Chapter 27, 31, PART 4, Appendix 1, 2, 4, 5, 9, 13
October 2016	SH(NA)-081264ENG-L	■Added functions External input/output forced on/off function, CC-Link IE Field Network Basic function, firmware update function ■Added or modified parts SAFETY PRECAUTIONS, Section 1.1, Chapter 3, Section 4.4, 6.4, 7.1, 23.5, 23.14, 24.6, 26.1, 26.2, 26.3, 27.1, 27.2, Chapter 28, Section 29.2, 29.4, 37.1, Chapter 38, Section 38.2, 38.3, 38.5, Appendix 1, 2, 3, 4, 5, 8, 9, 10, 12, 13 ■Deleted parts Chapter 32

Revision date	*Manual number	Description
January 2017	SH(NA)-081264ENG-M	■Added or modified parts TERMS, Section 10.1, 10.2, Chapter 31, Section 38.3, Appendix 1, 13
May 2017	SH(NA)-081264ENG-N	■Added or modified parts Section 4.4, Chapter 7, Section 7.1, Chapter 12, Section 12.1, 12.3, Chapter 27, Section 27.2, Appendix, 1, 4, 5, 8, 9, 12, 13, 14

Japanese manual number: SH-081224-N

This manual confers no industrial property rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2014 MITSUBISHI ELECTRIC CORPORATION

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

(1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.

(2) Even within the gratis warranty term, repairs shall be charged for in the following cases.

1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
2. Failure caused by unapproved modifications, etc., to the product by the user.
3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

(1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.

(2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to:

(1) Damages caused by any cause found not to be the responsibility of Mitsubishi.

(2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.

(3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.

(4) Replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

· For Safety CPUs

Please confirm the following product warranty details before using this product.

1. Limited Warranty and Product Support.

a. Mitsubishi Electric Company ("MELCO") warrants that for a period of eighteen (18) months after date of delivery from the point of manufacture or one year from date of Customer's purchase, whichever is less, Mitsubishi MELSEC Safety programmable logic controllers (the "Products") will be free from defects in material and workmanship.

b. At MELCO's option, for those Products MELCO determines are not as warranted, MELCO shall either repair or replace them or issue a credit or return the purchase price paid for them.

c. For this warranty to apply:

(1) Customer shall give MELCO (i) notice of a warranty claim to MELCO and the authorized dealer or distributor from whom the Products were purchased, (ii) the notice shall describe in reasonable details the warranty problem, (iii) the notice shall be provided promptly and in no event later than thirty (30) days after the Customer knows or has reason to believe that Products are not as warranted, and (iv) in any event, the notice must given within the warranty period;

(2) Customer shall cooperate with MELCO and MELCO's representatives in MELCO's investigation of the warranty claim, including preserving evidence of the claim and its causes, meaningfully responding to MELCO's questions and investigation of the problem, grant MELCO access to witnesses, personnel, documents, physical evidence and records concerning the warranty problem, and allow MELCO to examine and test the Products in question offsite or at the premises where they are installed or used; and

(3) If MELCO requests, Customer shall remove Products it claims are defective and ship them to MELCO or MELCO's authorized representative for examination and, if found defective, for repair or replacement. The costs of removal, shipment to and from MELCO's designated examination point, and reinstallation of repaired or replaced Products shall be at Customer's expense.

(4) If Customer requests and MELCO agrees to effect repairs onsite at any domestic or overseas location, the Customer will pay for the costs of sending repair personnel and shipping parts. MELCO is not responsible for any re-commissioning, maintenance, or testing on-site that involves repairs or replacing of the Products.

d. Repairs of Products located outside of Japan are accepted by MELCO's local authorized service facility centers ("FA Centers"). Terms and conditions on which each FA Center offers repair services for Products that are out of warranty or not covered by MELCO's limited warranty may vary.

e. Subject to availability of spare parts, MELCO will offer Product repair services for (7) years after each Product model or line is discontinued, at MELCO's or its FA Centers' rates and charges and standard terms in effect at the time of repair. MELCO usually produces and retains sufficient spare parts for repairs of its Products for a period of seven (7) years after production is discontinued.

f. MELCO generally announces discontinuation of Products through MELCO's Technical Bulletins. Products discontinued and repair parts for them may not be available after their production is discontinued.

2. Limits of Warranties.

a. MELCO does not warrant or guarantee the design, specify, manufacture, construction or installation of the materials, construction criteria, functionality, use, properties or other characteristics of the equipment, systems, or production lines into which the Products may be incorporated, including any safety, fail-safe and shut down systems using the Products.

b. MELCO is not responsible for determining the suitability of the Products for their intended purpose and use, including determining if the Products provide appropriate safety margins and redundancies for the applications, equipment or systems into which they are incorporated.

c. Customer acknowledges that qualified and experienced personnel are required to determine the suitability, application, design, construction and proper installation and integration of the Products. MELCO does not supply such personnel.

d. MELCO is not responsible for designing and conducting tests to determine that the Product functions appropriately and meets application standards and requirements as installed or incorporated into the end-user's equipment, production lines or systems.

e. MELCO does not warrant any Product:

(1) repaired or altered by persons other than MELCO or its authorized engineers or FA Centers;

(2) subjected to negligence, carelessness, accident, misuse, or damage;

(3) improperly stored, handled, installed or maintained;

(4) integrated or used in connection with improperly designed, incompatible or defective hardware or software;

(5) that fails because consumable parts such as batteries, backlights, or fuses were not tested, serviced or replaced;

(6) operated or used with equipment, production lines or systems that do not meet applicable and commensurate legal, safety and industry-accepted standards;

(7) operated or used in abnormal applications;

(8) installed, operated or used in contravention of instructions, precautions or warnings contained in MELCO's user, instruction and/or safety manuals, technical bulletins and guidelines for the Products;

(9) used with obsolete technologies or technologies not fully tested and widely accepted and in use at the time of the Product's manufacture;

(10) subjected to excessive heat or moisture, abnormal voltages, shock, excessive vibration, physical damage or other improper environment; or

(11) damaged or malfunctioning due to Acts of God, fires, acts of vandals, criminals or terrorists, communication or power failures, or any other cause or failure that results from circumstances beyond MELCO's control.

f. All Product information and specifications contained on MELCO's website and in catalogs, manuals, or technical information materials provided by MELCO are subject to change without prior notice.

g. The Product information and statements contained on MELCO's website and in catalogs, manuals, technical bulletins or other materials provided by MELCO are provided as a guide for Customer's use. They do not constitute warranties and are not incorporated in the contract of sale for the Products.
h. These terms and conditions constitute the entire agreement between Customer and MELCO with respect to warranties, remedies and damages and supersede any other understandings, whether written or oral, between the parties. Customer expressly acknowledges that any representations or statements made by MELCO or others concerning the Products outside these terms are not part of the basis of the bargain between the parties and are not factored into the pricing of the Products.
i. THE WARRANTIES AND REMEDIES SET FORTH IN THESE TERMS ARE THE EXCLUSIVE AND ONLY WARRANTIES AND REMEDIES THAT APPLY TO THE PRODUCTS.
j. MELCO DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

3. Limits on Damages.

a. MELCO'S MAXIMUM CUMULATIVE LIABILITY BASED ON ANY CLAIMS FOR BREACH OF WARRANTY OR CONTRACT, NEGLIGENCE, STRICT TORT LIABILITY OR OTHER THEORIES OF RECOVERY REGARDING THE SALE, REPAIR, REPLACEMENT, DELIVERY, PERFORMANCE, CONDITION, SUITABILITY, COMPLIANCE, OR OTHER ASPECTS OF THE PRODUCTS OR THEIR SALE, INSTALLATION OR USE SHALL BE LIMITED TO THE PRICE PAID FOR PRODUCTS NOT AS WARRANTED.
b. Although MELCO has obtained the certification for Product's compliance to the international safety standards IEC61508 and EN954-1/ISO13849-1 from TUV Rheinland, this fact does not guarantee that Product will be free from any malfunction or failure. The user of this Product shall comply with any and all applicable safety standard, regulation or law and take appropriate safety measures for the system in which the Product is installed or used and shall take the second or third safety measures other than the Product. MELCO is not liable for damages that could have been prevented by compliance with any applicable safety standard, regulation or law.
c. MELCO prohibits the use of Products with or in any application involving power plants, trains, railway systems, airplanes, airline operations, other transportation systems, amusement equipments, hospitals, medical care, dialysis and life support facilities or equipment, incineration and fuel devices, handling of nuclear or hazardous materials or chemicals, mining and drilling, and other applications where the level of risk to human life, health or property are elevated.
d. MELCO SHALL NOT BE LIABLE FOR SPECIAL, INCIDENTAL, CONSEQUENTIAL, INDIRECT OR PUNITIVE DAMAGES, FOR LOSS OF PROFITS, SALES, OR REVENUE, FOR INCREASED LABOR OR OVERHEAD COSTS, FOR DOWNTIME OR LOSS OF PRODUCTION, FOR COST OVERRUNS, OR FOR ENVIRONMENTAL OR POLLUTION DAMAGES OR CLEAN-UP COSTS, WHETHER THE LOSS IS BASED ON CLAIMS FOR BREACH OF CONTRACT OR WARRANTY, VIOLATION OF STATUTE, NEGLIGENCE OR OTHER TORT, STRICT LIABILITY OR OTHERWISE.
e. In the event that any damages which are asserted against MELCO arising out of or relating to the Products or defects in them, consist of personal injury, wrongful death and/or physical property damages as well as damages of a pecuniary nature, the disclaimers and limitations contained in these terms shall apply to all three types of damages to the fullest extent permitted by law. If, however, the personal injury, wrongful death and/or physical property damages cannot be disclaimed or limited by law or public policy to the extent provided by these terms, then in any such event the disclaimer of and limitations on pecuniary or economic consequential and incidental damages shall nevertheless be enforceable to the fullest extent allowed by law.
f. In no event shall any cause of action arising out of breach of warranty or otherwise concerning the Products be brought by Customer more than one year after the cause of action accrues.
g. Each of the limitations on remedies and damages set forth in these terms is separate and independently enforceable, notwithstanding the unenforceability or failure of essential purpose of any warranty, undertaking, damage limitation, other provision of these terms or other terms comprising the contract of sale between Customer and MELCO.

4. Delivery/Force Majeure.

a. Any delivery date for the Products acknowledged by MELCO is an estimated and not a promised date. MELCO will make all reasonable efforts to meet the delivery schedule set forth in Customer's order or the purchase contract but shall not be liable for failure to do so.
b. Products stored at the request of Customer or because Customer refuses or delays shipment shall be at the risk and expense of Customer.
c. MELCO shall not be liable for any damage to or loss of the Products or any delay in or failure to deliver, service, repair or replace the Products arising from shortage of raw materials, failure of suppliers to make timely delivery, labor difficulties of any kind, earthquake, fire, windstorm, flood, theft, criminal or terrorist acts, war, embargoes, governmental acts or rulings, loss or damage or delays in carriage, acts of God, vandals or any other circumstances reasonably beyond MELCO's control.

5. Choice of Law/Jurisdiction.

These terms and any agreement or contract between Customer and MELCO shall be governed by the laws of the State of New York without regard to conflicts of laws. To the extent any action or dispute is not arbitrated, the parties consent to the exclusive jurisdiction and venue of the federal and state courts located in the Southern District of the State of New York. Any judgment there obtained may be enforced in any court of competent jurisdiction.

6. Arbitration.

Any controversy or claim arising out of, or relating to or in connection with the Products, their sale or use or these terms, shall be settled by arbitration conducted in accordance with the Center for Public Resources (CPR) Rules for Non-Administered Arbitration of International Disputes, by a sole arbitrator chosen from the CPR's panels of distinguished neutrals. Judgment upon the award rendered by the Arbitrator shall be final and binding and may be entered by any court having jurisdiction thereof. The place of the arbitration shall be New York City, New York. The language of the arbitration shall be English. The neutral organization designated to perform the functions specified in Rule 6 and Rules 7.7(b), 7.8 and 7.9 shall be the CPR.

TRADEMARKS

Ethernet is a registered trademark of Fuji Xerox Co., Ltd. in Japan.
Microsoft, Microsoft Access, Excel, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, Windows NT, Windows Server, Windows Vista, and Windows XP are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.
In some cases, trademark symbols such as '™' or '®' are not specified in this manual.

SH(NA)-081264ENG-N(1705)MEE
MODEL: RCPU-U-OU-E
MODEL CODE: 13JX20

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

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.