MITSUBISHI ELECTRIC

Programmable Controller

MELSEC iQ-R series

MELSEC iQ-R SIL2 Process CPU Module User's Manual

-R08PSFCPU
-R16PSFCPU
-R32PSFCPU
-R120PSFCPU
-R6RFM
-R6PSFM

SAFETY PRECAUTIONS

(Read these precautions before using this product.)

Before using MELSEC iQ-R series programmable controllers, please read the manuals for the product and the relevant manuals introduced in those manuals carefully, and pay full attention to safety to handle the product correctly. If products are used in a different way from that specified by manufacturers, the protection function of the products may not work properly.

In this manual, the safety precautions are classified into two levels: "⚠WARNING" and "⚠CAUTION".

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

Under some circumstances, failure to observe the precautions given under "⚠CAUTION" may lead to serious consequences.

Observe the precautions of both levels because they are important for personal and system safety.

Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

[Design Precautions]

⚠ WARNING

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

[Precautions for using SIL2 Process CPUs]

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

[Design Precautions]

⚠ WARNING

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

[Precautions for using redundant function modules]

- The optical transmitter and receiver of the redundant function module use laser diodes (class 1 in accordance with IEC 60825-1: 2014/JIS C6802: 2014). Do not look directly at a laser beam. Doing so may harm your eyes.

[Design Precautions]

⚠ CAUTION

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

[Precautions for using SIL2 Process CPUs]

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

[Security Precautions]

⚠ WARNING

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

[Installation Precautions]

⚠ WARNING

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

[Installation Precautions]

⚠ CAUTION

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

[Wiring Precautions]

⚠ WARNING

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

[Wiring Precautions]

⚠ CAUTION

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

[Wiring Precautions]

⚠ CAUTION

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

[Precautions for using redundant function modules]

- For tracking cables to be used in the system, select the ones that meet the specifications in Page 69 Redundant Function Module. If not, normal data transmission is not guaranteed.

[Startup and Maintenance Precautions]

⚠ WARNING

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

[Startup and Maintenance Precautions]

⚠ CAUTION

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

[Startup and Maintenance Precautions]

⚠ CAUTION

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

[Precautions for using SIL2 Process CPUs]

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

[Operating Precautions]

⚠ CAUTION

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

[Computer Connection Precautions]

⚠ CAUTION

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

[Disposal Precautions]

⚠ CAUTION

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

[Transportation Precautions]

⚠ CAUTION

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

INTRODUCTION

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

Relevant products

Item	Model
CPU module	R08PSFCPU, R16PSFCPU, R32PSFCPU, R120PSFCPU
Redundant function module	R6RFM
SIL2 function module	R6PSFM

CONTENTS

SAFETY PRECAUTIONS 1
INTRODUCTION 12
RELEVANT MANUALS 23
TERMS 24
GENERIC TERMS AND ABBREVIATIONS 25

PART 1 PART NAMES

CHAPTER 1 CPU MODULE 28

CHAPTER 2 EXTENDED SRAM CASSETTE 31

CHAPTER 3 REDUNDANT FUNCTION MODULE 33

CHAPTER 4 SIL2 FUNCTION MODULE 35

PART 2 PROCEDURES BEFORE OPERATION

CHAPTER 5 START-UP PROCEDURE 38

5.1 Overview 38
5.2 Installing a Battery 46
Installation procedure 46
5.3 Inserting or Removing an Extended SRAM Cassette 47
Insertion procedure 47
Removal procedure 47
5.4 Inserting and Removing an SD Memory Card 48
Insertion procedure 48
Removal procedure 48
5.5 Wiring 49
Redundant function modules 49
Wiring to power supply modules 50
5.6 Creating a Project 51
5.7 Connecting a Personal Computer 52
5.8 Setting the System (System A or System B) 53
5.9 Writing Data to the Programmable Controller 55
5.10 Resetting the CPU Module 56
5.11 Executing the Program 57
5.12 Monitoring the Program 58

PART 3 SYSTEM CONFIGURATION

CHAPTER 6 SYSTEM CONFIGURATION 60

PART 4 SPECIFICATIONS

CHAPTER 7 PERFORMANCE SPECIFICATIONS 64

7.1 CPU Module 64
Hardware specifications 64
Programming specifications 66
7.2 Extended SRAM Cassette 68
7.3 Redundant Function Module 69
7.4 SIL2 Function Module 70

PART 5 CPU MODULE OPERATION

CHAPTER 8 BASIC CONCEPT 72

8.1 System 74
8.2 System Switching Between the Control System and Standby System 74
8.3 System Consistency Check 75
8.4 Operation Mode (Mode that Determines the Operation of the CPU Module) 75
8.5 Operation Modes of the System 75
8.6 Safety Operation Mode 76
Checking the safety operation mode 76
Operations restricted in SAFETY MODE 76
8.7 Determination of Control System/Standby System 77
When starting up both systems simultaneously 77
When starting up one system first 79
When one system is started automatically even though a tracking communication error has occurred 81
When starting up the previous control system as the control system 85
8.8 System State Transition After Start-up 86

CHAPTER 9 RUNNING A PROGRAM 87

9.1 Overview 87
9.2 Scan Configuration 88
Initial processing (when powered on or switched to RUN) 89
I/O refresh 89
Program operation (standard program) 90
Program operation (safety program) 90
END processing 93
9.3 Scan Time 94
Initial scan time 94
Constant scan 95
Device/label access service processing setting 97
9.4 Data Communication and I/O Processing 100
Data communication 100
I/O processing and response delay 100
Refresh mode 101
Direct mode 103
9.5 Program Flow 105
9.6 Program Execution Type 106
Initial execution type program 107
Scan execution type program 108

Fixed scan execution type program. . . . 108
Event execution type program. . . . 113
Standby type program 118
Execution type change 119
Group setting for refresh 121
9.7 Subroutine Program 122
9.8 Interrupt Program 123
Interrupt period setting. . . . 128
Interrupt enabled during instruction execution. . . . 129
Processing at interrupt program startup 130
Multiple interrupt function 133
Precautions 137

CHAPTER 10 CPU MODULE OPERATION PROCESSING 139

10.1 Operation Processing by Operating Status 139
10.2 Operation Processing When Operating Status Is Changed 141
Output mode at operating status change (STOP to RUN) 142
10.3 Operation Processing at Momentary Power Failure 144

CHAPTER 11 MEMORY CONFIGURATION OF THE CPU MODULE 145

11.1 Memory Configuration 145
Program memory/program cache memory 145
Device/label memory 147
Data memory 152
Refresh memory 152
CPU buffer memory 152
Signal flow memory 153
SD memory card 154
11.2 File Size Unit in Memory 156
11.3 Memory Operation 157
Initialization and value clear 157
11.4 Files 158
File types and storage memory 158
File operation available 159
File size 161

PART 6 FUNCTIONS

CHAPTER 12 FUNCTION LIST 164

CHAPTER 13 CLOCK FUNCTION 167

13.1 Time Setting 167
Clock data 167
Changing the clock data 168
Reading the clock data 169
Precautions on the clock data 169
13.2 Setting Time Zone 170
13.3 System Clock 171
Special relay/safety special relay used for system clock 171
Special register used for system clock 172

CHAPTER 14 WRITING DATA TO THE CPU MODULE 173
14.1 Writing Data to the Programmable Controller 173
14.2 Online Change 173
Online change (ladder block) 173
File batch online change 176
14.3 Precautions 177
CHAPTER 15 RAS FUNCTIONS 182
15.1 Scan Monitoring Function 182
Scan time monitoring time setting 182
Watchdog timer reset 182
Precautions 183
15.2 Self-Diagnostics Function 184
Self-diagnostics timing 184
How to check for errors (CPU module) 184
How to check for errors (SIL2 function module) 185
CPU module operation upon error detection setting 186
Safety diagnostic functions 189
15.3 Error Clear 190
15.4 Clearing Errors in the CPU Module of the Standby System from the CPU Module of the Control System 193
15.5 Event History Function 194
Event history setting 194
Logging of the event history 195
Viewing the event history 198
Clearing the event history 198
Precautions 198
15.6 Program Cache Memory Auto Recovery Function 199
CHAPTER 16 REMOTE OPERATION 200
16.1 Remote RUN/STOP/PAUSE 200
16.2 Remote RESET 201
16.3 Setting RUN-PAUSE Contacts 202
16.4 Precautions 202
CHAPTER 17 PID CONTROL/PROCESS CONTROL FUNCTION 203
17.1 PID Control Function 203
17.2 Process Control Function 204
Process control by using process control function blocks 204
Process control by using process control instructions 205
CHAPTER 18 SECURITY FUNCTION 206
18.1 Security Key Authentication for a CPU Module 208
18.2 User Authentication Function of CPU Modules 209
CHAPTER 19 ROUTING SETTING 210
19.1 Setting Method 210
19.2 Setting Example 211

CHAPTER 20 REDUNDANT FUNCTION 212
20.1 System Switching 212
System switching method 212
Operation at system switching 217
Execution availability of system switching 219
Check method of system switching information 220
Precautions 222
20.2 Tracking Transfer 225
Tracking data 226
Tracking block and tracking trigger 229
Setting procedure for tracking transfer 231
Tracking transfer setting 231
Safety tracking transfer setting 236
Tracking mode 237
Precautions 240
20.3 Memory Copy from Control System to Standby System 242
Memory copy using the engineering tool 245
Precautions 246
20.4 System Consistency Check 247
Files 249
Operating status 250
Modules on the main base unit 250
SD memory card 251
Safety operation mode 251
20.5 Program Execution in Both Systems 252
Operation of a program executed in both systems 254
Precautions 256
20.6 Redundant System Operation Setting 258
Standby system output setting 259
Setting to wait cyclic data receive after system switching 260
20.7 Redundant Function Module Communication Test 261
Execution procedure of the module communication test 261
CHAPTER 21 SWITCHING SAFETY OPERATION MODE 262
CHAPTER 22 IDENTIFICATION CHECK FOR SAFETY DATA 265
CHAPTER 23 SAFETY COMMUNICATION FUNCTION 267
23.1 Safety I/O Hold Time 268
CHAPTER 24 SLMP COMMUNICATIONS 270
CHAPTER 25 PRECAUTIONS ON PROGRAMMING 272
25.1 Restricted Instructions 272
Instructions that need to be executed again in a new control system 272
Instructions whose operations vary depending on tracking of the signal flow memory 275
Instructions that affect the status of another instruction when executed 278
Instructions causing different operation results between both systems 278
Precautions for using the COM or ZCOM instruction 279
Precautions for using the ADRSET instruction 279

25.2 Interrupt from modules . . . 280
25.3 Precautions for using the annunciator (F) . . . 280
25.4 Precautions on timers and timer function blocks . . . 281
25.5 Precautions on access to intelligent function module or external devices. . . . 282
25.6 Precautions on writing data from GOT or external devices . . . 284
25.7 Precautions on outputting in the middle of the scan. . . . 284

PART 7 DEVICES, LABELS, AND CONSTANTS

CHAPTER 26 STANDARD DEVICES 286

26.1 Standard Device List . . . 286
26.2 Device Setting . . . 288
Range of use of device points . . . 289
26.3 User Device . . . 290
Input (X) . . . 290
Output (Y) . . . 290
Internal relay (M) . . . 290
Latch relay (L) . . . 290
Link relay (B) . . . 291
Annunciator (F) . . . 291
Link special relay (SB) . . . 293
Edge relay (V) . . . 293
Timer . . . 294
Counter . . . 302
Data register (D) . . . 305
Link register (W) . . . 305
Link special register (SW) . . . 305
26.4 System Device . . . 306
Function device (FX/FY/FD) . . . 306
Special relay (SM) . . . 307
Special register (SD) . . . 307
26.5 Link Direct Device . . . 308
Specification method . . . 308
Specification range . . . 308
Difference from link refresh . . . 309
26.6 Module Access Device . . . 310
Precautions . . . 310
26.7 CPU Buffer Memory Access Device . . . 311
Specification method . . . 311
26.8 Index Register (Z/LZ) . . . 312
16-bit index modification . . . 312
32-bit index modification . . . 312
Device for which index modification can be performed . . . 313
Index register setting . . . 313
Combination of index modification . . . 314
Precautions . . . 314
26.9 File Register (R/ZR) . . . 316
Specification method . . . 316
Setting file registers . . . 317
Clearing file registers . . . 317

26.10 Refresh Data Register (RD) ... 318
Refresh memory setting ... 318
26.11 Nesting (N) ... 319
26.12 Pointer (P) ... 320
Global pointer ... 320
Local pointer ... 321
Pointer setting ... 322
26.13 Interrupt Pointer (I) ... 323
Interrupt factors of the interrupt pointer numbers ... 323
The priority for the interrupt pointer numbers and interrupt factors ... 324
26.14 Standard Global Device ... 325
26.15 Standard Local Device ... 325
26.16 Indirect Specification ... 329

CHAPTER 27 SAFETY DEVICES 330

27.1 Safety Devices ... 330
List of safety devices ... 330
Safety user device ... 331
Safety system devices ... 334
27.2 Safety Global Devices ... 335
27.3 Safety Local Devices ... 335

CHAPTER 28 STANDARD LABEL 336

28.1 Standard Global Label ... 336
28.2 Standard Local Label ... 337
28.3 Class ... 337
28.4 Data Types ... 338
28.5 Arrays ... 341
28.6 Structures ... 345
28.7 Label Access Setting from External Device ... 347
Configuration procedure ... 347
Label communication data storage location and operation at the time of each setting operation ... 347
File operation ... 347
28.8 Precautions ... 348

CHAPTER 29 SAFETY LABEL 350

29.1 Safety Label ... 350
Safety label types ... 350
Classes ... 351
Data types ... 352
Structures ... 352

CHAPTER 30 LATCH FUNCTION 353

30.1 Latch with Battery ... 353
Types of latch ... 353
Applicable devices and labels ... 354
Setting latch on devices ... 355
Setting latch on labels ... 359
Clearing latch range data ... 359
Precautions ... 359

CHAPTER 31 DEVICE/LABEL INITIAL VALUE SETTINGS 360
31.1 Setting Initial Device/Label Values. . . . 360
Setting initial device values . . . 360
Setting initial label values . . . 362
31.2 Applicable Devices/Labels . . . 363
31.3 Precautions . . . 363
CHAPTER 32 LABEL INITIALIZATION FUNCTION 364
32.1 Initialization of Labels After Rebuilt All (Reassignment) . . . 364
32.2 Label Initial Value Reflection Setting . . . 366
CHAPTER 33 CONSTANTS 368
33.1 Decimal Constant (K) . . . 368
33.2 Hexadecimal Constant (H) . . . 368
33.3 Real Constant (E). . . . 369
Setting range for real numbers . . . 369
Processing when operation is in progress. . . . 369
33.4 Character String Constant . . . 370
33.5 Notation of Constants . . . 371
PART 8 TROUBLESHOOTING
CHAPTER 34 TROUBLESHOOTING PROCEDURE 374
34.1 Troubleshooting with LED Indicators . . . 375
LED status of the CPU module . . . 375
34.2 Troubleshooting Using the Engineering Tool. . . . 376
System monitor . . . 376
Module diagnostics . . . 376
Event history . . . 376
CHAPTER 35 TROUBLESHOOTING BY SYMPTOM 377
35.1 When the POWER LED of the Power Supply Module Turns Off . . . 377
35.2 When the READY LED of the CPU Module Turns Off. . . . 377
35.3 When an Error Has Occurred in a Redundant Function Module . . . 378
When the RUN LED turns off . . . 378
When the ERR LED turns on or flashes . . . 378
When the L ERR LED turns on . . . 378
35.4 When an Error Has Occurred in a SIL2 Function Module . . . 379
When the READY LED turns off . . . 379
35.5 When Data Cannot Be Written to the Programmable Controller . . . 379
35.6 When Data Cannot Be Read from the Programmable Controller . . . 379
35.7 When the Operating Status of the CPU Module Cannot Be Changed . . . 380
35.8 When the Ethernet Function Cannot Be Used . . . 380
35.9 When Cyclic Data Is Turned Off at the Time of System Switching . . . 380
CHAPTER 36 MAINTENANCE AND INSPECTION 381
36.1 Module Replacement. . . . 381
Replacing a CPU module and SIL2 function module . . . 382
Replacing a power supply module. . . . 382

Replacing a redundant power supply module . . . 383
Replacing a redundant function module . . . 383
Replacing I/O modules . . . 383
Replacing an intelligent function module . . . 384
Procedure for remote head module replacement . . . 384
Replacing a main base unit . . . 385

CHAPTER 37 ERROR CODES 386

37.1 Error Code System . . . 386
37.2 Operation When an Error Occurs . . . 388
37.3 Safety Output Operation When Error Occurs . . . 388
37.4 How to Clear Errors . . . 388
37.5 List of Error Codes . . . 389
Codes of errors detected by the self-diagnostic function (1000H to 3FFFH) . . . 389
Codes of errors detected by other than the self-diagnostic function (4000H to 4FFFH) . . . 433
Codes of errors detected by other than the self-diagnostic function (6F00H to 6FFFH) . . . 462
Codes of errors detected by other than the self-diagnostic function (C000H to CFFFH) . . . 463

CHAPTER 38 EVENT LIST 464

38.1 How to Read the Event List . . . 464
38.2 Event List . . . 466

APPENDICES 473

Appendix 1 External Dimensions . . . 473
Appendix 2 Compliance with EMC and Low Voltage Directives . . . 476
Appendix 3 Functional Availability by CPU Module Model . . . 477
Appendix 4 List of Special Relay Areas . . . 479
Diagnostic information . . . 480
System information . . . 481
System clock . . . 482
Drive information . . . 483
Instruction related . . . 485
Ethernet function . . . 487
Online module change function . . . 488
Redundant function . . . 490
Safety information . . . 492
Appendix 5 List of Special Register Areas . . . 493
Diagnostic information . . . 494
System information . . . 505
System clock . . . 507
Fixed scan function information . . . 508
Drive information . . . 510
Instruction related . . . 512
Interrupt pointer mask pattern . . . 513
User authentication function . . . 513
Ethernet function . . . 514
Online module change function . . . 516
Redundant function . . . 517
Safety information . . . 522

Appendix 6 List of Safety Special Relay Areas 524
System clock 525
Safety information 526
Appendix 7 List of Safety Special Register Areas 527
System information 528
Safety information 528
Appendix 8 Buffer Memory 533
List of buffer memory addresses 533
Appendix 9 Processing Time 536
Instruction execution time 536
Program execution time. 536
END processing time. 541
Process control function processing time 548
Redundant function processing time 549
Appendix 10 Parameter List 557
System parameters 557
CPU parameters 557
Module parameters 559
Safety CPU parameters. 559
Memory card parameters 559
Appendix 11 Target List and Operation Details of the Device/Label Access Service Processing Setting 560
Target list 560
Operation details 562
Appendix 12 Added and Enhanced Functions 566

INDEX 567

REVISIONS 570
CONDITIONS OF USE FOR THE PRODUCT 571
WARRANTY 572
INFORMATION AND SERVICES 574
TRADEMARKS 574

RELEVANT MANUALS

The following manuals are relevant to this product.

Manual name [manual number]	Description
MELSEC iQ-R SIL2 Process CPU Module User's Manual [SH-082503ENG] (this manual)	Procedures before operation, specifications, devices, memory, functions, parameters, and troubleshooting of the CPU module
MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup) [SH-081256ENG]	Specifications, procedures before operation, system configuration, wiring, and communication examples of Ethernet, CC-Link IE Controller Network, and CC-Link IE Field Network
MELSEC iQ-R Ethernet User's Manual (Application) [SH-081257ENG]	Functions, parameter settings, programming, troubleshooting, I/O signals, and buffer memory of Ethernet
MELSEC iQ-R CPU Module Function Block Reference [BCN-P5999-0374]	Specifications of the MELSEC iQ-R series CPU module FBs
MELSEC iQ-R Ethernet, CC-Link IE, and MELSECNET/H Function Block Reference [BCN-P5999-0381]	Specifications of the following MELSEC iQ-R series module FBs: Ethernet-equipped module FBs, CC-Link IE TSN module FBs, CC-Link IE Controller Network module FBs, CC-Link IE Field Network module FBs, and MELSECNET/H network module FBs
MELSEC iQ-R Safety Function Block Reference [BCN-P5999-0815]	Specifications of the safety FBs
MELSEC iQ-R Programming Manual (Program Design) [SH-081265ENG]	Program specifications (ladder, ST, FBD/LD, and SFC programs)
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks) [SH-081266ENG]	Instructions for the CPU module and standard functions/function blocks
MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions) [SH-081749ENG]	Standard process function blocks, tag access function blocks, tag function blocks, and process control instruction designed for the process control
MELSEC iQ-R Programming Manual (Module Dedicated Instructions) [SH-081976ENG]	Dedicated instructions for the intelligent function modules
GX Works3 Operating Manual [SH-081215ENG]	System configuration, parameter settings, and online operations of GX Works3

This manual does not include detailed information on the following:

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

For details, refer to the following.

MELSEC iQ-R Module Configuration Manual

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

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

TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
Backup mode	A mode to continue operation in a redundant system by switching the standby system to the control system when an error occurs in the control system.
Buffer memory	Memory in an intelligent function module for storing data such as setting values and monitored values. When integrated into the CPU module, this memory refers to a memory for storing data such as setting values and monitored values of the Ethernet function, and data used for data communication of the multiple CPU system function.
Control 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.
Dedicated instruction	An instruction that simplifies programming for using functions of intelligent function modules
Device	A memory of a CPU module to store data. Devices such as X, Y, M, D, and others are provided depending on the intended use.
Engineering tool	A tool used for setting up programmable controllers, programming, debugging, and maintenance.
FB instance	A function block that is inserted to a sequence program
Global label	A label that is valid for all the program data when multiple program data are created in the project. There are two types of global label: a module specific label (module label), which is generated automatically by GX Works3, and an optional label, which can be created for any specified device.
Intelligent function module	A module that has functions other than input and output, such as an A/D converter module and D/A converter module
Label	A variable consisting of a specified string used in I/O data or internal processing
Module label	A label that represents one of memory areas (I/O signals and buffer memory areas) specific to each module in a given character string. For the module used, GX Works3 automatically generates this label, which can be used as a global label.
New control system	A system that has switched to control system from standby system after system switching
New standby system	A system that has switched to standby system from control system after system switching
Own system	The system that contains the CPU module that is used for descriptions.
POU	A unit that configures a program. Units are categorized and provided in accordance with functions. A program that is composed of POUs (program organization units) allows the lower-layer processing, when the program is multi-layered, to be divided into several units by processing and function, enabling the creation of programs based on each unit.
Program block	A group of POUs that configure a program
Program executed in both systems	A program that is executed in both CPU modules of the control system and the standby system
Redundant function module	A module that configures a redundant system and is used with a Process CPU (redundant mode) or a SIL2 Process CPU. The redundant function module model name is R6RFM.
Redundant system	A system consisting of two systems that have the same configuration (CPU module, power supply module, network module, and other modules). Even after an error occurs in one of the two systems, the other system takes over the control of the entire system. For details, refer to the descriptions of the redundant system in the following manual. MELSEC iQ-R Module Configuration Manual
Separate mode	A mode for system maintenance in a redundant system. This mode can maintain a redundant system without stopping control while the system is running.
Signal flow	The execution status that the last time an operation of a program or an FB is executed in each step.
SIL2 function module	A module that performs safety control and must be used with a SIL2 Process CPU. This module can only be used with the SIL2 Process CPU. The SIL2 function module model name is R6PSFM.
SIL2 mode	An operation mode of the I/O module and the intelligent function module to perform safety I/O at the SIL2 level. For details on the SIL2 mode, refer to the following. Manual for the I/O module or intelligent function module used
SIL2 Process CPU	A module that performs both standard control and safety control and is used with a SIL2 function module. This module is also used with a redundant function module and configures a redundant system. The SIL2 Process CPU models include the R08PSFCPU, R16PSFCPU, R32PSFCPU, and R120PSFCPU.
Standard/safety shared label	A label that can be used in both standard programs and safety programs. This label is used at data communications between a safety program and standard program.
Standby system	A backup system in a redundant system
System A	A system that is set as system A to distinguish two systems, which are connected with two tracking cables. When the two systems start up at the same time, this system will be a control system. System switching does not affect the system A/B setting.
System B	A system that is set as system B to distinguish two systems, which are connected with two tracking cables. When the two systems start up at the same time, this system will be a standby system. System switching does not affect the system A/B setting.
Tracking cable	An optical fiber cable used to connect two redundant function modules in a redundant system

GENERIC TERMS AND ABBREVIATIONS

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

Generic term/abbreviation	Description
Base unit	A main base unit, an extension base unit, an RQ extension base unit
CC-Link IE Controller Network-equipped module	An RJ71GP21-SX CC-Link IE Controller Network module, an RJ71GP21S-SX CC-Link IE Controller Network module, and the following modules when the CC-Link IE Controller Network function is used: • RJ71EN71 • RnENCPU
CC-Link IE Field Network-equipped master/local module	An RJ71GF11-T2 CC-Link IE Field Network master/local module and the following modules when the CC-Link IE Field Network function is used: • RJ71EN71 • RnENCPU
CPU module	R08PSFCPU, R16PSFCPU, R32PSFCPU, R120PSFCPU
Device supporting iQSS	A device which supports iQ Sensor Solution. For iQ Sensor Solution, refer to the following. 📖iQ Sensor Solution Reference Manual
Ethernet interface module with built-in CC-Link IE	RJ71EN71
Ethernet-equipped module	The following modules when the Ethernet function is used: • RJ71EN71 • CPU module
I/O module	An input module, an output module, an I/O combined module, and an interrupt module
Network module	Includes the following: • Ethernet interface module • CC-Link IE TSN master/local module • CC-Link IE Controller Network module • CC-Link IE Field Network master/local module • MELSECNET/H network module • MELSECNET/10 network module • RnENCPU (network part)
Power supply module	A MELSEC iQ-R series power supply module
RAS	Reliability, Availability, Serviceability. This term refers to the overall usability of automated equipment.
Remote head module	An RJ72GF15-T2 CC-Link IE Field Network remote head module
Standard CPU	A MELSEC iQ-R series CPU module that performs standard control (This term is used to distinguish from CPU modules that perform safety control.)
Standard programmable controller	A MELSEC iQ-R series module that performs standard control. (This term is used to distinguish a programmable controller from a safety programmable controller.)
Station sub ID number	An ID number of a sensor connected to a CC-Link-compatible communication module
Windows® 7 or later	Windows®7, Windows®8, Windows®8.1, Windows®10

MEMO

PART 1 PART NAMES

This part consists of the following chapters.

1 CPU MODULE

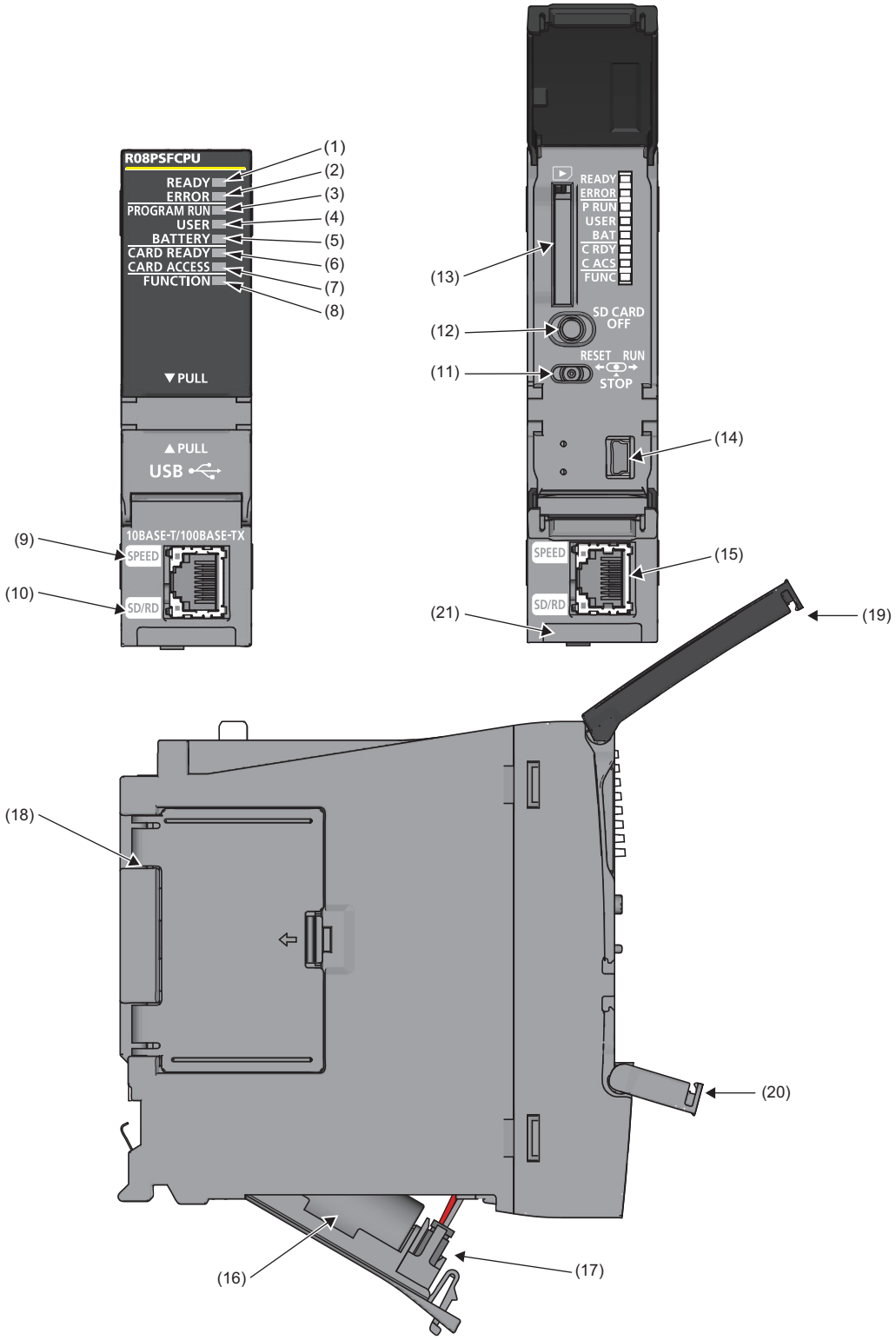
2 EXTENDED SRAM CASSETTE

3 REDUNDANT FUNCTION MODULE

4 SIL2 FUNCTION MODULE

1 CPU MODULE

The R08PSFCPU is used as an example.



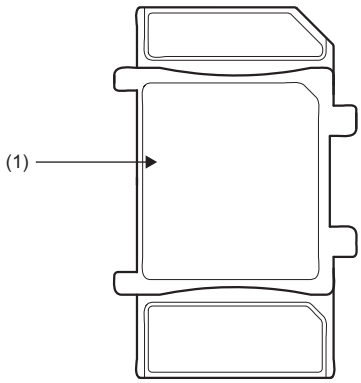
No.	Name	Description
(1)	READY LED	Indicates the operating status of the CPU module and the error level. (☞ Page 375 LED status of the CPU module) ● READY LED—ERROR LED status On—off: Normal operation On—on: Minor error On—flashing: Moderate error Flashing—on: Minor error (Changing module online) Flashing (every 2s)—off: Initial processing Flashing (every 400ms)—off: Changing module online Off—on/flashing: Major error
(2)	ERROR LED	
(3)	PROGRAM RUN LED	Indicates the operating status of the program. ■Control system (CTRL LED of the redundant function module: On) On: Being executed (RUN state) Flashing: Being suspended (PAUSE state) Off: Stopped (STOP state) or stop error ■Standby system (SBY LED of the redundant function module: On) [Backup mode] On: Being executed (programs being executed in both systems) Flashing: Being suspended (PAUSE state) (programs being executed in both systems) Off: Stopped (STOP state/RUN state/PAUSE state) (no program being executed in both systems) or stop error [Separate mode] On: Being executed (RUN state) Flashing: Being suspended (PAUSE state) or waiting for state transition to RUN (same as STOP state) Off: Stopped (STOP state) or stop error ■Systems not determined Flashing: Waiting for state transition to RUN by switch operation (same as STOP state) Off: Normal operation
(4)	USER LED	Indicates the status of the annunciator (F). (☞ Page 291 Annunciator (F)) On: Annunciator (F) ON Off: Normal operation
(5)	BATTERY LED	Indicates the battery status. Flashing: Battery low Off: Normal operation
(6)	CARD READY LED	Indicates the availability of the SD memory card. On: Available Flashing: Ready Off: Not available or not inserted
(7)	CARD ACCESS LED	Indicates the access status of the SD memory card. On: Being accessed Off: Not accessed
(8)	FUNCTION LED	Turns on or flashes when the following functions are used: • LED setting • Program restoration information write For the LED status, refer to the description of each function. ☞ Page 188 LED display setting
(9)	SPEED LED	Refer to the following. MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)
(10)	SD/RD LED	
(11)	RUN/STOP/RESET switch	A switch for controlling the operating status of the CPU module. (☞ Page 139 CPU MODULE OPERATION PROCESSING) RUN: Executes the program. STOP: Stops the program. RESET: Resets the CPU module. (Keep the switch in the RESET position for approximately one second.) Operate the RUN/STOP/RESET switch with your fingers. To prevent the switch from being damaged, do not use any tool such as a screwdriver.
(12)	SD CARD OFF button	A switch for disabling access to the SD memory card to remove it from the CPU module (☞ Page 48 Inserting and Removing an SD Memory Card)
(13)	SD memory card slot	A slot where an SD memory card is inserted
(14)	USB port*1	A connector for a USB-compatible peripheral (connector type: miniB)
(15)	Ethernet port	Refer to the following. MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)
(16)	Battery	A backup battery to hold clock data and to use the backup power function for the device/label memory
(17)	Battery connector pin	A pin for connecting a lead wire of the battery (To save the battery, the lead wire is disconnected from the connector before shipment.)
(18)	Cassette cover	A cover for the connector where an extended SRAM cassette is inserted (☞ Page 31 EXTENDED SRAM CASSETTE, ☞ Page 47 Inserting or Removing an Extended SRAM Cassette)

No.	Name	Description
(19)	LED cover	A cover for the LED indicators, SD memory card slot, and switches. Open this cover and insert or remove an SD memory card or set the RUN/STOP/RESET switch. Otherwise, keep the cover closed to prevent entry of foreign matter such as dust.
(20)	USB cover	A cover for the USB port. Open this cover and connect a USB-compatible peripheral. Otherwise, keep the cover closed to prevent entry of foreign matter such as dust.
(21)	Production information marking	Shows the production information (16 digits) of the module.

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

2 EXTENDED SRAM CASSETTE

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

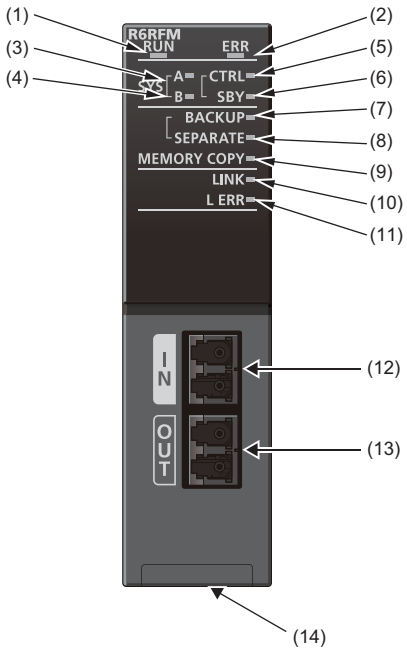


No.	Name	Description
(1)	Tab for cassette insertion/ removal	The part that is held when an extended SRAM cassette is inserted or removed (☞ Page 47 Inserting or Removing an Extended SRAM Cassette).

MEMO

3 REDUNDANT FUNCTION MODULE

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



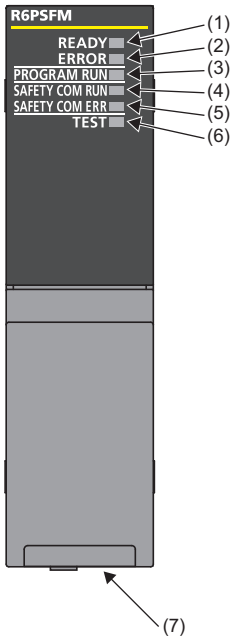
No.	Name	Description
(1)	RUN LED	Indicates the operating status. On: Normal operation Flashing: Changing module online or executing a module communication test Off: Error (☞ Page 378 When the RUN LED turns off)
(2)	ERR LED	Indicates the error status of the module. On: Error or module communication test completed with an error (☞ Page 378 When the ERR LED turns on or flashes) Flashing: Error (☞ Page 378 When the ERR LED turns on or flashes) Off: Normal operation
(3)	SYS A LED	Indicates the system A/B setting. On: System A Flashing: Parameter (set to system A) invalid Off: System B or not set
(4)	SYS B LED	Indicates the system A/B setting. On: System B Flashing: Parameter (set to system B) invalid Off: System A or not set
(5)	CTRL LED	Indicates the control status of the CPU module. Flashing: Control system Off: Standby system or not determined
(6)	SBY LED	Indicates the control status of the CPU module. On: Standby system Off: Control system or not determined
(7)	BACKUP LED	Indicates the operation mode. On: Backup mode Flashing: Cause of system switching failure being present*1 Off: Separate mode
(8)	SEPARATE LED	Indicates the operation mode. On: Separate mode Flashing: Cause of system switching failure being present*1 Off: Backup mode
(9)	MEMORY COPY LED	Indicates the memory copy status. On: Memory copy completed (standby system) Flashing (every 200ms): Memory copy being executed Flashing (every 1s): Memory copy error (standby system) Off: Memory copy not executed, or memory copy completed (control system)

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

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

4 SIL2 FUNCTION MODULE

This chapter describes the part names of the SIL2 function module.



No.	Name	Description
(1)	READY LED	Indicates the operating status of the CPU module and the error level of the safety control. (☞ Page 375 LED status of the CPU module). ● READY LED—ERROR LED status On—off: Normal operation On—on: Minor error On—flashing: Moderate error Off—on/flashing: Major error
(2)	ERROR LED	
(3)	PROGRAM RUN LED	Indicates the operating status of safety programs. On: Safety program being executed Off: Safety program not executed or stopped
(4)	SAFETY COM RUN LED	Indicates the status of safety communications. On: Safety communications being executed Off: Safety communications not executed or stopped
(5)	SAFETY COM ERR LED	Indicates the status of safety communications. On: Error during safety communications Off: No error
(6)	TEST LED	Indicates the status in TEST MODE. On: TEST MODE Flashing: SAFETY MODE (wait-for-restart) Off: SAFETY MODE
(7)	Production information marking	Shows the production information (16 digits) of the module.

MEMO

PART 2 PROCEDURES BEFORE OPERATION

This part consists of the following chapter.

5 START-UP PROCEDURE

5 START-UP PROCEDURE

This chapter describes the start-up procedure.

5.1 Overview

Take the following start-up procedure.

Point

The procedures for starting up both systems at a time are shown in this section.
Both system are started up at a time. However, starting up only one system is possible in the following cases.

- Debugging is required for only one system.
- One system is powered off before module replacement because the system is failed.

When starting up the systems one by one, apply the following procedures but replace the term "both systems" with "one system" to start up the control system. For start-up of the standby system, refer to the following.
☞ Page 44 Starting up the standby system

Installing/mounting/inserting procedure

1. Installing a battery

Install a battery to the CPU module of each system. (☞ Page 46 Installing a Battery)

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

Insert an extended SRAM cassette or an SD memory card into the CPU module of each system as needed. (☞ Page 47 Inserting or Removing an Extended SRAM Cassette, ☞ Page 48 Inserting and Removing an SD Memory Card)
Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card.

3. Mounting modules

Mount each module to base units. (📖MELSEC iQ-R Module Configuration Manual)

Wiring procedure

1. Wiring

Wire each module and external device.

Wiring target	Reference
Power supply module	📖MELSEC iQ-R Module Configuration Manual
Redundant function module	☞ Page 49 Redundant function modules
From the master/local module to the remote head module	• 📖MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup) • 📖MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Startup)
Intelligent function module	📖Manual for the intelligent function module used
I/O module	📖Manual for the I/O module used

Procedure for the remote head module side

1. Powering on the system

Check the following before powering on the system.

- A cable is correctly connected to the power supply.
- Power supply voltage is within the specified range.
- The remote head module is in the STOP state.

2. Creating a project

Activate the engineering tool and create a project. (MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Startup))

3. Connecting the personal computer and the remote head module

Connect the personal computer on which the engineering tool has been installed and the remote head module. (MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Startup))

4. Initializing the remote head module

Initialize the remote head module with the engineering tool. (MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Startup))

5. Setting parameters for the remote head module side

Set system parameters, CPU parameters, and module parameters of each module. (MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Application))

- Set system parameters and CPU parameters.
- Set module parameters of each module.

6. Writing the parameters to the remote head module

Write the parameters set using the engineering tool to the remote head module. (MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Startup))

7. Resetting the remote head module

Restart the system on the remote head module side with either of the following methods.

- Powering off and on the system
- Resetting the remote head module (MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Application))

Procedure for the CPU module side

1. Powering on the systems

Check each system for the following and power on the systems.

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

Check that the following LEDs turn on after the systems are powered on.

- Power supply module: POWER LED
- CPU module: READY LED
- SIL2 function module: READY LED
- Redundant function module: RUN LED
- Master/local module: RUN LED

Although the ERROR LED of each module flashes or turns on as follows in step 1, proceed to the next step.

- CPU module: The ERROR LED flashes.
- SIL2 function module: The ERROR LED flashes.
- Redundant function module: The ERR LED turns on.
- Master/local module: The ERR LED turns on.

2. Creating a project

Activate the engineering tool and create a project. (☞ Page 51 Creating a Project)

3. Connecting a personal computer and the CPU module

Connect a personal computer where an engineering tool has been installed to the CPU module. (☞ Page 52 Connecting a Personal Computer)

4. Initializing the CPU module

Initialize the CPU module using the engineering tool. (☞ Page 157 Initialization and value clear)

After initializing the CPU module of one system, connect the personal computer and the CPU module of the other system. Initialize the CPU module in the same way.

5. Setting parameters for the CPU module side

Set system parameters, CPU parameters, and module parameters of each module.

Point

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

6. Safety communication settings

Set the safety communication settings. (📖MELSEC iQ-R CC-Link IE Field Network User's Manual (Application))

7. Writing the system A/B setting

Set the system A or B using the engineering tool. (☞ Page 53 Setting the System (System A or System B))

8. Setting user information

Set the user information for the CPU modules in both systems and the project. (📖GX Works3 Operating Manual)

9. Creating programs

Create a safety program and standard program. (☞ Page 87 RUNNING A PROGRAM)

10. Writing data to the programmable controller

Write the set parameters and created program to the CPU modules of both systems using the engineering tool. (☞ Page 55 Writing Data to the Programmable Controller)

11. Resetting the CPU modules

Restart both systems in either of the following ways.

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

Point

For starting up only one system, perform the following operation on the engineering tool to start up the system as the control system within the time period set in "Other system Start-up Timeout Setting".

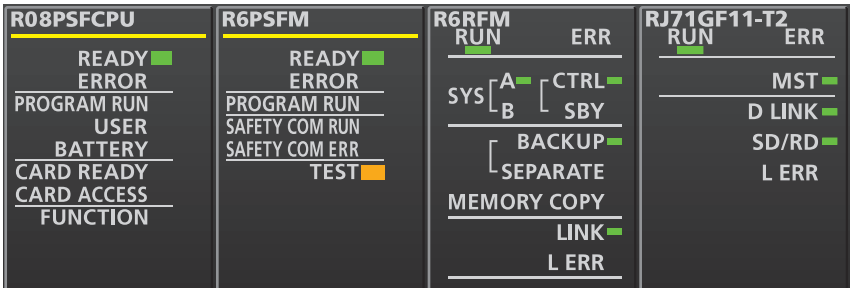
[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]

Select "Forced Start of Control System while Waiting for Other System to Start" and click the [Execute] button. (When "Other system Start-up Timeout Setting" of "CPU Parameter" has not been changed, a stop error occurs in 60 seconds.)

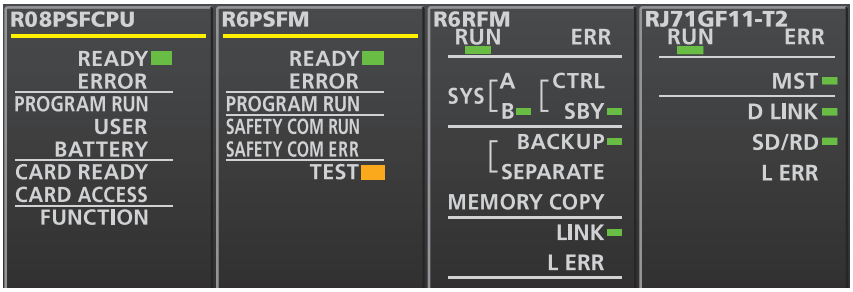
12. Checking the LEDs on the CPU module side

Check that the LEDs of each module are in the following states. The CARD READY LED status depends on whether an SD memory card has been inserted to each CPU module or not.

- Control system



- Standby system*1



*1 In a redundant master station system, the MST LED of the master/local module in the standby system flashes.

Point

When one system has been started up, a continuation error occurs since only the control system has been started up. Thus, the ERROR LED of the CPU module and the L ERR LED of the redundant function module turn on.

In addition, the BACKUP LED of the redundant function module flashes because a system switching disable cause exists.

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

- CPU module: ERROR LED (☞ Page 375 LED status of the CPU module)
- SIL2 function module: ERROR LED (☞ Page 375 LED status of the CPU module)
- Redundant function module: ERR LED, L ERR LED (☞ Page 378 When the ERR LED turns on or flashes, ☞ Page 378 When the L ERR LED turns on)
- Master/local module: ERR LED, L ERR LED (MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup))

Enabling the modules

1. Safety module operation

Check that the systems on the CPU module side and remote head module side are powered on, and then enable the modules set to SIL2 mode with the "Safety Module Operation" of the engineering tool. (📖Manual for the intelligent function module used, 📖Manual for the I/O module used)

2. Powering off the systems

After enabling the modules, power off the systems on the CPU module side and remote head module side.

3. Restarting the systems

Set the RUN/STOP/RESET switch of the CPU module and remote head module in both systems to the RUN position and power on the systems.

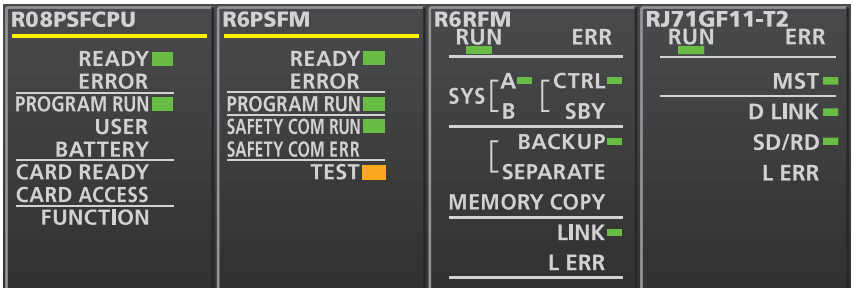
Procedure for checking the operation

1. Checking the operation

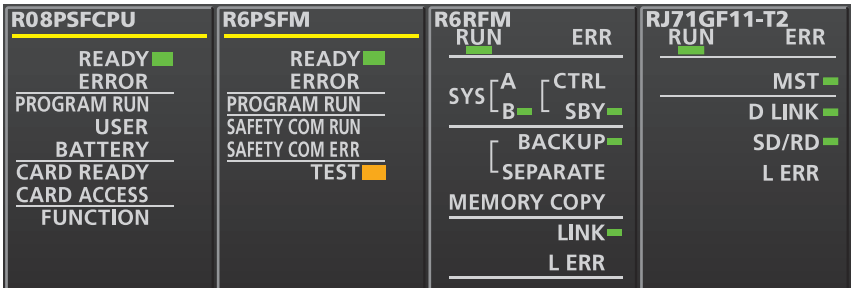
Check each module status and program operation used in the systems.

- Check if an error has occurred in each module.
- Check that the LEDs of each module are in the following states.

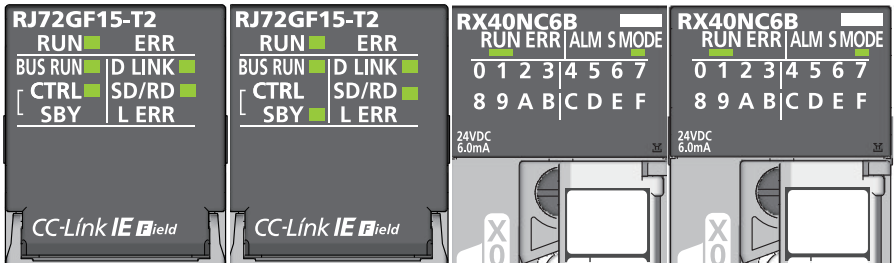
• Control system



• Standby system*1



• Remote head module side*2*3



*1 In a redundant master station system, the MST LED of the master/local module in the standby system flashes.

*2 In a redundant master station system, the following LEDs are always off since a redundant configuration is not made for the remote head module.
· CTRL LED
· SBY LED

*3 The input module with diagnostic functions is used as an example. For the LEDs of other modules, refer to the manual for the module used.

- Check if an error occurs in the CC-Link IE Field Network diagnostics. (MELSEC iQ-R CC-Link IE Field Network User's Manual (Application))
- Check the operation of safety programs and standard programs.

2. Switching the safety operation mode

Switch the safety operation mode to SAFETY MODE to run the system normally as a system using the CPU module. Set the CPU module to the STOP state before switching the mode. (Page 262 SWITCHING SAFETY OPERATION MODE)

Point

Setting the CPU module to the STOP state causes a continuation error due to the operating status mismatch and the error is detected in the standby system.

3. Executing a program

Power off the CPU modules and remote head module in both systems. After powering off, set the RUN/STOP/RESET switch of the CPU module and remote head module in both systems to the RUN position and power on the systems. Check that the PROGRAM RUN LED on the CPU module in the control system turns on.

Point

If the RUN/STOP/RESET switch of each CPU module is set to the RUN position while the CPU modules are powered on, a continuation error due to an operating status mismatch is detected in the CPU module of the standby system.

4. Monitoring the program

Check that the program operates normally using the engineering tool.

Starting up the standby system

Point

When only the control system is started up first, start up the standby system subsequently.
When both systems are started up simultaneously, this procedure is not required.

Start up the standby system while the control system keeps operating.
Perform the same procedures as those for starting up the control system.
☞ Page 38 Installing/mounting/inserting procedure
☞ Page 38 Wiring procedure
Perform step 1, 3, and 4 for the following procedure.
☞ Page 40 Procedure for the CPU module side

1. Powering off the system

Power off the standby system.

2. Connecting the redundant function modules

Connect the redundant function modules of both systems with tracking cables. (☞ Page 49 Redundant function modules)

3. Starting up the system

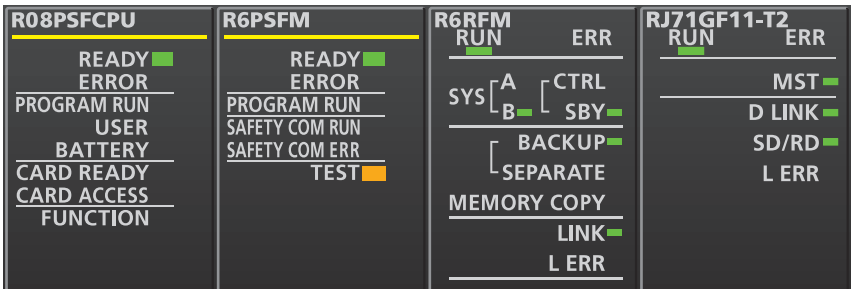
Power on the standby system.
Connect the personal computer to the control system, and execute the memory copy function (from the control system to the standby system) using the engineering tool by following the procedure below. (☞ Page 242 Memory Copy from Control System to Standby System)

[Online] ⇨ [Redundant PLC Operation] ⇨ [Redundant Operation]
Select "Memory Copy" and click the [Execute] button.
Upon completion of the memory copy function, the MEMORY COPY LED of the redundant function module in the standby system turns on.
After that, reset the CPU module of the standby system and set it to the RUN state.

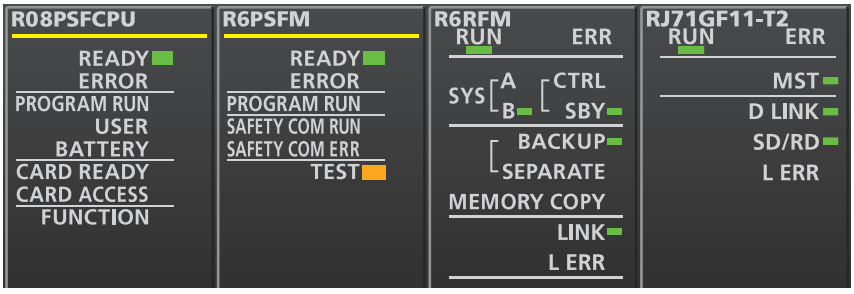
4. Checking the LEDs on the CPU module side

Check that the LEDs of each module are in the following states. The CARD READY LED status depends on whether an SD memory card has been inserted to each CPU module or not.

• Control system



• Standby system



5. Clearing errors

If the ERROR LED of the CPU module in the control system is on, clear the error using the engineering tool. (☞ Page 184 Check method using the engineering tool)

6. Checking the operation

☞ Page 42 Procedure for checking the operation

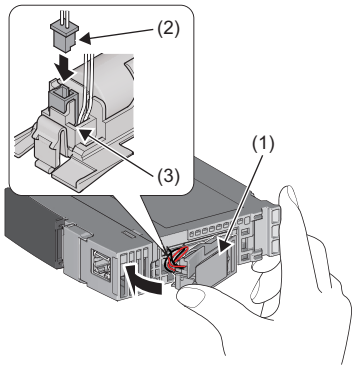
5.2 Installing a Battery

Install a battery to the CPU module.

Installation procedure

Q6BAT

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



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

■Precautions

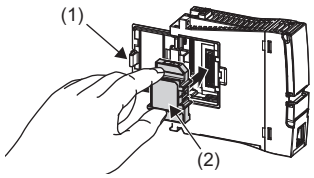
- When using the CPU module with the battery removed, check that the jack (3) of the CPU module is inserted in the socket on the battery cover located on the bottom of the CPU module.
- Do not drop or apply strong shock to the module and the battery.

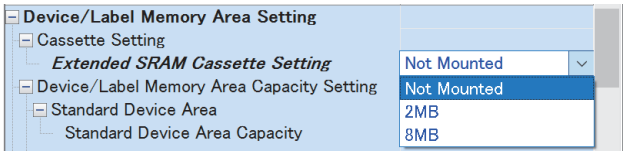
5.3 Inserting or Removing an Extended SRAM Cassette

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

Insertion procedure

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





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

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

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

Precautions

- The availability of the extended SRAM cassette differs depending on the CPU module used. For the availability, refer to the performance specifications of the extended SRAM cassette. (☞ Page 68 Extended SRAM Cassette)
- When the extended SRAM cassette is inserted or removed, all the data such as devices and the file register stored in the device/label memory are erased. Back up the data in the programmable controller before replacing the cassette.
- If the capacity of the extended SRAM cassette differs before and after the replacement, the ERROR LED of the CPU module may flash. But, it is not an error. Change the capacity setting in the CPU parameters. (Refer to step 5 above.)

Restriction

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

Removal procedure

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

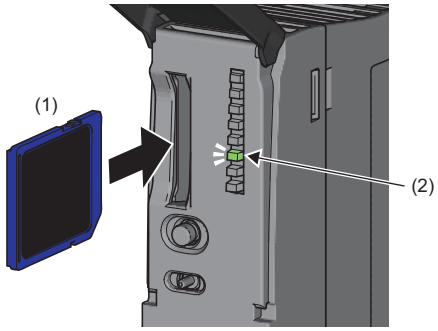
1. Read the data on the device/label memory from the CPU module, and save it in advance using the engineering tool. (When the extended SRAM cassette is removed, all of the data on the device/label memory are erased.)
2. Power off the programmable controller.
3. Remove the CPU module from the base unit,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

5.4 Inserting and Removing an SD Memory Card

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

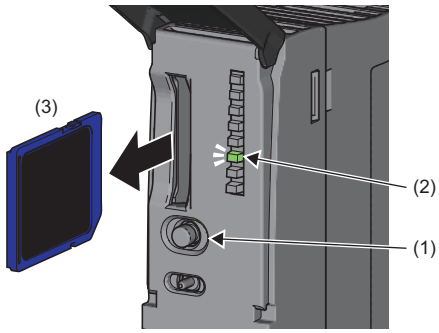
Insertion procedure

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



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

Removal procedure



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

*1 SM606 (SD memory card forced disable instruction) can also disable access to an SD memory card. For details, refer to the following.
☞ Page 483 Drive information

Precautions

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

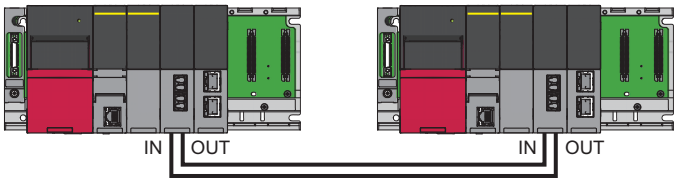
5.5 Wiring

Redundant function modules

This section describes the wiring to redundant function modules.

Wiring method

Connect the tracking cables from the OUT connector of a redundant function module to the IN connector of the other redundant function module.



Point

For the specifications of the tracking cables connected to redundant function modules, refer to the following.

☞ Page 69 Redundant Function Module

Connecting/disconnecting tracking cables

■Connection procedure

1. Pay attention to the orientation of the tracking cable connector and insert the connector to a redundant function module until it clicks. A redundant function module has one IN connector and one OUT connector. Connect the IN connector of system A and the OUT connector of system B, and connect the OUT connector of system A and the IN connector of system B.
2. Pull each cable lightly and check that it has been connected securely.

■Disconnection procedure

1. Disconnect the tracking cables while pressing the connector hook.

Precautions

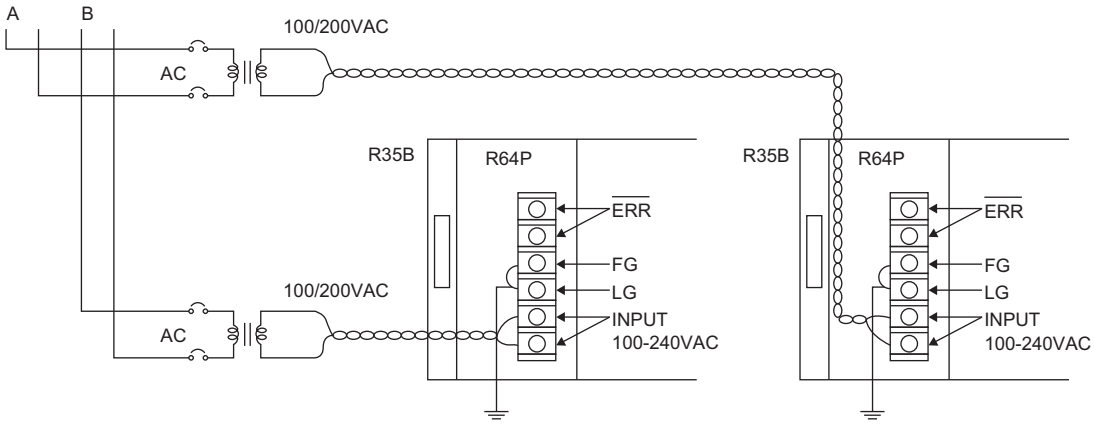
- There are restrictions on the cable bending radius. For details, refer to the specifications of the tracking cables used.
- Place the cables in a duct or clamp them. If not, dangling cables may swing or inadvertently be pulled, resulting in damage to the module or cables or malfunction due to poor contact.
- When connecting tracking cables, pay attention not to touch optical fiber cores of the connectors on the cables and module sides and prevent dirt and dust from adhering to them. If oil on hands, dirt, and dust adhere to the optical fiber cores, the transmission loss increases and tracking may not work properly.
- Hold the connector of the tracking cables to connect or disconnect the cable. Pulling the cable connected to the module may result in malfunction or damage to the module or cable or malfunction due to poor contact.

Wiring to power supply modules

This section describes the wiring to power supply modules.
The terminal block of each power supply module has a screw size of M4. Wire cables to the terminal block with the applicable solderless terminal RAV1.25-4 or RAV2-4.
Separately supply power to the system A and B.

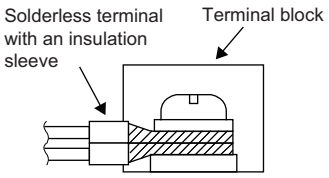
Wiring example

The following figure shows a wiring example of power cables to each main base unit and ground cables.
For wiring examples for each power supply module, refer to the manual included with the power supply module. (📖 Before Using the Product)



Point

- 100VAC, 200VAC and 24VDC wires must be twisted starting from the terminal connected, and connect modules at the shortest distance. Also, use the thickest wire (maximum 2㎟) to reduce the voltage drop.
- For the wiring to a terminal block, use a solderless terminal.
- To prevent the short-circuit because of loosening screws, use the solderless terminal with an insulation sleeve of 0.8mm or less. Note that up to two solderless terminals can be connected per terminal block.

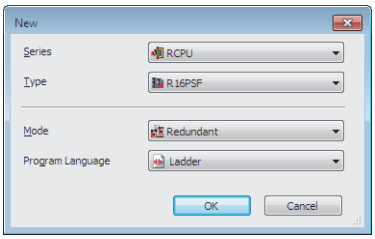


- Ground the LG and FG terminals after short-circuiting them. Failure to do so may cause the terminals to be susceptible to noise. The LG terminal has a half potential of the input voltage.
- When two redundant power supply modules operate in parallel as a redundant power supply system, it is recommended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

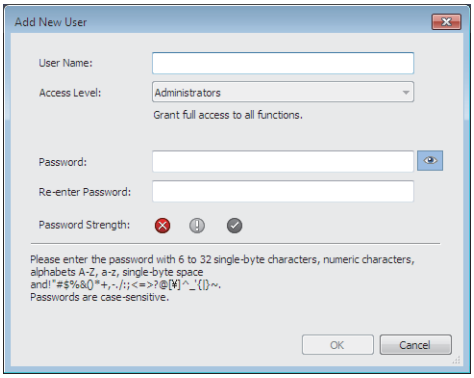
5.6 Creating a Project

Activate the engineering tool and create a project.

[Project] ⇨ [New]



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



2. When creating a new project, registration of user information for the user authentication function is required. Set a user name and password.
(GX Works3 Operating Manual)

Point

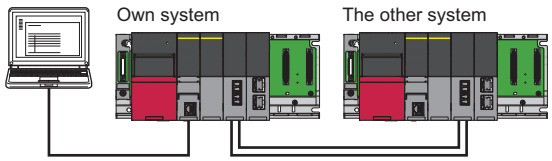
Users with an access level of "Developers", "Users", and "Assistant Developers" can be added. For details on the access level, refer to the following.
GX Works3 Operating Manual

5.7 Connecting a Personal Computer

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

Procedure

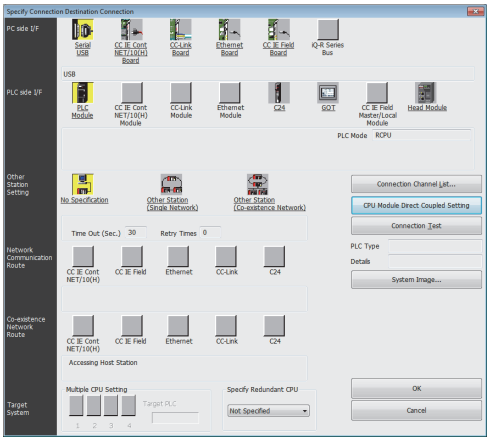
Connect a personal computer directly to the CPU module, following the procedure below.



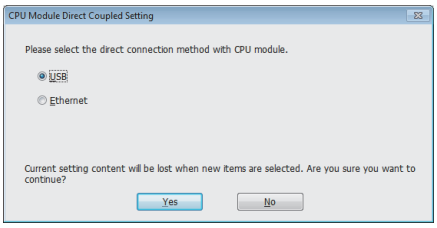
1. Connect a personal computer to the CPU module using a USB cable or Ethernet cable. This CPU module becomes the own system.

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

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



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



5. Set "Specify Redundant CPU" to "Not Specified" on the "Specify Connection Destination Connection" window and click the [Connection Test] button to check that the personal computer has been connected to the CPU module.

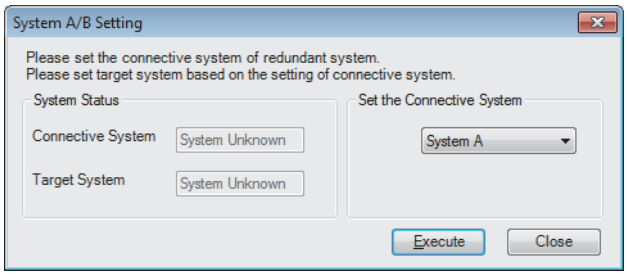
Install a USB driver to use a USB cable for the first time. (GX Works3 Operating Manual)

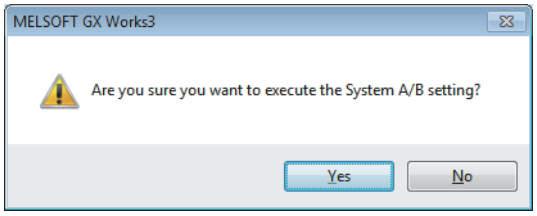
5.8 Setting the System (System A or System B)

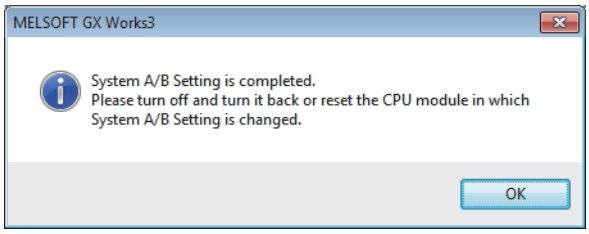
Set the system A or B using the engineering tool and write the system settings to the CPU module.

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

Setting procedure







1. Set the CPU module to the STOP state.
2. In the "System A/B Setting" window, select system A or system B for the own system.
3. Click the [Execute] button.
4. Click the [Yes] button on the window shown on the left. Check that the SYS A LED or SYS B LED of the redundant function module flashes in accordance with the setting.
5. To change the system A/B setting, power off and on the system or reset the CPU module, and then click the [OK] button. Check that the SYS A LED or SYS B LED of the redundant function module turns on in accordance with the new setting.

Point

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

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

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

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

Checking method

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

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

Point

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

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

Precautions

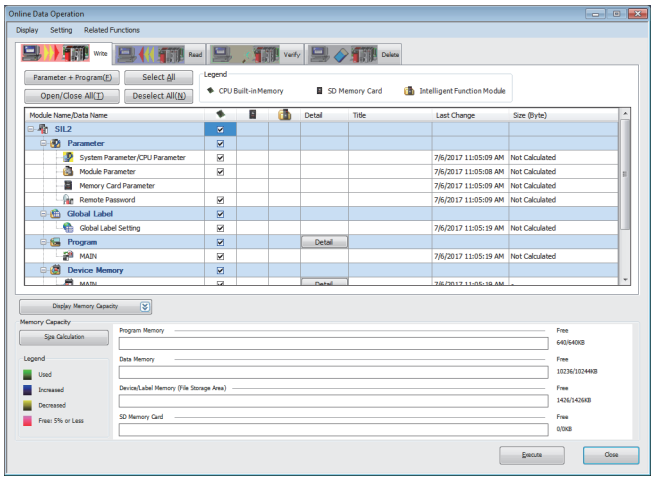
- Do not power off the system or reset the CPU module during the system A/B setting. The system A/B setting may not be reflected properly. If not reflected properly, set the system again.
- For the system A/B setting, use the CPU module that is connected with a USB cable or an Ethernet cable. If the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

5.9 Writing Data to the Programmable Controller

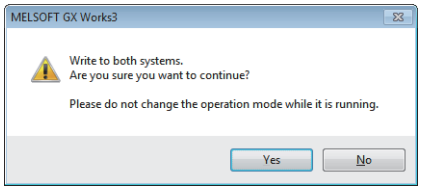
Write the set parameters and the created program to the CPU module.

[Online] ⇨ [Write to PLC]

Procedure



1. Select system parameters, CPU parameters (standard/safety), module parameters, and program files (standard/safety) on the "Online Data Operation" window. When FBs are used, select the corresponding FB/FUN files (standard/safety).
2. Click the [Execute] button.



3. Click the [Yes] button. *1
4. After the write processing completes, click the [Close] button.

*1 When one of both systems is started up first, the window asking for writing the data to only the connected system appears.

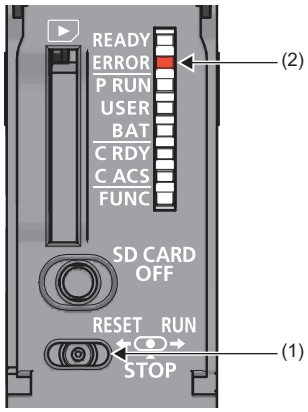
Point

- To operate CPU modules, write system parameters, CPU parameters (standard/safety), and program files (standard/safety). To operate I/O modules and intelligent function modules, write module parameters or module extension parameters.
- When a parameter setting is changed, reset the CPU module.
- Use of the [Select Favorites] button enables users to easily select frequently-used files, such as the system parameter file, CPU parameter file, and program file. Register items as favorites on the window opened by selecting [Setting] ⇨ [Register Favorites Selection] on the menu bar.

5.10 Resetting the CPU Module

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

Procedure



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

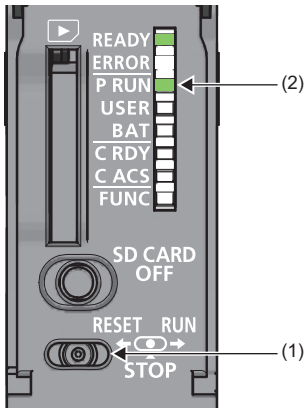
Point

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

5.11 Executing the Program

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

Procedure



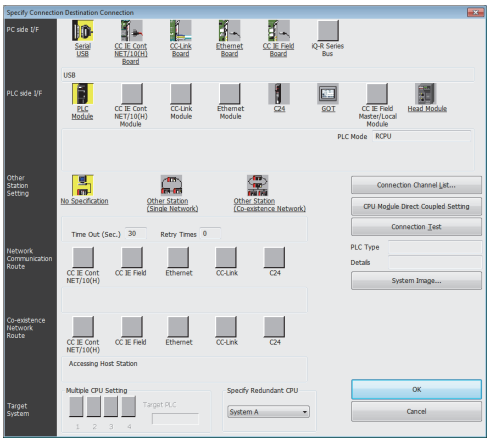
1. Set the RUN/STOP/RESET switch (1) to the RUN position.
2. Check that the PROGRAM RUN LED (P RUN) (2) turns on.

5.12 Monitoring the Program

Monitor the program operation using the engineering tool.

Change the connection destination with the engineering tool and check the operating status of the system A or B.

[Online] ⇨ [Current Connection Destination]



1. Select a system in "Specify Redundant CPU" on the "Specify Connection Destination Connection" window.
2. Click the [Connection Test] button to check whether the CPU module of the selected system has been connected.

For how to check the operation, refer to the following.

GX Works3 Operating Manual

PART 3 SYSTEM CONFIGURATION

This part consists of the following chapter.

6 SYSTEM CONFIGURATION

6 SYSTEM CONFIGURATION

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

MELSEC iQ-R Module Configuration Manual

Combinations of the SIL2 Process CPU/SIL2 function module and other modules

When using the SIL2 Process CPU/SIL2 function module together with other modules, ensure that all the modules meet Class Ⅲ or the SELV.

MEMO

MEMO

PART 4 SPECIFICATIONS

This part consists of the following chapter.

7 PERFORMANCE SPECIFICATIONS

7 PERFORMANCE SPECIFICATIONS

7.1 CPU Module

This section describes the specifications of the CPU module.

Hardware specifications

Item			R08PSF	R16PSF	R32PSF	R120PSF
Operation control method			Stored program cyclic operation			
I/O control mode			Refresh mode (The direct access input/output is available by specifying the direct access input/output (DX, DY).)			
Instruction processing time	LD instruction		0.98ns			
	MOV instruction		1.96ns			
Memory capacity	Program capacity		80K steps (320K bytes) (For safety programs: 40K steps (160K bytes))	160K steps (640K bytes) (For safety programs: 40K steps (160K bytes))	320K steps (1280K bytes) (For safety programs: 40K steps (160K bytes))	1200K steps (4800K bytes) (For safety programs: 40K steps (160K bytes))
	Program memory		320K bytes (For safety programs: 160K bytes)	640K bytes (For safety programs: 160K bytes)	1280K bytes (For safety programs: 160K bytes)	4800K bytes (For safety programs: 160K bytes)
	Device/label memory[*1]		1178K bytes	1710K bytes	2306K bytes	3370K bytes
	Signal flow memory	Area for standard/ safety programs	10K bytes	20K bytes	40K bytes	150K bytes
		Area for standard/ safety function blocks	256K bytes			
	Data memory		5M bytes	10M bytes	20M bytes	40M bytes
	CPU buffer memory		1024K bytes (512K words) (including the built-in function information area capacity 4M bytes (2K words))			
	Refresh memory		2048K bytes[*2]			
Number of storable files[*5]	Program memory (P: number of program files, FB: number of FB files)		380 (including safety programs) (P: 252, FB: 128 (One FB file can store 64 function blocks.))			
	Program memory (P: number of safety program files, FB: number of safety FB files)		48 (P: 32, FB: 16 (One FB file can store 64 function blocks.))			
	Device/label memory (file storage area)		323 (with or without an extended SRAM cassette)[*3]			
	Data memory		512[*4]			
	SD memory card		NZ1MEM-2GBSD: 256[*4] NZ1MEM-4GBSD, NZ1MEM-8GBSD, NZ1MEM-16GBSD: 32767[*4]			
Number of storable folders[*5]	Data memory		512[*4]			
	SD memory card		NZ1MEM-2GBSD: 256[*4] NZ1MEM-4GBSD, NZ1MEM-8GBSD, NZ1MEM-16GBSD: 32767[*4]			
USB port			USB2.0 High Speed (miniB) × 1			
Ethernet port			Refer to the following. MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)			
Clock function			Year, month, date, hour, minute, second, and day of the week (automatic leap year adjustment) -1.00 to +1.00s/d at 0 to 55℃			
Allowable momentary power failure time			The time differs depending on the power supply module used. (MELSEC iQ-R Module Configuration Manual)			
Internal current consumption (5VDC)			0.76A			
External dimensions	Height		106mm (Base unit mounting side: 98mm)			
	Width		27.8mm			
	Depth		110mm			

Item	R08PSF	R16PSF	R32PSF	R120PSF
Weight	0.20kg			

*1 The capacity of device area, label area, latch label area, and file storage area can be changed in parameter. The capacity of the device/label memory can be increased by inserting an extended SRAM cassette. (☞ Page 148 Device/label memory area setting)

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

*3 System files are included.

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

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

Point

Wait for five seconds or longer after power-off and power on the system again. If the interval between the power-off and the power-on is short, the module may not start up.

Programming specifications

Item			R08PSF	R16PSF	R32PSF	R120PSF
Programming language			• Ladder diagram (LD) • Structured text (ST)*5 • Function block diagram (FBD/LD)*5			
Programming supporting function			• Function block (FB) • Label programming (local/global)			
Program operation	Execution type	Standard program	• Initial execution type • Scan execution type • Fixed scan execution type • Event execution type • Standby type			
		Safety program	• Fixed scan execution type			
	Type of interrupt	Standard program	• Interrupt using the internal timer (I28 to I31) • Interrupt by a module			
Number of executable programs		Standard program	252 (including safety programs)			
		Safety program	32			
Number of FB files		FB file	128, including the number of safety FB files (One FB file can store 64 function blocks.)			
		Safety FB file	16 (One safety FB file can store 64 function blocks.)			
Tact performance	Constant scan		0.2 to 2000ms (The value can be set in increments of 0.1ms.)			
	Fixed scan interrupt	Interrupt using the internal timer (I28 to I31)	0.5 to 1000ms (The value can be set in increments of 0.5ms.)			
Timer performance	Low-speed timer		1 to 1000ms (Default: 100ms)			
	High-speed timer		0.01 to 100ms (Default: 10ms)			
	Long timer*5		0.001 to 1000ms (Default: 0.001ms)			
Number of I/O points			4096 points*1			
Number of user device points (default)	Input (X)*5		12288 points (fixed)*2			
	Output (Y)*5		12288 points (fixed)*2			
	Internal relay (M)*5		12288 points (user-changeable)*3			
	Link relay (B)*5		8192 points (user-changeable)*3			
	Link special relay (SB)*5		2048 points (user-changeable)*3			
	Annunciator (F)*5		2048 points (user-changeable)*3			
	Edge relay (V)*5		2048 points (user-changeable)*3			
	Timer device	Timer (T)*5	1024 points (user-changeable)*3			
		Long timer (LT)*5	1024 points (user-changeable)*3			
	Retentive timer device	Retentive timer (ST)*5	0 points (user-changeable)*3			
		Long retentive timer (LST)*5	0 points (user-changeable)*3			
	Counter device	Counter (C)*5	512 points (user-changeable)*3			
		Long counter (LC)*5	512 points (user-changeable)*3			
	Data register (D)*5		18432 points (user-changeable)*3			
	Link register (W)*5		8192 points (user-changeable)*3			
	Link special register (SW)*5		2048 points (user-changeable)*3			
	Latch relay (L)*5		8192 points (user-changeable)*3			
Number of safety user device points (default)	Safety input (SA\X)*6		8192 points (user-selectable to either 8192 or 12288 points)			
	Safety output (SA\Y)*6		8192 points (user-selectable to either 8192 or 12288 points)			
	Safety internal relay (SA\M)*6		6144 points (user-changeable)*3			
	Safety link relay (SA\B)*6		4096 points (user-changeable)*3			
	Safety timer (SA\T)*6		512 points (user-changeable)*3			
	Safety retentive timer (SA\ST)*6		0 points (user-changeable)*3			
	Safety counter (SA\C)*6		512 points (user-changeable)*3			
	Safety data register (SA\D)*6		12288 points (user-changeable)*3			
	Safety link register (SA\W)*6		4096 points (user-changeable)*3			

Item		R08PSF	R16PSF	R32PSF	R120PSF
Number of system device points	Special relay (SM)*5	4096 points (fixed)			
	Special register (SD)*5	4096 points (fixed)			
	Function input (FX)*5	16 points (fixed)			
	Function output (FY)*5	16 points (fixed)			
	Function register (FD)*5	5 points × 4 words (fixed)			
Number of safety system device points	Safety special relay (SA\SM)*6	4096 points (fixed)			
	Safety special register (SA\SD)*6	4096 points (fixed)			
Number of file register points (default)	File register (R/ZR)*5	0 points (user-changeable)*3			
Number of index register points (default)	Index register (Z)*5	20 points (user-changeable, up to 24 points)			
	Long index register (LZ)*5	2 points (user-changeable, up to 12 points)			
Number of pointer points	Pointer (P)*5 (global/local) (default)	8192 points (user-changeable, up to 16384 points)			16384 points (user-changeable, up to 32768 points)
	Interrupt pointer (I)*5	1024 points (fixed)			
Number of link direct device points	Link input (J□\X□)*5	16384 points maximum*4			
	Link output (J□\Y□)*5	16384 points maximum*4			
	Link relay (J□\B□)*5	32768 points maximum*4			
	Link register (J□\W□)*5	131072 points maximum*4			
	Link special relay (J□\SB□)*5	512 points maximum*4			
	Link special register (J□\SW□)*5	512 points maximum*4			
Number of module access device points	Intelligent function module device (U□\G□)*5	268435456 points maximum*4			
Number of CPU buffer memory access device points	Buffer memory (U3E□\G□)*5	268435456 points (Max.)*4			
Number of refresh data register points (default)	Refresh data register (RD)*5	524288 points (1048576 points maximum)			
Number of nesting points	Nesting (N)	15 points			

*1 This item is used according to the I/O assignment based on the number of occupied I/O points of each module when I/O modules and intelligent function modules are mounted. (X/Y0 to FFF)(MELSEC iQ-R Module Configuration Manual)

*2 This is the number of points that can be used in programs.

*3 For the setting range, refer to the following.

☞ Page 288 Device Setting

*4 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.

*5 These devices cannot be used in safety programs.

*6 These devices cannot be used in standard programs.

7.2 Extended SRAM Cassette

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

Item		NZ2MC-2MBSE (ECC-compatible)*1	NZ2MC-8MBSE (ECC-compatible)
Capacity		2M bytes	8M bytes
Number of insertions and removals		Limited to 50 times	
External dimensions	Height	49mm	
	Width	32mm	
	Depth	18.5mm	
Weight		20g	

*1 It can be used with engineering tool version "1.065T" or later.

7.3 Redundant Function Module

This section describes the specifications of the redundant function module.

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

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

Point

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

- Type: Multimode optical fiber (GI)
- Model: QG series

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

7.4 SIL2 Function Module

This section describes the specifications of the SIL2 function module.

Hardware specifications

Item			R6PSFM
Operation control method			Stored program cyclic operation
Memory capacity	Program capacity	Safety program	40K steps (160K bytes)
	Program memory	Safety program	160K bytes
	Device/label memory	Safety program	80K bytes
Number of occupied I/O points			16 points*1
Buffer memory			4096K bytes
Allowable momentary power failure time			The time differs depending on the power supply module used. (MELSEC iQ-R Module Configuration Manual)
Internal current consumption (5VDC)			0.67A
External dimensions	Height		106mm (Base unit mounting side: 98mm)
	Width		27.8mm
	Depth		110mm
Weight			0.16kg

*1 The use of all the I/O signals is prohibited because they are used by the system. These I/O signals are not used for safety control, and thus they do not cause a malfunction of the safety control system.

Point

Wait for five seconds or longer after power-off and power on the system again. If the interval between the power-off and the power-on is short, the module may not start up.

PART 5 CPU MODULE OPERATION

This part consists of the following chapters.

8 BASIC CONCEPT

9 RUNNING A PROGRAM

10 CPU MODULE OPERATION PROCESSING

11 MEMORY CONFIGURATION OF THE CPU MODULE

8 BASIC CONCEPT

The SIL2 Process CPU and SIL2 function module comply with the following standards.

Standard	Remarks
IEC61508 Parts 1-7: 2010(SIL2)	SIL2, Type B, HFT=1

This product meets the requirements for use in applications that adhere to the following standards:

Standard
IEC 61511-1: 2016+AMD1: 2017 CSV (SIL2)

Certificate for safety standards

The certificate for the safety standards has been obtained with the combination of product versions shown below.

Product	Model	Hardware version*1	Firmware version	Software version
SIL2 Process CPU	• R08PSFCPU • R16PSFCPU • R32PSFCPU • R120PSFCPU	08	14	—
SIL2 function module	R6PSFM	05	14	—
GX Works3	SW1DND-GXW3-E	—	—	1.110Q

*1 The hardware version can be checked with the 3rd and 4th digits from the left of the production information.

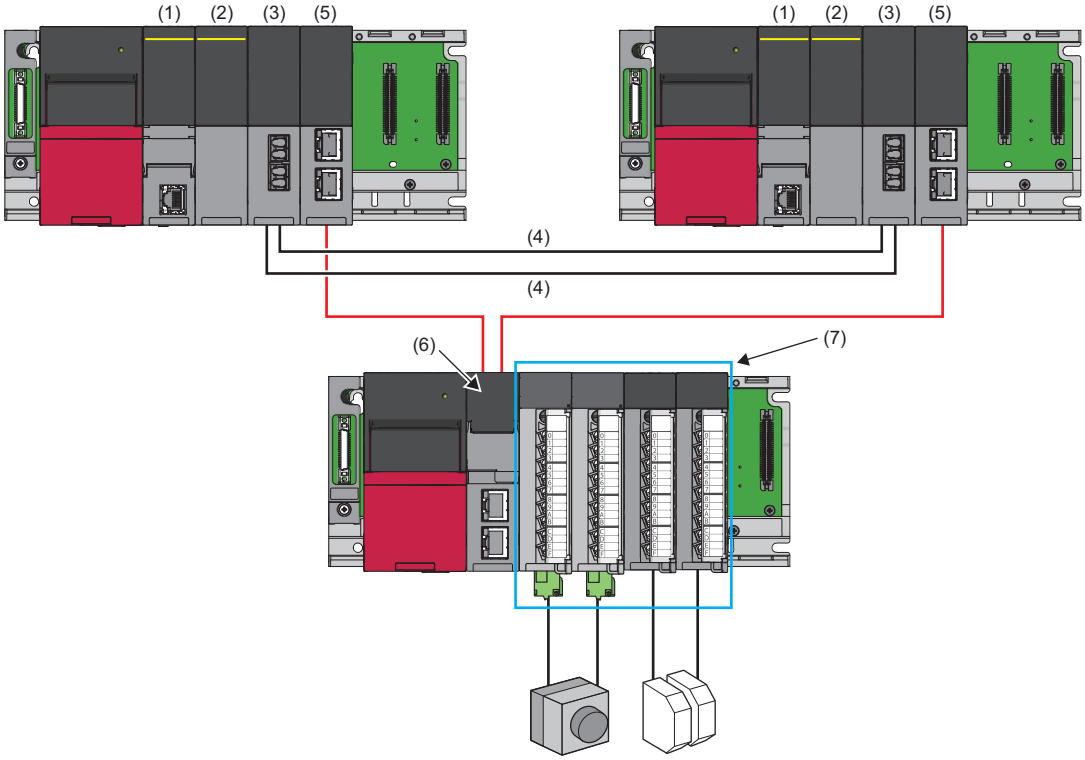
The above information is current at the time of publication. For information on the latest version that has been certified as compliant with the safety standards, please contact your local Mitsubishi Electric representative.

Use the SIL2 Process CPU with the SIL2 function module as a pair. Programs for safety control and for standard control can be executed simultaneously in one system. These products can be used to construct a safety function for general industrial machinery.
Use the SIL2 Process CPU with the redundant function module as a pair to configure a redundant system.
Mount the modules on the slots from the CPU slot to slot number 1 in the order of the SIL2 Process CPU, SIL2 function module, and redundant function module.

Point

The mounting positions and order are as described above. They cannot be changed.

I/O modules and intelligent function modules set to SIL2 mode are mounted to the intelligent device station (remote head module) on CC-Link IE Field Network.



(1) SIL2 Process CPU
(2) SIL2 function module
(3) Redundant function module
(4) Tracking cable
(5) CC-Link IE Field Network master/local module
(6) Remote head module
(7) Modules set to SIL2 mode*1

*2 For details on modules set to SIL2 mode, refer to the following.
MELSEC iQ-R Module Configuration Manual
Manuals for the I/O module or intelligent function module used

8.1 System

System A and system B

One system is specified as a system A and the other is specified as a system B to distinguish between two systems connected with tracking cables. Set the system A or B with the engineering tool. (☞ Page 53 Setting the System (System A or System B))

Point

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

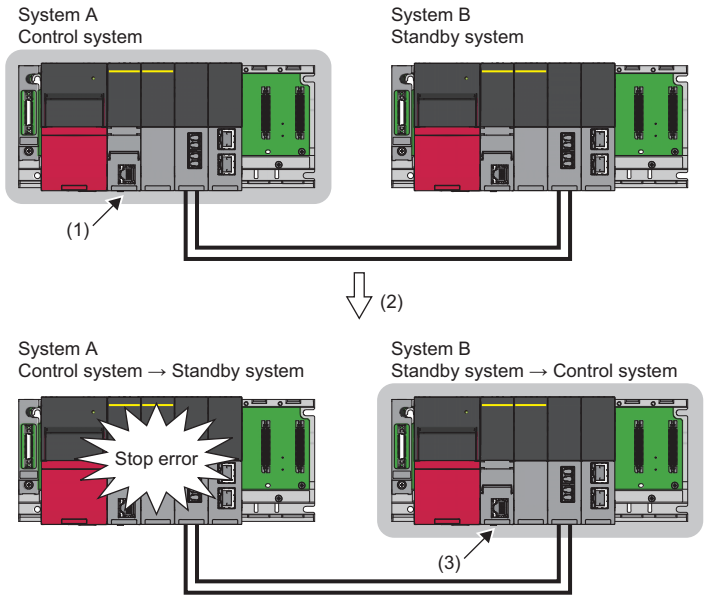
Control system and standby system

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 77 Determination of Control System/Standby System)

8.2 System Switching Between the Control System and Standby System

Data link is performed between the redundant function modules connected with tracking cables and data required for operation is transferred (tracking transfer) at every scan from the control system to the standby system. If an error occurs in the control system, the standby system will function as the new control system and continue the control using the data that the system has received.



(1) The control system is operating without errors. (Data is transferred to the standby system from the control system at every scan.)
(2) A stop error occurs on the control system.
(3) The standby system now functions as the new control system and continues the control using the data that the system has received.

For details on the system switching, refer to the following.

☞ Page 212 System Switching

8.3 System Consistency Check

Whether both systems have the same configuration is checked to switch the system and continue the operation without causing a system failure. (☞ Page 247 System Consistency Check)

8.4 Operation Mode (Mode that Determines the Operation of the CPU Module)

This mode determines the operation of the CPU module and is fixed to redundant mode[*1]. (☞ Page 51 Creating a Project)

*1 Use this mode when a redundant system is built.

8.5 Operation Modes of the System

This mode determines the operation of the system and is fixed to backup mode.

Operation mode	Description
Backup mode	When an error or failure has occurred in the control system, the standby system is switched to the control system to continue the operation. Even if a system failure has occurred in the control system, the standby system continues the operation because data is transferred from the control system to the standby system at every scan.

8.6 Safety Operation Mode

There are two safety operation modes: TEST MODE and SAFETY MODE. Running normally as a safety system or changing a safety program and safety parameters can be selected.

Safety operation mode	Description
SAFETY MODE	Mode for operating the safety system controlled by the CPU module. In this mode, safety programs and safety parameters of the CPU module cannot be changed. Only device data values in safety programs can be changed.
TEST MODE	Mode for performing maintenance (such as setting changes and tests) of the safety system controlled by the SIL2 Process CPU. In this mode, safety programs and safety parameters of the CPU module can be changed. Device data can be changed by performing a device test.
SAFETY MODE (wait-for-restart)*1	The system switches to this mode during the period of time from immediately after safety operation mode is changed from TEST MODE into SAFETY MODE until the CPU module is powered off or is reset. Even if safety operation mode is switched to SAFETY MODE, safety operation mode remains in this mode until the CPU module is powered off and on or is reset. Programs are not executed in this mode. An error occurs if the CPU module is in the RUN state.

*1 The same restrictions apply to this mode as those for SAFETY MODE.

Checking the safety operation mode

The safety operation mode can be checked in the following ways.

- SIL2 function module TEST LED (☞ Page 35 SIL2 FUNCTION MODULE)
- "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 CPU module (SA\SD205) (☞ Page 527 List of Safety Special Register Areas)

Operations restricted in SAFETY MODE

When the safety operation mode is set to SAFETY MODE, the following operations cannot be performed.

Operation		Restrictions
Writing data to the programmable controller		Data cannot be written to files relating to safety control such as safety programs or safety CPU parameters. (☞ Page 159 File operation available)
Device/label test (value changes)		Device/label tests (value changes) cannot be conducted from external devices for the following. (This includes value changes from the test function or monitor windows, and change requests from other external devices.) • Safety global devices • Safety local devices • Safety global labels • Safety local labels • Standard/safety shared labels
Memory operation		Memory operations for the CPU built-in memory (including extended SRAM cassette) cannot be performed from the engineering tool. (☞ Page 157 Memory Operation) However, SD memory card initialization is possible.
Security function	Security key authentication function	The following operations cannot be performed for the CPU module or extended SRAM cassette. • Registration • Deletion
	File password function	The following operations cannot be performed for the CPU module. • Registration • Change • Deletion

8.7 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

Whether each system is the control system or standby system is determined when the both systems are started up by powering off and on or reset of the CPU module and then ready for tracking communications. The safety operation mode (TEST MODE or SAFETY MODE) of the system does not affect the determination of the system types.

■When both systems are started up simultaneously

The system A operates as the control system and the system B as the standby system.

"Simultaneously" here means that one system is started up within three seconds after the other system is started.

■When one of the system A and system B is started up first

The system will wait for the start-up of the other system in three seconds. (☞ Page 79 The system waiting for the start-up of the other system)

When communications for both systems are established, system A operates as the control system and system B as the standby system.

Checking method

Check the LED of each redundant function module to check the status of the control system/standby system.

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

Point

Users can check the status of the control system/standby system with the engineering tool. (GX Works3 Operating Manual)

- System monitor
- Monitor status bar

Precautions

■If the system A/B settings have not been set for both systems or the same system setting has been set for both systems

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

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

If both systems are restarted simultaneously in a system where the system A operates as the standby system due to a stop error and the system B normally operates as the control system, a stop error may occur on both systems.
If the stop error cause of the system A is a program error, a stop error occurs on the system A again after the systems are restarted. Even if the system B normally operates as the control system before the restart, a stop error also occurs on the system B if a mismatch between files is detected in the system consistency check after the restart. As a result, a stop error occurs on both systems.
In this case, eliminate the cause of the error that has occurred on the system A and restart and restore both systems.

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

Do not power off the CPU module in the other system.
The system may start up without checking the system consistency even when the conditions between the both systems do not match. (☞ Page 248 Execution timing)

■When the configurations between the both systems do not match

When the configurations (such as insertion of an SD memory card) between the both systems do not match, it may take time to start communications between the both systems.
Check that the both system configurations match in the process of starting up both systems at a time, and then start up a system. (☞ Page 38 START-UP PROCEDURE)

■Using an SD memory card

Do not power off or reset the CPU module, or remove an SD memory card during an access to the SD memory card. A stop error may occur on both systems and the systems may not be started as a system.
If the CPU module is powered off or reset or an SD memory card is removed during an access to the SD memory card, the data in the SD memory card may be corrupted. In this case, diagnostics of the SD memory card (such as a file system check or restoration processing) is performed when the CPU module is powered off and on or is reset.
During diagnostics of the SD memory card, tracking communications are disabled. If diagnostics of the SD memory card on the other system are not completed in the timeout period set in "Other system Start-up Timeout Setting", a stop error occurs on the own system. If diagnostics of the SD memory card takes time on the other system, a stop error occurs on the other system as well. In this case, a stop error occurs on both systems and the systems cannot be started as a redundant system. Restart and restore both systems.

When starting up one system first

One of the system A and system B can be started up first.

How to start one system first

Perform one of the following operations to the CPU module that is waiting for the start-up of the other system. The system of the CPU module will be started as the control system.

■Online operation

Perform the following operation on the engineering tool.

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

Select "Forced Start of Control System while Waiting for Other System to Start" and click the [Execute] button.

■Switch operation

Set the RUN/STOP/RESET switch of the CPU module to the RUN, STOP, and then the RUN position again.

Set "Control System Start-up Setting (Switch Operation)" to "Enable" in the CPU parameter in advance. (☞ Page 258 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 258 Redundant System Operation Setting)

The system waiting for the start-up of the other system

When the CPU module is started up and tracking communications cannot be established with the other system, the CPU module will start waiting for the start-up of the other system in three seconds.

Both of the CTRL LED and SBY LED of the redundant function module turn off because the system of the CPU module is not the control system or standby system yet. The BACKUP LED flashes because a system switching disable cause has been generated.

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

■Operation while the system is waiting for the start-up of the other system

Even though the RUN/STOP/RESET switch of a CPU module is set to RUN while the system is waiting for the start-up of the other system, the CPU module is in the STOP state because the system has not yet been set to control system or standby system. Thus, the CPU module does not execute programs.

The following table lists the refresh operations to be performed while the system is waiting for the start-up of the other system.

Type	Operation
I/O refresh	Only the input refresh is performed. The output refresh is not performed.
Network module link refresh	From the network module to the CPU module and vice versa, data in SB/SW is transferred but data in the devices other than SB/SW is not.
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 258 Redundant System Operation Setting)

When a timeout occurs, a stop error will occur. To prevent occurrence of a stop error, set "Other system Start-up Timeout Setting" of "CPU Parameter" to "Not Set".

Precautions

- To start up the system that is waiting for the start-up of the other system as the control system, check that the other system is not operating as the control system.
- When tracking communications are established by connecting the other system with a tracking cable or powering on or resetting the CPU module in the other system, the CPU module that has been waiting for the start-up of the other system enters the RUN state and executes programs. Thus, always check that program execution by the CPU module does not cause any problems before powering on the other system or connecting the other system with a tracking cable.
- When the CPU module that has been waiting for the start-up of the other system becomes ready for tracking communications and a stop error has occurred in the other system, a stop error occurs in that CPU module as well. In this case, take a corrective action against the error of the other system, and turn off and on or reset the CPU module in both systems.
- When the system waits for the start-up of the other system, tracking communications cannot be made because the other system is off or either of tracking cables has problems. Check that the other system is on or tracking cables have no problems. (☞ Page 378 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 78 Using an SD memory card)

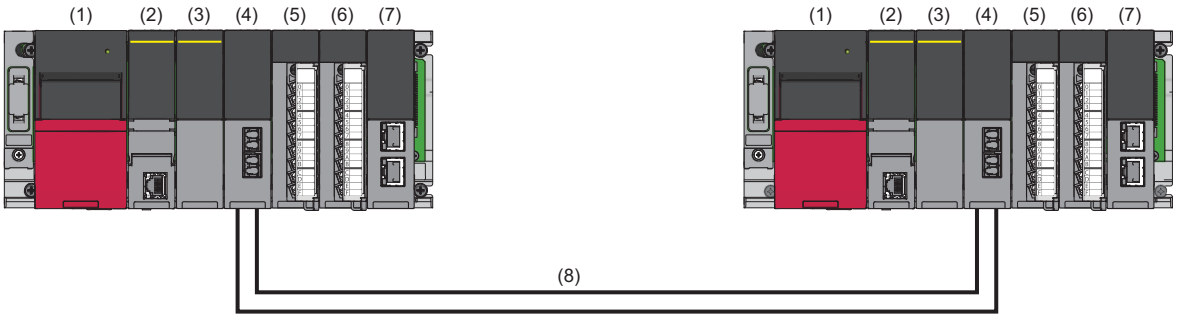
When one system is started automatically even though a tracking communication error has occurred

When the other system is powered off[*1] or there is an error with tracking cables at a system start, the CPU module enters the state that waits for the other system to start. The following shows examples, such as a system configuration and a program, to start up either of two systems using external signals without waiting for the other system to start, and prevent both systems from operating as control systems.

*1 Instead of configuring the system described below, configuring a redundant power supply system is recommended when taking a measure against failure of the power supply module in the other system.

System configuration

The following figure shows a configuration example of the system.

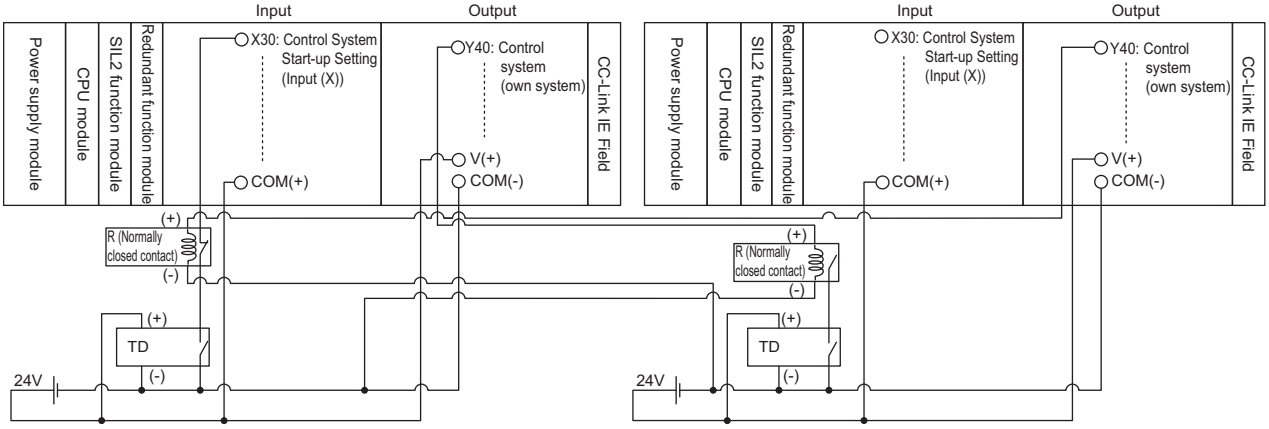


(1) Power supply module (R61P)
(2) CPU module (RnPSFCPU)
(3) SIL2 function module (R6PSFM)
(4) Redundant function module (R6RFM)
(5) Input module (RX40C7)
(6) Output module (RY40NT5P)
(7) Tracking cable
(8) CC-Link IE Field Network master/local module (RJ71GF11-T2)[*1]

*1 Starting up one system is allowed without the CC-Link IE Field Network master/local module.

Wiring example

The following figure shows a wiring example.



- Supply the 24V power using the 24V external power supply. Use the same power source as the one that supplies power to the power supply module in each system.
- TD is an on delay timer wired externally. Connect the output signal wire of the on delay timer to a relay (normally closed contact). To prevent both systems from starting up simultaneously, configure different timer settings for system A and system B.
- R is a relay (normally closed contact) wired externally. This relay connects the output signal wire of the on delay timer and the output device (Y40: Control system (own system)). The output signal wire of the relay is input to X30.

■I/O signals

The following table lists the details on the I/O signals.

Device No.	Signal name
X30	Control System Start-up Setting (Input (X)). With the timer wired externally, this bit turns on after a certain time. When the output Y of the other system is off (control system (own system)) at that time, the system starts as the control system.
Y40	Control system (own system)

■Setting time of the external on delay timer

For the external timer, with the following equation as a guide, set a longer time than the time until both systems start up so that this function (Automatic start-up at tracking communication error) is not executed when tracking communications are normally performed. In addition, set different times to system A and system B so that the times of both systems are not up simultaneously.

- Time set for the external timer[*1] = ❶ Start-up time of the CPU module (Time from power-on until RUN) + ❷ One scan time + ❸ Time lag of power-on + ❹ α + ❺ β
  - ❶ Start-up time of the CPU module: Time from when the CPU module is powered on until when the CPU module enters the RUN state when tracking communication is normal
  - ❷ One scan time: Time until when Y40 (Control system (own system)) is refreshed
  - ❸ Time lag of power-on: Time to add to the external timer of the system that has started up first when two systems start up one by one. It adjusts the activation timing of the external timer.
  - ❹ α: Margin for variation in start-up time of the CPU module. Set a sufficient margin to accommodate the variation.
  - ❺ β: Time to add to the timer of either of two systems so that the times of both systems are not up simultaneously

*1 If the time set for the external timer is shorter than the time determined by the above equation or an identical time is set to system A and system B, one system cannot recognize whether the other system has started up as the control system and both systems may start up as control systems.

Parameter settings

The following shows parameter settings.

■System parameter

Set the system parameter according to the system configuration in "I/O Assignment Setting".

Slot	Module Name	Module Status Setting	Points	Start XY
Base				
CPU	R32PSFCPU(Host Station)			3E00
0(*-0)	R6PSFM	No Setting	16 Points	0000
1(*-1)	R6RFM	No Setting	32 Points	0010
2(*-2)	RX40C7	No Setting	16 Points	0030
3(*-3)	RY40NT5P	No Setting	16 Points	0040

■CPU parameter (program settings)

Set this program example (MAIN in this example) in "Program Setting" as follows.

Execute Order	Program Name	Execution Type		Refresh Group Setting	Device/File Use or not	Both Systems Program Executions Setting
		Type	Detailed Setting Information			
1	MAIN	Scan		(Do not Set)	<Detailed Setting>	Both Systems Executions

- Set "Execution Type" to "Scan".
- Set "Both Systems Program Executions Setting" to "Both Systems Executions".

■CPU parameter (redundant settings)

Set the CPU parameter in "Redundant System Settings" as follows.

Item	Setting	
Redundant Behavior Setting		
Watching Standby System Setting	Enable	
Standby System Output Setting	Disable	
Backup Mode Setting		
Operating Status Check	Check	
Safety Operation Mode Check	Check	
Auto Memory Copy Setting	Disable	
Control/Standby System Start-up Setting		
Other system Start-up Timeout Setting	Not Set	← (1)
Time-out Period	60 Second	
Control System Start-up Setting(Switch Operation)	Disable	
Control System Start-up Setting(Input(X))	Enable	← (2)
Input(X)	X30	← (3)
Tracking Setting		
Signal Flow Memory Tracking Setting	Transfer	
Tracking Device/Label Setting	Detailed setting	← (4)
Tracking Block No.1 Auto Transfer Setting	Transfer Automatically	
Device/Label Detailed Setting	<Detailed Setting>	

(1) Set "Not Set" in "Other system Start-up Timeout Setting".
(2) Set "Enable" in "Control System Start-up Setting (Input (X))".
(3) Set "X30" in "Input (X)".
(4) Set "Detailed setting" in "Tracking Device/Label Setting".

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

Tracking Block No. 1

Total Global Device 1.5K Word

File Register Setting

File Name

No.	Device	Points (Decimal)	Start	End
1	X	12224	40	2FFF
2	Y	12208	50	2FFF

← (5)

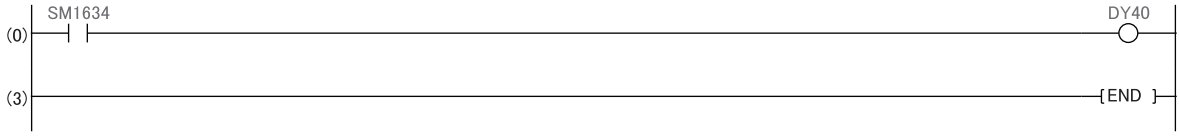
(5) Do not include X30 to X3F and Y40 to Y4F, which are used in the program example, in the tracking transfer range.

Point

- To switch enabling and disabling "Control System Start-up Setting (Input (X))", set a switch for the input to X30 so that "Control System Start-up Setting (Input (X))" is enabled only when the switch is on.
- When using this system, set the RUN/STOP/RESET switch to RUN to operate the ladder program.

Program example

The following figure shows a program example and the overview of the operation.



■Output of the control system (own system)

(0) The other system is notified of the start-up of the own system as the control system by turning on Y40 (Control system (own system)) using the direct access output when the own system operates as the control system (SM1634 is on), or by turning off Y40 when the own system does not operate as the control system.

Point

After the automatic start-up, to enable the system switching, eliminate the cause of tracking communication error and restart the CPU module in the control system or in the state that waits for the other system to start.

When starting up the previous control system as the control system

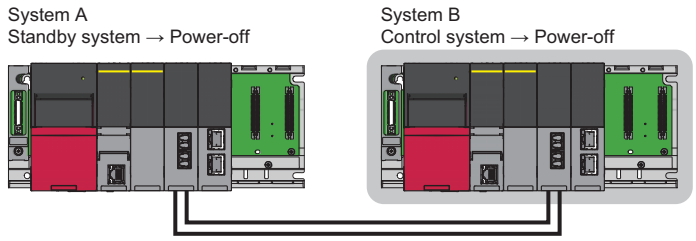
The system A is always specified as the control system when both systems are started up simultaneously. Even though both systems are temporarily powered off due to a power failure or other causes while the system B is operating as the control system, the system A is started up as the control system when both systems are powered on again.
To start up the system B that was the previous control system as the control system again, create a program that uses SM1636 (Previous control system identification flag). For a system where a network module has been mounted, execute the SP.CONTSW instruction after a network module on the other system is started up.

Program example

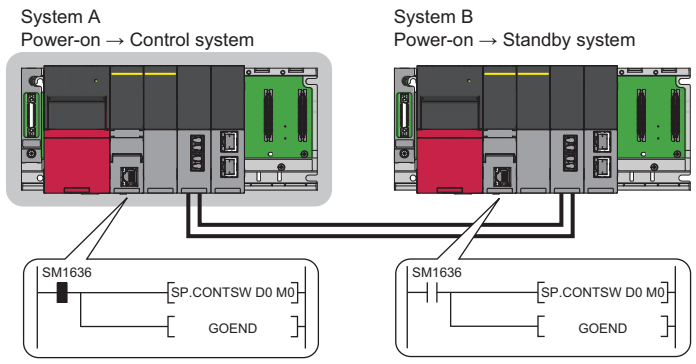
Executing a system switching instruction when the CPU module in the system A is set to the RUN state for the first time switches the system B to the control system. Turn on SM1646 (System switching by a user) in advance.



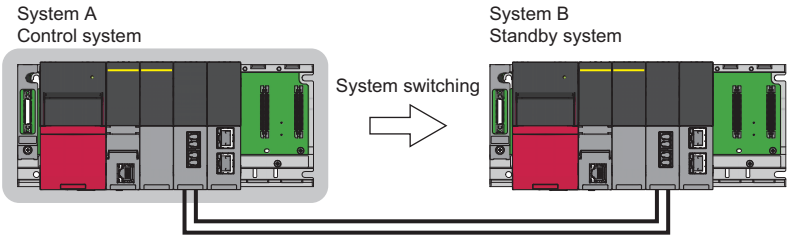
Operation example



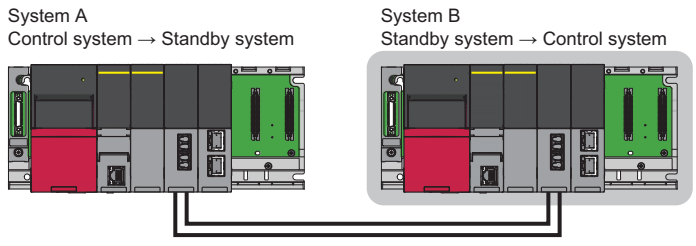
1. Both systems are temporarily powered off due to a power failure or other causes while the system B is operating as the control system.



2. The system A is started as the control system when both systems are powered on. SM1636 turns on for only one scan after the CPU module in the system A is set to the RUN state.



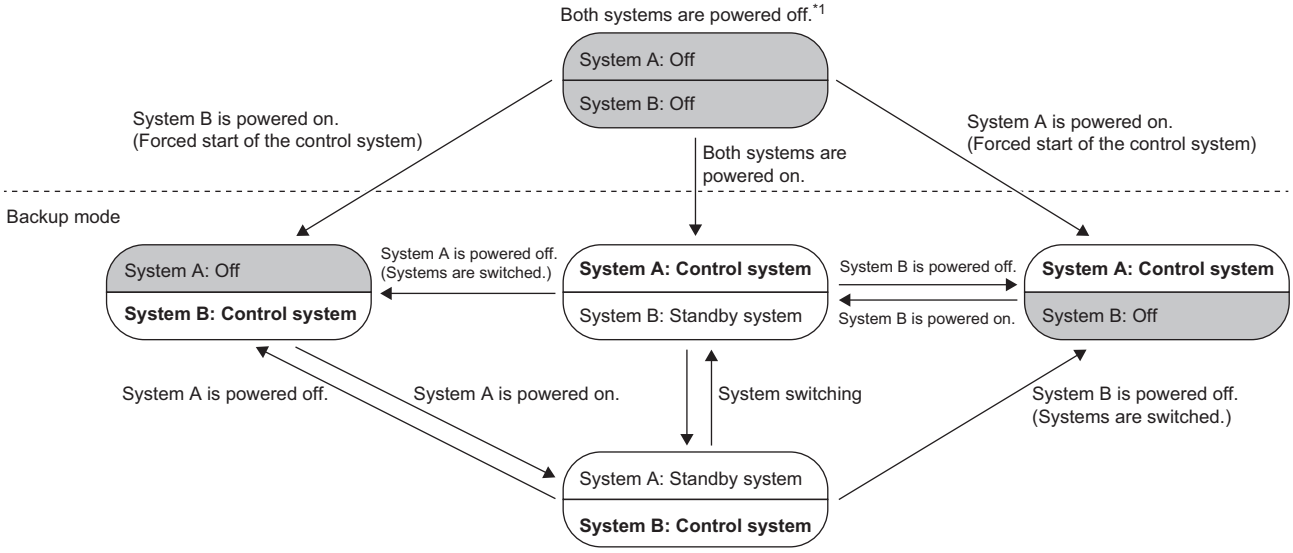
3. The system switching is performed by the SP.CONTSW instruction.



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

8.8 System State Transition After Start-up

The following figure shows the state transition of the system after start-up.

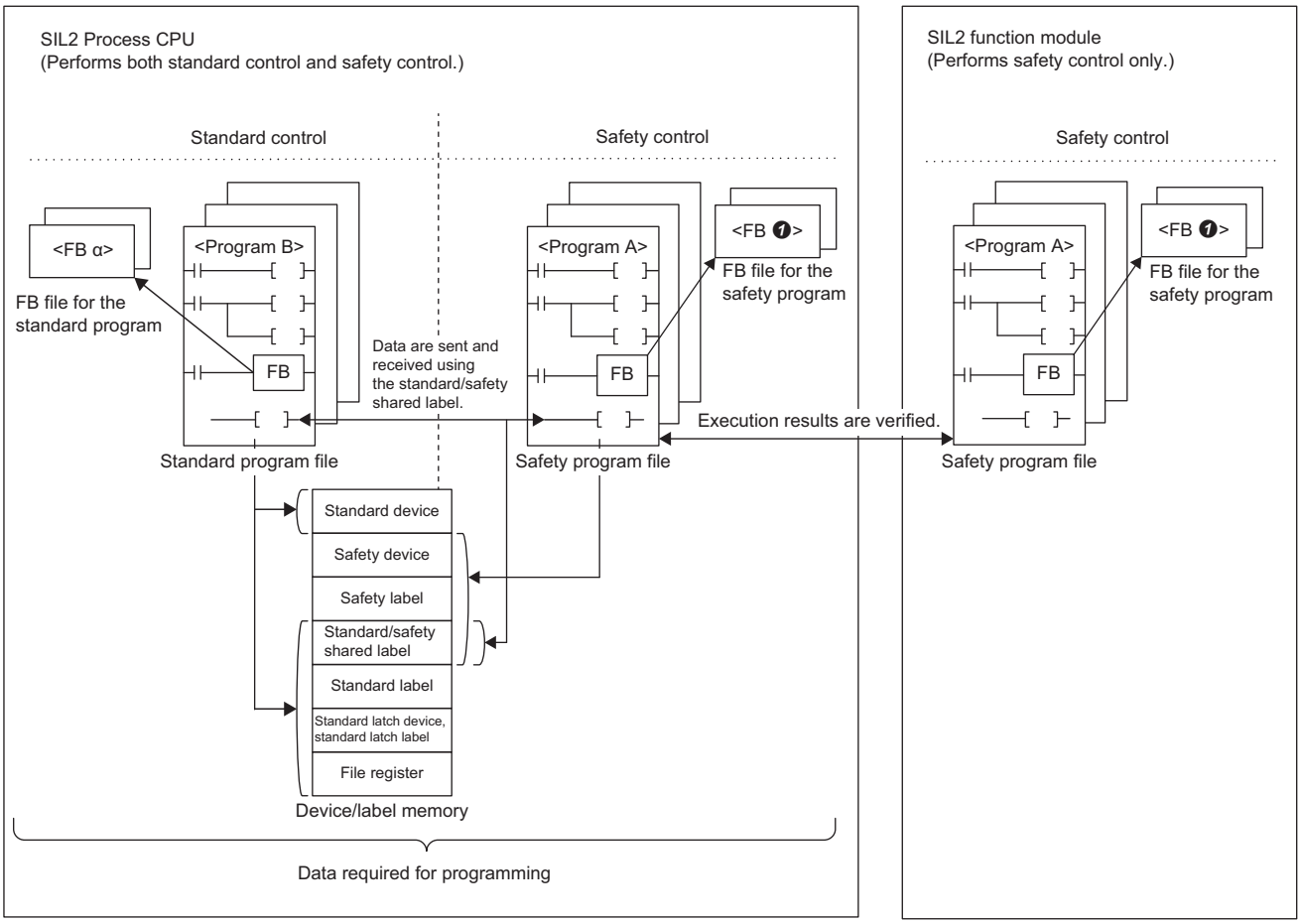


*1 When both systems are powered off, both systems enter the power-off state regardless of the system type.

9 RUNNING A PROGRAM

9.1 Overview

The CPU module executes standard programs and safety programs, and the SIL2 function module executes safety programs only. During operations, the CPU module verifies the execution results of safety programs of both the CPU module and the SIL2 function module.



Devices/labels that can be used in safety programs

Only safety devices and safety labels can be used in safety programs.

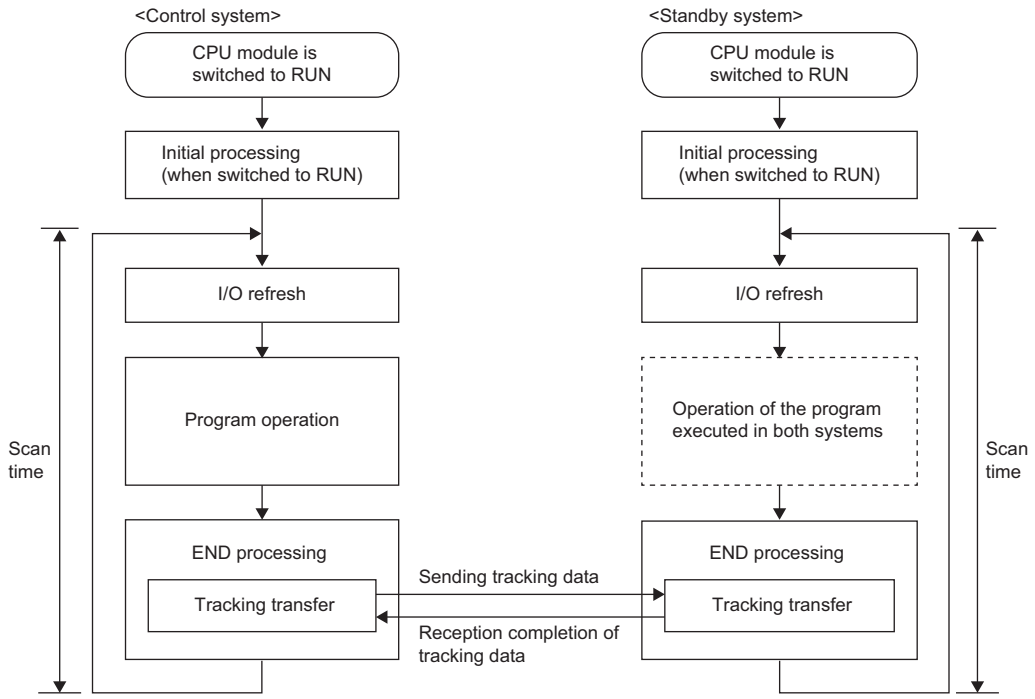
- Safety device: ☞ Page 330 SAFETY DEVICES
- Safety label: ☞ Page 350 SAFETY LABEL

9.2 Scan Configuration

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

Tracking transfer is performed during the END processing. (☞ Page 225 Tracking Transfer)

The following figure shows scan configurations of when both systems are started up simultaneously.



Both systems perform different processing.

○: Performed, ×: Not performed

Processing	Control system	Standby system
Initial processing/initial processing (when switched to RUN)	○	○
I/O refresh	○	○*1
Program operation	○	×*2
Tracking transfer	○	○
END processing	○	○

*1 Only the input refresh is performed. Whether or not to perform the output refresh depends on "Standby System Output Setting" of "CPU Parameter". (☞ Page 258 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 252 Program Execution in Both Systems)

Initial processing (when powered on or switched to RUN)

For the initial processing (when powered on or switched to RUN), the following processes are performed:
○: Performed, ×: Not performed

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

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

*2 For details on the initial label value setting, refer to the following.
☞ Page 364 LABEL INITIALIZATION FUNCTION

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

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

I/O refresh

The module performs the following before starting program operation.

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

Point

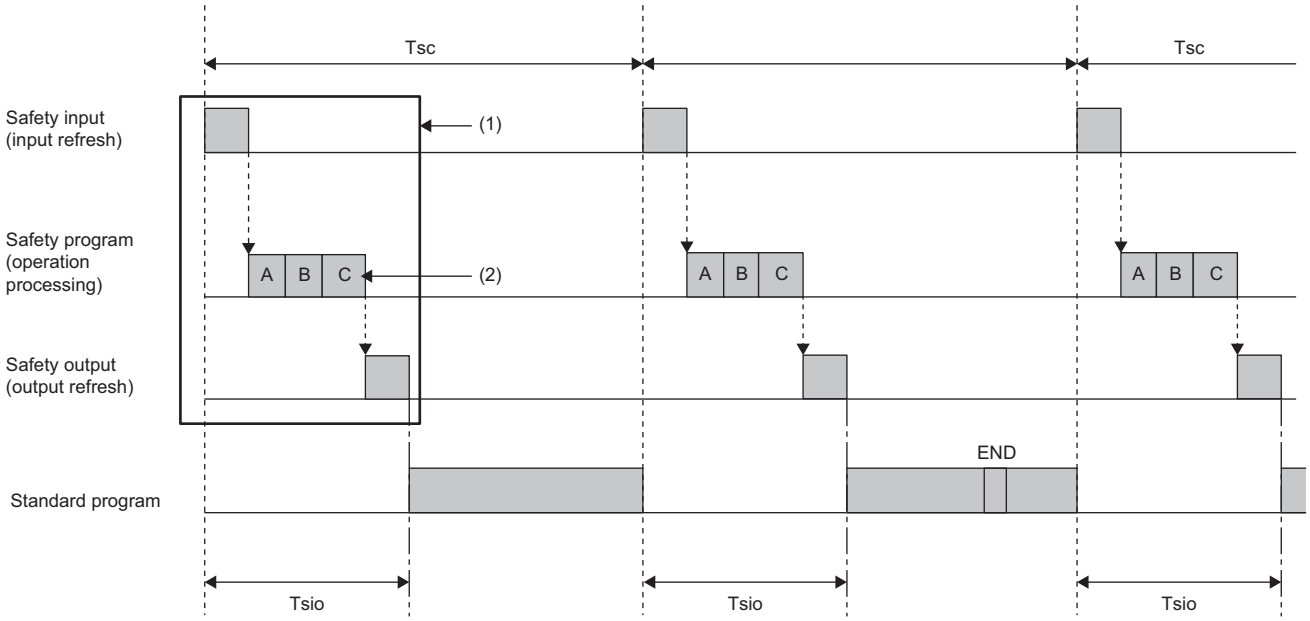
While constant scan is in progress, I/O refresh is performed after the waiting time for constant scan expires.

Program operation (standard program)

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 122 Subroutine Program)

Program operation (safety program)

Safety programs are executed at every safety cycle. Safety cycle processing is performed in the order of safety input (refresh) processing → safety program → safety output (refresh) processing. (☞ Page 91 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 in the following areas.
 • CPU module: SD1890, SD1891
 • SIL2 function module: Un\G62, Un\G63
(2) Execute all safety programs. (Execute safety programs A, B, and C.)

Point

- After executing a rising/falling instruction, the instruction will be valid until the safety program for the next safety cycle processing is executed. (MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))
- In safety programs, the block number of the file register (R) is not saved or restored. Data in the index register (Z, LZ) are not saved or restored, either.
- Even if no safety program has been registered, or the operating status of CPU modules 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 cycle time

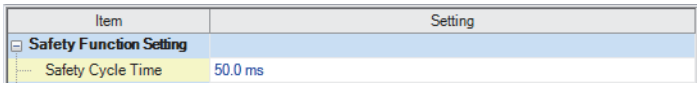
A safety cycle time is a timing for executing safety programs and performing safety input/output processing.

■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.	10.0 to 1000.0ms (unit: 0.1ms)	50.0ms

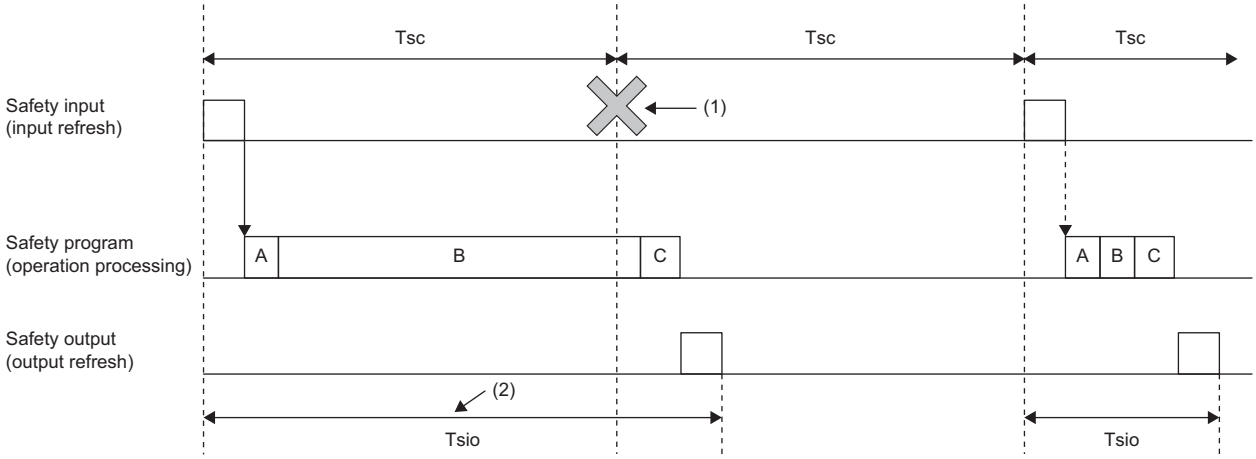
Point

If an instruction with long processing time is executed in the standard program, the start of safety cycle processing is delayed. To prioritize safety cycle processing, set "Enable" for the enabling interrupt setting during command execution. (☞ Page 129 Interrupt enabled during instruction execution)

When the program does not complete within the safety cycle time

Whether execution of the safety program is completed within the safety cycle time is monitored, and if the program execution time extends into the next safety cycle time, an error occurs in the CPU module and SIL2 function module. Note that if the execution time extends into the next safety cycle time, the 'safety input → safety program operation' following program completion is not executed, a safety cycle processing error (error code: 1A01H) is detected in the CPU module, and that operation is executed in the subsequent safety cycle time. Also, if the program execution time extends into the next safety cycle time, the safety cycle processing time execution cycle error flag turns on, and the safety cycle processing time execution cycle error count is incremented by one as follows.

Item	Safety cycle processing time execution cycle error flag	Safety cycle processing time execution cycle error count	Cause of increments of safety cycle processing time execution cycle error count
CPU module	SM1888	SD1888	Occurrence of an error code: 1A00H or 1A01H
SIL2 function module	—	Un\G54	Occurrence of an error code: 1A00H



Tsc: Safety cycle time

Tsio: Safety program + Safety input/output processing time

(1) If safety input/output and safety program execution are not completed within the safety cycle time, interrupts during this period are ignored. Execute safety input/output and safety programs from the next interrupt.

(2) If the time from safety input to safety output exceeds the safety cycle time, a continuation error will occur after safety processing is complete.

- CPU module: SM1888 turns ON, and the value in SD1888 is incremented by one.
- SIL2 function module: The value in Un\G54 is incremented by one.

END processing

The CPU module performs the following processing.

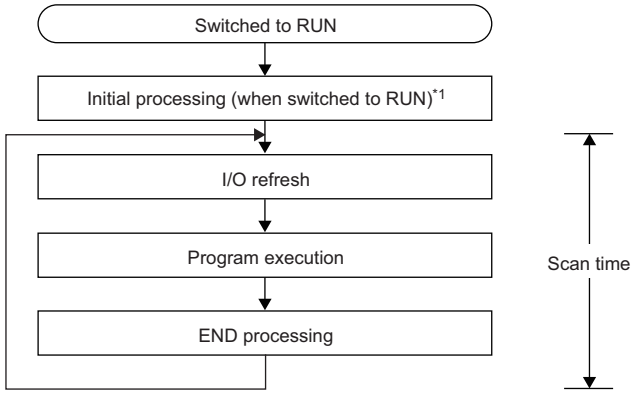
- Network module link refresh
- Intelligent function module refresh
- Instruction end processing (including dedicated instruction for the module)
- Device latch processing
- The device latch processing is a kind of service processing such as read and write of devices, labels, and program access files (☞ Page 560 Target List and Operation Details of the Device/Label Access Service Processing Setting).
- Watchdog timer reset (☞ Page 182 Watchdog timer reset)
- Refresh between CPU modules (for the multiple CPU system configuration)
- Self-diagnostics processing
- Sets a value to the special relay/special register (for those with the set timing specified as the END processing timing)

Point

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

9.3 Scan Time

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



*1 The initial scan time includes this processing.

Initial scan time

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

How to check the initial scan time

See below.

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

Initial scan time monitoring

Monitoring is performed with the initial scan time monitoring time. (☞ Page 182 Scan time monitoring time setting)

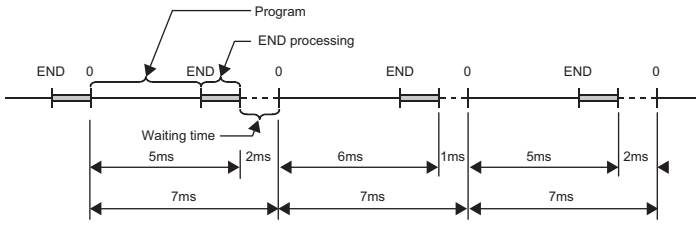
■Precautions on the initial scan time monitoring time

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

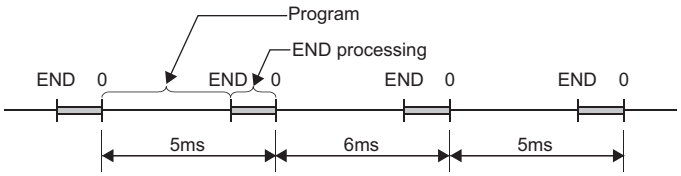
Constant scan

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

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



- When constant scan is not set

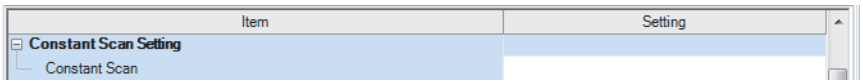


Setting constant scan

Constant scan can be set.

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

Window



Displayed items

Item	Description	Setting range	Default
Constant Scan	Sets the constant scan time.	0.2 to 2000ms (unit: 0.1ms)	—

Point

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

Accuracy of constant scan

The accuracy of the constant scan is 0.01ms. However, if processing, which should be executed during the waiting time from the completion of the END processing to the start of the next scan, is being executed, the constant scan cannot finish even if the constant scan time is reached. The constant scan may increase by the program execution time of the interrupt factor. (☞ Page 96 Precautions)

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 549 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 190 Error Clear)

For program executed in both systems

The constant scan function is invalid for the standby system.

Precautions

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

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

Device/label access service processing setting

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

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

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

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

Point

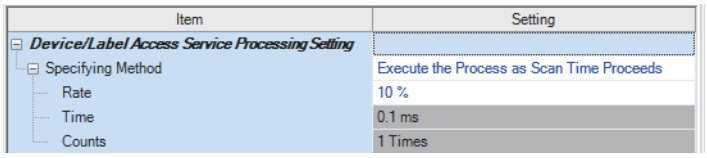
For functions to which the device/label access service processing setting can be applied, refer to the list targeted for device/label access service processing setting. (☞ Page 560 Target list)

Setting method

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

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

Window



Displayed items

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

Operations enabled by setting details

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

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

*1 This term indicates how long the scan time will be extended at its maximum due to the device/label access service processing.

*2 This term indicates how much the scan time will be varied due to the device/label access service processing.

*3 This term indicates how long the response time will take after a request for the device/label access service processing is received from an engineering tool or the like.

*4 This term indicates how much the response time will be varied depending on the contents of requests for the device/label access service processing from engineering tools or the like.

*5 When this function is enabled, monitoring is performed between execution of programs, and therefore values of ongoing operation may be read out. (☞ Page 99 When "Execute END Processing between Programs" is enabled)

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

☞ Page 562 Operation details

Precautions

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

■Functions that may prolong the scan time

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

- Online change
- Device/buffer memory batch monitor
- File register write or read*1
- Safety local devices (label) batch/registration monitor
- Current value change of the safety labels/safety devices
- Switching safety operation mode
- Identification check for safety data

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

■When constant scan is set

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

■When "Execute END Processing between Programs" is enabled

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

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

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

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

9.4 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 101 Refresh mode)

Refresh timing	Description
When END processing is performed	The refresh processing is performed at every END processing*1.
At the execution time of specified program	The refresh processing is performed before and after specified program execution. (☞ Page 121 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 103 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.

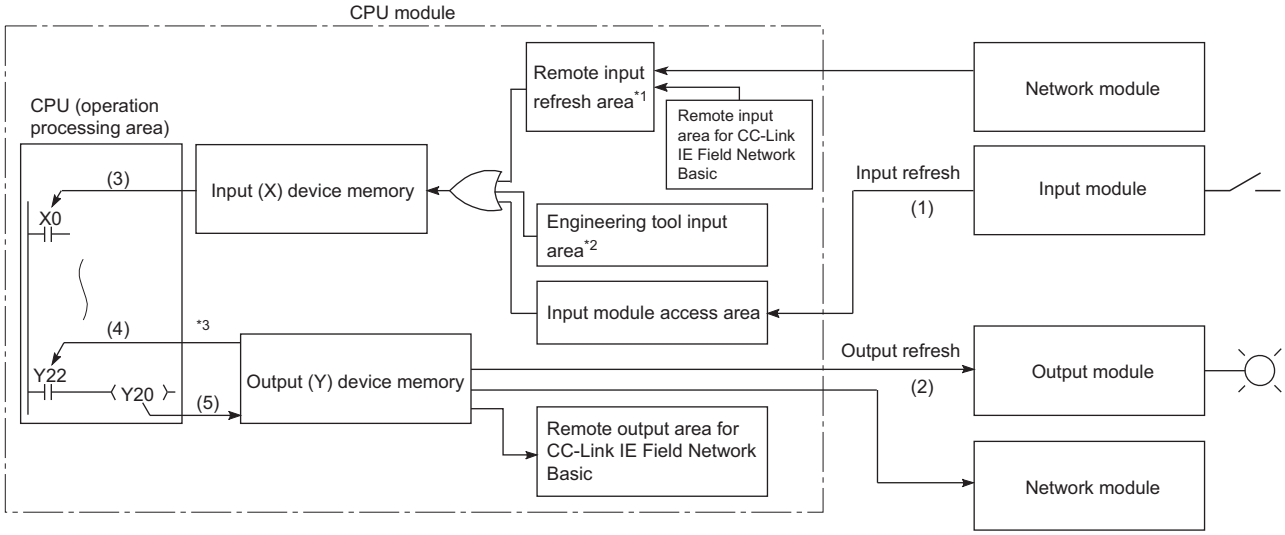
Item	Refresh mode	Direct mode
Input/output modules	Enabled	Enabled
Input/output of intelligent function modules		
Remote I/O in CC-Link IE Controller Network, CC-Link IE Field Network, or CC-Link		Disabled

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, or CC-Link.

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

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

Item	Description
Input refresh	At a specified timing (at END instruction execution or before specified program execution), input data are collectively read out from the input modules (1), the OR processing with the engineering tool input area and remote input refresh area is executed, and then the data are stored in the input (X) device memory.
Output refresh	At a specified timing (at END instruction execution or before specified program execution), data in the output (Y) device memory (2) are collectively output to the output module.
Execution of an input contact instruction	Input data in the input (X) device memory (3) are read out and the program is executed.
Execution of an output contact instruction	Output data in the output (Y) device memory (4) are read out and the program is executed.
Execution of the OUT instruction	The operation result of the program (5) are stored to the output (Y) device memory.

■Input

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

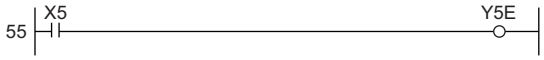
■Output

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

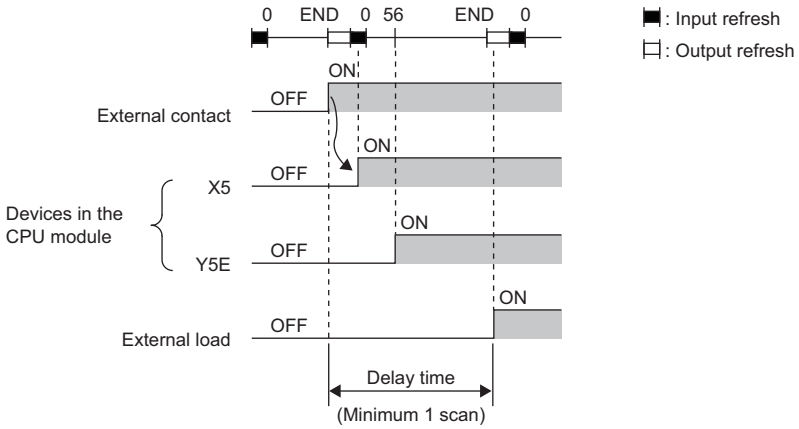
Response delay

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

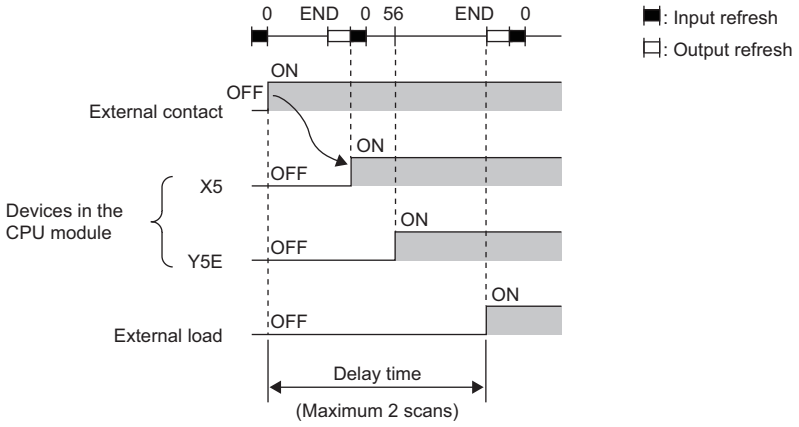
[Example] A program that turns on the output Y5E when the input X5 turns on



• Y5E turns on the earliest



• Y5E turns on the latest

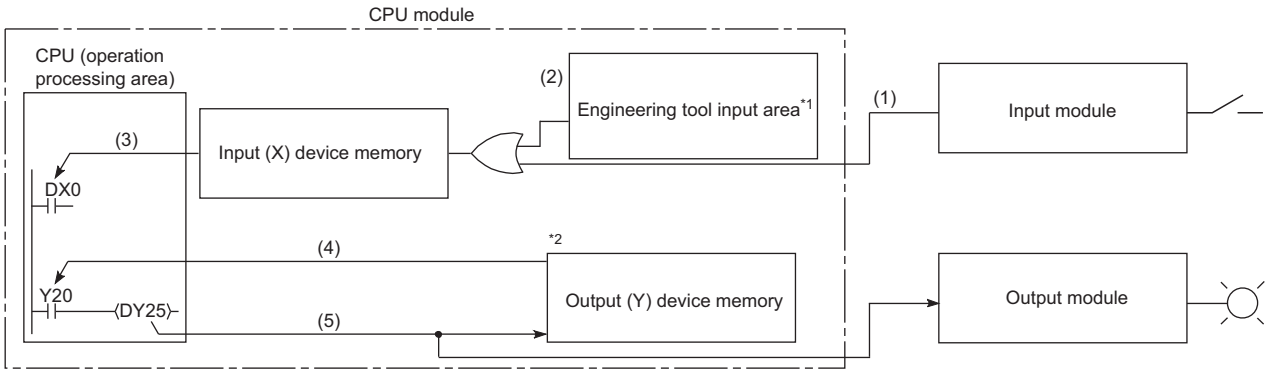


Direct mode

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

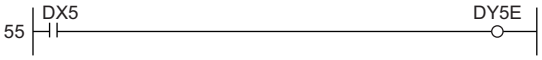
*2 Data in the output (Y) device memory can be turned on or off by the following:
- Test operation of an engineering tool
- Writing data from the network module
- Writing data from an external device using SLMP

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.

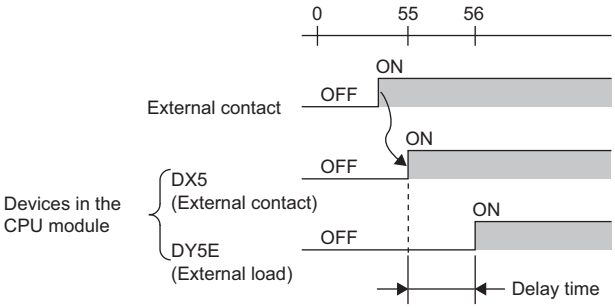
Response delay

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.

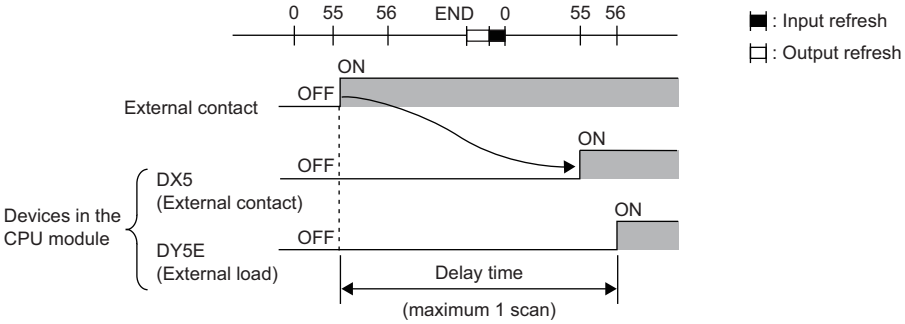
[Example] A program that turns on the output DY5E when the input DX5 turns on



• Y5E turns on the earliest

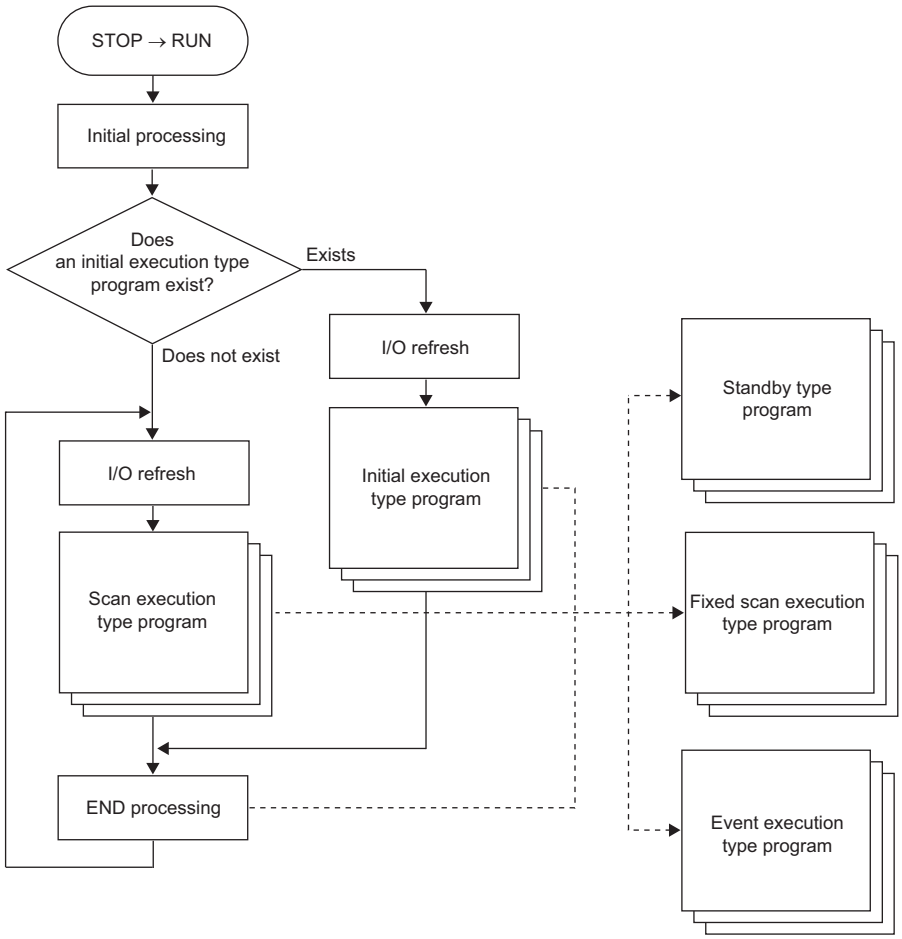


• Y5E turns on the latest



9.5 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 106 Program Execution Type, ☞ Page 119 Execution type change)



Point

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

9.6 Program Execution Type

Set the execution condition of the program. (☞ Page 119 Execution type change)

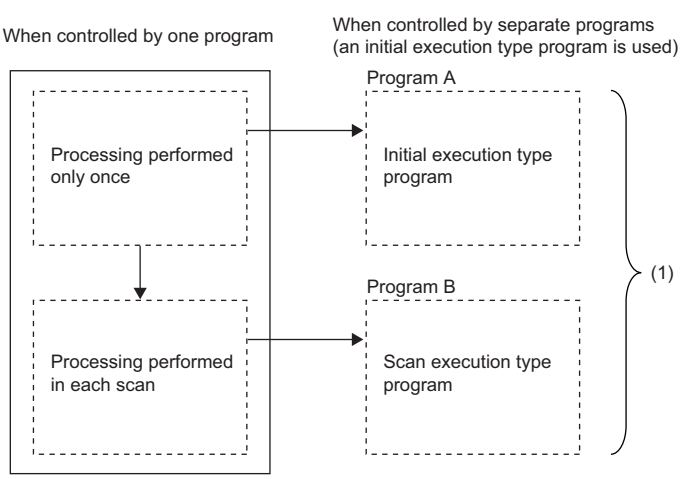
Program	Execution type
Standard program	• Initial • Scan • Fixed scan • Event • Standby
Safety program	Fixed scan

Safety programs are executed as fixed scan execution type programs. However, because they perform safety control, the operation differs from standard fixed scan execution type programs in the following ways.

- They are unaffected by disabling interrupt instructions (EI instruction, DI instruction, and IMASK instruction), and are executed based on the safety cycle time.
- Of all interrupt programs that can be created by customers, the execution of safety programs is given highest priority.
- Safety cycle processing including safety programs prioritizes fixed periodicity. If the safety cycle time is exceeded, and safety cycle processing is executed, processing is not performed in succession, but based on the interrupt timing of the next cycle time.

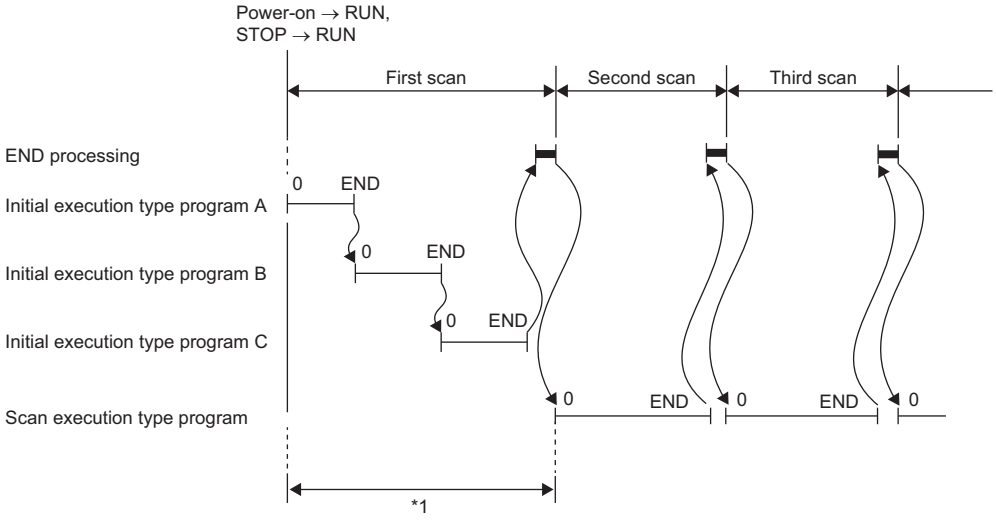
Initial execution type program

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



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

Note that the execution time of the initial execution type program is the same as the initial scan time. (☞ Page 94 Initial scan time) When multiple initial execution type programs are executed, the execution time of the initial execution type program will be the total time until all the initial execution type programs finish execution.



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

Precautions

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

■Restrictions on program creation

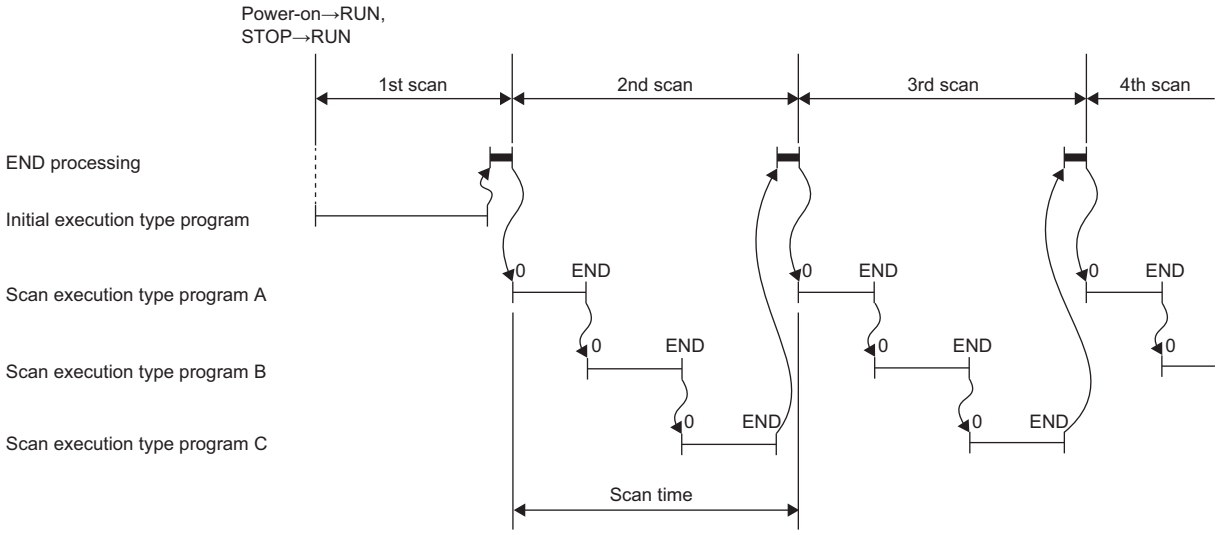
In the initial execution type program, do not use an instruction which requires several scans to complete (i.e., an instruction with a completion device).

Ex.

e.g. SEND and RECV instructions

Scan execution type program

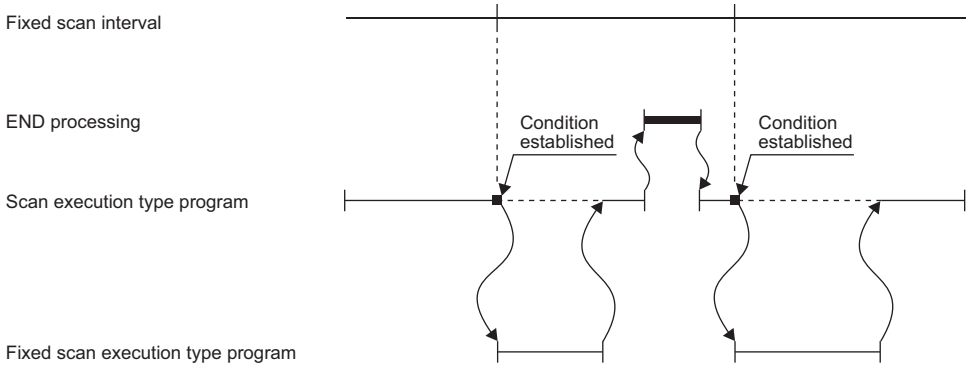
Scan execution type program is executed only once per every scan starting from the scan following the scan in which the initial execution type program was executed.



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

Fixed scan execution type program

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



Point

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

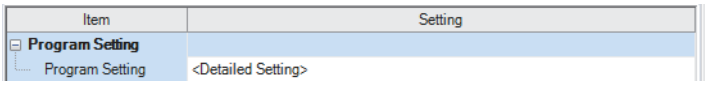
Fixed scan interval setting

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

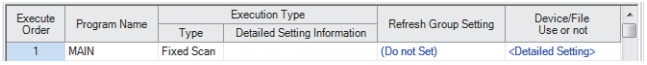
[CPU Parameter] ⇨ [Program Setting]

Operating procedure

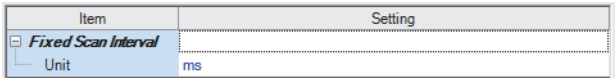
"Program Setting" window



"Detailed Setting" window



"Fixed Scan Execution Type Detailed Setting" window



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

Displayed items

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

Operation when the execution condition is satisfied

The following describes operation of the program.

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

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

■If there are multiple fixed scan execution type programs

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

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

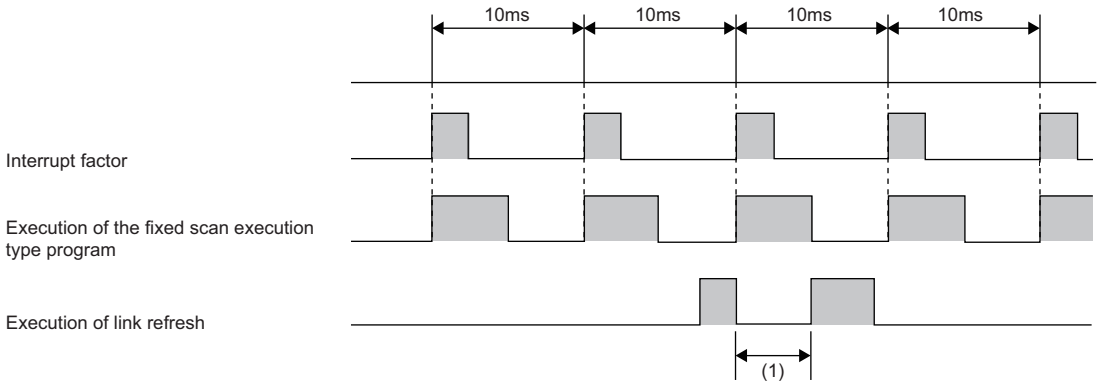
The program is executed according to the fixed scan execution mode setting.

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

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

■If an interrupt factor occurs during link refresh

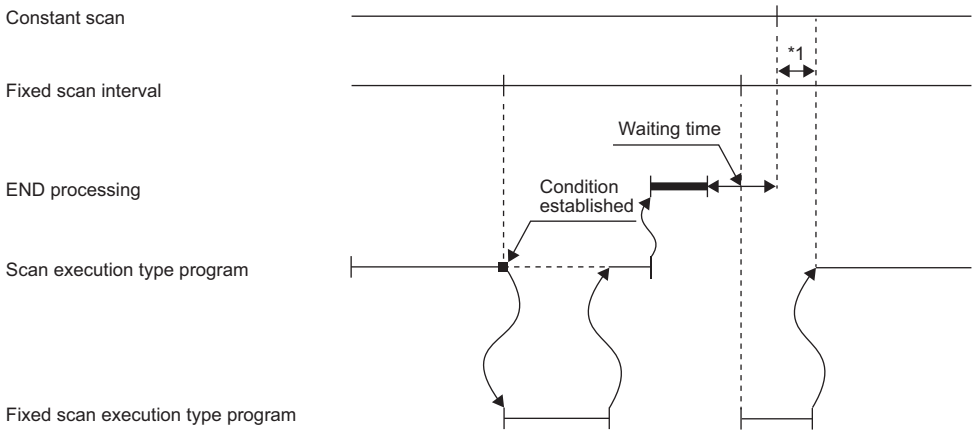
The link refresh is suspended and the fixed scan execution type program is executed. Even while the station-based block data assurance is enabled for cyclic data during refresh of such links as CC-Link IE Field Network, if the fixed scan execution type program uses a device specified as the refresh target,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



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

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

The fixed scan execution type program is executed.



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

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

If an interrupt program (including an interrupt which triggers the event execution type program) is triggered while the fixed scan execution type program is being executed, the program operates in accordance with the interrupt priority. (☞ Page 133 Multiple interrupt function)

Processing when the fixed scan execution type program starts

The same processing as when the interrupt program starts. (☞ Page 130 Processing at interrupt program startup)

Fixed scan execution mode

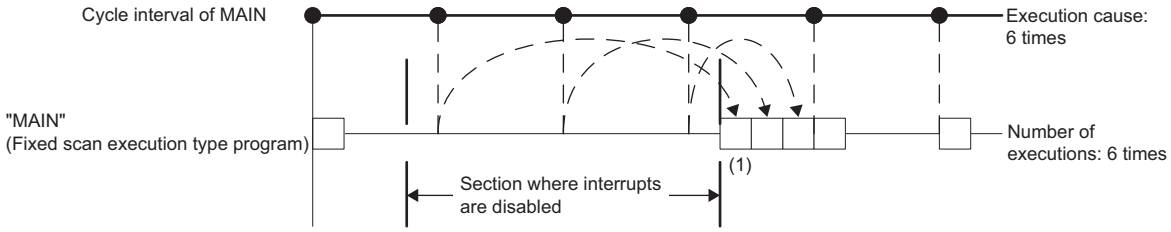
For fixed scan interrupts (I28 to I31, I48, I49) triggered by the fixed scan execution type program or the internal timer of the CPU module, this mode specifies the program execution operation that is performed when more than one interrupt occurs. (☞ Page 124 Operation upon occurrence of an interrupt 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

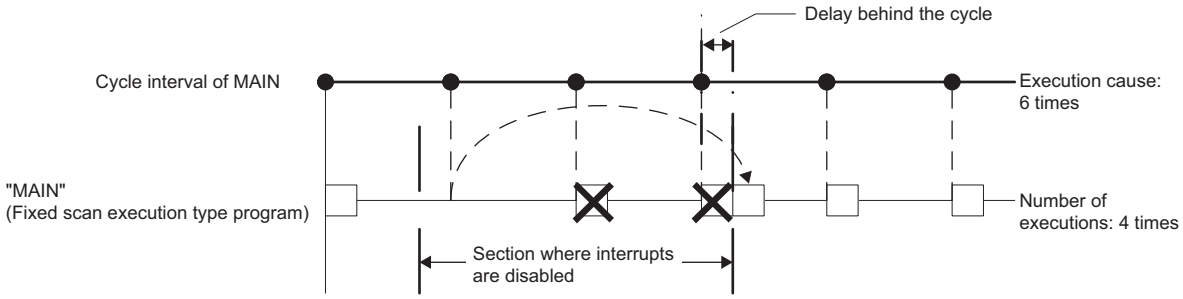
The program is executed for all the pending number of executions so that it can be executed the same number of times as execution causes.



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

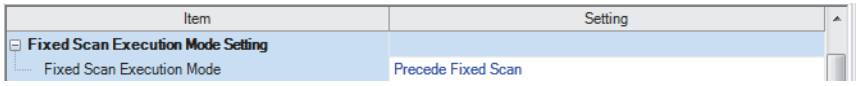


■Fixed scan execution mode setting

Use the fixed scan execution mode setting.

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

Window

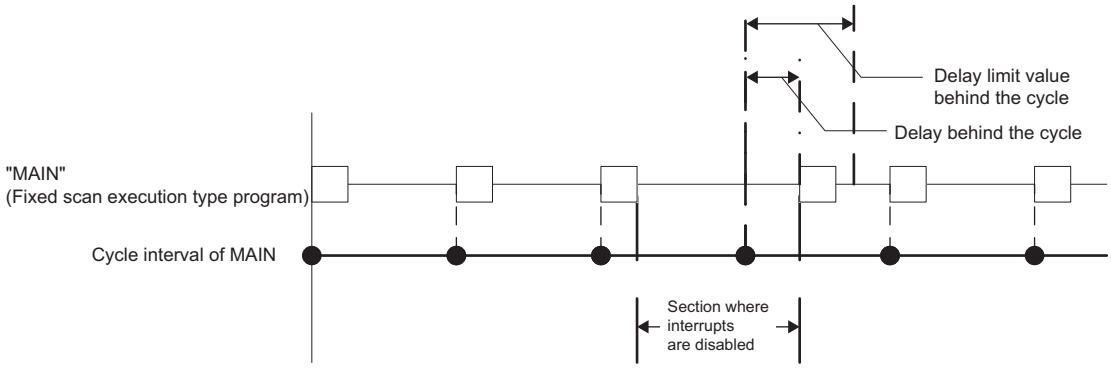


Displayed items

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

■Delay limit value behind a cycle

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



The delay limit value behind a cycle differs as follows.

• For the fixed scan execution type program

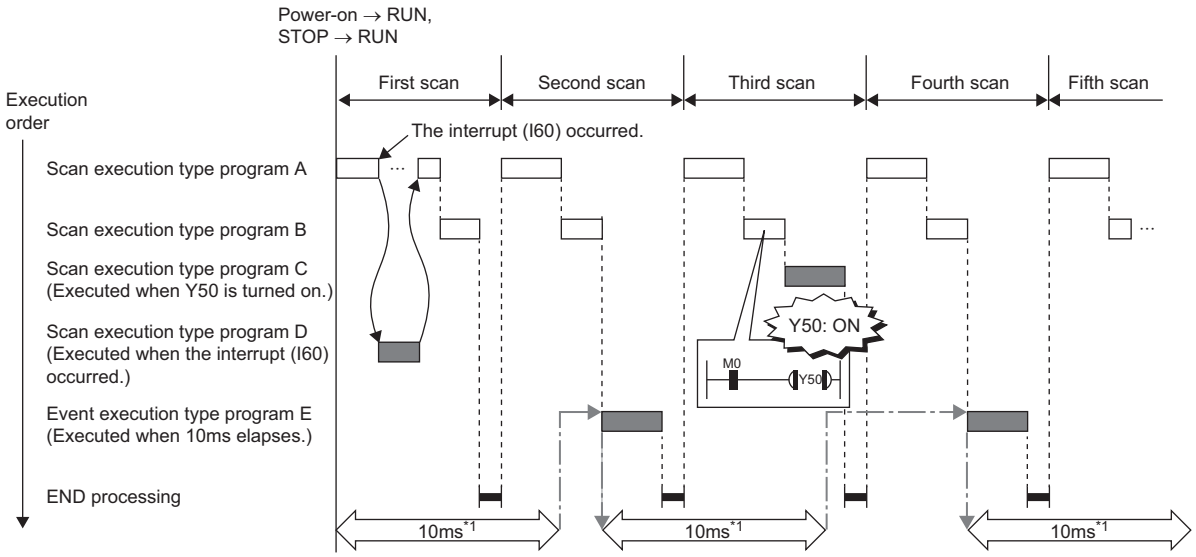
The value is the greatest common divisor of the fixed scan intervals of all the fixed scan execution type programs. For example, if there are fixed scan execution type programs with 2ms, 4ms, and 12ms scan periods, the delay limit value behind a cycle is 2ms.

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

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

Event execution type program

This type of program starts execution when triggered by a specified event. (☞ Page 114 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.



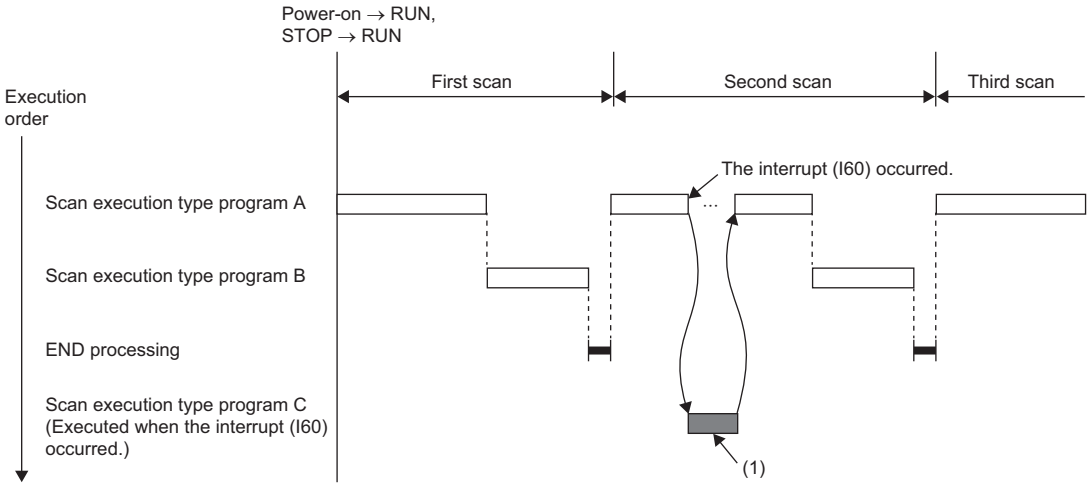
*1 Measurement of elapsed time is 10ms or more because it is determined depending on the scan time.

Trigger type

The following lists the triggers for the event execution type programs. (☞ Page 117 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.

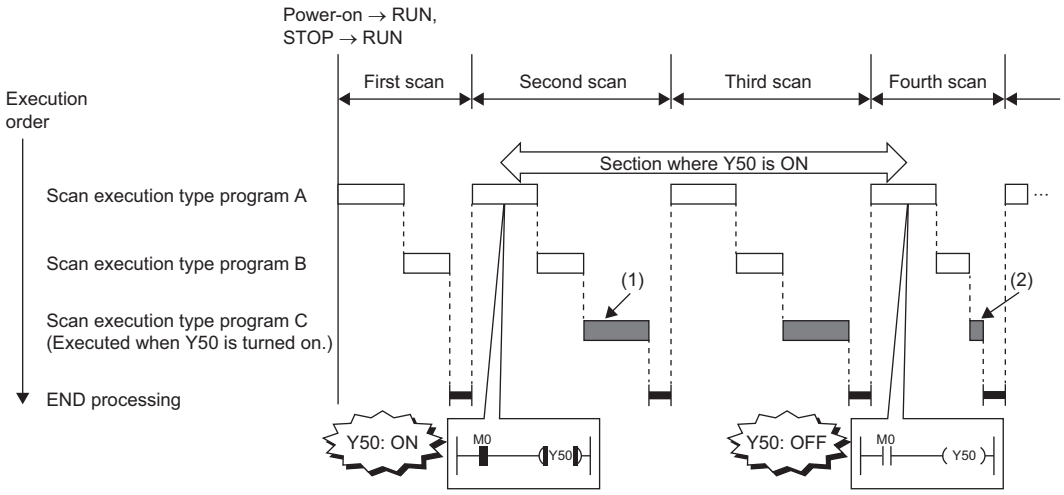
Interrupt pointers (I) which can be specified are I0 to I15, I28 to I31, I44, I45, I48, I49, I50 to I1023.

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 124 Operation upon occurrence of an interrupt factor)

■Bit data ON (TRUE)

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



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

Applicable devices are as follows.

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

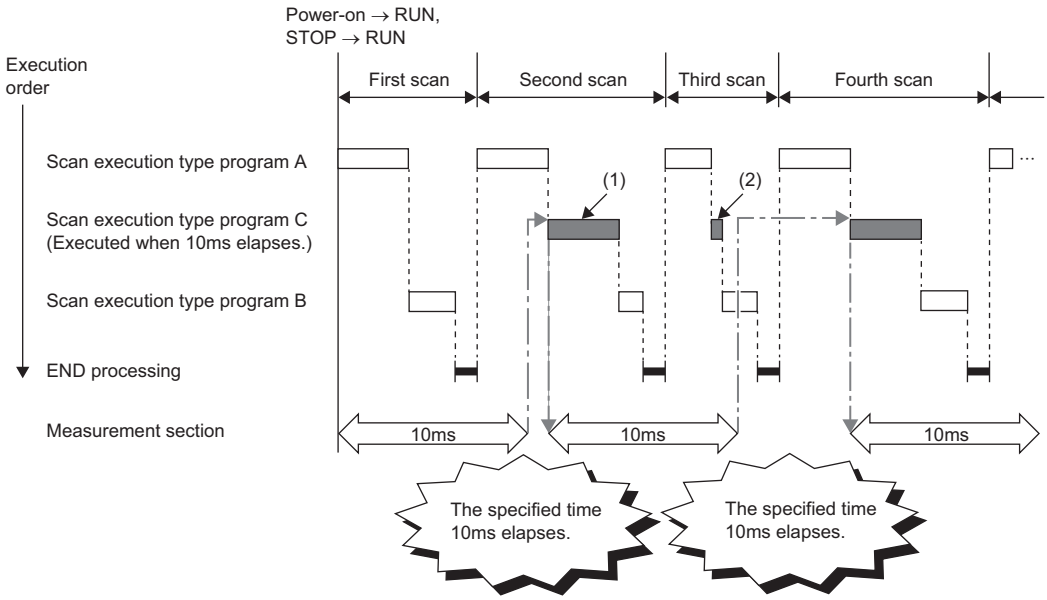
*1 A local device or index-modified device cannot be specified.
*2 This is not available when "Use File Register of Each Program" is enabled. However, if no file register is assigned or if specification is out of the range, interrupt is not be turned on (TRUE).

Point

To measure the timer and the retentive timer continuously even when the trigger execution condition of the event execution type program is not met, use the long timer (LT) and the long retentive timer (LST). The timer (T) and the retentive timer (ST) can be used in the event execution type program. However, they do not measure time when the trigger execution condition of the event execution type program is not met. Set "Clear Output and Current Value of Timer" to "Clear" when using the timer (T) and the retentive timer (ST).
(☞ Page 117 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).

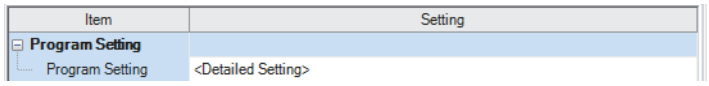
Trigger setting

Use the event execution type detail setting.

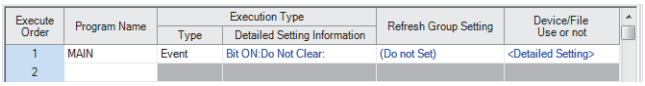
[CPU Parameter] ⇨ [Program Setting]

Operating procedure

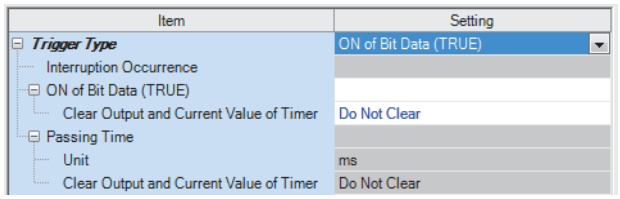
"Program Setting" window



"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

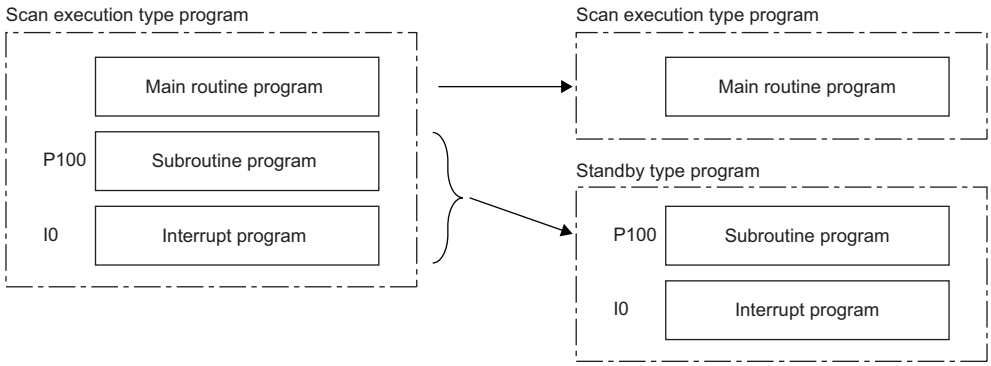
Item		Description	Setting range	Default
Interruption occurrence		Sets the interrupt pointer used as the trigger.	I0 to I15, I28 to I31, I44, I45, I48, I49, I50 to I1023	—
ON of Bit Data (TRUE)		Sets the device used as the trigger.	☞ Page 115 Bit data ON (TRUE)	—
Clear Output and Current Value of Timer		Sets that the current values of the output (Y), timer (T), and long timer (LT) used in this program are cleared at the execution turn of the event execution type program that comes after the specified bit data is OFF.	• Do Not Clear • Clear	Do Not Clear
Passing Time	Unit	Sets the time passed.	• When "ms" is selected: 1 to 65535ms (in units of 1ms) • When "s" is selected: 1 to 65535s (in units of 1s)	ms
	Clear Output and Current Value of Timer	Sets that the current values of the output (Y), timer (T), and long timer (LT) used in this program are cleared at the execution turn of the event execution type program that comes after the specified time passes.	• Do Not Clear • Clear	Do Not Clear

Standby type program

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

Librarization of programs

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



Program arrangement change

Prepare programs available in all systems to use them only when necessary. For example, a program set in advance as the standby type with a parameter can be changed to scan execution type and executed. (☞ Page 119 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.

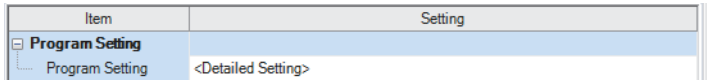
Using parameter settings

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

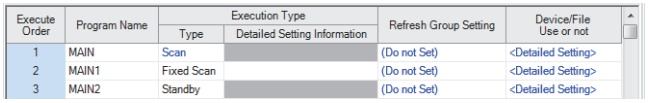
[CPU Parameter] ⇨ [Program Setting] ⇨ [Detailed Setting]

Operating procedure

"Program Setting" window



"Detailed Setting" window



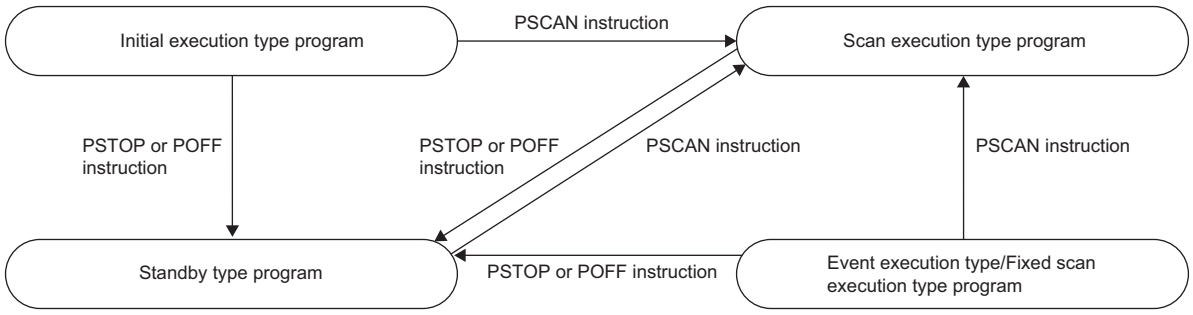
1. Click "Detailed Setting" on the "Program Setting" window.

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

Using an instruction

Use the following instructions.

- PSCAN
- PSTOP
- POFF



■Change timing

The execution type change timing is shown below.

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

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

■Precautions

The following lists the precautions when changing the execution type.

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

MELSEC iQ-R Programming Manual (Program Design)

Group setting for refresh

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

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

Refresh group setting

A group number is set to each program.

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

Window

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

Displayed items

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

Point

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

Refresh setting of each module

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

[Module Parameter] ⇨ [Refresh Setting]

Window

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

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

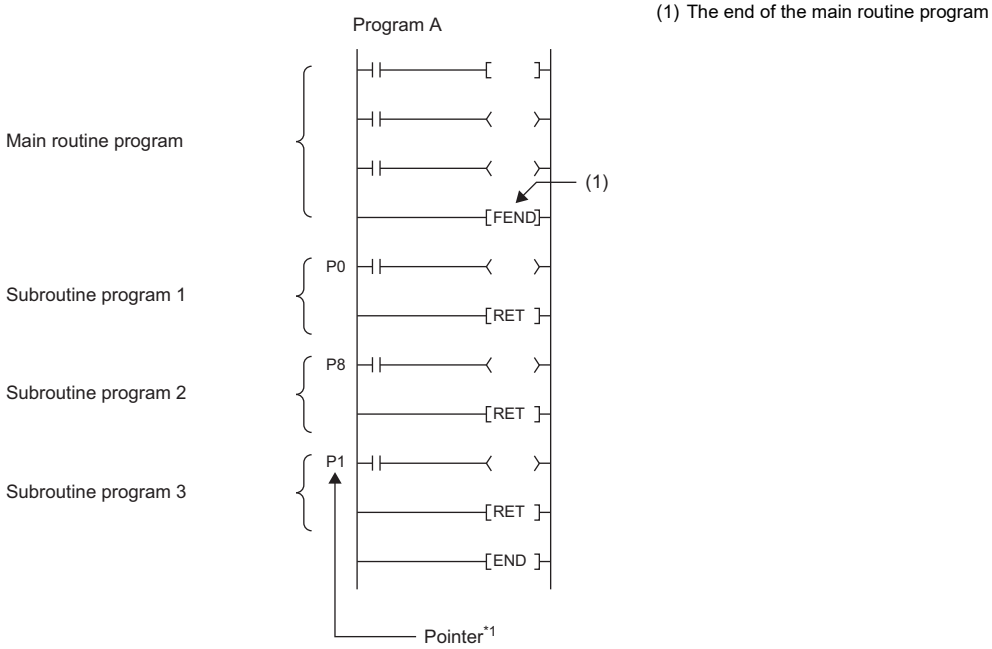
Point

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

9.7 Subroutine Program

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

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



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

Point

- Making a program a standby type allows it to be managed as a separate program. (☞ Page 118 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 319 Nesting (N))

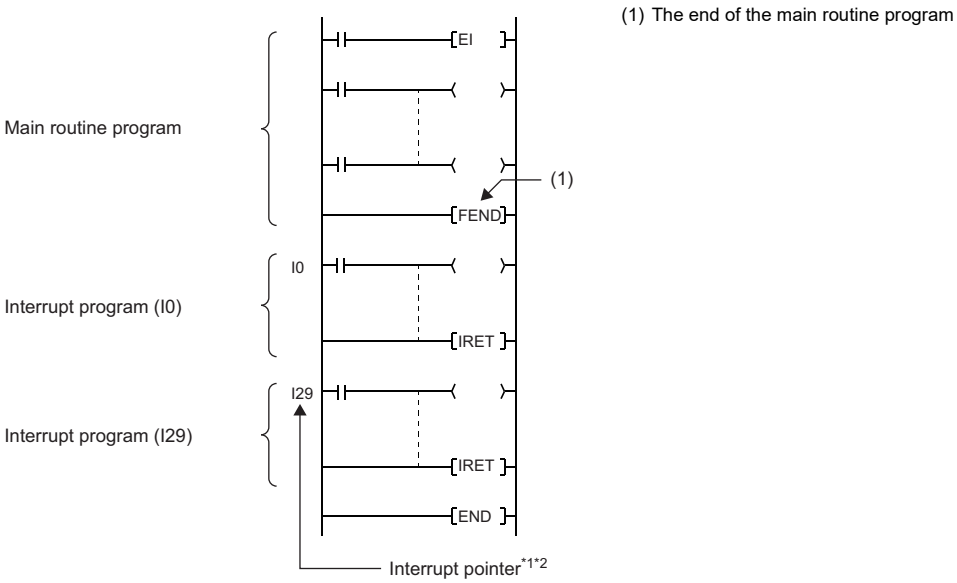
Precautions

The following lists the precautions when using the subroutine program.

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

9.8 Interrupt Program

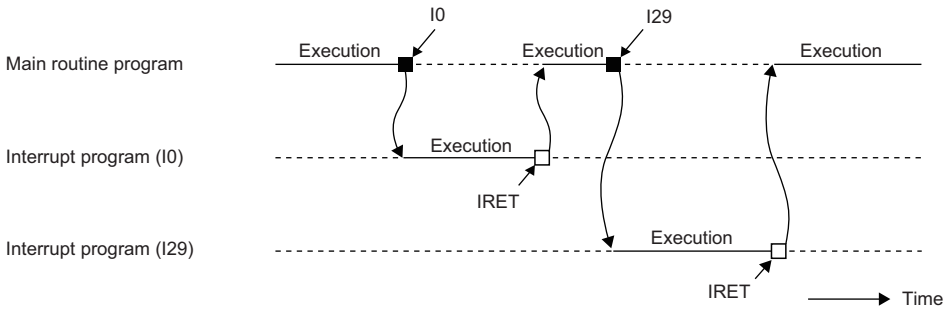
A program from an interrupt pointer (I) through the IRET instruction.



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

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

When an interrupt factor occurs, the interrupt program corresponding to its interrupt pointer number is executed. (☞ Page 324 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

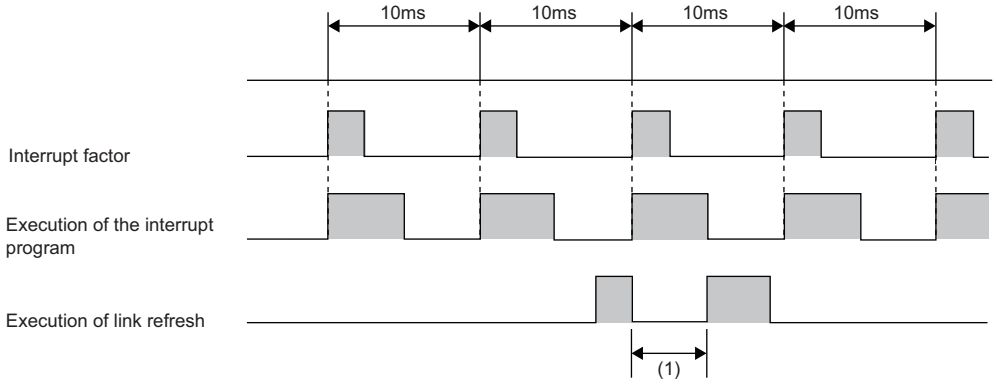
Making a program a standby type allows it to be managed as a separate program. (☞ Page 118 Standby type program)

Operation upon occurrence of an interrupt factor

The following shows the operation when an interrupt factor occurs.

■If an interrupt factor occurs during link refresh

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



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

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

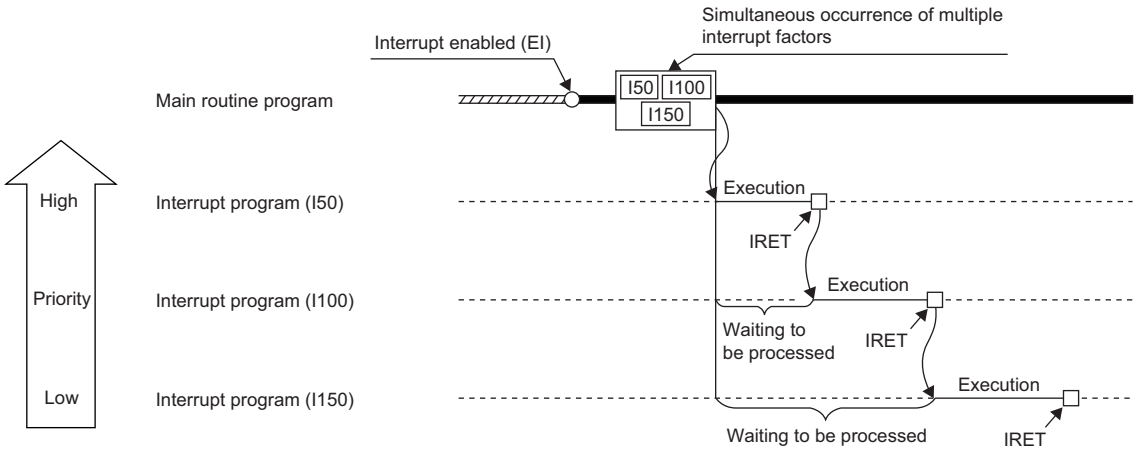
The interrupt program for the interrupt factor is executed.

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

If an interrupt such as a fixed scan execution type program (including an interrupt which triggers the event execution type program) is triggered while an interrupt program is being executed, the program operates in accordance with the interrupt priority. (☞ Page 133 Interrupt priority)

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

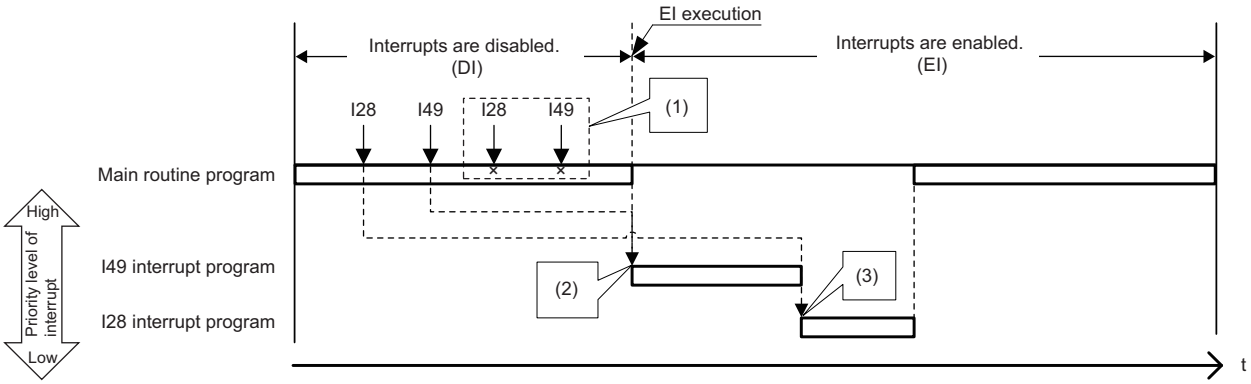
The interrupt programs are executed in the order of priority. If multiple interrupts with the same interrupt priority occur simultaneously, the interrupt programs are executed in the order of interrupt priority. (☞ Page 136 Multiple interrupt execution sequence)



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

• For I0 to I15, I28 to I31, I50 to I1023

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

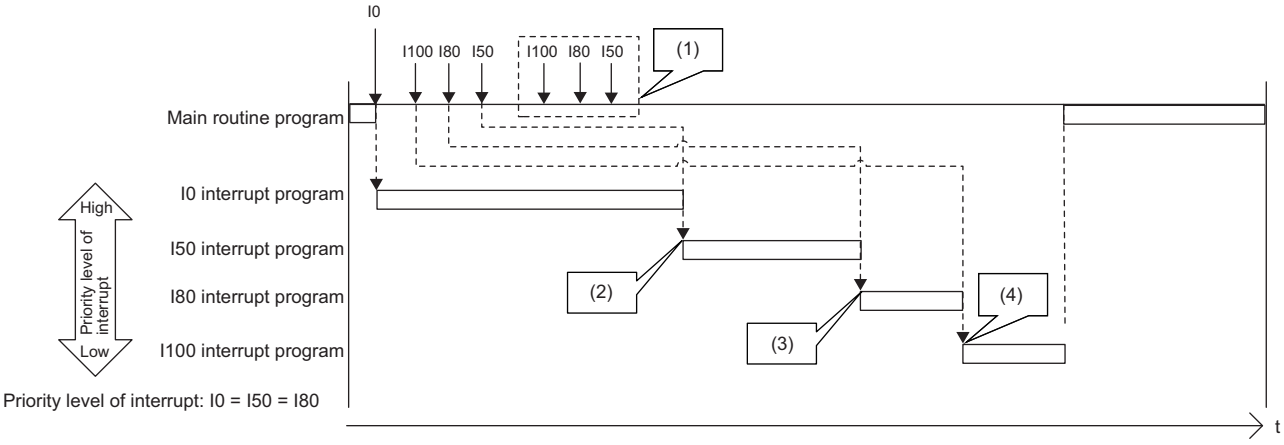


(1) The second and following interrupt factors that occur while interrupts have been disabled (DI) are not memorized.
(2) When interrupts are enabled, interrupts are executed in order from I15 because it has a higher priority.
(3) I28 is executed. (I15 is not executed for the second time.)

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

• For I0 to I15 and I50 to I1023

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

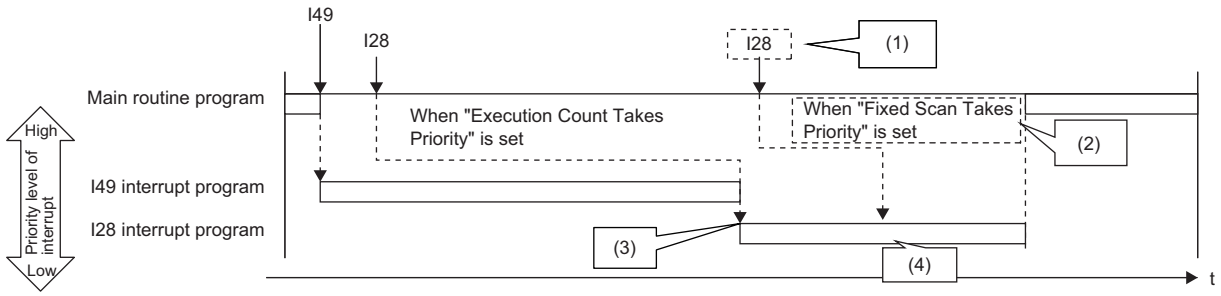


(1) The second and following interrupt factors that occur while interrupts have been disabled (DI) are not memorized.
(2) After the interrupt in execution is completed, interrupts are executed in order from I50 because it has a higher priority. Although both I50 and I80 have the same priority level, I50 is executed ahead of I80 because I50 has a higher priority order.
(3) I80 is executed. (I50 is not executed for the second time.)
(4) I100 is executed. (I80 is not executed for the second time.)

• For I28 to I31

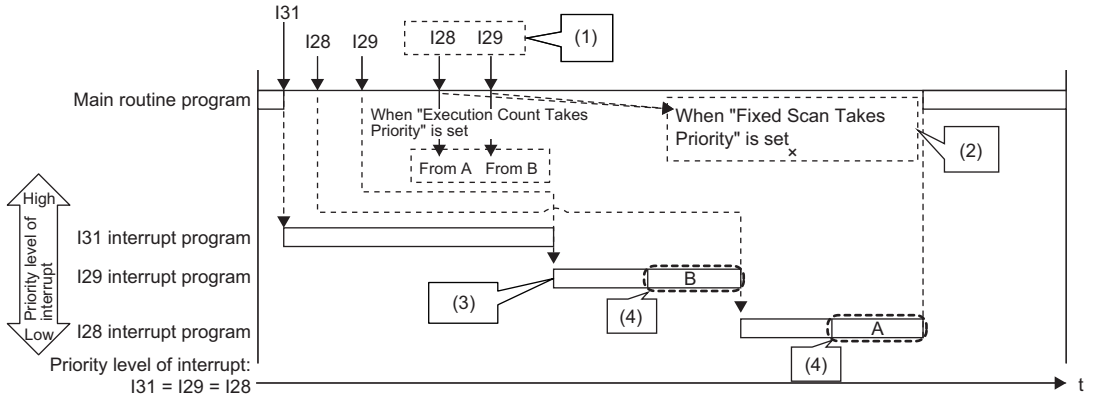
The interrupt factor that has occurred is memorized. After the running interrupt program finishes, the interrupt program corresponding to the factor will be executed. If the same interrupt factor occurs multiple times, it will be memorized once but operation at the second and later occurrences depends on setting of the fixed scan execution mode (☞ Page 111 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

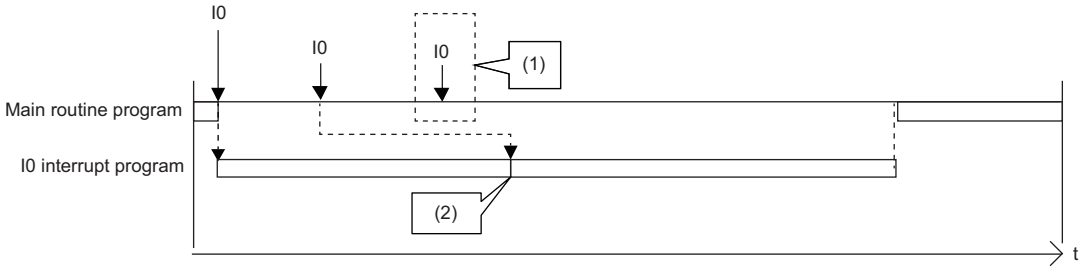


(1) The second and following interrupt factors that occur while interrupts are being executed operate according to the setting of the fixed scan execution mode.
(2) The program is not executed for the second time.
(3) After the interrupt in execution is completed, interrupts are executed in order from I29 because it has a higher priority.
(4) The second interrupt is executed.

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

• For I0 to I15 and I50 to I1023

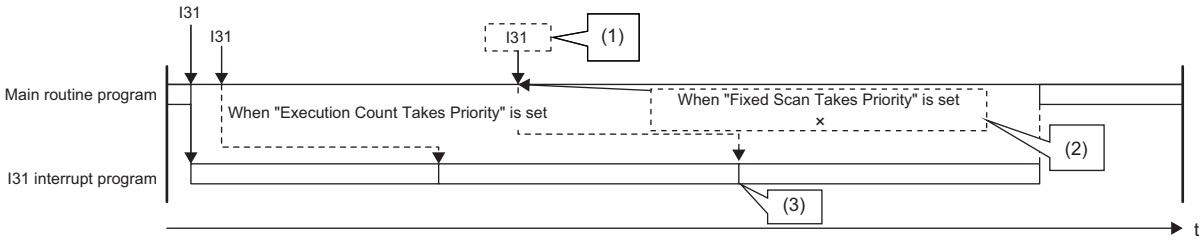
The interrupt factor that has occurred is memorized, and the interrupt program corresponding to the factor will be executed when the interrupt is enabled. Even if the same interrupt factor occurs multiple times, it will be memorized only once.



(1) The second and following interrupt factors that occur while interrupts are executed are not memorized.
(2) After the interrupt in execution is completed, the first interrupt program is executed.

• For I28 to I31

The interrupt factor that has occurred is memorized. After the running interrupt program finishes, the interrupt program corresponding to the factor will be executed. If the same interrupt factor occurs multiple times, it will be memorized once but operation at the second and later occurrences depends on setting of the fixed scan execution mode (☞ Page 111 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.

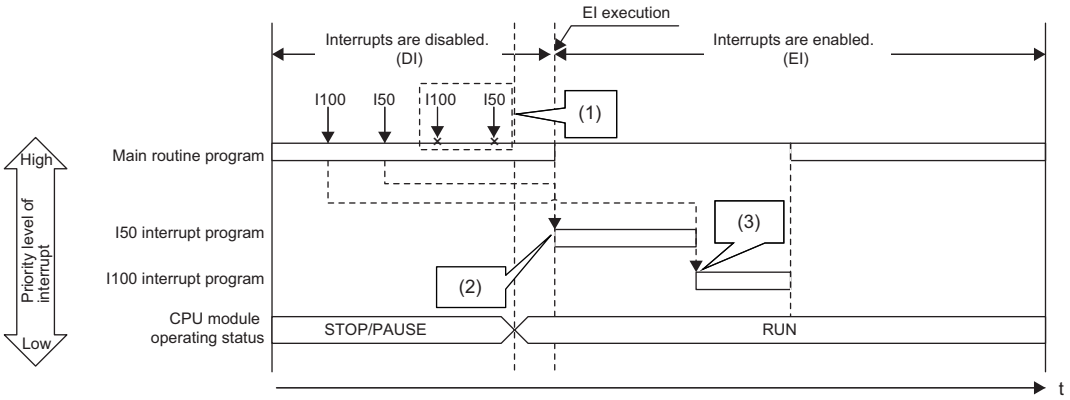


(1) The second and following interrupt factors that occur while interrupts are being executed operate according to the setting of the fixed scan execution mode.
(2) The program is not executed for the second time.
(3) The second interrupt is executed.

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

• For I0 to I15, I28 to I31, I50 to I1023

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



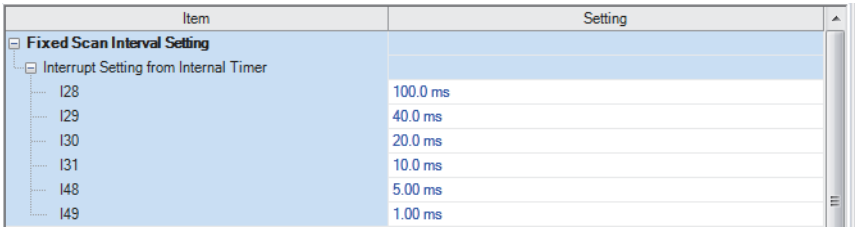
(1) The second and following interrupt factors that occur while the CPU module is in the STOP state are not memorized.
(2) When interrupts are enabled by changing the operating status of the CPU module from STOP to RUN, interrupts are executed in order from I50 that has a higher priority level.
(3) I100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

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

Restriction

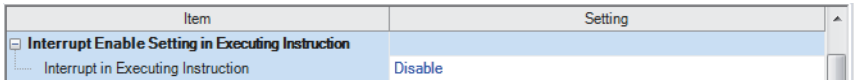
The interrupt pointer (I) cannot be used in safety programs.

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 137 Precautions)	• Disable • Enable	Disable

Processing at interrupt program startup

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

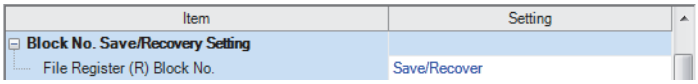
- Saving/restoring of the file register (R) block number
- Saving/restoring of the index register (Z, LZ)

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

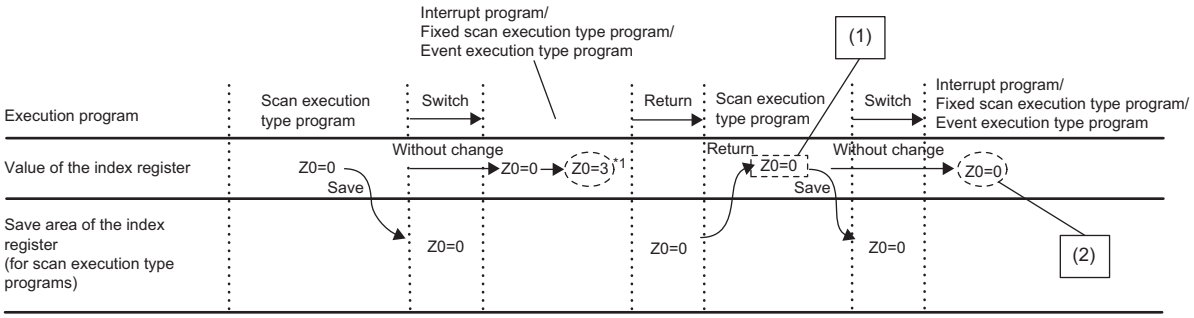
Point

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 536 Overhead time when executing the interrupt program). To change the block number of the file register (R) while "Not Saved/Recovered" is enabled, the block number of that file register (R) must be saved/restored by the program.

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

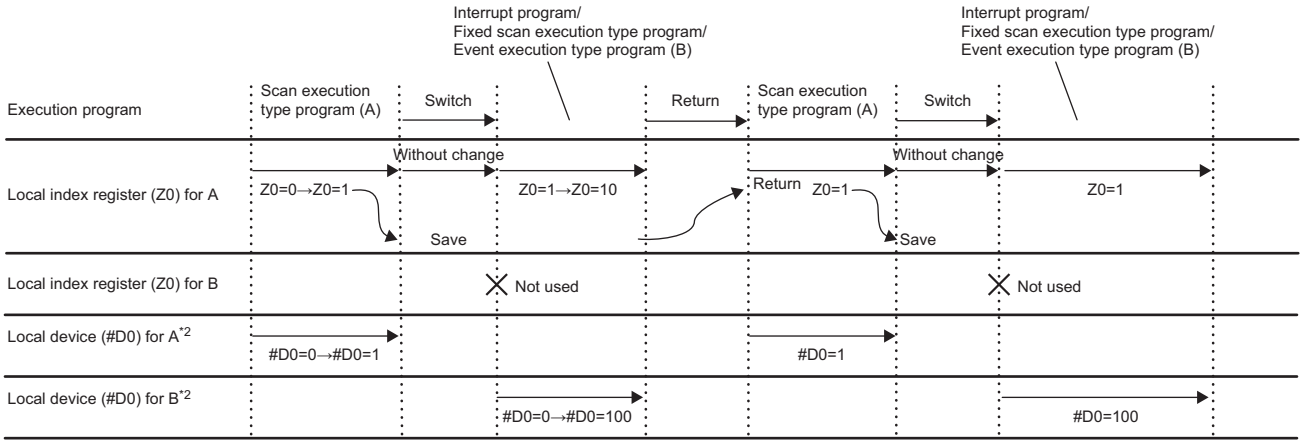
When an interrupt program starts, the value of the index register (Z, LZ) of the running program is saved. When the interrupt program finishes, and the saved value is restored to the running program. Note that when an interrupt program starts, the local index register (Z, LZ) is not switched to the different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

- Saving/restoring of the index register



(1) This value can be used as a value specific to the scan execution type program. (Using the value in an interrupt program is not needed to be concerned.)
(2) This value can be used only at the moment an interrupt program, fixed scan execution type program, or event execution type program is executed. (The value cannot be used continuously.)

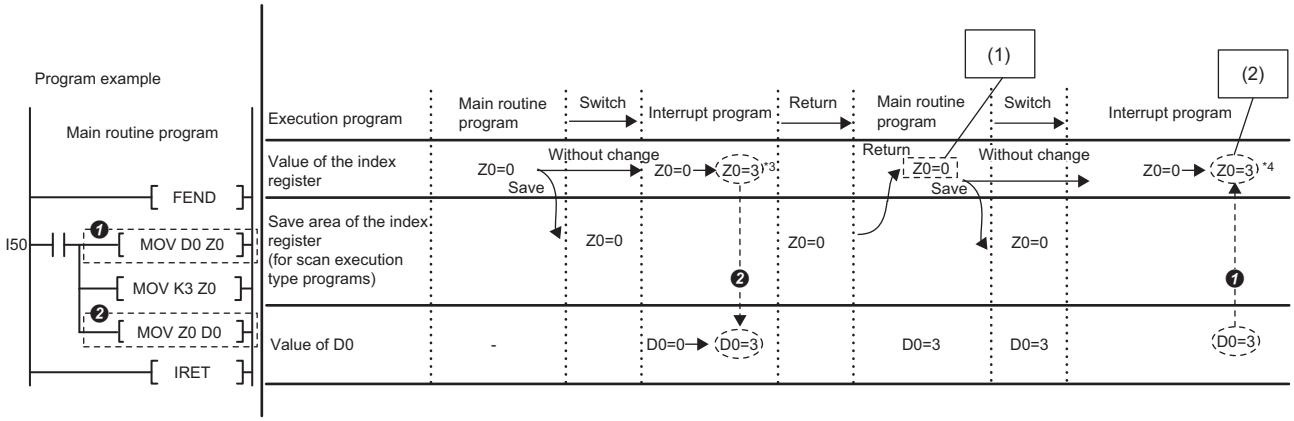
- Operations of the local index register and local device



*1 The value of Z0 is changed by the interrupt program, fixed scan execution type program, or event execution type program.
*2 For when SM777 is on (when using the local device of the program file of storage location)

Point

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

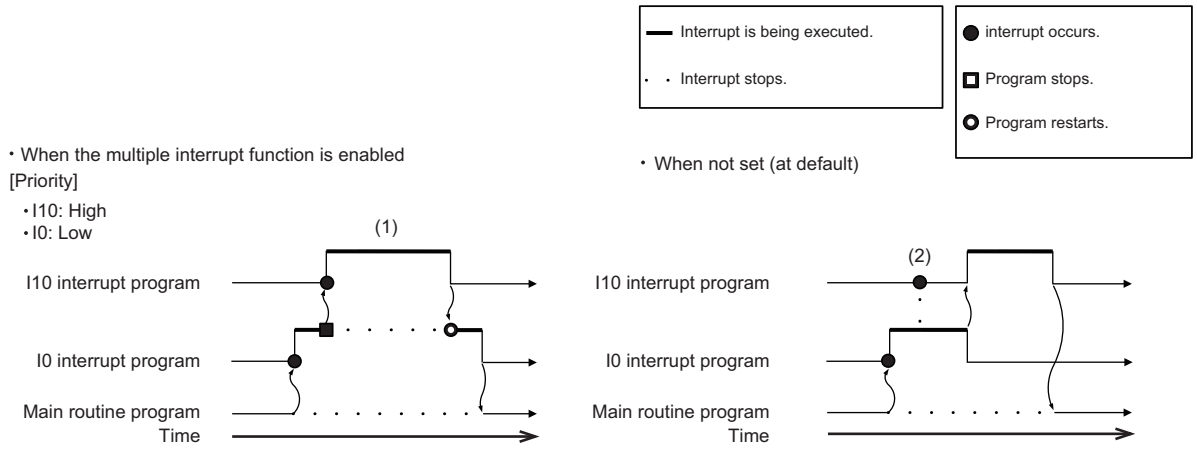


*3 A value in Z0 changes at MOV D0 Z0.
*4 Z is zero when the program ❶ and ❷ are not added.

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

Multiple interrupt function

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



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

Interrupt priority

If the interrupt priority of a program for which its execution condition has been satisfied is higher than that of the running program, the programs are executed in accordance with their interrupt priority. If the interrupt priority of the new program is the same or lower, it enters the waiting status until the running program finishes. The interrupt priority 4 listed below cannot be changed, whereas the interrupt priorities 5 to 8 can. (☞ Page 134 Interrupt priority setting)

Interrupt priority		Interrupt factor (Interrupt pointer)	Execution sequence at simultaneous occurrence	Changeability
High ↑ ↓ Low	4	Interrupt by internal timer (I28 to I31), Fixed scan execution type program	I31 → I30 → I29 → I28 → Fixed scan execution type program	Unchangeable (Fixed)
	5 to 8	Interrupt from module (I0 to I15, I50 to I1023)	I0 ··· → ··· I1023	Changeable

Restriction

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.

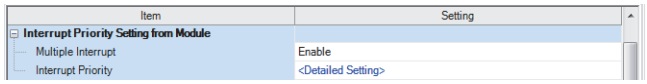
Interrupt priority setting

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

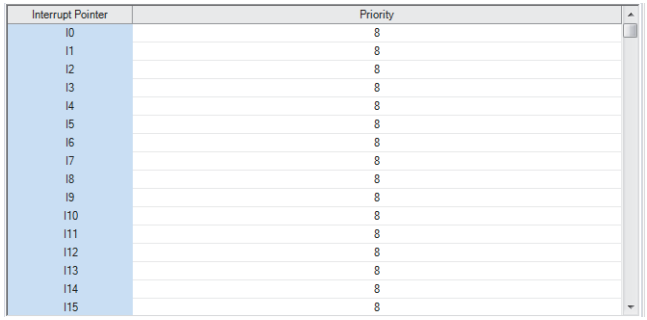
[CPU Parameter] ⇨ [Interrupt Settings] ⇨ [Interrupt Priority Setting from Module]

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.

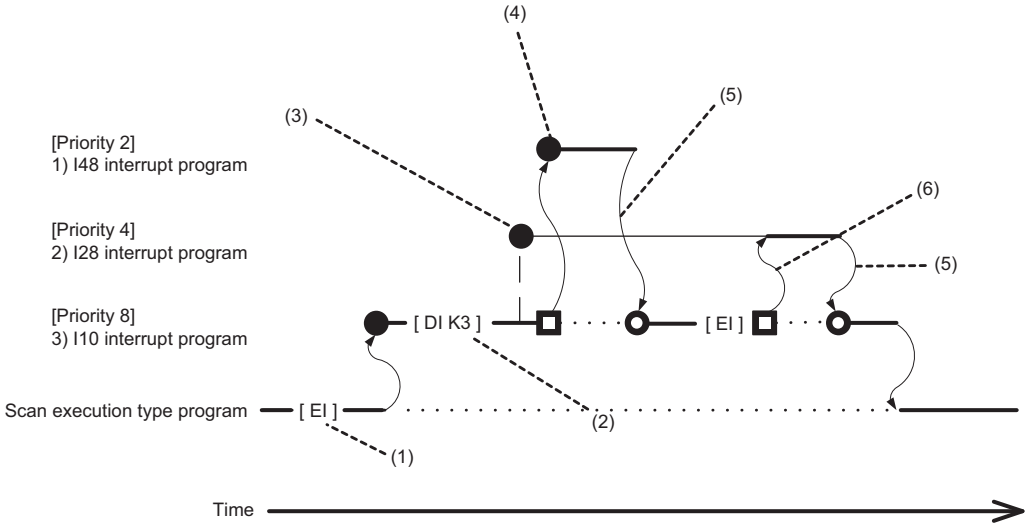
Ex.

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	4	I28, I29, I30, I31
	5	I101
	6	I0, I50, I100
	7	I1020
	8	An interrupt pointer among I0 to I15 or I50 to I1023 with the priority other than 5 to 7



(1) Interrupt is enabled.
(2) Interrupts with priority 5 to 8 are disabled.
(3) I50 is not executed because interrupts with priority 5 to 8 are disabled.
(4) I28 is executed because its interrupt priority is higher.
(5) Return from interrupt. Execution of the interrupted I10 resumes.
(6) High-priority interrupt I50 is executed because interrupts with priority 5 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.

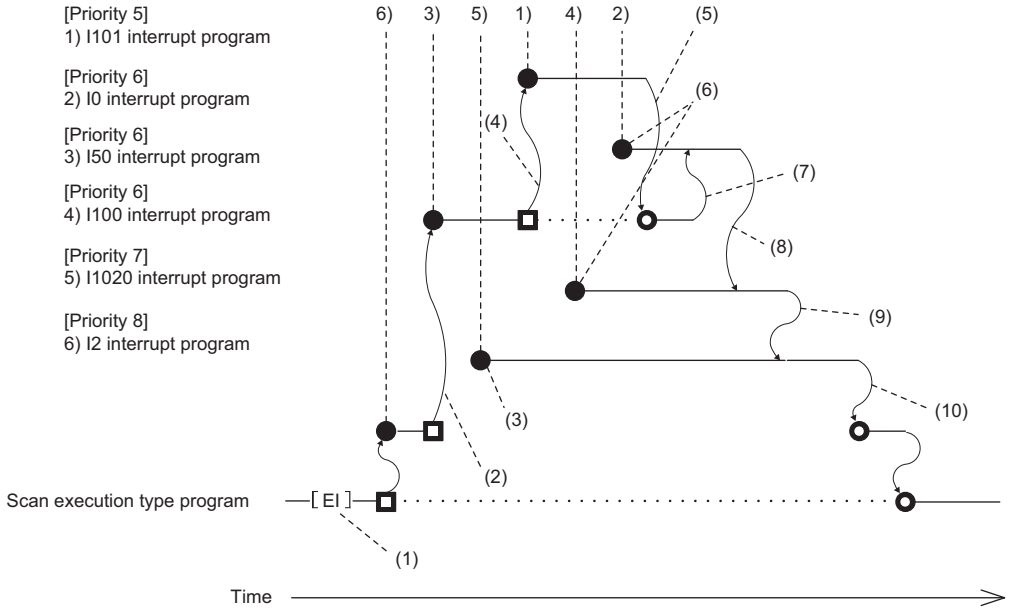
Ex.

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	4	I28, I29, I30, I31
	5	I101
	6	I0, I50, I100
	7	I1020
	8	An interrupt pointer among I50 to I1023 with the priority other than 5 to 7



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

Precautions

The precautions for the interrupt program are mentioned below.

Restrictions on program creation

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

Data inconsistency

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

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

If the interrupt accuracy does not increase

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

- Limit each character string used in a program to 32 characters or less in length.
- Decrease the multiplex interrupts.

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 129 Interrupt enabled during instruction execution)

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

Delay in interrupt programs

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

Devices and labels used by interrupt programs

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

Internal timer interrupt

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

10 CPU MODULE OPERATION PROCESSING

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

- RUN state
- STOP state
- PAUSE state

10.1 Operation Processing by Operating Status

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

Operation processing in RUN state

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

■Output at the time of entering RUN state

Depending on the setting of output (Y) at the time of change from STOP state to RUN state, either of the following is output. (☞ Page 142 Output mode at operating status change (STOP to RUN))

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

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

Operation processing in STOP state

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

■Output at the time of entering STOP state

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

Operation processing in PAUSE state

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

Operation processing based on CPU module operating status

Operation processing based on the CPU module operating status is the same as that of the standard CPU. (☞ Page 139 Operation Processing by Operating Status)

■Safety communications processing based on CPU module operating status

The following table shows safety communications processing based on the CPU module operating status.

CPU module 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 in the CPU module.

■Operation processing when operating status of the CPU module changes

The following table shows operation processing in the safety program when the operating status of the CPU module is changed.

CPU module operating status	CPU module processing			
	Safety program	External output	Safety device data*1	
			Other than SA\Y	SA\Y
STOP → RUN	Executes safety programs.	Outputs the value after safety program execution.	Retains the safety device data status immediately before the RUN state. Clears safety local devices.	Retains the SA\Y value immediately before the RUN state.
RUN → STOP	Executes the program up to the END instruction and then stops.	Turns off all safety communication outputs.*3	Retains the device data status immediately before the STOP state.	Turns off all SA\Y values.*2
RUN → PAUSE	Stops the operation after the execution of one scan.	Turns off all safety communication outputs.*3	Retains the device memory status immediately before the PAUSE state.	Turns off all SA\Y values.*2
PAUSE → RUN	Executes the program from the start.	Outputs the value after safety program execution.	Retains the device data status immediately before the RUN state. Clears safety local devices.*2	Retains the device data status immediately before the RUN state.
PAUSE → STOP	Operation remains stopped.	All safety communication outputs remain off.*3	Retains the device data status immediately before the STOP state.	Retains the SA\Y value immediately before the STOP state.
STOP → PAUSE	Operation remains stopped.	All safety communication outputs remain off.*3	Retains the device data status immediately before the PAUSE state.	Retains the SA\Y value immediately before the PAUSE state.

*1 Assignment device data for standard/safety shared labels is processed in the same way as operation processing with standard programs.

*2 Values for safety devices when the CPU module 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 141 Operation Processing When Operating Status Is Changed

10.2 Operation Processing When Operating Status Is Changed

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

CPU module operating status	CPU module processing			
	Program	External output	Device memory	
			Other than Y	Y
STOP → RUN	Executes the program from the start.	Determines the status depending on the setting of "Output Mode Setting at STOP to RUN" in the CPU parameter. (☞ Page 142 Output mode at operating status change (STOP to RUN))	Retains the device memory status immediately before the RUN state. However, if a device/label initial value has been set, this initial value is set.*1 Clears local devices.	Determines the status depending on the setting of "Output Mode Setting at STOP to RUN" in the CPU parameter. (☞ Page 142 Output mode at operating status change (STOP to RUN))
RUN → STOP	Executes the program up to the END instruction and then stops.	Saves the output (Y) status immediately before the STOP state, and all points turn off.	Retains the device memory status immediately before the STOP state.	Saves the output (Y) status immediately before the STOP state, and all points turn off.
RUN → PAUSE	Stops the operation after the execution of one scan.	Retains the output (Y) status immediately before the PAUSE state.	Retains the device memory status immediately before the PAUSE state.	Retains the output (Y) status immediately before the PAUSE state.
PAUSE → RUN	Executes the program from the start.	Retains the output (Y) status immediately before the RUN state.	Retains the device memory status immediately before the RUN state. Clears local devices.	Retains the output (Y) status immediately before the RUN state.
PAUSE → STOP	Operation remains stopped.	Saves the output (Y) status immediately before the STOP state, and all points turn off.	Retains the device memory status immediately before the STOP state.	Saves the output (Y) status immediately before the STOP state, and all points turn off.
STOP → PAUSE	Operation remains stopped.	Retains the output (Y) status immediately before the PAUSE state.	Retains the device memory status immediately before the PAUSE state.	Retains the output (Y) status immediately before the PAUSE state.

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

Output mode at operating status change (STOP to RUN)

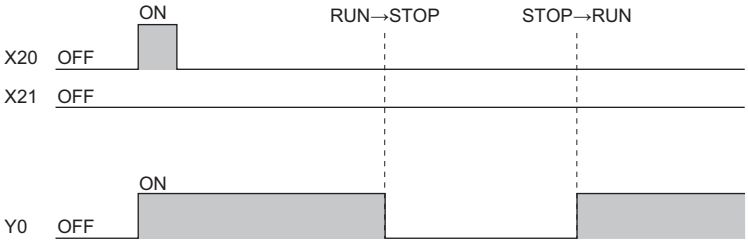
When the operating status changes from RUN to STOP, for example, the CPU module internally stores the status of the outputs (Y) to turn them all off.
However, the safety output (SA\Y) for the safety output device (SA\Y) is off, regardless of this setting. If the value is changed with a device test or others while the CPU module is in the STOP state, the value will be applied when the status of the CPU module changes from STOP to RUN.

Operation when the operating status is changed from STOP to RUN

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

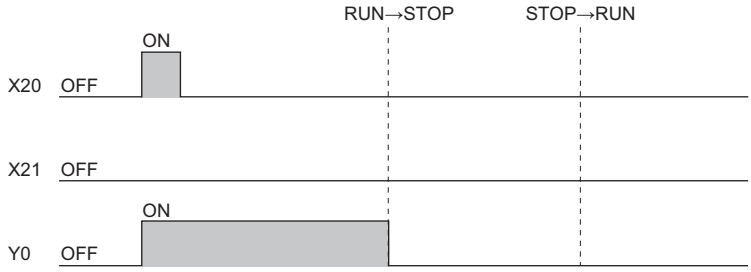
■Outputting the output (Y) status before STOP

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



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

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

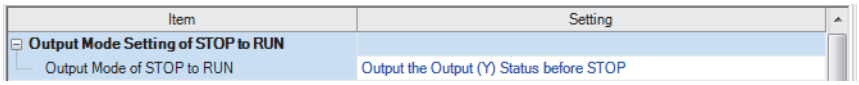


Setting the output mode

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

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

Window



Displayed items

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

10.3 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 182 Scan time monitoring time setting)

Momentary power failure exceeding the allowable momentary power failure time

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

11 MEMORY CONFIGURATION OF THE CPU MODULE

11.1 Memory Configuration

The following shows the memory configuration of the CPU module.

Point

- The usage of the memory can be checked from the engineering tool. (GX Works3 Operating Manual)
- The number of writes for the program memory and data memory is limited to 100000 times. When using the function that the target file is written, pay attention to the number of writes. (☞ Page 158 File types and storage memory) The current number of writes for the program memory and data memory can be checked in the special register (SD630/SD631 and SD634/SD635). (☞ Page 510 Drive information)

Program memory/program cache memory

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

*1 This memory is used for program operations.

- Powered-on
- Reset operation

Application

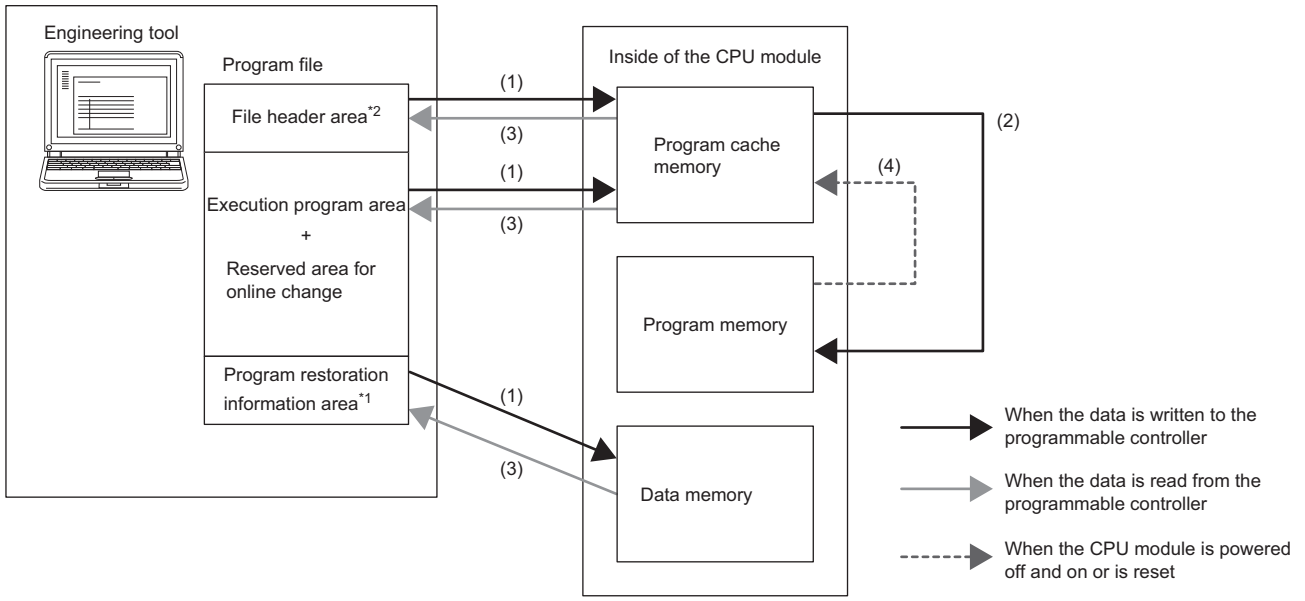
The program memory and program cache memory store execution programs of program files and FB (function block) files. Execution programs include instruction codes, statements, and notes.

Data to be allocated

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

Data allocation and procedure of read/write operations

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



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

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

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

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

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

Point

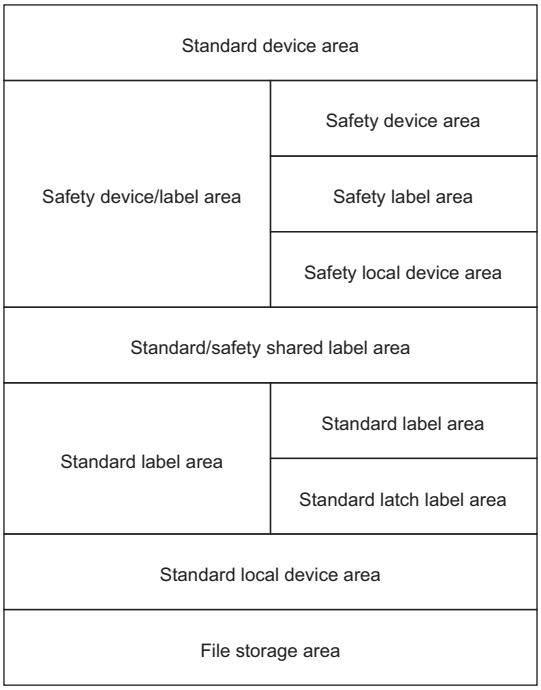
For read/write operations from/to the programmable controller, refer to the following.

GX Works3 Operating Manual

Device/label memory

In addition to standard devices and standard labels, the data such as safety devices and safety labels are allocated to each data area of the device/label memory.

The safety device area, safety label area, safety local device area, and standard/safety shared label area are allocated between the standard device area and standard label area. The capacities for each area can be changed in the device/label memory area setting by using the engineering tool. (☞ Page 148 Device/label memory area setting)



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.

Device/label memory area setting

The capacity of each area on the device/label memory can be specified.

■Default capacity of each area

The default capacity of each area is as follows.

Item	R08PSFCPU	R16PSFCPU	R32PSFCPU	R120PSFCPU
Standard device area	40K words	40K words	40K words	40K words
Safety device area	20K words	20K words	20K words	20K words
Safety label area	20K words	20K words	20K words	20K words
Safety local device area	0K word	0K word	0K word	0K word
Standard/safety shared label area	10K words	10K words	10K words	10K words
Standard label area	40K words	50K words	90K words	110K words
Standard latch label area	2K words	2K words	4K words	4K words
Standard local device area	0K word	0K word	0K word	0K word
File storage area	457K words	713K words	969K words	1481K words

Point

The standard local device area capacity obtained by subtracting the total capacity of the standard device area, standard label area, standard latch label area, file storage area, and safety device/safety label area capacity, and the total standard/safety shared label area from the total device/label memory is set. Note, however, that even if the total capacity of the standard device area, standard label area, and safety device/safety label area capacity is smaller than the following, the remaining capacity cannot be assigned to the standard local device area. (The area which is smaller than the following is the area not used.)

- R08PSFCPU: 50K words
- R16PSFCPU: 60K words
- R32PSFCPU: 70K words
- R120PSFCPU: 90K words

Point

The safety local device area capacity obtained by subtracting the total capacity of the safety device area and safety label area from the safety device/label memory capacity is set.

■Setting range of capacity of each area

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

*1 The remaining capacity of other areas is automatically set as the capacity of the standard local device area and safety local device area. (☞ Page 148 Default capacity of each area)

- R08PSFCPU

Area	Setting range of capacity of each area		
	Without an extended SRAM cassette	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 588K words	2 to 1612K words	2 to 4684K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 586K words	0 to 1610K words	0 to 4682K words
Standard latch label area	0 to 544K words	0 to 1568K words	0 to 4640K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 544K words	0 to 1568K words	0 to 4640K words

• R16PSFCPU

Area	Setting range of capacity of each area		
	Without an extended SRAM cassette	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 854K words	2 to 1878K words	2 to 4950K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 852K words	0 to 1876K words	0 to 4948K words
Standard latch label area	0 to 800K words	0 to 1824K words	0 to 4896K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 800K words	0 to 1824K words	0 to 4896K words

• R32PSFCPU

Area	Setting range of capacity of each area		
	Without an extended SRAM cassette	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 1152K words	2 to 2176K words	2 to 5248K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 1150K words	0 to 2174K words	0 to 5246K words
Standard latch label area	0 to 1088K words	0 to 2112K words	0 to 5184K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 1088K words	0 to 2112K words	0 to 5184K words

• R120PSFCPU

Area	Setting range of capacity of each area		
	Without an extended SRAM cassette	With an extended SRAM cassette (2MB)	With an extended SRAM cassette (8MB)
Standard device area	2 to 1684K words	2 to 2708K words	2 to 5780K words
Safety device area	1 to 40K words	1 to 40K words	1 to 40K words
Safety label area	0 to 39K words	0 to 39K words	0 to 39K words
Safety local device area	Setting disabled	Setting disabled	Setting disabled
Standard/safety shared label area	0 to 40K words	0 to 40K words	0 to 40K words
Standard label area	0 to 1682K words	0 to 2706K words	0 to 5778K words
Standard latch label area	0 to 1600K words	0 to 2624K words	0 to 5696K words
Standard local device area	Setting disabled	Setting disabled	Setting disabled
File storage area	0 to 1600K words	0 to 2624K words	0 to 5696K words

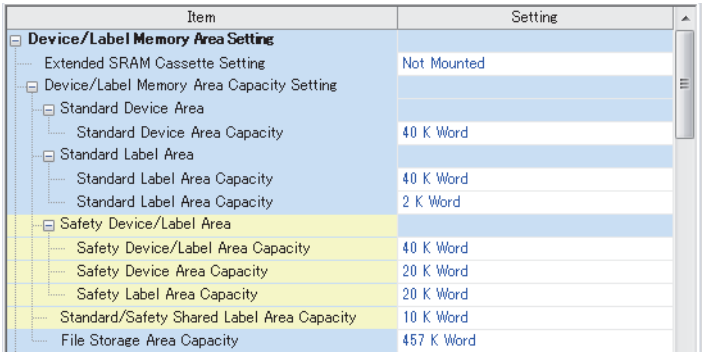
■Setting method

The capacity of each data area allocated within 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			When using an extended SRAM cassette, select the capacity of the cassette.	• Not Mounted • 2MB*1 • 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 148 Setting range of capacity of each area	☞ Page 148 Default capacity of each area
	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		Sets the capacity of the file storage area to be used for storing register files and others.		

*1 Check the version of the engineering tool. (☞ Page 566 Added and Enhanced Functions)

Point

Note that the total of the capacity of each area (including the capacity of the local device area/safety local device area) and safety special relay/safety special register area capacity should not exceed the capacity of the device/label memory. (☞ Page 64 Hardware specifications)

• 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*2	Increment for setting
Bit	Input	X	X0 to X2FFF	—
	Output	Y	Y0 to Y2FFF	—
	Internal relay	M	M0 to M94674943	64 points
	Link relay	B	B0 to B5A49FFF	64 points
	Annunciator	F	F0 to F32767	64 points
	Link special relay	SB	SB0 to SB5A49FFF	64 points
	Edge relay	V	V0 to V32767	64 points
	Latch relay	L	L0 to L32767	64 points
Word	Timer	T	T0 to T5259711	32 points
	Retentive timer	ST	ST0 to ST5259711	32 points
	Long timer	LT	LT0 to LT1479295	1 point
	Long retentive timer	LST	LST0 to LST1479295	1 point
	Counter	C	C0 to C5259711	32 points
	Long counter	LC	LC0 to LC2784543	32 points
	Data register	D	D0 to D5917183	4 points
	Link register	W	W0 to W5A49FF	4 points
	Link special register	SW	SW0 to SW5A49FF	4 points

*2 This range is the maximum when the R120PSFCPU with an extended SRAM cassette (8MB) (NZ2MC-8MBSE) is used. The number of points varies depending on the model of the CPU module and whether to use an extended SRAM cassette.

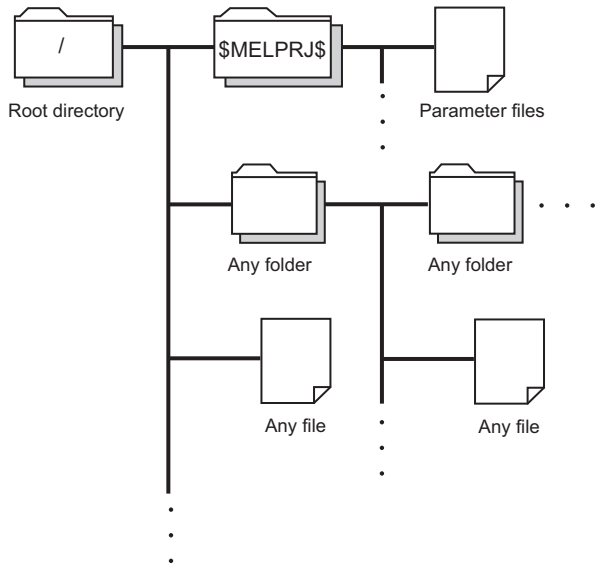
• Safety device area setting range

The number of points of each safety device used in safety programs and capacity in which the total number of safety device points can be stored are set. Set the total number of safety device points within the safety device area range.

Type	Device name	Symbol	Range of use	Increment for setting
Bit	Safety input	SA\X	SA\X0 to SA\X2FFF	—
	Safety output	SA\Y	SA\Y0 to SA\Y2FFF	—
	Safety internal relay	SA\M	SA\M0 to SA\M638975	64 points
	Safety link relay	SA\B	SA\B0 to SA\B9BFFF	64 points
Word	Safety timer	SA\T	SA\T0 to SA\T35487	32 points
	Safety retentive timer	SA\ST	SA\ST0 to SA\ST35487	32 points
	Safety counter	SA\C	SA\C0 to SA\C35487	32 points
	Safety data register	SA\D	SA\D0 to SA\D39935	4 points
	Safety link register	SA\W	SA\W0 to SA\W9BFF	4 points

Data memory

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



Point

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

Precautions

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

Refresh memory

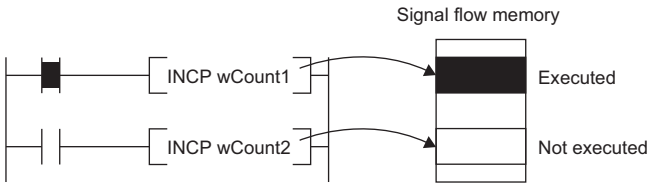
This memory is used to store data used to refresh communication with the intelligent function module. (☞ Page 318 Refresh Data Register (RD))

CPU buffer memory

This memory is used by the Ethernet function. (☞ Page 533 Buffer Memory)

Signal flow memory

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



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

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

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

MELSEC iQ-R Programming Manual (Program Design)

Point

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

- Contact instruction: LDP, LDF, ANDP, ANDF, ORP, ORF, LDPI, LDFI, ANDPI, ANDFI, ORPI, and ORFI
- Association instruction: MEP, MEF
- Output instruction: SET F, RST F, PLS, PLF, FF
- Rising edge execution instruction: □P (such as INCP and MOVP), SP.□, ZP.□, GP.□, JP.□, SORTD(_U), DSORTD(_U), LEDR, DUTY, TIMCHK, XCALLSCJ

SD memory card

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

Point

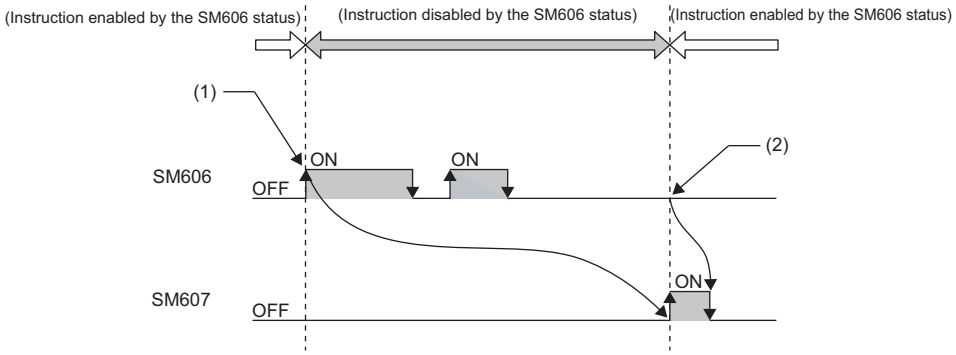
For details on how to insert or remove the SD memory card, refer to the following.

☞ Page 48 Inserting and Removing an SD Memory Card

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

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

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



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

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

Point

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

■Operation of each function accessing the SD memory card

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

Function being executed	SD memory card forced disable instruction executed during access to the SD memory card	Access made to the SD memory card after the SD memory card is disabled
• Access to device comment or label in SD memory card • Device/label initial value operation at status change from STOP to RUN	The SD memory card becomes disabled after the executed function is completed.	An error occurs in the CPU module.*1
Access to SD memory card by engineering tool, SLMP, FTP function	The function is responded to with an error.	The function is responded to with an error.
Instruction to access SD memory card	The SD memory card becomes disabled after the instruction is completed.	The instruction is completed with an error.
iQ Sensor Solution data backup/restoration function	After the backup or restore operation is completed, the SD memory card is disabled.	The function is responded to with an error.

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

Precautions

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

■Precautions when forcibly disabling the SD memory card

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

11.2 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
R08PSFCPU	128 bytes	512 bytes	2048 bytes
R16PSFCPU			4096 bytes
R32PSFCPU			8192 bytes
R120PSFCPU			16384 bytes

Point

Data is written in the unit of the file size (cluster size). For example, when 464 bytes of CPU parameter is written to the data memory on the R08PSFCPU, it is written as 2048 bytes of data because the file size unit of the data memory is 2048 bytes.

File size unit of each SD memory card

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

11.3 Memory Operation

Initialization and value clear

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

GX Works3 Operating Manual

Items to be specified in the engineering tool				Target
Initialization	Data memory			Deletes all the folders and files in the program memory and data memory.
	Device/label memory			Deletes all the files in the file storage areas in the device/label memory.
	SD memory card			Deletes all the folders and files in the SD memory card.
Value clear	Device, label	Zero clear		Excluding devices and labels with latch specified, clears the following to zero: X, Y, M, B, F, SB, V, 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 file to zero.
		Zero clear excluding Latch (2)		Clears the file register files other than Latch (2) to zero.
	Device / label / file register latch clear			Clears devices, labels, and file 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 CPU module. (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).

11.4 Files

This section lists the files used by the CPU module.

File types and storage memory

The following table summarizes the types of files stored in the CPU module and 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	
Standard program		◎*3*5	×	◎*3*5	×	ANY_STRING.PRG
Standard FB file		○*3	×	○*3	×	ANY_STRING.PFB
Standard CPU parameter		×	×	◎	×	CPU.PRM
System parameter		×	×	◎	×	SYSTEM.PRM
Standard module parameter		×	×	○	×	UNIT.PRM
Module extension parameter		×	×	○	○	• UEXmmmnn.PRM*1 • UEXmmm00.PPR*4
Module-specific backup parameter*8		×	×	○	○	UBPmmmnn.BPR
Memory card parameter		×	×	×	○	MEMCARD.PRM
Device comment		×	×	○	○	ANY_STRING.DCM
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		×	○	×	○*2	ANY_STRING.QDR
Event history		×	×	○	○	EVENT.LOG
Device data storage file		×	×	○	○*2	DEVSTORE.QST
General-purpose data		×	×	○	○	ANY_STRING.CSV/BIN
Remote password		×	×	○	○	00000001.SYP
Safety program*6		◎*3*5	×	◎*3*5	×	ANY_STRING.SPG
Safety FB file*6		○*3	×	○*3	×	ANY_STRING.SPB
Safety CPU parameter*6*7		×	×	◎	×	CPU.SPR
Safety module parameter*6		×	×	○	×	UNIT.SPR
Safety global label setting file*6		×	×	○	×	GLBLINF.SIF
Standard/safety shared label setting file*6		×	×	○	×	S_GLBLINF.SIF

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

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

*3 When this file is stored in the built-in memory of the CPU module, it is divided into program memory and data memory and stored. (☞ Page 162 Configuration of a program file)

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

*5 One or more of either standard program or safety program is required.

*6 The file is also stored in the SIL2 function module.

*7 This parameter relates to safety control inside the CPU parameters. (☞ Page 559 Safety CPU parameters)
This item is yellow in the engineering tool CPU parameters.

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

File operation available

This section describes file operations which are available for each file type.

In TEST MODE

The following table lists file operations which are available for each file type in TEST MODE.

○: Available, ×: Not available, —: N/A

File type		Operation from engineering tool			Operation with SLMP and FTP server function			Operation with instructions in standard programs	
		Write	Read	Delete	Write	Read	Delete	Write	Read
Standard program		○*1	○	○*2	×	×	×	—	—
Standard FB file		○*2	○	○*2	×	×	×	—	—
Standard CPU parameter		○*2	○	○*2	×	×	×	—	—
System parameter		○*2	○	○*2	×	×	×	—	—
Standard module parameter		○*2	○	○*2	×	×	×	—	—
Module extension parameter		○*2	○	○*2	×	×	×	—	—
Module-specific backup parameter		○	○	○	×	×	×	○	○
Memory card parameter		○*2	○	○*2	×	×	×	—	—
Device comment		○	○	○*2	×	×	×	—	—
Initial device value		○	○	○*2	×	×	×	—	—
Standard global label setting file		○*4	○	○*2	×	×	×	—	—
Initial label value file	Initial global label value file	○	○	○*2	×	×	×	—	—
	Initial local label value file	○	○	○*2	×	×	×	—	—
File register		○	○	○*2	×	×	×	○	○
Event history		—	—	—	×	×	×	—	—
Device data storage file		○*4*5	○	○*4*5	×	×	×	○	○
General-purpose data		○	○	○	○	○	○	○	○
Remote password		○*2	○	○*2	×	×	×	—	—
Safety program		○*2	○	○*2	×	×	×	—	—
Safety FB file		○*2	○	○*2	×	×	×	—	—
Safety CPU parameter*3		○*2	○	○*2	×	×	×	—	—
Safety module parameter		○*2	○	○*2	×	×	×	—	—
Safety global label setting file		○*2	○	○*2	×	×	×	—	—
Standard/safety shared label setting file		○*2	○	○*2	×	×	×	—	—

*1 When the CPU module is in the STOP state, the file operation is performed by writing the file to the programmable controller. In the RUN state, it is performed by online change. Note that only the standard programs registered in the parameter can be written when the operating status of the CPU module is RUN.

*2 Available only when the CPU module operating status is STOP/PAUSE. If any file operation is performed in the RUN state, the operating status of the CPU module is changed by the remote STOP function, and then the operation continues.

*3 This parameter relates to safety control inside the CPU parameters. (☞ Page 559 Safety CPU parameters)
This item is yellow in the engineering tool CPU parameters.

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

*5 When the target memory for the file operation is the SD memory card, the operation can be performed even while the operating status of the CPU module is RUN.

In SAFETY MODE

The following table lists file operations which are available for each file type in SAFETY MODE.

○: Available, ×: Not available, —: N/A

File type		Operation from 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*4	○	○*2	×	×	×	—	—
Standard CPU parameter		×	○	×	×	×	×	—	—
System parameter		×	○	×	×	×	×	—	—
Standard module parameter		×	○	×	×	×	×	—	—
Module extension parameter		○*2	○	○*2	×	×	×	—	—
Module-specific backup parameter		○	○	○	×	×	×	○	○
Memory card parameter		○*2	○	○*2	×	×	×	—	—
Device comment		○	○	○*2	×	×	×	—	—
Initial device value		○	○	○*2	×	×	×	—	—
Standard global label setting file		○*2*4	○	○*2	×	×	×	—	—
Initial label value file	Initial global label value file	○	○	○*2	×	×	×	—	—
	Initial local label value file	○	○	○*2	×	×	×	—	—
File register		○	○	○*2	×	×	×	○	○
Event history		—	—	—	×	×	×	—	—
Device data storage file		○*5*6	○	○*5*6	×	×	×	○	○
General-purpose data		○	○	○	○	○	○	○	○
Remote password		○*2	○	○*2	×	×	×	—	—
Safety program		×	○	×	×	×	×	—	—
Safety FB file		×	○	×	×	×	×	—	—
Safety CPU parameter*3		×	○	×	×	×	×	—	—
Safety module parameter		×	○	×	×	×	×	—	—
Safety global label setting file		×	○	×	×	×	×	—	—
Standard/safety shared label setting file		×	○	×	×	×	×	—	—

*1 When the CPU module is in the STOP state, the file operation is performed by writing the file to the programmable controller. In the RUN state, it is performed by online change. Note that only the standard programs registered in the parameter can be written when the operating status of the CPU module is RUN.

*2 Available only when the CPU module operating status is STOP/PAUSE. If any file operation is performed in the RUN state, the operating status of the CPU module is changed by the remote STOP function, and then the operation continues.

*3 This parameter relates to safety control inside the CPU parameters. (☞ Page 559 Safety CPU parameters)
This item is yellow in the engineering tool CPU parameters.

*4 Writing is possible in file units, however, it is not possible to select files together with parameter files or files relating to safety and then write to the programmable controller. If writing in file units, write after clearing the selection of other parameter files or files relating to safety, or write after switching to TEST MODE.

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

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

File size

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

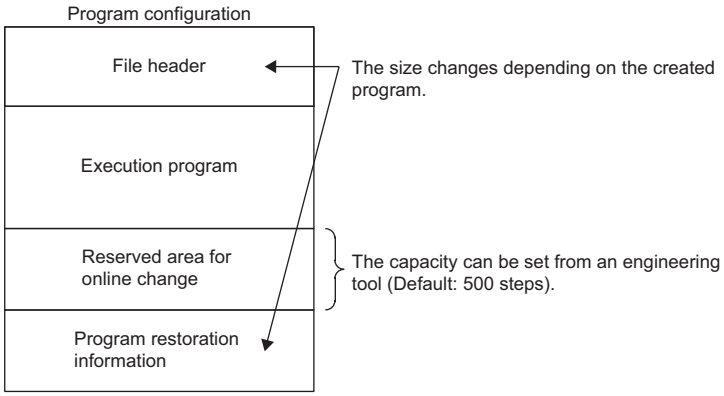
File type		File size
Standard program		Approx. 4248 bytes minimum (only END instruction + 500 steps reserved for online program change)
Standard FB file		Approx. 4984 bytes minimum (non-processing FB + 500 steps reserved for online program change)
Standard CPU parameter		824 bytes minimum
System parameter		112 bytes minimum
Standard module parameter		The size differs depending on the module used. For example, the size is 1036 bytes at minimum when the Ethernet function is used.
Module extension parameter		The size differs depending on the module used. For example, the size is 65572 bytes when the predefined protocol support function is used with the Ethernet function.
Module-specific backup parameter		Manual for the module used
Memory card parameter		124 bytes minimum
Device comment		220 bytes minimum (when a device comment with 20 characters is set in a device)
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.
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)

Configuration of a program file

The following figure shows the configuration of a program.

■Configuration of a program

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

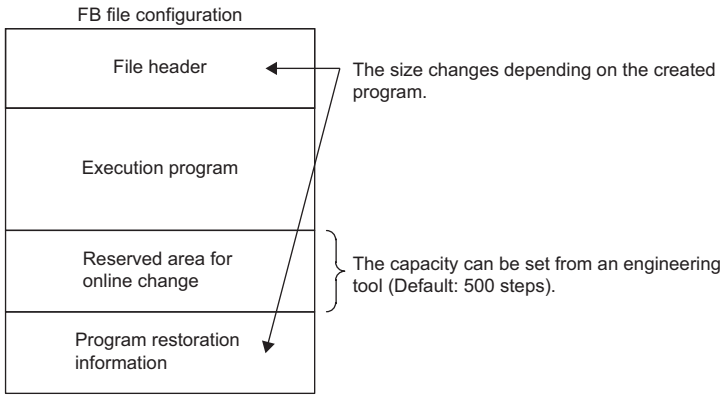


When the program is stored in the CPU module, file header, execution program, and reserved area for online change are allocated to the program memory. Program restoration information is allocated to the data memory. The following table shows the destination memory to which each area is allocated when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

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

■FB file

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



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

PART 6 FUNCTIONS

This part consists of the following chapters.

12 FUNCTION LIST

13 CLOCK FUNCTION

14 WRITING DATA TO THE CPU MODULE

15 RAS FUNCTIONS

16 REMOTE OPERATION

17 PID CONTROL/PROCESS CONTROL FUNCTION

18 SECURITY FUNCTION

19 ROUTING SETTING

20 REDUNDANT FUNCTION

21 SWITCHING SAFETY OPERATION MODE

22 IDENTIFICATION CHECK FOR SAFETY DATA

23 SAFETY COMMUNICATION FUNCTION

24 SLMP COMMUNICATIONS

25 PRECAUTIONS ON PROGRAMMING

12 FUNCTION LIST

The following table lists the functions of the CPU module.

Restriction

Some functions have restrictions on the firmware version of the CPU module used or the version of the engineering tool used. (☞ Page 566 Added and Enhanced Functions)

Function		Description	Reference
Constant scan		Repeatedly runs a program while retaining the scan time constant.	Page 95 Constant scan
Device/label access service processing setting		Sets the time or execution timing of the device/label access service processing performed in the END processing in parameter.	Page 97 Device/label access service processing setting
Interrupt function	Multiple interrupt function	When an interrupt occurs while an interrupt program triggered by another cause is running, stops the program if its priority is lower than that of the new interrupt, and runs the higher-priority program whenever its execution condition is satisfied.	Page 133 Multiple interrupt function
Output mode setting when the status changed from STOP to RUN		Sets the output (Y) status when the operating status of the CPU module is switched from STOP to RUN.	Page 142 Output mode at operating status change (STOP to RUN)
Device/label memory area setting		Sets the capacity of each area in the device/label memory.	Page 148 Device/label memory area setting
SD memory card forced disable		Forcibly disables an access to the SD memory card without powering off the system even when the function that accesses the SD memory card is being executed.	Page 154 How to forcibly disable the SD memory card with a special relay
Clock function		Manages time for the system functions such as time stamps for the event history function and date information for the data logging function.	Page 167 CLOCK FUNCTION
Writing data to the CPU module	Writing data to the programmable controller	Writes the specified data of the project with the engineering tool to the memory of the CPU module.	• Page 173 Writing Data to the Programmable Controller • 📖GX Works3 Operating Manual
	Online change (ladder block)	Writes the part of a program edited on the ladder editor using the engineering tool to the running CPU module in units of ladder blocks. Edited contents spanning multiple files or multiple portions can be written to the CPU module at once.	• Page 173 Online change (ladder block) • 📖GX Works3 Operating Manual
	Online change (files)	Writes programs and other data to the running CPU module in units of files.	• Page 176 File batch online change • 📖GX Works3 Operating Manual
RAS function	Scan monitoring function	Detects a hardware failure or program error by monitoring that the END processing is performed within a set scan time.	Page 182 Scan Monitoring Function
	Self-diagnostics function	Self-diagnoses the CPU module to see whether an error exist or not.	Page 184 Self-Diagnostics Function
	Error clear	Batch-clears all the continuation errors being detected.	Page 190 Error Clear
	Event history function	Collects operations executed and errors detected from the modules, and saves them in the CPU module. The saved logs can be checked in chronological order.	Page 194 Event History Function
	Program cache memory auto recovery function	Automatically detects and recovers the data stored in the program cache memory of the CPU module at the time of program execution if the data is overwritten due to the factors such as excessive electrical noise.	Page 199 Program Cache Memory Auto Recovery Function
Remote operation		Remotely controls the operating status of the CPU module without switch operation.	Page 200 REMOTE OPERATION
PID control function		Performs PID control by using the PID operation instruction.	📖MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)
Process control function		A process control program can be easily created by using process control function blocks.	📖MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions)

Function		Description	Reference
Security function	Block password function	Prevents unauthorized access to programs (in units of POUs).	GX Works3 Operating Manual
	Security key authentication function	Prevents unauthorized access to programs (in units of program files) or unauthorized execution of programs.	
	File password function	Prevents unauthorized reading/writing of data from/to files.	
	IP filter function	Identifies the IP address of external devices over Ethernet, and blocks access from an invalid IP address.	• GX Works3 Operating Manual • MELSEC iQ-R Ethernet User's Manual (Application)
	Remote password function	Limits access from external devices to the CPU module to the specific communication route via Ethernet.	• GX Works3 Operating Manual • MELSEC iQ-R Ethernet User's Manual (Application)
	User authentication function of CPU modules	Prevents an unauthorized access to a program or parameters written to a CPU module. Operations are restricted by registering a user name/password for a CPU module.	Page 209 User Authentication Function of CPU Modules
Routing setting		Sets the communication path required for transient communication to the stations on the different network.	Page 210 ROUTING SETTING
Redundant function	System switching	Switches the systems between the control system and the standby system to continue operation of the redundant system when a failure or an error occurs in the control system. For debugging and maintenance purpose, users can switch the systems at any desired timing.	Page 212 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 225 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 242 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 247 System Consistency Check
	Program execution in both systems	Detects an error in the external device or network of the systems (control system and standby system) respectively by executing a program that diagnoses external devices and networks in both systems.	Page 252 Program Execution in Both Systems
	Redundant system operation setting	Sets the redundant system operation in the redundant system settings of the CPU parameter.	Page 258 Redundant System Operation Setting
	Redundant function module communication test	Checks the redundant function module itself for errors when its communications are unstable.	Page 261 Redundant Function Module Communication Test
Safety operation mode		There are two modes for the CPU module: 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 262 SWITCHING SAFETY OPERATION MODE
Identification check for safety data		Checks if the project data created using the engineering tool and the data in the CPU module are the same, and confirms that the program executed in SAFETY MODE is the one written by the user.	Page 265 IDENTIFICATION CHECK FOR SAFETY DATA
Safety communication function		Communicates data between the CPU module and modules supporting safety functions using safety protocols.	Page 267 SAFETY COMMUNICATION FUNCTION
Label access setting from external device		Enables communications specifying the global label name from the GOT and external devices using SLMP.	Page 347 Label Access Setting from External Device
Latch function	Latch with a battery	Holds the device/label data in the CPU module even at power-off.	Page 353 Latch with Battery
Initial device/label value setting		Sets the initial values of devices and labels used in the program directly (not via the program) to the devices, labels, and buffer memory areas of intelligent function modules.	Page 360 DEVICE/LABEL INITIAL VALUE SETTINGS

Function		Description	Reference
Label initialization function	Label initialization after converting all programs (reassignment)	All the label areas will be initialized (initial values are set if the values have been set, or the labels are cleared to zero if not) when the CPU module is powered off and on or the operating status of the CPU module is switched from STOP to RUN after data is rebuilt (reassigned) with the engineering tool and then written to the programmable controller.	Page 364 Initialization of Labels After Rebuilt All (Reassignment)
	Label initial value reflection setting	With the default settings, initial label values are not set in labels when the operating status of CPU module is switched from STOP to RUN even though the label initial values have been set for the labels. This function can set whether or not to set label initial values when the operating status of the CPU module is switched from STOP to RUN.	Page 366 Label Initial Value Reflection Setting
Ethernet function		Accesses the CPU module over Ethernet. Using this function, the CPU module can connect to MELSOFT products and GOTs, perform socket communications, and transfer FTP files.	MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)
SLMP communication function		Enables accesses to devices/labels and remote operations from external devices other than the engineering tool by sending request messages using SLMP.	SLMP Reference Manual

13 CLOCK FUNCTION

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

13.1 Time Setting

The clock continues operating with the internal battery of the CPU module while the CPU module is powered off or during power failure longer than the allowable momentary power failure time.
The time of the standby system is synchronized with the time of the control system. The time setting using the time setting function (SNTP client) is available only for the control system.

Clock data

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

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

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

Changing the clock data

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

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

Point

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

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

Using the engineering tool

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

Using SM/SD

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

Using the instruction

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

Automatic change using the SNTP server

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

Reading the clock data

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

- Using SM/SD
- Using the instruction

Using SM/SD

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

Using the instruction

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

Precautions on the clock data

The following lists the precautions on the clock data.

When using the product for the first time

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

Modifying the clock data

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

Range of the clock data

The clock data must be written within the following range.

Page 167 Clock data

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

Ex.

Operating status of the CPU module when impossible date is set

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

Execution target specification

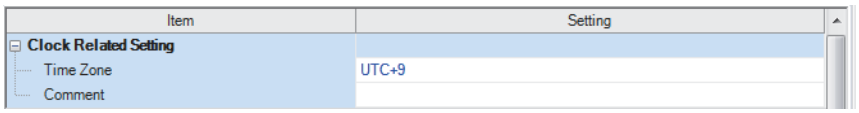
When setting clock data for the Safety CPU at the "Set Clock" window of the engineering tool, "All Stations Specified" cannot be selected as an execution target.

13.2 Setting Time Zone

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

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

Window



Displayed items

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

Point

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

13.3 System Clock

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

Special relay/safety special relay used for system clock

Special relay used for system clock are as follows (☞ Page 482 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
SA\SM400	Always On
SA\SM401	Always Off
SA\SM444	On at the first safety program execution
SA\SM1800	On for only the first safety program execution after system switching (standby system to control system)

Precautions

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

Special register used for system clock

Special register used for system clock are as follows (☞ Page 507 System clock)

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

14 WRITING DATA TO THE CPU MODULE

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

14.1 Writing Data to the Programmable Controller

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

GX Works3 Operating Manual

14.2 Online Change

The following table lists the types of the online change.

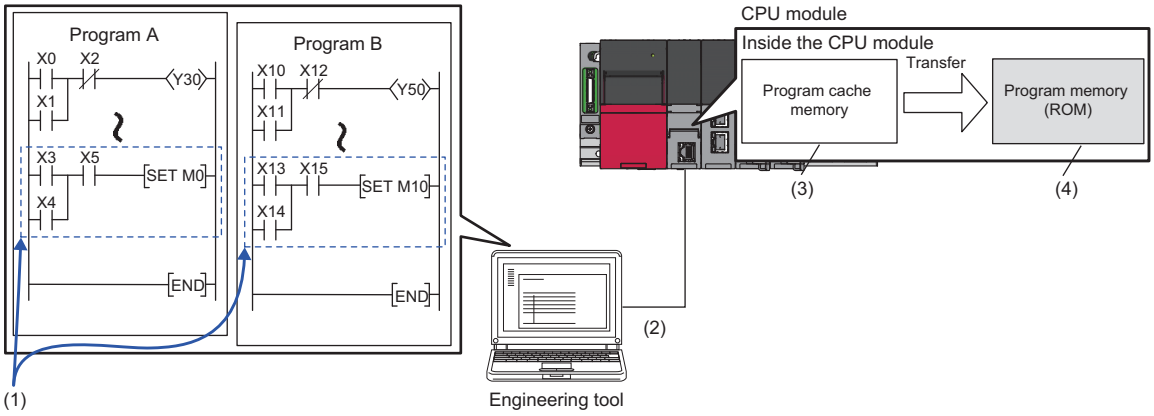
Type		Description	Reference
Online change*1	Online change (ladder block)*2	Changes and writes a part of the program and data online.	• Page 173 Online change (ladder block) • GX Works3 Operating Manual
File batch online change		Writes data in file units online.	• Page 176 File batch online change • GX Works3 Operating Manual

*1 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.

*2 Online changes (ladder block) are not possible in safety programs.

Online change (ladder block)

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



(1) Portion edited in the engineering tool

(2) The changed ladder block is written to the running CPU module.

(3) The program contents in the program cache memory are changed.

(4) After changing the program contents in the program memory, they are automatically transferred to the program memory.

Point

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

GX Works3 Operating Manual

Editable contents

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

■Editable contents within a program block

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

■Editable contents within FB definition

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

Item	Reference
Precautions for adding a local label	📖Precautions when local labels are added into the MELSEC iQ-R series function blocks (FA-A-0232)
Instructions that refer to the signal flow memory	☞ Page 153 Signal flow memory

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

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

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

■Editable contents within FUN definition

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

■Editable contents for each program component

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

Range changeable in a single session

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

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

Reserved area for online change

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

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

Required free space in the data memory

If the data memory does not have enough free space when the online change (ladder block) is executed, an error occurs. The free space of the data memory for the online change (ladder block) is equal to the size of program restoration information. The size of the program restoration information can be checked from the memory capacity calculation of the engineering tool.

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

Setting the initial value for registering/changing label definition

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

■Initial value setting availability

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

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

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

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

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

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

Reflecting program change to other systems

When the online change (ladder block) is performed on the CPU module in one system, the change is also reflected on the CPU module in the other system.

Writing of program restoration information

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.

File batch online change

This function writes programs and other data to the running CPU module in units of files.

For the operating procedure and the execution condition of the file batch online change, refer to the following.

GX Works3 Operating Manual

14.3 Precautions

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

Prohibited operation (Turning off or resetting the CPU modules)

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

Online change failure

During an online change, avoid the following.

- Switching the operating status of the CPU module from STOP (PAUSE) to RUN
- Switching systems
- Disconnection of the tracking cable
- Power-off or reset of the CPU module in the control system or standby system

When any of the above conditions is satisfied during an online change, a file mismatch is detected in the system consistency check or the online change fails.

■Action for when an online change has failed

If an online change has failed, take the following actions.

1. Get the CPU module ready for an online change and execute the online change with the same data as before the failure.

- If the online change succeeds, the action is completed.
- If the online change fails again, take the following actions.

2. Verify the data in the CPU module of the control system with the programmable controller from the engineering tool, and check whether the online change has been completed successfully.

- When the verification result is mismatched, the online change has not been completed successfully. Connect the engineering tool to the CPU module of the control system, execute an online change only to the control system, and proceed to step 3.
- When the verification result is matched but the program memory transfer window was not displayed at the online change, the possible cause is that data transfer to the program memory has not been completed successfully. Batch-write files only to the control system, and proceed to step 3.
- When the verification result is matched and the program memory transfer window is displayed (the online change is completed successfully) during online change, proceed to step 3.

3. Execute the memory copy from the control system to the standby system. When the memory copy has failed, power off and on or reset the CPU module of the standby system and execute the memory copy again. (☞ Page 242 Memory Copy from Control System to Standby System)

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

Operation from engineering tools

■Operations cannot be executed simultaneously

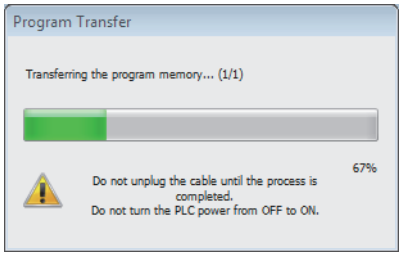
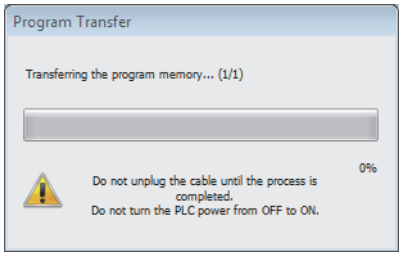
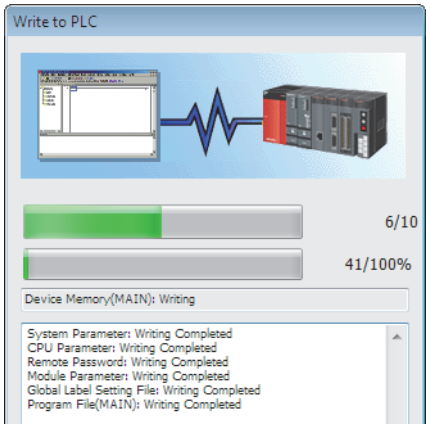
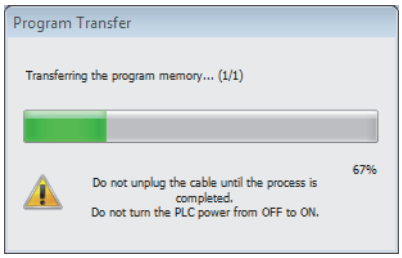
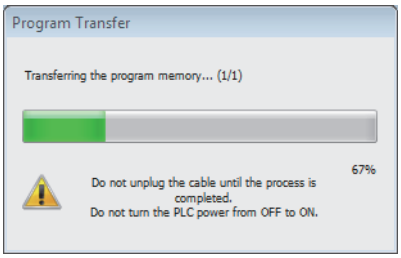
The file batch online change or online change (ladder block) cannot be executed simultaneously with the following operations from the engineering tool.

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

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

■Writing during the program memory transfer

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

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

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

When the online change (ladder block) is used

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

■When deleting the OUT instruction which is on

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

■Program file not registered in program setting

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

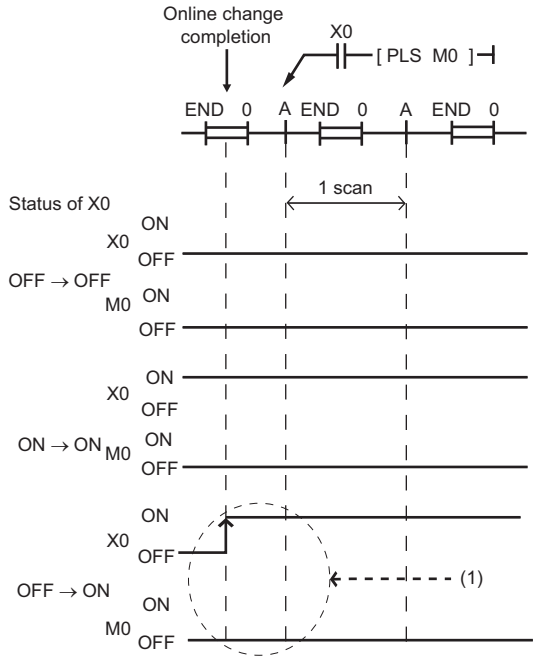
■Instructions which do not operate correctly

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

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

- Rising instruction

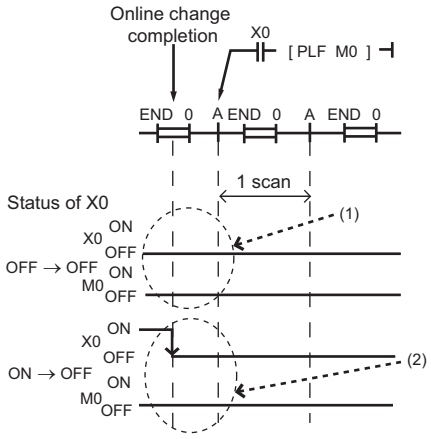
When a rising instruction exists within the range to be changed, the rising instruction will not be executed even if the execution condition (OFF to ON) is satisfied after completion of the online change (ladder block) or writing data to the programmable controller.



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

• Falling instruction

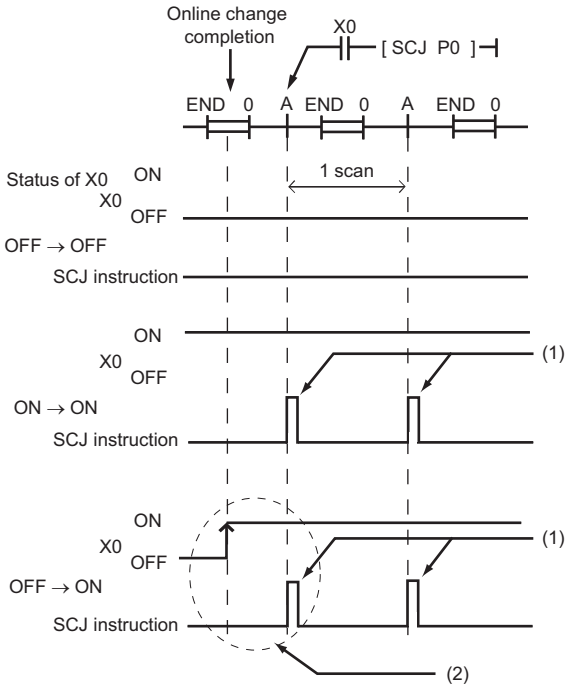
When a falling instruction exists within the range to be changed, the falling instruction will not be executed even if the execution condition (ON to OFF) is satisfied after completion of the online change (ladder block) or writing data to the programmable controller.



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

• SCJ instruction

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



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

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

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

■Scan monitoring function during online changes

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

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

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

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

Note the following:

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

☞ Page 174 Editable contents

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

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

During the file batch online change

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

■Writing the label data

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

■Falling instruction in the subroutine type FB program

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

15 RAS FUNCTIONS

15.1 Scan Monitoring Function

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

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

Scan time monitoring time setting

Set the scan time monitoring time.

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

Window



Displayed items

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

Watchdog timer reset

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

Precautions

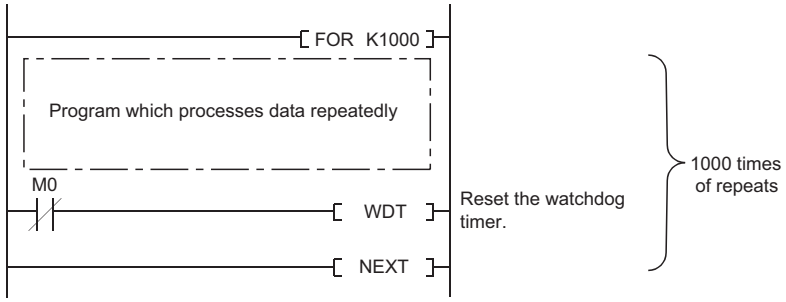
The following lists the precautions on the scan monitoring function.

Measurement error of watchdog timer

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

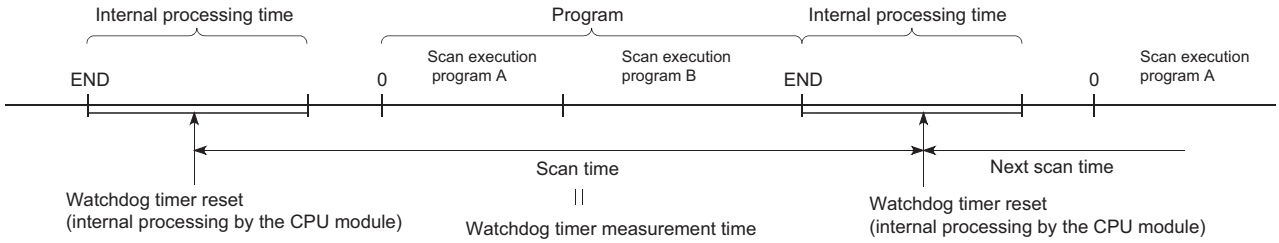
Resetting the watchdog timer when repeatedly executing a program

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



Scan time when the WDT reset instruction is used

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



Precautions specific to redundant system

- 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 to be received. Therefore, a time taken for tracking data to be received does not need to be considered in the scan time monitoring time setting.

15.2 Self-Diagnostics Function

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

Self-diagnostics timing

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

How to check for errors (CPU module)

The following describes how to check for errors in the CPU module.

Check method using the special relay and special register

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

Check method using LED

The error occurrence conditions can be checked through the lighting conditions of ERROR LED. (☞)Page 375 LED status of the CPU module

Check method using the engineering tool

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

■Existing errors

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

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

■Error history

Occurred errors are logged in the event history. (☞ Page 194 Event History Function)

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

How to check for errors (SIL2 function module)

The following describes how to check for errors in the SIL2 function module.

Check method using the buffer memory

When the SIL2 function module detects an error, the corresponding error code is stored in Un\G0 (Latest self-diagnostics error code). If multiple errors are detected, the latest error code is stored in Un\G0. Up to 16 error codes can be stored in Un\G10 (Self-diagnostics error code 1) to Un\G25 (Self-diagnostics error code 16). (The error codes of the 17th error and later are not stored.)

Check method using LED

Like intelligent function modules, the error status can be checked with the ERROR LED. (☞ Page 375 LED status of the CPU module)

Check method 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)

CPU module operation upon error detection setting

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

Mode when an error is detected

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

■Mode for stopping the operation of CPU module

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

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

Point

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

■Mode for continuing the operation of CPU module

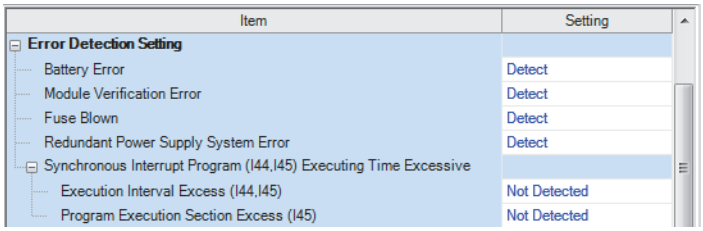
If an error has been detected, the program other than programs (instruction) where an error was generated is executed.

Error detection setting

Set whether or not to detect errors.

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

Window



Displayed items

Item	Description	Setting range	Default
Battery Error	Sets whether or not to detect the battery error.	• Detect • Not Detected	Detect
Module Verification Error*1	Sets whether or not to detect a module verification error.		
Fuse Blown	Sets whether or not to detect a fuse blown of the controlled module.		
Redundant Power Supply System Error*2	Sets whether or not to detect a power supply module error (power off or power supply failure in either system) in the redundant power supply system.		

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

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

■Applicable errors to the error detection setting

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

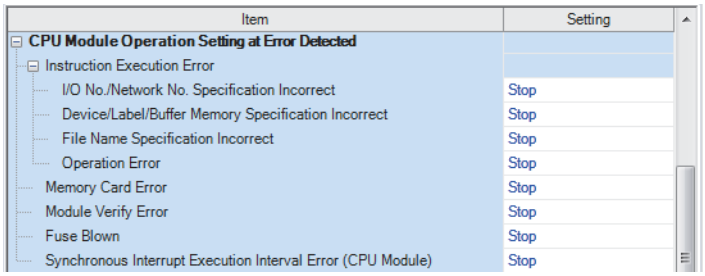
Error name	Error code
Power shutoff (either of the redundant power supply modules)	1010H
Failure (either of the redundant power supply modules)	1020H
Battery error	1090H
Module verification error	2400H, 2401H
Fuse blown error	2420H

CPU module operation upon error detection setting

Set the CPU module operation upon error detection.

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

Window



Displayed items

Item		Description	Setting range	Default
Standard program instruction execution error	I/O No./Network No. Specification Error	Sets the CPU module operation when a specification error of I/O numbers/Network numbers has been detected.	• Stop • Continue	Stop
	Device/Label/Buffer Memory Specification Incorrect	Sets the CPU module operation when a specification error of Device/Label/Buffer memory has been detected.		
	File Name Specification Incorrect	Sets the CPU module operation upon a file name specification error.		
	Operation Error	Sets the CPU module operation upon an operation error.		
Memory Card Error		Sets the CPU module operation upon a memory card error.		
Module Verification Error*1		Sets the CPU module operation upon a module verification error.		
Fuse Blown*1		Sets the CPU module operation upon a fuse blown error of controlled module.		

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

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

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

Error name	Error code
Memory card error	2120H, 2121H
Module verification error	2400H, 2401H
Fuse blown error	2420H
I/O number or network number specification error	2800H, 2801H, 2802H, 2803H, 2804H, 2805H, 2806H, 2807H, 2810H
Device, label, or buffer memory specification error	2820H, 2821H, 2822H, 2823H, 2824H
File name specification error	2840H, 2841H, 2842H
Operation error	3400H, 3401H, 3402H, 3403H, 3404H, 3405H, 3406H, 3420H, 3421H, 3422H, 3423H, 3440H, 3441H, 34A0H

CPU module operation setting

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

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

Window

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

Displayed items

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

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

■Applicable errors to the CPU module operation setting

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

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

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.		

Safety diagnostic functions

The following table lists the safety-specific diagnostic functions among the self-diagnostic functions for the CPU module.

Item[*1]		Description	Diagnostic timing	Error code
Memory diagnosis	RAM diagnosis	Detects errors occurring at program memory, device memory, and memory used by the system.	• At powering off and on • The CPU module is reset • At END processing	3C20H, 3C21H, 3C22H, 3C2FH, 3C33H
	F/W diagnosis	Diagnoses whether the firmware stored in the ROM is corrupt.	• At powering off and on • The CPU module is reset	3C10H
Program diagnosis	Operation circuit diagnosis	Detects errors at operation circuits used to operate safety programs.	• At powering off and on • The CPU module is reset • At END processing	3C15H
	File verification	Detects errors in stored safety programs and safety parameters.	• At powering off and on • The CPU module is reset • At END processing	2180H, 36E1H, 3C33H, 3C34H
		Diagnoses whether safety programs and safety parameters stored in both the CPU module and SIL2 function module are the same.	• At powering off and on • The CPU module is reset	2180H, 3640H
Operation result verification	Output data verification	Verifies output results calculated separately in the CPU module and SIL2 function module.	At safety cycle processing	3E60H
System diagnosis	Time monitoring	Monitors the execution status of the respective CPU module and SIL2 function module processing.	Always	3E61H
	Microcomputer diagnosis	Detects microcomputer internal register errors.	• At powering off and on • The CPU module is reset • At END processing	3C00H, 3C01H
Power supply voltage monitoring	Power supply voltage monitoring	Detects voltage errors to prevent operation with voltage outside the guaranteed operating range.	Always	— (Performs a shutdown.)
	Power supply voltage monitoring/ shutdown circuit diagnosis	Monitors whether the circuit monitoring the power supply voltage and circuit used to perform shutdown are functioning properly.	• At powering off and on • The CPU module is reset • At END processing	3E10H, 3E11H
Reset circuit monitoring	Reset circuit monitoring diagnosis	Diagnoses whether the reset has been performed correctly.	• At powering off and on • The CPU module is reset	3E12H
Clock stop detection	Clock diagnosis	Detects clock errors (frequency errors).	• At powering off and on • The CPU module is reset • At END processing	3C16H
CRC calculation circuit diagnosis	CRC calculation circuit diagnosis	Diagnoses whether the CRC calculation circuit is capable of correctly calculating CRC.	• At powering off and on • The CPU module is reset • At END processing	3E01H
Module	Module OS diagnosis	Verifies whether the CPU module and SIL2 function module are functioning normally without running out of control.	• At powering off and on • The CPU module is reset • At END processing	3C11H, 3C17H

*1 Self-diagnostics is performed in both the CPU module and SIL2 function module.

15.3 Error Clear

This function clears all the existing continuation errors at once.

Errors that can be cleared

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

■CPU module

Error name	Error code
Power shutoff	1000H
Power shutoff (either of the redundant power supply modules)	1010H
Failure (either of the redundant power supply modules)	1020H
Invalid power supply module	1030H
Power supply module configuration error	1031H
ROM write count error	1080H
Battery error	1090H
Memory card access error	1100H
SNTP clock setting error	1120H
Default gateway/gateway IP address error	1124H
Own node port number error	1128H
Open specification port number error	1129H
Specified IP address error	112DH
Connection establishment failed	112EH
Socket communications response send error	1133H
TCP connection timeout	1134H
IP address error	1152H
Connection number acquisition error	1155H
Receive buffer securement error	1157H
UDP/IP send failed	1165H
TCP/IP send failed	1166H
Unsend data send error	1167H
Redundant IP address error	1180H
Module moderate error	1200H
Another CPU module moderate error	1220H
Annunciator ON	1800H
Operation continuation error	1810H
Receive queue full	1830H
Receive processing error	1831H
Transient data error	1832H
Constant scan time error	1900H
Safety cycle time exceeded	1A00H
Safety cycle processing error	1A01H
Destination station error in safety communications	1A40H[*1]
Parameter mismatch in safety communication destination station	1A50H[*1]
Device mismatch in safety communication destination station	1A51H[*1]
Device version mismatch in safety communication destination station	1A52H[*1]
Safety protocol version mismatch in safety communication destination station	1A53H[*1]
Timeout in safety communications	1A60H[*1], 1A61H[*1], 1A62H[*1], 1A63H[*1], 1A64H[*1], 1A65H[*1], 1A66H[*1], 1A80H
Safety communications receipt data error	1A70H[*1], 1A71H[*1], 1A72H[*1], 1A73H[*1]
Network configuration mismatch	1B00H
System consistency check error (operating status)	1B20H
System consistency check error (safety operation mode)	1B21H

Error name	Error code
Redundant system error	1B40H, 1B42H, 1B43H
Standby system CPU module error	1B60H, 1B61H
Tracking communications disabled	1B70H
Tracking communication error	1B71H, 1B78H
Tracking transfer error	1B80H, 1B81H, 1B82H
Redundant function module error	1BA0H
System switching error	1BD0H, 1BD1H
Memory card error	2120H, 2121H
Module verification error	2400H, 2401H
Fuse blown error	2420H
Module major error	2441H, 2442H, 2450H
Another CPU module major error	2461H, 2462H, 2470H
I/O number or network number specification error	2800H, 2801H, 2802H, 2803H, 2804H, 2805H, 2806H, 2807H, 2810H
Device, label, or buffer memory specification error	2820H, 2821H, 2822H, 2823H, 2824H
File name specification error	2840H, 2841H, 2842H
Operation error	3400H, 3401H, 3402H, 3403H, 3404H, 3405H, 3406H, 3420H, 3421H, 3422H, 3423H, 3424H, 3440H, 3441H, 34A0H

*1 If a safety communications error (minor error) occurs between safety stations, a safety station interlock will be triggered for safety communications between relevant stations, and safety communications will not be resumed until the safety station interlock is released. The safety station interlock is released with safety special register areas (safety station interlock release request for each safety connection). (☞ Page 527 List of Safety Special Register Areas)
For details on the safety station interlock status, refer to the following.
MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

■SIL2 function module

Error name	Error code
ROM write count error	1080H
Safety cycle time exceeded	1A00H
Destination station error in safety communications	1A40H*1
Parameter mismatch in safety communication destination station	1A50H*1
Device mismatch in safety communication destination station	1A51H*1
Device version mismatch in safety communication destination station	1A52H*1
Safety protocol version mismatch in safety communication destination station	1A53H*1
Timeout in safety communications	1A60H*1, 1A61H*1, 1A62H*1, 1A63H*1, 1A64H*1, 1A65H*1, 1A66H*1, 1A80H*1
Safety communications receipt data error	1A70H*1, 1A71H*1, 1A72H*1, 1A73H*1

*1 If a safety communications error (minor error) occurs between safety stations, a safety station interlock will be triggered for safety communications between relevant stations, and safety communications will not be resumed until the safety station interlock is released. The safety station interlock is released with safety special register areas (safety station interlock release request for each safety connection). (☞ Page 527 List of Safety Special Register Areas)
For details on the safety station interlock status, refer to the following.
MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

How to clear errors (CPU module)

The following describes how to clear errors in the CPU module.

■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. Eliminate the cause of the detected continuation error.
3. Turn on SM50 (Error reset) to clear the errors. When multiple continuation errors have occurred, all the errors are cleared at once.

How to clear errors (SIL2 function module)

The following describes how to clear errors in the SIL2 function module. The errors in the other system cannot be cleared. Clear the errors separately in each system.

■Using the engineering tool

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

■Using the buffer memory

Clear errors using operations of buffer memory.

1. Check the standard program and identify the detected continuation error using Un\G0 (Latest self-diagnostic error code).
2. Eliminate the cause of the detected continuation error.
3. Set 1 in Un\G50 (Error clear) in the standard program to clear the error. *1

*1 After the error is cleared, 0 is automatically set in Un\G50 by the system.

Precautions

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

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

15.4 Clearing Errors in the CPU Module of the Standby System from the CPU Module of the Control System

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

Error clearing procedure

Use SM1679 to clear errors.

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

Precautions

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

15.5 Event History Function

The CPU module collects and stores event information from each module, such as errors detected by the module, operations performed on the module, and network errors. *1
Events that occur on the SIL2 function module are not retained in the event history.
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

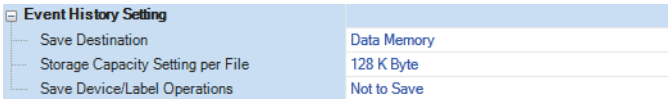
The event history information is constantly collected regardless of the operating state of the CPU module. There are occasions, however, when the event history information cannot be collected due to a major error in a module, a base unit error, a cable failure, or some other cause.

Event history setting

Under normal circumstances, the event history function can be used with its default settings and need not be manually configured. The storage memory and size settings for event history files can be changed as needed. (☞ Page 195 Event history file)

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

Window



Displayed items

Item	Description	Setting range	Default
Save Destination	Specify the storage location of event history files. (☞ Page 195 Storage memory)	• Data Memory • Memory Card	Data Memory
Set Save Volume of Per File	Specify the storage capacity per event history file. (☞ Page 196 File size)	1 to 2048K bytes (in 1K bytes)	128K Byte

Point

For use in any of the following environments, use of an SD memory card for the save destination memory is recommended because the number of events that occur becomes large.

- When file data write operations and device/label data write operations are performed frequently
- When the communication status changes frequently
- When device data write operations are periodically performed from a programmable controller of another station or from another CPU module

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

Logging of the event history

This section describes events saving for the event history.

Event history file

The storage memory and file size for event history files can be changed in event history setting. (☞ Page 194 Event history setting)

■Storage memory

Choose either the data memory or SD memory card.

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

■File size

The size for event history files can be changed in event history setting (☞ Page 194 Event history setting). If the storage size exceeds the specified size, records are deleted in order from the oldest one and the latest one is stored. An event history file size is obtained from the following calculation formula.

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

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

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

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

[Calculation formula]

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

Ex.

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

Operating procedure

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

Element		Size (byte)
File header		20
Event history management information		12
Event to be saved	Power-on and reset	40
	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

■When files are created

An event history file is created when:

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

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

Point

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

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

Operation	Operation for the event history
Memory initialization	When this event occurs, the event history is stored into the internal memory. If the internal memory reaches the maximum number of event history records it can store, all subsequent events are lost. (☞ Page 197 Loss of event history information)
Event history creation	The event history, which has been stored in the internal memory during absence of the event history file, is stored into the data memory or the SD memory card (If any event was lost, it is logged as "*HST LOSS*").

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

Operation	Operation for the event history
Removal of the SD memory card	When this event occurs, the event history is stored into the internal memory. If the internal memory reaches the maximum number of event history records it can store, all subsequent events are lost. (☞ Page 197 Loss of event history information)
Installation of the SD memory card	The event history, which have been stored in the internal memory during absence of the SD memory card, is stored to the SD memory card. If the re-inserted SD memory card contains an event history file of the same file size, the CPU module continues to store the event history information. If the file size is different, the CPU module removes the existing event history file and creates a new event history file.

■When parameters take effect

Any changed parameters take effect when:

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

Point

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

Loss of event history information

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

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 (main base unit). 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.

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 466 Event List)

Viewing the event history

The event history can be viewed using the menus of the engineering tool. For operating procedures and how to interpret the displayed information, refer to the following:

GX Works3 Operating Manual

Clearing the event history

The event history can be cleared from the event history window. Once the event history is cleared, the CPU module deletes all the event history information stored in the specified storage memory. For operating procedures and other details, refer to the following:

GX Works3 Operating Manual

Point

When clearing of the event history is executed, the "Event history clear" (20200) is logged. When an event history file is executed clear during the logging restriction of the event history, the "Event history logging restriction" (00421) is also logged.

Precautions

Change of the event history setting

The following table shows the operation when the event history setting is changed after the CPU module starts, and written to the CPU module.

Changed item	Operation
Save Destination	With the original file left as is, an event history file is generated in the enabled save destination (drive).
Set Save Volume of Per File	The event history file is discarded and regenerated.
Save Device/Label Operations	With the original file left as is, EVENT.LOG is generated when the file is not saved, and EVEN2.LOG is generated when the file is saved.

Therefore, to save a past event history, before changing the event history setting, save data by using the [Create File] button in the event history window of the engineering tool.

[Diagnostics] ⇨ [System Monitor] ⇨ [Event History] button ⇨ [Create File] button

Device data write operation from another station or another CPU module

When "Save Device/Label Operations" is set in Event History Setting, many event updates occur. Depending on the timing, events that occurred cannot be checked with the engineering tool.

15.6 Program Cache Memory Auto Recovery Function

If the contents of memory of the CPU module have been rewritten by itself due to the factors such as excessive electrical noise, the program cache memory recovers the corresponding areas automatically during the run of the program. This function becomes active with RUN state of the CPU module and works during the run of the program. No special settings are required as the system performs this task automatically.

16 REMOTE OPERATION

In a system using the CPU module, the behavior of a remote operation (with the engineering tool) differs depending on the operations.

Operation	Behavior
Remote RUN*1/Remote STOP/Remote PAUSE	The CPU module operating status of a system specified in the transfer setup of the engineering tool or both systems can be changed.
Remote RESET	The CPU modules of both systems can be reset by performing remote RESET operation on the control system CPU module. Only the CPU module of the standby system can be reset by performing remote RESET operation on the CPU module of the standby system.

*1 When the safety operation mode is SAFETY MODE, the safety global devices, safety global labels, safety local labels, and standard/safety shared labels are not cleared.

16.1 Remote RUN/STOP/PAUSE

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

Remote operation performed on the CPU module of a system specified in the transfer setup of the engineering tool

When the following item is selected in "Execution Target", the remote operation is performed only on the CPU module of a system specified in the transfer setup of the engineering tool.

- Currently Specified Station

[Online] ⇨ [Remote Operation]

Remote operation performed on both systems

When the following item is selected in "Specify Execution Target", 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 "Specify Group No." to perform remote operation on the CPU modules of both systems.

16.2 Remote RESET

The CPU modules of both systems can be reset by performing remote RESET operation on the CPU module of the control system. Only the CPU module of the standby system can be reset by performing remote RESET operation on the CPU module of the standby system.
Select "Control System" or "Standby System" in the transfer setup, and select "Currently Specified Station/Specify Both Systems" in "Specify Execution Target" of "Remote Operation".

Precautions

- When the CPU module of the control system is in the STOP state and the CPU module of the standby system is in the RUN state, performing remote RESET operation on the CPU module of the control system causes system switching. To prevent system switching in remote RESET operation, perform remote RESET operation after setting both of the CPU modules to the STOP state.
- When remote operation is performed on the CPU module of the control system or standby system through another path, the CPU module of the standby system cannot be reset by performing remote RESET operation on the CPU module of the control system. To perform remote RESET operation on the CPU module of the control system or standby system, use the path that was used to perform the remote operation on the CPU module of the standby system. Cancel the remote operation first and perform remote RESET operation.
- When performing remote RESET operation on both systems, select "Control System" in "Specify Redundant CPU". When "Not Specified" is selected, an error may occur depending on the timing of the operating status change of each CPU module.

[Online] ⇨ [Current Connection Destination] ⇨ [Specify Redundant CPU]

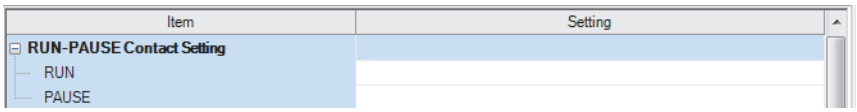
- When the safety operation mode is SAFETY MODE, the safety global devices, safety global labels, safety local labels, and standard/safety shared labels are not cleared.

16.3 Setting RUN-PAUSE Contacts

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

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

Window



Displayed items

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

Point

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

16.4 Precautions

The following describes the precautions on the remote operation in a system using the CPU module.

- 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.

17 PID CONTROL/PROCESS CONTROL FUNCTION

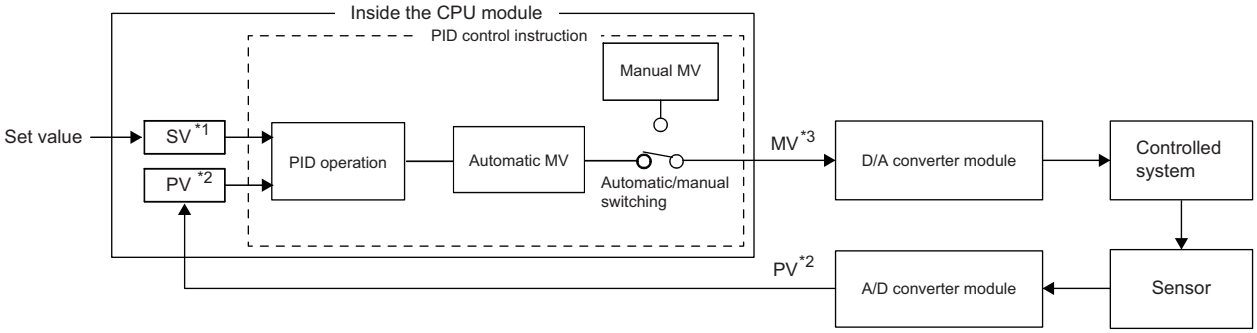
This chapter describes the PID control/process control function.

Restriction

The PID control/process control function cannot be used in safety programs.

17.1 PID Control Function

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



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

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

Point

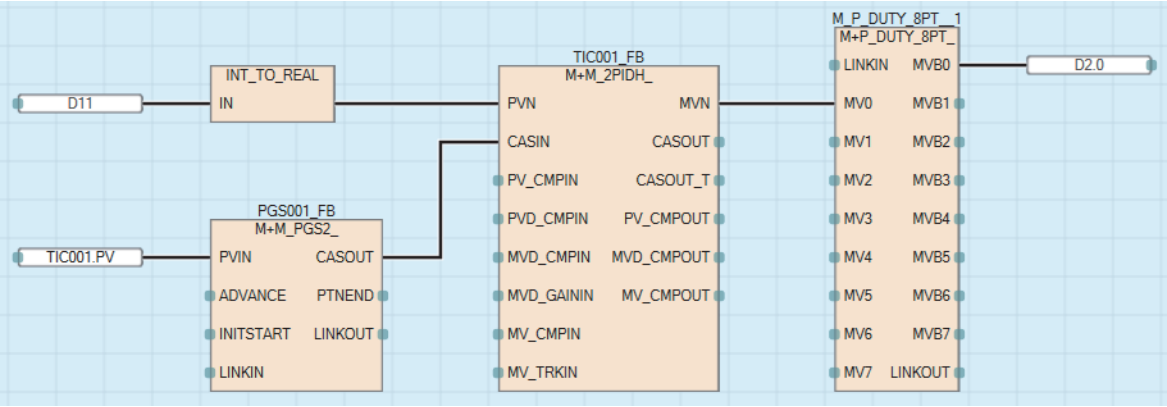
For the PID control, use the PID operation instruction. To execute the same PID control as that of the MELSEC-Q series and MELSEC-L series, use the PID control instructions.
For details on the comparison between the PID operation instruction and PID control instructions, instruction specifications, PID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

17.2 Process Control Function

This chapter describes the process control function.

Process control by using process control function blocks

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



Process control function blocks have features as follows.

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

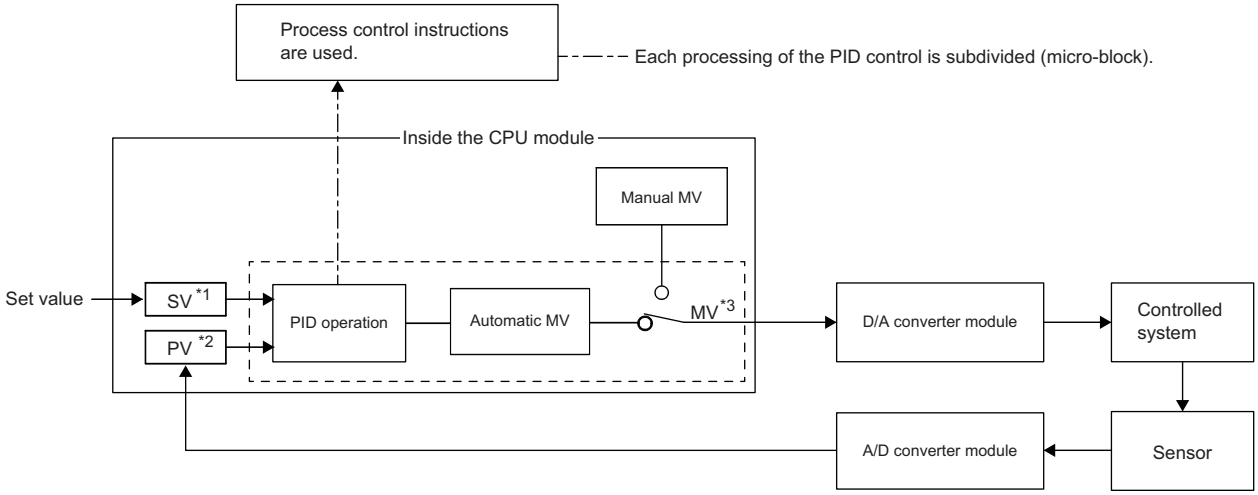
Point

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

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

Process control by using process control instructions

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



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

Process control instructions have the following features:

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

Point

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

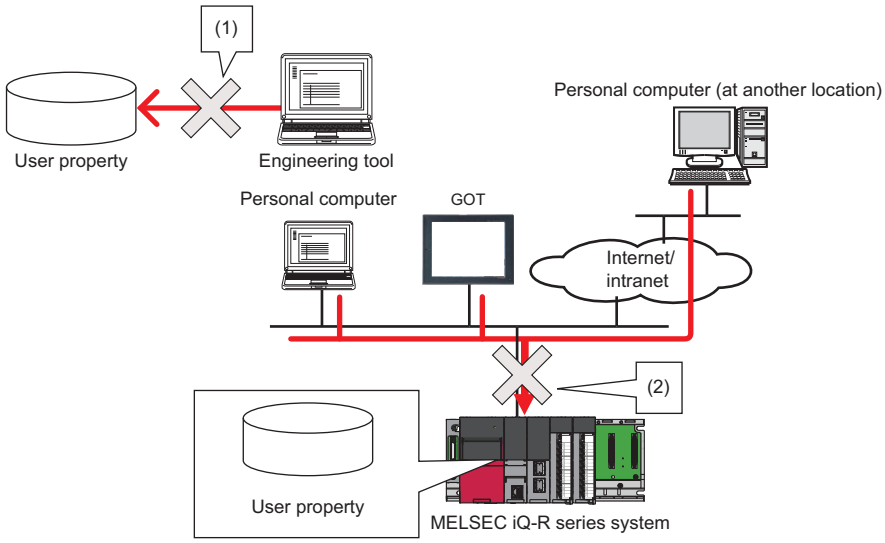
MELSEC iQ-R Programming Manual (Process Control Function Blocks/Instructions)

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

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

18 SECURITY FUNCTION

This function serves to protect the user property stored in a personal computer and the user property inside modules in the MELSEC iQ-R series system against threats such as theft, tampering, faulty operation, and unauthorized execution due to the unauthorized access by an outsider. Use an appropriate security function according to the purpose as shown in the following table:



(1) Protection is provided against theft, tampering, and faulty operation resulting from unauthorized access by an outsider. (Protection for user property on a personal computer)

(2) Protection is provided against theft, tampering, faulty operation, and unauthorized execution resulting from unauthorized access by an outsider. (Protection for user property in CPU modules)

Data to be protected		Purpose	Function	Reference
GX Works3	Project	Prevents unauthorized access to programs (in units of POU). (A password is used.)	Block password function	GX Works3 Operating Manual
		Prevents unauthorized access to programs (in units of program file). (A security key is used.) Project data locked with a security key can only be viewed with an engineering tool for which the same security key has been registered.	Security key authentication function for a project	
		Restricts access to projects on the personal computer with a combination of user name and password. The security level can be set individually for each user.	User authentication function for a project	
CPU module	Program	Prevents unauthorized execution of programs. (A security key is used.) Programs locked with a security key can only be executed at modules for which the same security key has been set.	Security key authentication function for a CPU module	Page 208 Security Key Authentication for a CPU Module
	Program, parameter	Restricts access to programs or parameters in the CPU module with a combination of user name and password. The security level can be set individually for each user.	User authentication function of CPU modules*1	Page 209 User Authentication Function of CPU Modules
	File	Prevents unauthorized read and write of files. (A password is used.)	File password function*1	• GX Works3 Operating Manual • MELSEC iQ-R Ethernet User's Manual (Application) • SLMP Reference Manual • MELSEC iQ-R Serial Communication Module User's Manual (Application)
	Access	Blocks access from an invalid IP address by identifying the IP address of an external device via Ethernet.	IP filter function	MELSEC iQ-R Ethernet User's Manual (Application)
		Restricts access via Ethernet that is taking a different route from specific communication routes. (A password is used.)	Remote password function*1	

*1 These functions disable the password authentication for a certain duration of time after a certain number of failed authentication attempts. For details, refer to the manuals for each function.

Point

If a personal computer with a security key registered is abused by an outsider, there is no way to prevent the outflow of the program property, and thus the user needs to take adequate measures as shown below:

- Preventive measures against the theft of a personal computer (for example, wire locking)
- Management of users of the personal computer (for example, deletion of unnecessary user accounts, strict control of login information, and implementation of fingerprint authentication)

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

- Import the registered security key to another personal computer.
- Export the registered security key to a file and store the file in a safe place.

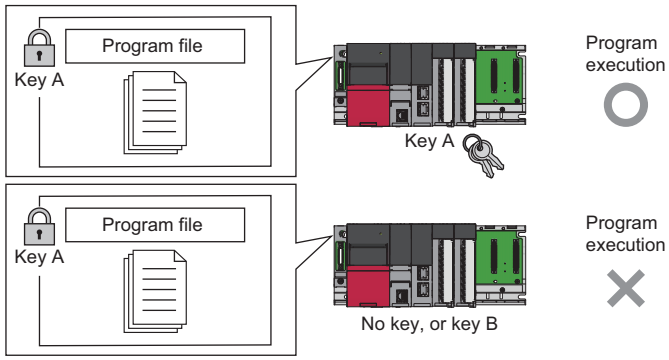
The following information in the CPU module can be deleted by selecting [Online] ⇨ [User Authentication] ⇨ [Initialization of all PLC Data].

- User management information set with user authentication function (User authentication function returns to disabled status.)
- Security key
- All program memory, data memory files
- All files in file storage areas in device/label memory
- Safety operation mode status (returns to TEST MODE)

18.1 Security Key Authentication for a CPU Module

This function prevents an unauthorized execution of programs written in a CPU module.

The operations are restricted by detecting a mismatch between the security key of the program file written in a CPU module and the security key of a CPU module.



If the security key of only one program does not match the security key of the CPU module, all the other programs in the CPU module are not executed as well.

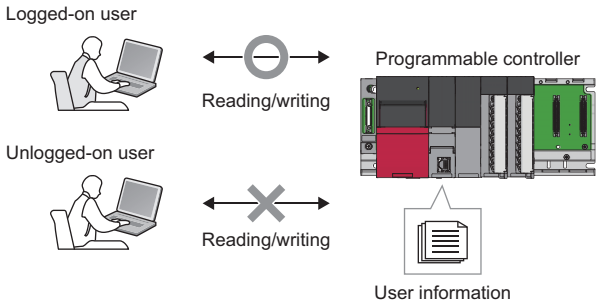
The security key written in the CPU module is retained even after power-off of the CPU module.

For details, refer to the following.

GX Works3 Operating Manual

18.2 User Authentication Function of CPU Modules

This function prevents an unauthorized access to a program or parameters written to a CPU module. Operations are restricted by registering a user name/password for a CPU module.
To access a CPU module, logging on with the registered user information (user name/password) is required.



For details on the user authentication function of CPU modules, refer to the following.
GX Works3 Operating Manual

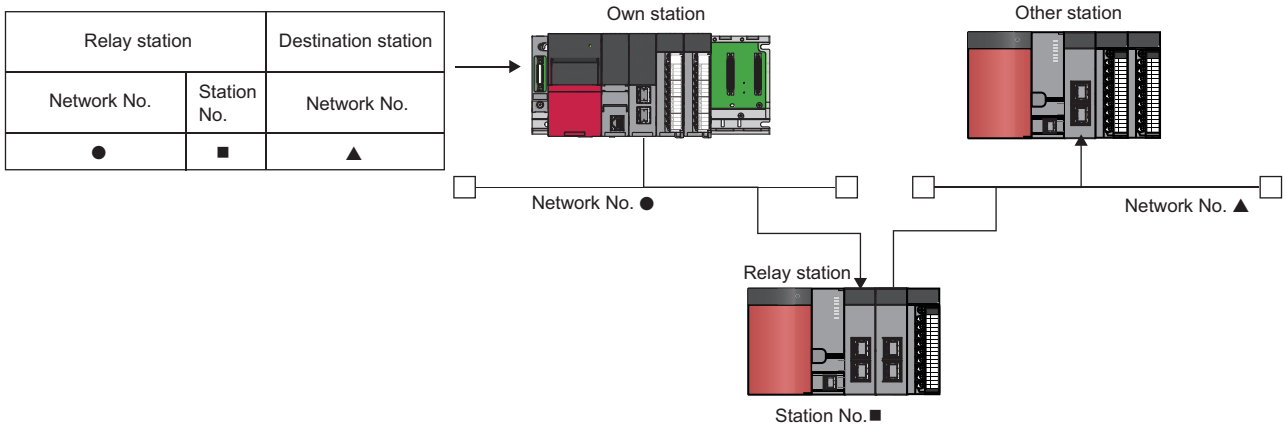
Point

- The user information registered in a CPU module needs to match with the user information of a project.
- For RnPSFCPUs (firmware version 12 or later), enhanced measures to reduce vulnerability are taken and partial changes are made to the following functions.
  · Functions and operations that require user authentication
  · Writing user information to a CPU module
  · Logging on to a CPU module

The enhanced vulnerability-measures are taken for GX Works3 (version 1.105K or later) as well.
Communications with the vulnerability-measures taken are performed by using an RnPSFCPU (firmware version 12 or later) and GX Works3 (version 1.105K or later) and enabling the setting to communicate only with GX Works3 with the vulnerability-measures enhanced version.
However, when that setting is enabled, some functions of the intelligent function module that read files, such as the user authentication and recording functions of the GOT sequence program monitor (circuit display), may become unavailable.
To use the GOT sequence program monitor (circuit display) or the intelligent function module that reads out files, disable the above-mentioned setting.
For details on the user authentication function, refer to the following.
GX Works3 Operating Manual

19 ROUTING SETTING

The user can configure any communication route to perform transient transmission to stations in a different network. This setting can be used when the system has a network module which does not support dynamic routing or when it is necessary to clearly specify a communication route.

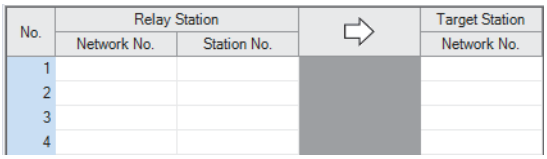


19.1 Setting Method

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 No.	Sets the network number of the first relay station to pass through to the destination station network.	1 to 239	—
	Station No.	Sets the station number of the first relay station to pass through to the destination station network.	0 to 120	—
Target Station	Network No.	Sets the network number of the final arrival network.	1 to 239	—

Point

The S(P).RTWRITE instruction can be used to temporarily change or add a routing setting during operation (setting made by the S(P).RTWRITE instruction is cleared when the CPU module is powered off or reset). Also, the S(P).RTREAD instruction can be used to read setting details of the parameters. For details on these instructions, refer to the following.

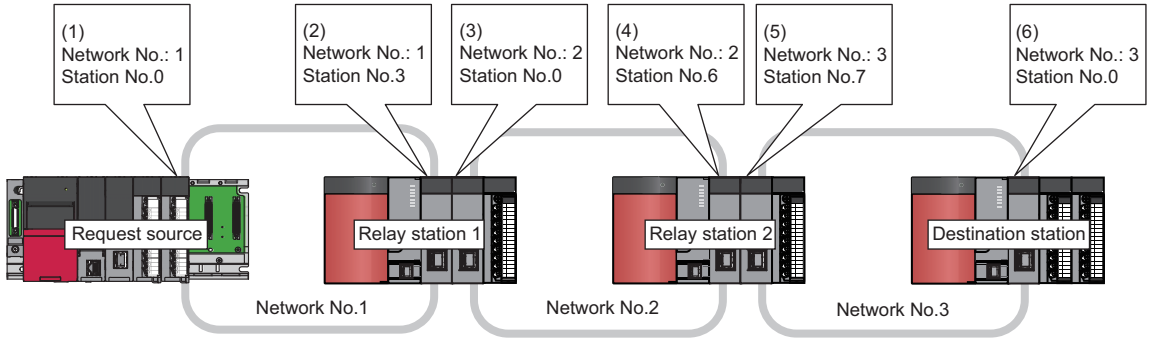
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

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.

20 REDUNDANT FUNCTION

20.1 System Switching

This function switches the systems between the control system and the standby system to continue operation of the redundant system when a failure or an error occurs in the control system. For debugging and maintenance purpose, users can switch the systems at any desired timing.

System switching method

The following two methods are available for system switching: automatic system switching (automatically performed by a system) and manual system switching (performed by a user).
The following table lists the system switching types, causes, and priority of when multiple causes are simultaneously generated.

System switching type	System switching cause	Priority	
Automatic system switching	Power-off, reset, hardware failure of the CPU module	High ↑ ↓ Low	1
	Stop error of the CPU module		2
	System switching request from a network module		3
Manual system switching	System switching request by the SP.CONTSW instruction		4
	System switching request from the engineering tool		5

- When multiple system switching causes are simultaneously generated, the systems are switched according to the cause with higher priority. The switching cause determined according to the priority is stored in the event history and SD1643 (System switching cause).
- A manual system switching request is executed for the CPU module of the control system.

Point

The systems are not switched even when the CPU module is set to the STOP state. They are switched when any of the above system switching causes is generated.

Automatic system switching

A system judges whether system switching is required or not, and automatically switches the systems between the control system and the standby system as required.

■System switching due to power-off, reset, or hardware failure of the CPU module

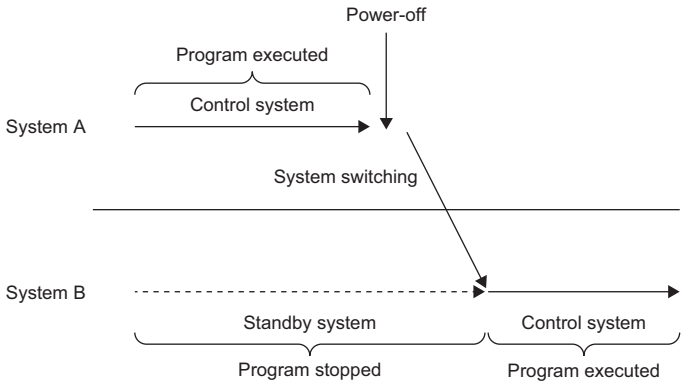
The standby system CPU module monitors the control system status. If the current control system is unable to control the system in the following situations, the standby system CPU module is switched to the control system and the new control system CPU module continues the control over the system.

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

*1 The CPU module of the control system is switched to the standby system even when the current standby system is not ready to switch. (☞ Page 219 Execution availability of system switching)

Ex.

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

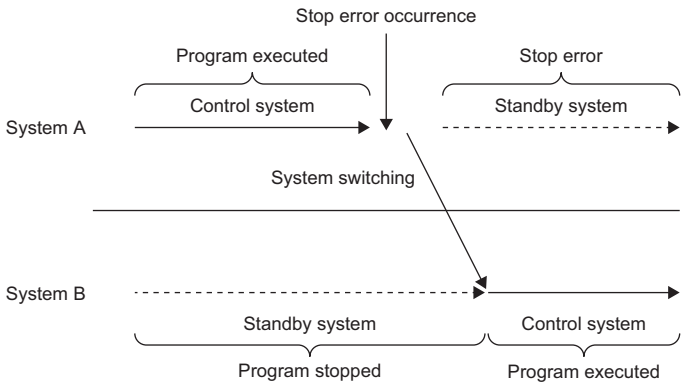
■Stop error of the CPU module

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

*1 If a WDT error has occurred, the CPU module of the control system is switched to the standby system even when the current standby system is not ready to switch. (☞ Page 219 Execution availability of system switching)

Ex.

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

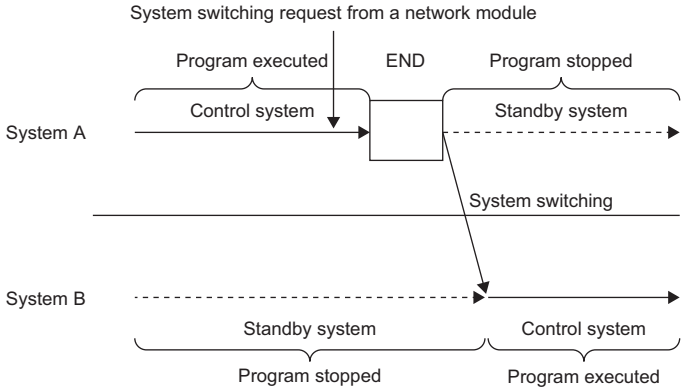


■System switching request from a network module

A control system network module requests the CPU module to switch systems when a communication error or disconnection is detected. When the CPU module of the control system receives a system switching request from the network module, the systems are switched after the END processing.

Ex.

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



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 224 When the cable for the network module is disconnected)

Manual system switching

The user can manually switch the systems between the control system and the standby system at a desired timing.

Point

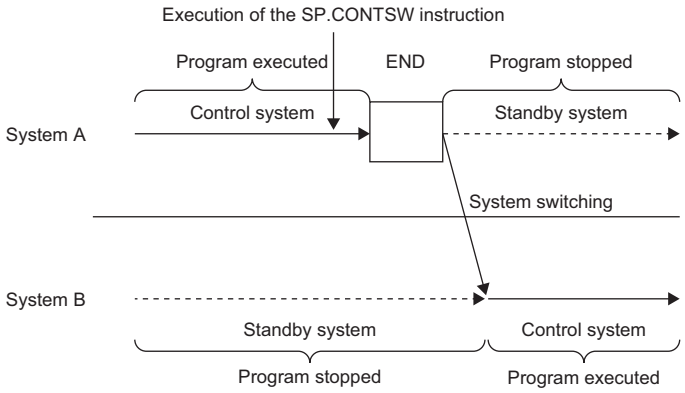
- After turning on SM1646 (System switching by a user), perform the manual system switching in the control system.
- When the manual system switching is disabled by the DCONTSW instruction, execute the ECONTSW instruction. The system switching is enabled in the initial status.

■System switching by the SP.CONTSW instruction

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

Ex.

Operation of system switching by the SP.CONTSW instruction

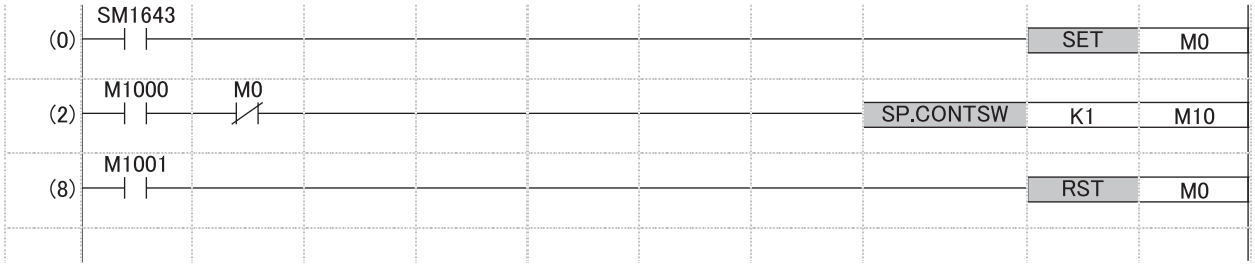


Point

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

When using the SP.CONTSW instruction, create a program that does not execute the SP.CONTSW instruction again on the CPU module of the new control system by using SM1643 (ON for only one scan after system switching (standby system to control system)), as shown below.

M1000: System switching command, M1001: Clear signal



For details on the SP.CONTSW instruction, refer to the following.

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

■System switching request from the engineering tool

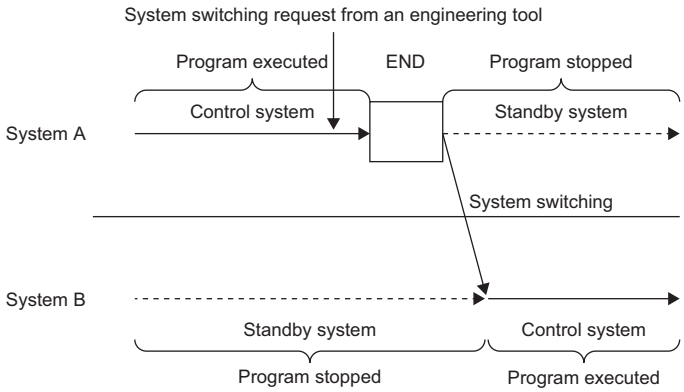
When the engineering tool sends a system switching request to the CPU module of the control system, the systems are switched after the END processing.

Switch the systems from the "Redundant Operation" window of the engineering tool.

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

Ex.

Operation of system switching from the engineering tool



Operation at system switching

The following table shows the operations of the CPU modules of when the control system and the standby system are switched.

These operations are for when both systems are operating and the operating status of the CPU modules are the same.

Item			CPU module of the new control system	CPU module of the new standby system
Program execution	Standard program	Initial execution type program	This type of program is not executed. However, when an initial execution type program has not been completed on the old control system at system switching, it is executed again from the start.	The program execution stops.*1
		Scan execution type program	This type of program is executed from the step 0.	
		Fixed scan execution type program	The fixed scan execution interval is measured from 0.	The program execution stops.
		Standby type program	This type of program is not executed.	
		Event execution type program	• Interrupt program: The program is executed when an interrupt factor is generated. • ON of bit data (TRUE): The program is executed when a specified factor is generated. • Passing time: A specified time is measured from 0.	
	Safety program	Fixed scan execution type program	The fixed scan execution interval is measured from 0.	
Tracking transfer			Tracking data is transferred to the new standby system.	Tracking data is received. However, when a stop error occurs on the new standby system, the new standby system does not receive the tracking data.
Online change	Online change (ladder block)	Standard program	Write operation at system switching continues.	
		Safety program	Safety programs cannot be changed online.	
	File batch online change	Standard program	Write operation at system switching continues.	
		Safety program	Safety programs cannot be changed online.	
Device/label memory (standard label, safety label, standard/safety shared label)			The state before system switching is held.	
Signal flow memory (standard/safety program)		With tracking setting	The signal flow of the old control system is reflected.	The state before system switching is held.
		Without tracking setting	The signal flow of the tracking transfer targets is turned on.	
Setting the device/initial label value			The values are not set.	
Special relay (SM), special register (SD)			The state before system switching is held. However, when the CPU module is in the RUN state, SD520 to SD531 and SD1890 to SD1895 are cleared.	
Safety special relay (SA\SM), safety special register (SA\SD)			The state before system switching is held. The safety special relay areas/safety special register areas related to safety communications (SA\SM1008, SA\SM1016, SA\SM1024, SA\SM1032, SA\SM1040, SA\SM1048, SA\SM1056, SA\SM1064, SA\SM1088, SA\SD1008 to SA\SD1071, SA\SD1088, SA\SD1089, SA\SD1090 to SA\SD1097, SA\SD1104 to SA\SD1223, SA\SD1232 to SA\SD1359, and SA\SD1600 to SA\SD1663) are cleared.	
Output (Y)			The state before system switching is held.	The state is held when the standby system output setting of the redundant system operation setting is enabled. If the setting is disabled, the output (Y) is turned off and then the output refresh stops.
Safety output device (SA\Y)				The safety output device (SA\Y) is turned off regardless of the standby system output setting.
Safety output	CC-Link IE Field Network (CPU module → Master/local module)		After a safety program which is the fixed scan execution type is operated using transferred device data, data is output.*2	Refresh processing is stopped because the safety connection is disconnected.

Item	CPU module of the new control system	CPU module of the new standby system
Local device setting (standard/safety program)	This setting is in accordance with the parameter settings.	No operation is performed because the program does not operate.*1
File register setting	The file register setting before system switching is held.	
Direct access input (DX)	Data is input when an instruction using the direct access input (DX) is executed in the program after system switching.	
Direct access output (DY)	Data is output when an instruction using the direct access output (DY) is executed in the program after system switching.	
FROM/TO instructions	These instructions are executed when the instruction execution condition is satisfied in the program after system switching.	No operation is performed because the program does not operate.*1
Instruction that requires several scans	• This instruction is executed in the program execution after system switching. • When the systems are switched while the instruction is being executed in a program executed in both systems, the instruction execution continues and the completion device turns on at a completion of the instruction execution.	• No operation is performed because the program does not operate.*1 • When the systems are switched while the instruction is being executed, the instruction execution continues. However, the completion device does not turn on in the new standby system, and it turns on in the control system after the next system switching.*1
Constant scan	Constant scan is enabled.	Constant scan is disabled.

*1 The operation varies in a program executed in both systems. (☞ Page 255 Operation at system switching)

*2 The safety program may be executed multiple times up until safety output is performed from the CPU module to the network module.

Execution availability of system switching

The following tables show the execution availability of system switching.

○: Switching possible, ×: Switching not possible

Status of system using the CPU module		Execution availability of system switching						
		Automatic system switching					Manual system switching	
		Power-off or reset of the CPU module	Hardware failure of the CPU module	Stop error of the CPU module		System switching request from a network module*3	System switching request by the SP.CONTSW instruction*3	System switching request from the engineering tool*4
				WDT error	Error other than WDT error			
Normal operation or continuation error in the standby system		○	○	○	○	○	○	○
Tracking communications disabled (cable disconnection)		×	○*6	○*6	×	×	×	×
Power-off, reset, hardware failure of the CPU module of the standby system		×	○*6	○*6	×	×	×	×
Stop error in the standby system	WDT error	○*8	○*6	○*6	×	×	×	×
	Error other than WDT error	○*8	○	○*6	×	×	×	×
Network error detected in the standby system*1		○	○	○	○	×	×	×
During memory copy from control system to standby system		○*8	○	○*6	×	×	×	×
During online program change		○	○	○	○	○*2*5	×	×
Operating status mismatch of the CPU modules in both systems		○	○	○	○	×	×	×
Mismatch between the safety operation mode of both systems		○	○	○	○	×	×	×
Error in safety tracking data detected		×	○*6	○*6	×	×	×	×
During system switching*7		○	○	○	○	○	×	×
During online module change	Redundant function module	×	○*6	○*6	×	×	×	×
	Other modules	○	○	○	○	○	○	○
System switching disabled by the DCONTSW instruction		○	○	○	○	○	×	×

*1 When the group specification has been set in the standby system, a network error is not detected if communication is available with the line of an Ethernet-equipped module after a communication error has occurred on the other Ethernet-equipped module. (MELSEC iQ-R Ethernet User's Manual (Application))

*2 After the online change has been completed, the system switching cause is detected and the systems are switched. However, when an online change is being executed only in the standby system, the systems cannot be switched.

*3 If system switching is disabled when a system switching request is sent, a continuation error occurs and a cause of a system switching failure is stored in SD1644 (Cause of system switching failure).

*4 If system switching is disabled when a system switching request is sent, the error code corresponding to a cause of the system switching failure is returned.

*5 When the systems are switched during an online program change, a mismatch between the files is detected in the system consistency check and a stop error may occur on the new standby system. (☞ Page 249 Files)

*6 Although the CPU module of the control system is switched to the standby system, the current standby system is not switched.

*7 The initial processing of the safety function after system switching completes is also included.

*8 When the safety control data is set to be tracked, a stop error is detected only in the CPU module of the standby system (error code: 37E0H is detected when memory copy is being executed), which may result in the detection of an error in the safety tracking data. In this case, if the CPU module of the control system is powered off or reset, system switching becomes disabled.

Occurrence of a cause of the system switching failure

The BACKUP LED flashes when a cause of the system switching failure has been generated.

Point

The cause of the BACKUP LED flashing can be checked in SD1642 (BACKUP/SEPARATE LED flashing cause). Check SD1642 and eliminate the cause to flash the LED.

Check method of system switching information

The following table lists the check methods of system switching information at system switching (automatic system switching and manual system switching).

Check method	Information	Reference
Event history	System switching result, system switching cause, and control system/standby system transition	Page 220 Event history
Special relay (SM)/Special register (SD)	System switching result and detailed information	Page 221 Special relay/Special register
CTRL LED and SBY LED of the redundant function module	System switching result	Page 221 CTRL LED and SBY LED of the redundant function module

When the systems have been switched, check the switching cause or detailed information and take corrective action to restore the system to a normal state as required.

Event history

The information related to system switching results, system switching cause, and control system/standby system transitions can be checked with the event history of the engineering tool.

When the systems are switched, the following items are stored in the event history of both systems.

- Automatic system switching: "System switching (system)" (event code: 00F00)
- Manual system switching: "System switching (user)" (event code: 2B000)

When the systems are switched by the SP.CONTSW instruction, the system switching instruction ID number specified with the SP.CONTSW instruction is also stored.

Special relay/Special register

System switching results and detailed information can be checked with the special relay and special register.

- Whether the systems have been switched or not can be checked by checking SD1649 (System switching cause (when the systems are successfully switched)). When the systems have been switched, the system switching cause is stored in SD1649 of the control system and standby system.
- If the systems have not been switched even after a system switching cause is generated, the cause why the systems have not been switched can be checked by checking SD1644 (Cause of system switching failure). The switching cause is stored in SD1643 (System switching cause) of the control system.

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

- Special relay (☞ Page 490 Redundant function)
- Special register (☞ Page 517 Redundant function)

■Special relay

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

○: Stored, ×: Not stored

SM No.	Name	Storage status at system switching	
		CPU module of the new control system	CPU module of the new standby system
SM1637	System switching detection (standby system to control system)	○	×
SM1643	ON for only one scan after system switching (standby system to control system)	○	×
SM1644	ON for only one scan after system switching (control system to standby system)	×	○
SM1645	System switching request from a network module	○	○
SM1646	System switching by a user	×	×

■Special register

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

○: Stored, ×: Not stored

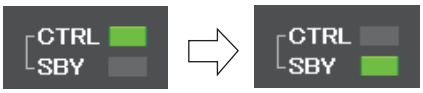
SD No.	Name	Storage status at system switching	
		CPU module of the new control system	CPU module of the new standby system
SD1642	BACKUP/SEPARATE LED flashing cause	×	×
SD1643	System switching cause	×	○
SD1644	Cause of system switching failure*1	×	×
SD1645	System switching request status from a network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

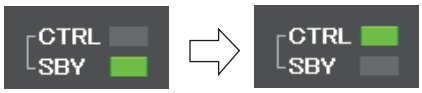
CTRL LED and SBY LED of the redundant function module

System switching results can be checked with the CTRL LED and SBY LED on the front of the redundant function module.

Control system → Standby system



Standby system → Control system



Precautions

The following describes the precautions on system switching.

Item	Description	Reference
Error in the redundant function module	• When an error has been detected on the redundant function module, the control system and standby system continue operating without being switched. • When a communication error has been detected in the communications between the redundant function module and a CPU module, the systems are switched.	Page 223 Error in the redundant function module
When both systems operate as standby systems	• If a communication error is generated due to a tracking cable error during system switching, both systems may operate as standby systems. • If a WDT error or a hardware failure of the CPU module has occurred in the control system while a stop error exists in the standby system, both systems operate as standby systems.	Page 223 When both systems operate as standby systems
When both systems operate as control systems	If the tracking cable and the network cable that is connected to the control system network module are simultaneously pulled out or disconnected, both systems may operate as control systems. In this case, connect the tracking cable again. After a stop error has occurred in the system B, power off and on or reset the system B.	—
Scan time monitoring function	During system switching, scan time monitoring with the watchdog timer is interrupted. Thus, in a scan in which the systems are switched, no error is detected even if the scan time monitoring time has elapsed. A time taken for system switching does not need to be considered in the scan time monitoring time (WDT) setting.	—
System switching disabled by online change	Manual system switching is disabled during an online change.	Page 223 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 224 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 224 System switching during initial processing/initial processing (when switched to RUN)
CC-Link control at system switching	When the master operating station is switched with a program by using the CC-Link standby master function, the CC-Link control can be continued even after system switching.	MELSEC iQ-R CC-Link System Master/Local Module User's Manual (Application)
When a stop error is generated in a new standby system at system switching	If a hardware failure is detected in the CPU module or SIL2 function module in a new standby system during a system switching, it may take two seconds longer for the safety control to start in the new control system compared to the normal system switching.	—

Error in the redundant function module

- When an error has been detected on a redundant function module, a continuation error occurs on the CPU module, and the control system and standby system continue operating without being switched. Check the error code, and perform online module change if the redundant function module has failed. (MELSEC iQ-R Online Module Change Manual)
- When the redundant function module was removed from the base unit or a failure has occurred in the base unit and a communication error has been detected in the communication between the redundant function module and a CPU module, a stop error occurs on the CPU module and the systems are switched. In this case, check the error code and eliminate the error cause.

Restriction

When the CC-Link IE Field Network module has not been mounted and a communication error has been detected in the communications between the redundant function module and the CPU module, the systems may not be switched.
To switch the systems even though the redundant function module has been removed from the base unit or a failure has occurred in the base unit, mount the CC-Link IE Field Network module.

When both systems operate as standby systems

- If a communication error is generated due to a tracking cable error during system switching, both systems may operate as standby systems. When an error occurs with the tracking cable, each L ERR LED of the redundant function modules turns on. In this case, replace the tracking cable with a new one as soon as possible. If both systems operate as standby systems, connect a new tracking cable properly and turn off and on or reset the CPU module of one system so that the other system will operate as the control system.
- If a WDT error or a hardware failure of the CPU module has occurred in the control system while a stop error exists in the standby system, the control system is switched to the standby system and the both systems operate as standby systems. In this case, eliminate error causes, and then power off and on or reset the CPU modules in the both systems.

The following table lists the operations of when both systems operate as standby systems.

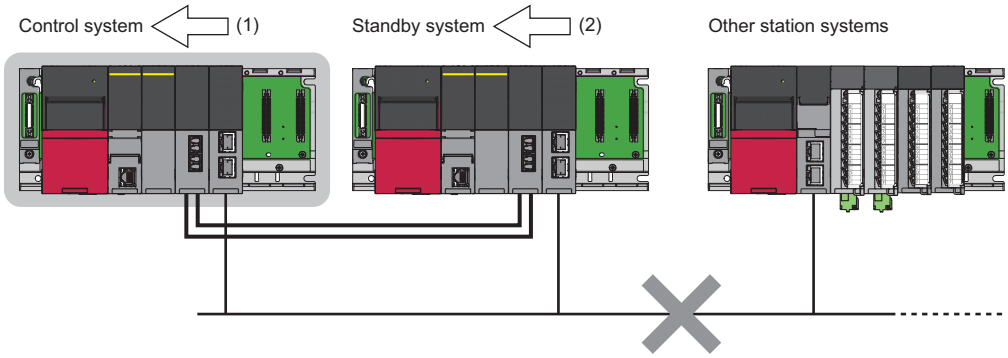
Item		Operation of both systems
LED of the redundant function module	BACKUP LED	Flashing
	CTRL LED	Off
	SBY LED	On
Special relay	SM1635 (Standby system judgment flag)	On
System switching request	System switching request from a network module	System switching disabled
	System switching request by the SP.CONTSW instruction	
	System switching request from the engineering tool	

System switching disabled by online change

Manual system switching is disabled during an online change.
To disable the manual system switching during an online change, the CPU module is set to the manual system switching disabled state before an online change starts. After the online change is completed, it is set to the manual system switching enabled state.
If a communication with the engineering tool is interrupted due to communication cable disconnection or other causes during an online change, the online change cannot be completed, and the CPU module remains in the manual system switching disabled state. When the CPU module is in the manual system switching disabled state, the systems cannot be switched by manual system switching or a system switching request from a network module.
If an online change has failed, refer to the following and take actions.
☞ Page 177 Action for when an online change has failed

When the cable for the network module is disconnected

If a network module cable is disconnected, the systems may not be switched depending on the timing of error detection on the control system and the timing on the standby system.



(1) When the control system detects cable disconnection first, the systems are switched.
(2) When the standby system detects cable disconnection first, the systems are not switched. At this time, a continuation error occurs in the CPU module of the control system and the BACKUP LED of the redundant function module flashes because a system switching disable cause exists.

In both case, replace the network module cable with a new one and clear the network error.

System switching during initial processing/initial processing (when switched to RUN)

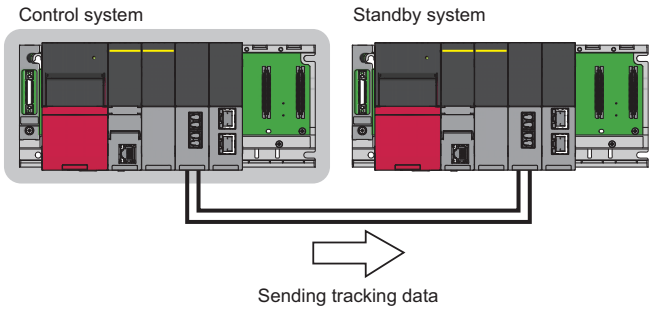
When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the control system, the systems are switched after the initial processing or initial processing (when switched to RUN). Even if the initial processing or initial processing (when switched to RUN) takes time and a communication error occurs, the systems are switched after the initial processing or initial processing (when switched to RUN).

When a system switching cause has been generated during initial processing or initial processing (when switched to RUN) on the standby system, whether the systems are switched or not depends on the system switching cause.

System switching cause	Whether to switch the systems
Power-off, reset, hardware failure of the CPU module	The systems are switched after the initial processing or initial processing (when switched to RUN).
Stop error of the CPU module	
System switching request from a network module	The systems are not switched.
System switching request by the SP.CONTSW instruction	
System switching request from the engineering tool	

20.2 Tracking Transfer

This function transfers the control data from the control system to the standby system and maintains the consistency of the data in the two systems to continue operation of the redundant system when a failure or an error occurs in the control system.



- The CPU module of the control system sends tracking data to the CPU module of the standby system before the END processing. The tracking data includes special relay (SM), special register (SD), PID control instruction information, and system data in addition to device (standard/safety device)/label (standard label, safety label, and standard/safety shared label) data. (☞ Page 226 Tracking data)
- The CPU module of the standby system receives the tracing data, and then reflects the received data on the device/label memory.
- Standard device/standard label data can be divided and transferred as required. To transfer device/label data within a desired range, set a range of standard global devices to be transferred or whether to transfer standard local devices/ standard global labels/standard local labels for each tracking block and turn on the tracking triggers (SD1667 to SD1670) assigned for each tracking block.
- Up to 1M-word standard device/standard label data can be transferred in one scan.
- When a stop error has occurred, only system data is transferred. (☞ Page 228 System data)

Point

- In the CPU parameters, "Tracking Device/Label Setting" under "Tracking Setting" in "Redundant System Settings" is set to "Transfer collectively" by default. Thus, tracking transfer is performed without setting the parameter. (☞ Page 233 Batch transfer)
- When a program file is converted with the process control extension setting enabled, "Tracking Device/ Label Setting" is automatically set to "Detailed setting". In the tracking block No.64, the range of the file register specified in the system resource of the process control extension setting is registered. When the system is operating in backup mode, the tracking transfer trigger (bit 15 of SD1670) corresponding to the tracking block No.64 automatically turns on at every scan. (☞ Page 232 When the "Process Control Extension Setting" is enabled)

Restriction

When a tracking transfer is not performed, the possible causes are the following:

- The CPU module of the standby system has been powered off or reset.
- Hardware failure of the CPU module has occurred.*1
- An error has occurred in the redundant function module.*1
- A WDT error has occurred.*1
- Tracking cables has been pulled out or disconnected.

*1 When the tracking communication line properly works, "Link-up" (event code: 00100) may be stored in the event history.

Tracking data

The following table lists the tracking data that can be transferred from the control system to the standby system.

○: Transfer possible, ×: Transfer not possible

Item		Transfer*4	Reference
Device data	User device	○	Page 227 Devices that can be specified
	Safety user device	○*3	
	Special relay	○ (Auto transfer)	
	Safety special relay	×	
	Special register	○ (Auto transfer)	
	Safety special register	×	
	Index register	○	
	File register	○	
	Refresh data register	○	
Global label*2 (Except module labels)	Standard global label	○	Page 233 Tracking device/label setting
	Safety global label	○*3	Page 236 Safety tracking transfer setting
	Standard/safety shared label	○*3	
Local device	Standard local device	○*1	Page 233 Tracking device/label setting
	Safety local device	○*3	Page 236 Safety tracking transfer setting
Local label	Standard local label	○*1	Page 233 Tracking device/label setting
	Safety local label	○*3	Page 236 Safety tracking transfer setting
Signal flow memory	Standard signal flow memory	○*1	Page 232 Tracking transfer setting for the signal flow memory
	Safety signal flow memory	○*3	Page 236 Safety tracking transfer setting
PID control instruction information		○ (Auto transfer)	Page 228 PID control instruction information
CPU buffer memory		×	—
System data		○ (Auto transfer)	Page 228 System data

*1 The data that is used in a program executed in both systems is not transferred. (☞ Page 252 Program Execution in Both Systems)

*2 Global labels/safety global labels with devices assigned are not transferred as global labels/safety global labels. They are transferred according to the tracking settings/safety tracking settings of the assigned devices.

*3 Tracking transfer is performed when "Transfer" is set in "Safety Tracking Setting" of "Redundant System Settings" in the CPU parameters. It is performed when the safety operation mode is either TEST MODE or SAFETY MODE.

*4 During online change on both systems or on the control system only, only system data is transferred. During online change on the standby system only, data is transferred according to the table.

Devices that can be specified

The following table lists the data that can be specified for tracking transfer.

○: Specifiable, ×: Not specifiable, —: Not settable as a local device

Classification	Device name	Transfer	
		Global device	Local device
User device	Input (X)	○	—
	Output (Y)	○	—
	Internal relay (M)	○	○
	Link relay (B)	○	—
	Annunciator (F)	○	—
	Link special relay (SB)	○	—
	Edge relay (V)	○	○
	Step relay (S)	×	—
	Timer (T)	○	○
	Retentive timer (ST)	○	○
	Long timer (LT)	○	○
	Long retentive timer (LST)	○	○
	Counter (C)	○	○
	Long counter (LC)	○	○
	Data register (D)	○	○
	Link register (W)	○	—
	Link special register (SW)	○	—
	Latch relay (L)	○	—
Safety user device	Safety input (SA\X)	○	—
	Safety output (SA\Y)	○	—
	Safety internal relay (SA\M)	○	○
	Safety link relay (SA\B)	○	—
	Safety timer (SA\T)	○	○
	Safety retentive timer (SA\ST)	○	○
	Safety counter (SA\C)	○	○
	Safety data register (SA\D)	○	○
	Safety link register (SA\W)	○	—
System device	Function input (FX)	×	—
	Function output (FY)	×	—
	Function register (FD)	×	—
	Special relay (SM)	×*1	—
	Special register (SD)	×*1	—
Safety system device	Safety special relay (SA\SM)	×	—
	Safety special register (SA\SD)	×	—
Index register	Index register (Z)	○	○
	Long index register (LZ)	○	○
File register	File register (R)	×	—
	File register (ZR)	○	—
Refresh data register	Refresh data register (RD)	○	—

*1 The data is automatically transferred regardless of parameter settings. (☞ Page 228 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 479 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 493 List of Special Register Areas

■PID control instruction information

The PID control data that is specified with the PIDINIT or S.PIDINIT instruction is transferred. (📖MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))

■System data

The data related to the system, such as system switching and operation mode change, is transferred.

Tracking block and tracking trigger

The devices or labels of a specified range is transferred by setting a range of devices or labels to be transferred for each tracking block and turning on the tracking trigger which is assigned for each tracking block. Note that only standard devices/labels can be transferred for each tracking block. (Safety devices/labels cannot be transferred for each tracking block.)

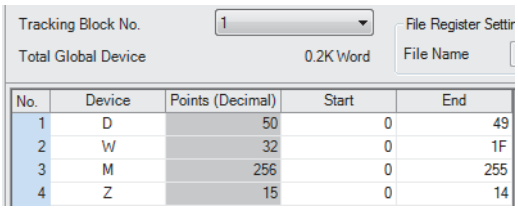
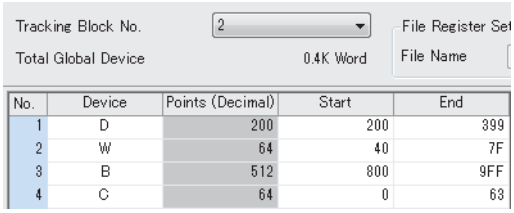
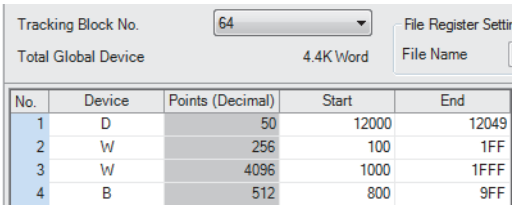
Tracking block

A tracking block is used for setting a tracking transfer range of global devices and whether or not to transfer local devices/global labels/local labels.

- Up to 64 blocks (No.1 to 64) are available.
- Up to 2048 devices can be set in a block. Up to 2048 devices can be transferred in a single tracking.
- The available capacity of tracking devices or labels is 1M words per block.

Ex.

When only the tracking block No.3 has no setting

Tracking block No.	Setting example
1	 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" and "Safety Tracking Setting" in the CPU parameters. (☞ Page 231 Tracking transfer setting, ☞ Page 236 Safety tracking transfer setting)
2. Add a program that controls tracking triggers. (☞ Page 230 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

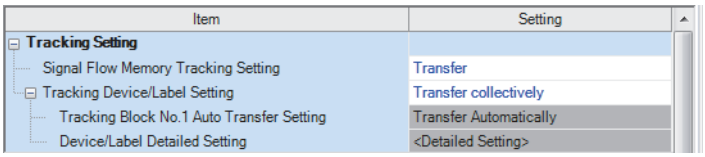
To continue the same control as before even after system switching, configure the settings in "Tracking Setting" so that all data necessary for the program is transferred.

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 232 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 233 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 229 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 234 Detailed setting)	—	—

Point

Set "Detailed setting" of "Tracking Device/Label Setting" for the following purposes.

- To shorten a tracking transfer time
- To add tracking data, such as the file register (ZR)
- To transfer each data set individually

When the "Process Control Extension Setting" is enabled

Settings and operations are as follows:

- When a program file is converted, "Tracking Device/Label Setting" is automatically set to "Detailed setting".
- In the tracking block No.64, the range of the file register specified in the system resource of the process control extension setting is registered. Because the other devices are not automatically registered, specify the other tracking block numbers for those devices. The devices and their ranges set in "Device/Label Memory Area Setting" can be imported at a time by clicking the [Device Setting Reflection] button in the "Global Device Setting" window. (☞ Page 235 Global device setting)
- To transfer the tracking block No.64, the bit 15 of SD1670 automatically turns on at every scan when the system is operating in backup mode. Do not turn off the bit 15 of SD1670.

Tracking transfer setting for the signal flow memory

By transferring the signal flow memory, operations of rising/falling instructions in the old control system are taken over to the new control system even after system switching.

Point

"Signal Flow Memory Tracking Setting" is set to "Transfer" by default. Transferring the signal flow memory is recommended. For the operation not to transfer it, refer to the following.

☞ Page 275 Instructions whose operations vary depending on tracking of the signal flow memory

■Tracking of individual POUs

The following table shows whether the signal flow memory is transferred or not for each POU.

○: Transferred, ×: Not transferred, —: No signal flow memory

POU			Both systems program executions setting	
			Control system execution	Both systems executions
Program block			○	×
Function block	Macro type		○	×
	Subroutine type	Global FB	○	○*1
		Local FB	○	×
Function			—	

*1 To prevent the signal flow memory in the standby system from being overwritten in a both systems execution program, use a local FB. When a global FB is used, the signal flow memory in the standby system is overwritten with the signal flow memory in the control system.

Tracking device/label setting

The following two methods are available for transferring devices and labels: automatically transferring all the devices and labels in a batch or transferring specified devices and labels of a specified tracking block.

■Batch transfer

When "Tracking Device/Label Setting" is set to "Transfer collectively", the following devices and labels are assigned to the tracking block No.1 and automatically transferred.

Type	Description
Global device	■Bit device • Input (X) • Output (Y) • Internal relay (M) • Link relay (B) • Edge relay (V) • Latch relay (L) ■Word device • Timer (T) • Long timer (LT) • Retentive timer (ST) • Long retentive timer (LST) • Counter (C) • Long counter (LC) • Data register (D) • Link register (W) • Index register (Z) • Long index register (LZ)
Local device*1	All local devices
Global label*2	All the global labels assigned to the device/label memory
Local label*1	All local labels

*1 The data used in both systems execution programs is not transferred.

*2 Global labels with devices assigned are not transferred as global labels. They are transferred according to the tracking settings of the assigned devices. To transfer global labels with devices assigned, specify the assigned global devices in the global device setting. (☞ Page 235 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 234 Detailed setting)

Restriction

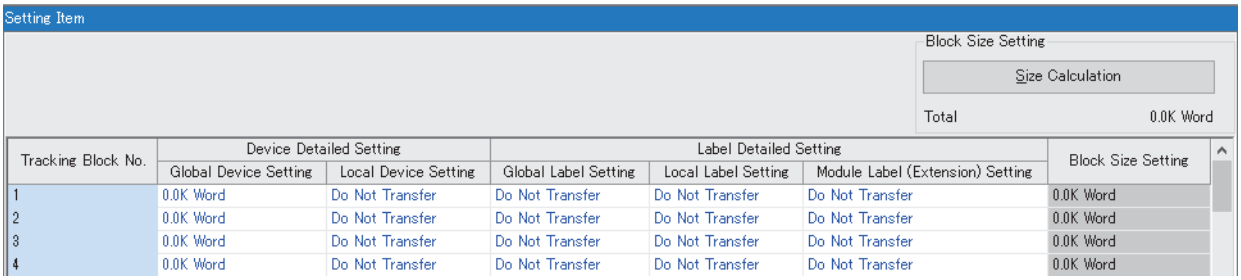
After setting "Transfer collectively", perform a test operation in the system design phase and check if the size of tracking data is 1M words or smaller. If the size of the tracking data is larger than 1M words, a stop error occurs when the CPU module is powered off and on or reset.

■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 235 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 227 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

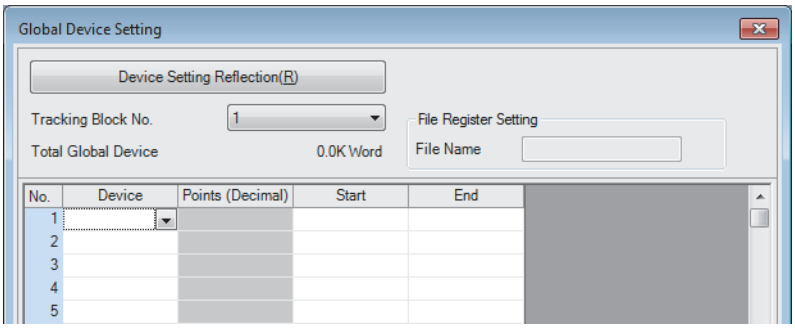
When setting devices and labels for tracking transfer, click the [Size Calculation] button to check if the tracking data capacity in one scan is equal to or less than 1M words.

■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 227 Devices that can be specified)
Start/End	Specify a range of devices to be transferred.

Point

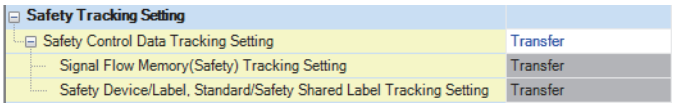
To input devices and their ranges set in "Device/Label Memory Area Setting" in a batch, click the [Device Setting Reflection] button.

Safety tracking transfer setting

Set tracking transfer of the information required for safety control.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Safety Tracking Setting]

Window



Displayed items

Item		Description	Setting range	Default
Safety Control Data Tracking Setting	Signal Flow Memory (Safety) Tracking Setting	Set whether to transfer the data to be used for safety control or not. "Signal Flow Memory (Safety) Tracking Setting" and "Safety Device/Label, Standard/Safety Label Tracking Setting" are set according to "Safety Control Data Tracking Setting".	• Do Not Transfer • Transfer	Transfer
	Safety Device/Label, Standard/ Safety Label Tracking Setting			

Tracking mode

The following two modes are available for tracking.

Item	Description
Synchronous tracking mode	Tracking data is always transferred to the standby system once every scan of the control system. During a tracking transfer from the control system to the standby system, the next scan does not start in the control system.
Asynchronous tracking mode	When a tracking transfer from the control system is to be performed and the previous tracking is still in progress, the tracking transfer from the control system is canceled and the previous tracking continues. The control system starts the next scan without waiting for notifications of data reception/reflection completion from the standby system.

The tracking mode depends on the CPU module operating status.

CPU module operating status of standby system and control system		Tracking mode
Control system	Standby system	
RUN	RUN	Synchronous tracking mode*1
	STOP, PAUSE	Asynchronous tracking mode
STOP, PAUSE	RUN	
	STOP, PAUSE	

*1 When the RUN/STOP/RESET switch of the CPU module of each system is set to the RUN position and both systems are powered on, data is transferred in the asynchronous tracking mode at first. After the tracking data is reflected to the standby system, the mode is switched to the synchronous tracking mode.

In the following conditions, the tracking mode is the asynchronous tracking mode.

- During an online change
- When the systems are switched
- When tracking communications disabled is detected

Point

Even if the safety operation mode is changed, the tracking mode will not change.

Effect on the scan time

The following describes the effect on the scan time depending on the tracking mode.

Point

For the calculation method for an increase in the scan time due to tracking transfer, refer to the following.

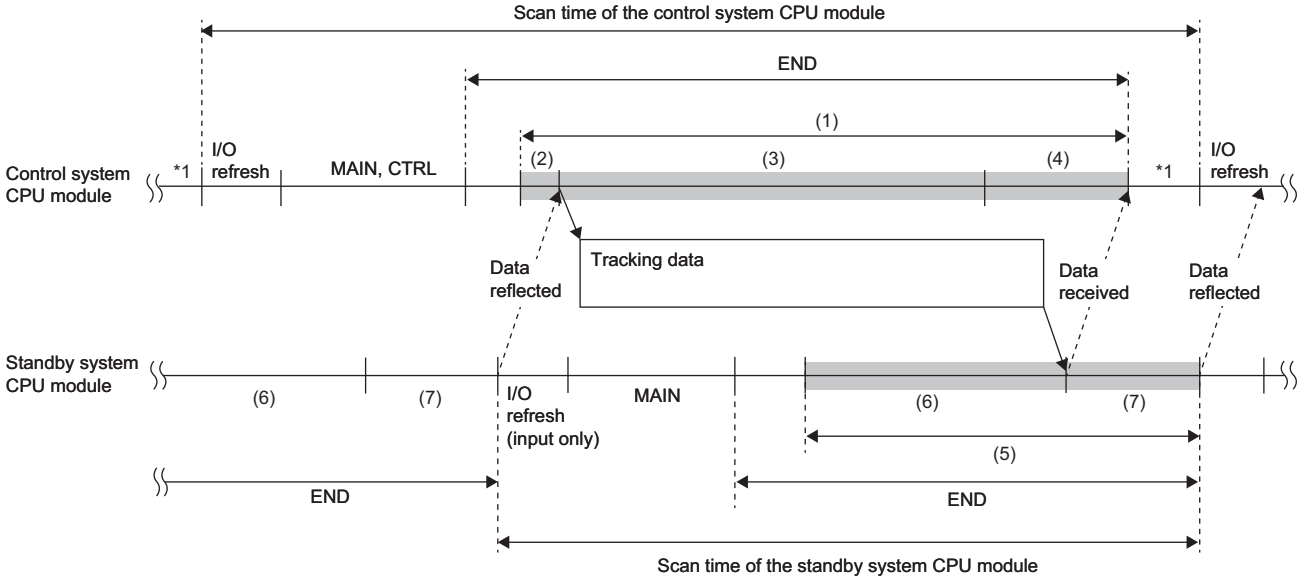
☞ Page 549 Increase in the scan time due to tracking transfer

■Synchronous tracking mode

In the synchronous tracking mode, tracking transfer is always performed once every scan during the END processing. Until the standby system receives the tracking data sent from the control system, the next scan is not started. Thus, the scan time of the control system and standby system increases by the time for tracking data send/receive processing.

Ex.

CTRL: Program executed in the control system, MAIN: Program executed in both systems



*1 If the constant scan is used, waiting time for the constant scan generated.

In the control system, the scan time is extended by the following tracking send processing time.

Item		Description
(1) Tracking send processing	(2) Waiting for completion of tracking data reflection	After receiving a notification of the reflection completion from the standby system, the control system sends the tracking data.
	(3) Sending tracking data	The control system sends the tracking data.
	(4) Waiting for completion of tracking data reception	The control system waits for a notification of receive completion from the standby system. After receiving the notification of the receive completion from the standby system, the control system starts another END processing.

In the standby system, the scan time increases by the following tracking receive processing time.

Item		Description
(5) Tracking receive processing	(6) Waiting for tracking data reception	The standby system receives the tracking data from the control system. After receiving the tracking data, the standby system notifies the control system of the receive completion and reflects the tracking data.
	(7) Reflecting tracking data	The standby system reflects the tracking data. After the reflection completion, the standby system notifies the control system of the reflection completion and starts another END processing.

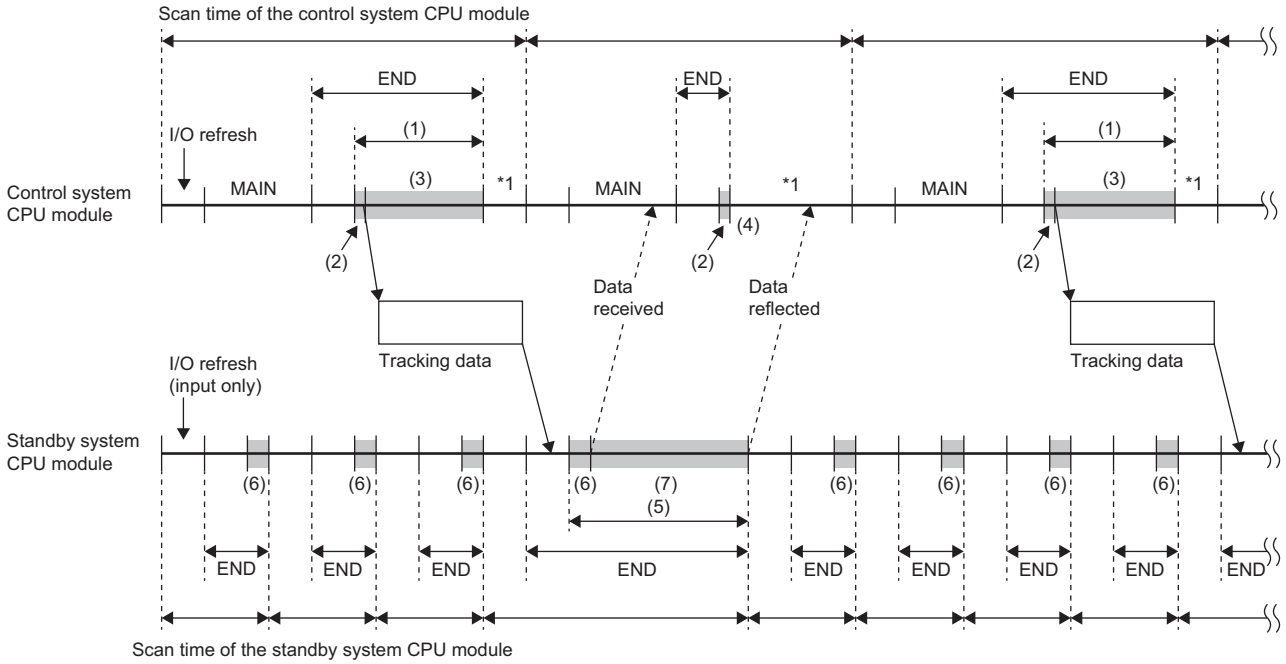
■Asynchronous tracking mode

In the asynchronous tracking mode, the control system starts the next scan without waiting for notifications of data reception/reflection completion from the standby system.

Unlike the synchronous tracking mode, the scan time is not affected by waiting time for data reception/reflection completion. When the standby system does not receive the tracking data from the control system, the standby system starts the next scan.

Ex.

When the control system CPU module is in the RUN state and the standby system CPU module is in the STOP state



*1 If the constant scan is used, waiting time for the constant scan generated.

In the control system, the scan time is extended by the following tracking send processing time.

Item		Description
(1) Tracking send processing	(2) Waiting for completion of tracking data reflection	After receiving a notification of the reflection completion from the standby system, the control system sends the tracking data. When a notification of reflection completion is not arrived (4), the control system does not send tracking data in the scan.
	(3) Sending tracking data	The control system sends the tracking data. After completing the send, the control system starts another END processing without waiting for a notification of receive completion from the standby system.

In the standby system, the scan time increases by the following tracking receive processing time.

Item		Description
(5) Tracking receive processing	(6) Waiting for tracking data reception	The standby system receives the tracking data from the control system. When the standby system does not receive the tracking data, the standby system starts the next scan. After receiving the tracking data, the standby system notifies the control system of the receive completion and reflects the tracking data.
	(7) Reflecting tracking data	The standby system reflects the tracking data. After the reflection completion, the standby system notifies the control system of the reflection completion and starts the next scan.

■When the mode is switched from the asynchronous tracking mode to the synchronous tracking mode

When the mode is switched from the asynchronous tracking mode to the synchronous tracking mode, the standby system receives the tracking data twice in one scan. Therefore, the scan time of the standby system is extended by the following time.

Scan time of the standby system: Standby system scan time × 2 + Control system scan time

Precautions

Operation at power-on

When the RUN/STOP/RESET switch of the CPU module of each system is set to the RUN position and both systems are powered on, the control system CPU module starts in the STOP state and switched to the RUN state after reflecting the tracking data is completed in the standby system CPU module. (The same operation is performed when one system is powered on while the other system is waiting for the start-up of the other system.)

Since the operating status of the CPU module in the control system and that of the CPU module in the standby system do not march until the operating status of the CPU module in the control system is switched from the STOP state to RUN state, the BACKUP LED of the redundant function module flashes.

When the time of the initial processing and that of the initial processing (when switched to RUN) of the CPU module in the standby system are longer than those of the CPU module in the control system, the operating status of the CPU module in the control system may not be immediately switched to the RUN state after power-on.

Device data used by the CPU module of the new control system

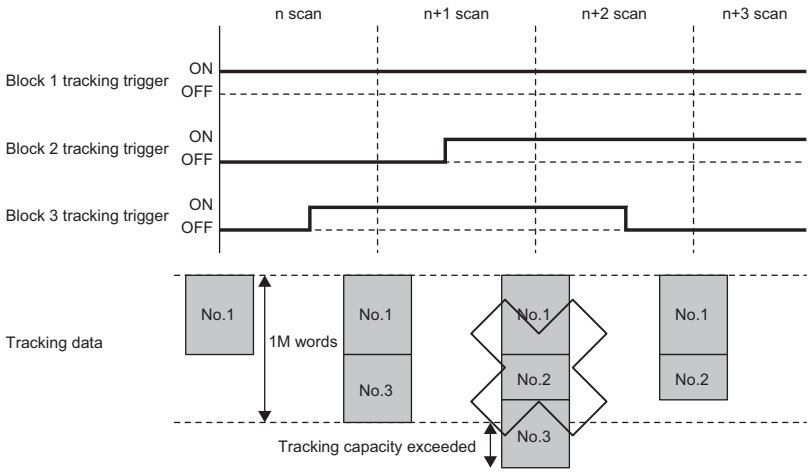
After system switching, the CPU module of the new control system starts operations by using the device/label data received in tracking transfer. The following table shows device/label data used by the CPU module of the new control system by tracking data receive status at system switching.

Device/label initial value	No tracking data received	Tracking data received once at least
Not set	Operation starts based on the latched device/label data.	Operation starts based on the device/label data that is transferred from the old control system. When the synchronous tracking mode is used as a tracking mode, the data in the old control system at the final scan start is used by the new control system.
Set	Operation starts based on the device/label data that is set with device/label initial values.	

Data capacity for tracking transfer

Up to 1M words of device/label data can be transferred in one scan. Set the tracking data capacity within 1M words.

If the data capacity exceeds 1M words, standard global devices/standard local devices/standard global labels/standard local labels are not transferred in the scan. In this case, check which tracking trigger turned on in the event history of the engineering tool and set the tracking data capacity within 1M words.



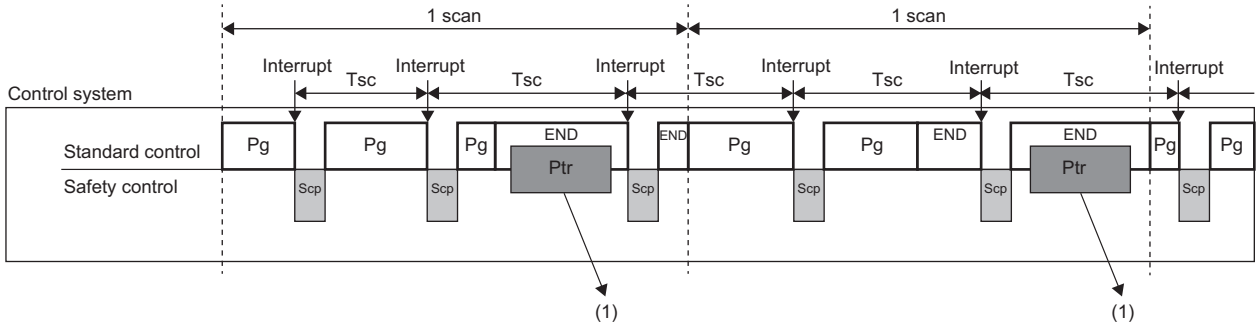
Even though the size of the tracking data to be transferred is 1M words or less, the size of the data may become larger than 1M words depending on the label type or data type to be used after the data or all the data is converted with the engineering tool. When configuring "Tracking Setting" with the CPU parameter, click the [Size Calculation] button in "Detailed setting" of "Device/Label Detailed Setting" to check if the size of the tracking data is 1M words or less and transferred in one scan. (☞ Page 234 Detailed setting)

To reduce the size of the tracking data, consider the following.

- Exclude devices/labels that are not required to continue the system operation.
- Divide the tracking data into multiple blocks to transfer the data in multiple scans.

Influence on tracking data due to safety cycle time

Safety cycle processing is performed at the interval of the specified safety cycle time, and the tracking transfer is performed during the END processing. If the safety cycle time set in the parameter is shorter than the scan time, safety cycle processing may be performed multiple times in a single scan. In such case, safety devices and safety labels immediately before the tracking transfer are transferred to the other system as tracking data.



Tsc: Safety cycle time
Pg: Standard program
Scp: Safety cycle processing
Ptr: Tracking
END: END processing
(1) Tracking transfer to the standby system is performed.

To transfer all data of safety cycle processing to the other system as tracking data, set a longer time than the scan time for the safety cycle time. (☞ Page 269 Setting method)

When data is different between the control system and the standby system

When executing the tracking transfer, files such as program (standard/safety), FB file (standard/safety), CPU parameter (standard/safety), global label setting (standard/safety), standard/safety shared label setting must be the same between the CPU modules of the both systems. (☞ Page 242 Memory Copy from Control System to Standby System)

If there is any difference, standard global devices, safety global devices, system data, and PID control instruction information are transferred. *1

*1 Standard global devices and safety global devices are transferred even when the tracking transfer settings differ between the both systems. However, devices that are included in the device range settings of the control system CPU module and not included in the device range settings of the standby system CPU module are not reflected to the standby system CPU module. (The standby system CPU module reflects values according to the device range settings of the standby system.)

If the files differ between the control system and the standby system, the following tracking transfer stops.

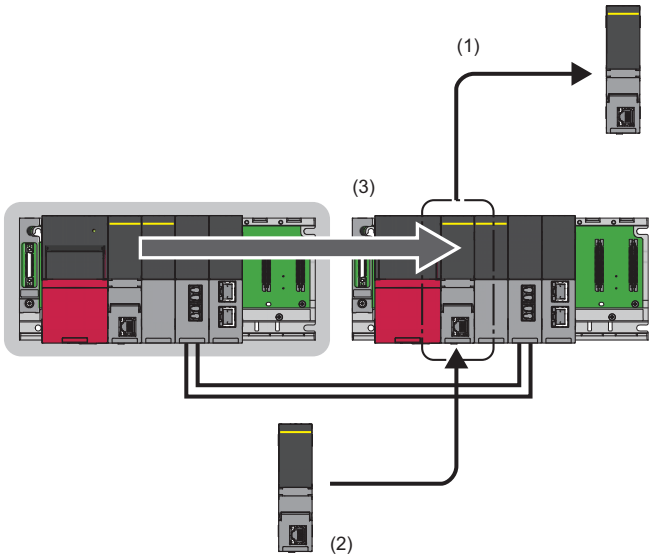
Condition	Data for which the tracking transfer is stopped
When there is a difference in any of standard CPU parameter file, standard program file, standard FB file, or standard global label setting file	• Standard local device • Standard global label • Standard local label • Standard signal flow memory
When there is a difference in any of safety CPU parameter file, safety program file, safety FB file, safety global label setting file, or standard/safety shared label setting file	• Safety local device • Safety global label • Safety local label • Standard/safety shared label • Safety signal flow memory

20.3 Memory Copy from Control System to Standby System

This function transfers data, such as programs and parameters, of the CPU module in the control system to the CPU module in the standby system to match the memory data between the two CPU modules.

Ex.

Replacing the CPU module of the standby system using this function



(1) Remove the CPU module of the standby system.
(2) Mount a new CPU module.
(3) Execute the memory copy function to match the memory data between the two CPU modules.

Data in the following memory is copied.

- Program memory
- Device/label memory (only the file register file area)
- Data memory (only the files in the system folder ($MELPRJ$))
- SD memory card (only the files in the system folder ($MELPRJ$))
- System memory (copying only the user management information file)

For the CPU built-in memory, the memory copy is executed regardless of whether or not there are differences between the control system and standby system. For the SD memory card, the memory copy is executed only when there are differences between the control system and standby system.

Point

For the module replacement method and the maintenance procedure using this function, refer to the following.
☞ Page 381 Module Replacement

Files to be copied

The following table lists the files to be copied by the memory copy function.

○: Copied, ×: Not copied, —: Cannot be stored

File type		Copy			
		CPU built-in memory			SD memory card
		Program memory	Device/label memory	Data memory	
Standard program		○	—	○	—
Standard FB file		○	—	○	—
Standard CPU parameter		—	—	○	—
System parameter		—	—	○	—
Standard module parameter		—	—	○	—
Module extension parameter		—	—	○	○
Module-specific backup parameter		—	—	×	×
Memory card parameter		—	—	—	○
Device comment		—	—	○	○
Initial device value		—	—	○	—
Standard global label setting file		—	—	○	—
Initial label value file	Initial global label value file	—	—	○	—
	Initial local label value file	—	—	○	—
File register		—	○	—	×
Event history		—	—	×	×
Device data storage file		—	—	○	×
General-purpose data		—	—	×	×
Remote password		—	—	○	○
Safety program		○	—	○	—
Safety FB file		○	—	○	—
Safety CPU parameter		—	—	○	—
Safety module parameter		—	—	○	—
Safety global label setting file		—	—	○	—
Standard/safety shared label setting file		—	—	○	—

Point

When the memory copy is executed, files are copied after the memory of the standby system that has differences is initialized. (☞ Page 246 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 to an extended SRAM cassette, the replaced CPU module can take over the security key by simply inserting the extended SRAM cassette to which the security key has been written. In this case, writing the security key to the replaced CPU module after completion of the memory copy is not required.

Execution methods

There are three methods for executing the memory copy.

Item	Description	Reference
Memory copy using the engineering tool	The memory copy is executed by an online operation of the engineering tool connected to the CPU module of the control system.	Page 245 Memory copy using the engineering tool

■Check method using the special relay/special register

The execution status and completion status of memory copy can be checked using the following special relay/special register areas.

SM/SD number	Name
SM1654	Memory copy being executed
SM1655	Memory copy completion
SD988*1	Memory copy completion status (latch)
SD1654	Memory copy completion status

*1 This is a latch area. Even when the system is powered off and on or reset, the result of the memory copy executed immediately before power-on or reset can be checked.

For details on each special relay and special register area, refer to the following.

☞ Page 479 List of Special Relay Areas, ☞ Page 493 List of Special Register Areas

Execution conditions

The memory copy can be executed regardless of the operating status (RUN, STOP, PAUSE, stop error) of the CPU module in both control and standby systems. However, execution availability differs depending on the safety operation mode.

○: Operation possible, ×: Operation not possible

Safety operation mode of each system		Execution availability
Control system	Standby system	
TEST MODE	TEST MODE	○
	SAFETY MODE	×
SAFETY MODE	TEST MODE	○
	SAFETY MODE	×

Memory copy using the engineering tool

The execution procedure for the memory copy using the engineering tool is described below.

Execution procedure

1. Connect the engineering tool to the CPU module of the control system.
2. Open the "Redundant Operation" window of the engineering tool.

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

3. On the control system CPU module of memory copy source, a user with the Administrator access level of the user authentication function logs on. (Execute the memory copy only in the logged on state.)
4. Select "Memory Copy" in the "Redundant Operation" window, and click the [Execute] button. During the memory copy, the MEMORY COPY LED of the redundant function module flashes (in 200ms intervals) in both systems.

Control system



Standby system



5. When the memory copy is completed, the MEMORY COPY LED turns off in the control system and turns on in the standby system.

Control system



Standby system



6. Power off and on or reset the standby system. The MEMORY COPY LED turns off in the standby system.

Control system



Standby system



Precautions

The following are the precautions for the memory copy function.

Automatic memory initialization

The initialization operation for each memory differs.

Memory	Initialization operation
Program memory, device/label memory	Regardless of whether or not there are differences, execute the memory copy after the memory is initialized.
Data memory	Regardless of whether or not there are differences, the "$MELPRJ$" folder is deleted from the memory and then the memory copy is executed.
SD memory card	If there are differences, the "$MELPRJ$" folder is deleted from the SD memory card and then the memory copy is executed.

When memory copy is executed, the ERROR LED of the standby system CPU module may flash. This phenomenon is not an error. *1

*1 When a stop error has been occurred in the standby system, no error is detected.

Follow the procedure to turn off and on or reset the standby system CPU module.

Also, back up the data that is not supported by the memory copy and is stored in the "$MELPRJ$" folder, such as event history, before executing the memory copy.

Restrictions

■Restrictions on memory copy execution

Do not execute the memory copy in the following cases.

- The CPU module of the standby system is powered off or being reset.
- The initial processing (when switched to RUN) is being performed.
- The tracking cable has an error or is not connected.
- The redundant function module has an error.
- The two systems cannot perform communications via the redundant function module.
- The model of the CPU module in the control system differs from that of the CPU module in the standby system.
- An online operation is being performed by the engineering tool.
- The systems are being switched.
- A module (a redundant function module or any module of the standby system) is being changed online.
- The CPU module is set as the standby system.
- The safety operation mode of the standby system is SAFETY MODE.
- Memory copy cannot be executed with the access level of the operator. (It is executed by a user other than an Administrator.)
- A communication timeout occurs in tracking communications.

■Restrictions during memory copy execution

Do not execute the following operations during memory copy execution.

- Online operation from the engineering tool
- RUN/STOP/RESET switch operation
- Removing the SD memory card
- Online module change
- Powering off or resetting the systems
- Disconnecting tracking cables

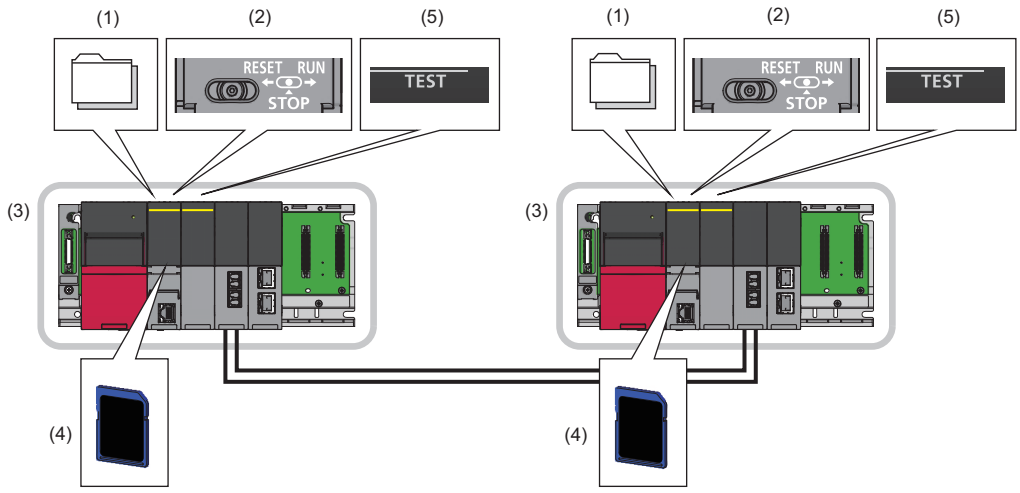
Errors during memory copy execution

When the memory copy is completed with an error, the MEMORY COPY LED of the redundant function module turns off in the control system and the MEMORY COPY LED flashes (in 1s interval) in the standby system. In this case, the memory copy has not been normally executed to the CPU module of the standby system.

Check the error code stored in SD1654 (Memory copy completion status), eliminate the error cause, and execute the memory copy again. (☞ Page 493 List of Special Register Areas)

20.4 System Consistency Check

This function checks whether the system configurations and files in the CPU modules are the same between the control system and the standby system.



The function checks the following items.

Number in figure	Item	Description	Reference
(1)	Files	Parameter file, program file, and other files	Page 249 Files
(2)	Operating status	Operating status of the CPU module (RUN/STOP/PAUSE)	Page 250 Operating status
(3)	Modules on the main base unit	CPU module and other modules mounted on the main base unit	Page 250 Modules on the main base unit
(4)	SD memory card	Insertion status of an SD memory card and status of the write protect switch	Page 251 SD memory card
(5)	Mode of safety control	Safety operation mode	Page 251 Safety operation mode

Execution timing

The following table lists the execution timing of the system consistency check.

Item	Execution timing
Files	• When both systems are simultaneously powered on or reset • When the other system is powered on or reset while one system is waiting for the other system to start • When the standby system is turned on or reset while the control system is operating • When the operating status of the CPU module is switched from STOP to RUN • When the END processing is performed • When the online program change is completed • When the systems are switched • When a tracking cable is reconnected
Operating status*1	• When the standby system is turned on or reset while the control system is operating • When the operating status of the CPU module is switched from STOP to RUN • When the END processing is performed • When a tracking cable is reconnected
Modules on the main base unit	• When both systems are simultaneously powered on or reset • When the other system is powered on or reset while one system is waiting for the other system to start • When the standby system is turned on or reset while the control system is operating • When a tracking cable is reconnected
SD memory card	• When both systems are simultaneously powered on or reset • When the other system is powered on or reset while one system is waiting for the other system to start • When the standby system is turned on or reset while the control system is operating
Safety operation mode*1	• When the standby system is turned on or reset while the control system is operating • When the operating status of the CPU module is switched from STOP to RUN • When the END processing is performed • When a tracking cable is reconnected

*1 Checking this item can be disabled by setting "Backup Mode Setting" of "Redundant Behavior Setting". (☞ Page 258 Redundant System Operation Setting)

The function is not executed in the following cases.

- One of the systems has not started up (powered off, reset, or initial processing being performed).
- A stop error has occurred in the CPU module.
- The memory copy is being executed.

Files

Whether files are the same between the control system and the standby system is checked.

The following table shows whether or not to perform the check on each file type.

○: Checked, ×: Not checked, —: Storage not possible

File type		Check target memory	
		Built-in memory of CPU module	SD memory card*3
Program*1		○	—
FB file*1		○	—
CPU parameter		○	—
System parameter		○	—
Module parameter		○	—
Module extension parameter		○	○
Module-specific backup parameter		×	×
Memory card parameter		—	○
Device comment		×	×
Initial device value		○	○
Global label setting file		○	—
Initial label value file	Initial global label value file	○	○
	Initial local label value file	○	○
File register file*2		○	×
Event history		×	×
Device data storage file		×	×
General-purpose data		×	×
Remote password		○	×
Safety program		○	—
Safety FB file		○	—
Safety CPU parameter		○	—
Safety module parameter		○	—
Safety global label setting file		○	—
Standard/safety shared label setting file		○	—

*1 Reserved area for online change is also checked. (☞ Page 162 Configuration of a program file)

*2 Whether a file exists or not is checked. The file contents are not checked.

*3 Files are not checked at the point when an SD memory card is inserted during operation.

Operation of when a mismatch is detected

A stop error occurs in the CPU module of the standby system.

Perform either of the following operations to match the files in the CPU modules of both systems.

- Writing the correct project to the standby system (☞ Page 55 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 242 Memory Copy from Control System to Standby System)

Operating status

Whether the operating status (RUN/STOP/PAUSE) of the CPU module is the same between the control system and the standby system is checked.

Operation of when a mismatch is detected

A continuation error occurs in the CPU module of the standby system. The BACKUP LEDs of the redundant function modules of the control system and standby system flash because a cause of the system switching failure has occurred.

Backup mode setting

To prevent a continuation error from being detected when the operating status is changed during operation, set the system consistency check so that the consistency check on the operating status is not performed.
With this setting, a continuation error is not detected for the standby system CPU module even if the CPU modules of both systems are in different statuses.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Redundant Behavior Setting] ⇨ [Backup Mode Setting]

Window



Displayed items

Item	Description	Setting range	Default
Operating Status Check	Set whether to perform the consistency check for the operating statuses of both systems or not.	• Do Not Check • Check	Check

Modules on the main base unit

Whether the following items are the same between the control system and the standby system is checked.

- Model of the CPU module
- Models and types of modules mounted on each slot
- Safety protocol version settings

Operation of when a mismatch is detected

A stop error occurs in the CPU module of the standby system. If a stop error is detected when both systems are simultaneously powered on or reset, a stop error occurs in the CPU module of the control system as well. ("Simultaneously" here means when one system is started up within three seconds of the other.)

SD memory card

Whether the insertion status of an SD memory card and the status of the write protect switch are the same between the control system and the standby system is checked. The type and capacity of an SD memory card is not checked.

Point

The insertion status of an SD memory card is checked even when only the standby system is powered off and on or reset while the control system is operating. When an SD memory card is used, it is recommended to remove it only when necessary.

Operation of when a mismatch is detected

If a mismatch of the installation of the SD memory cards or the status of the write protect switch is detected, a stop error occurs on the standby system CPU module. If a mismatch is detected when the CPU modules of both systems are simultaneously turned on or reset, a stop error occurs on the control system CPU module as well.

Safety operation mode

Whether the safety operation modes (SAFETY MODE/TEST MODE) of the CPU modules of both systems are the same is checked.

Operation of when a mismatch is detected

If a mismatch is detected between the safety operation modes, a continuation error occurs on the standby system CPU module.
The BACKUP LEDs of the redundant function modules of the control system and standby system flash because a cause of the system switching failure has occurred.

Backup mode setting

When "Do Not Check" is set in "Operating Status Check", a continuation error is not detected for the standby system CPU module even if the safety operation modes are different between the both systems.

[CPU Parameter] ⇨ [Redundant System Settings] ⇨ [Redundant Behavior Setting] ⇨ [Backup Mode Setting]

Window



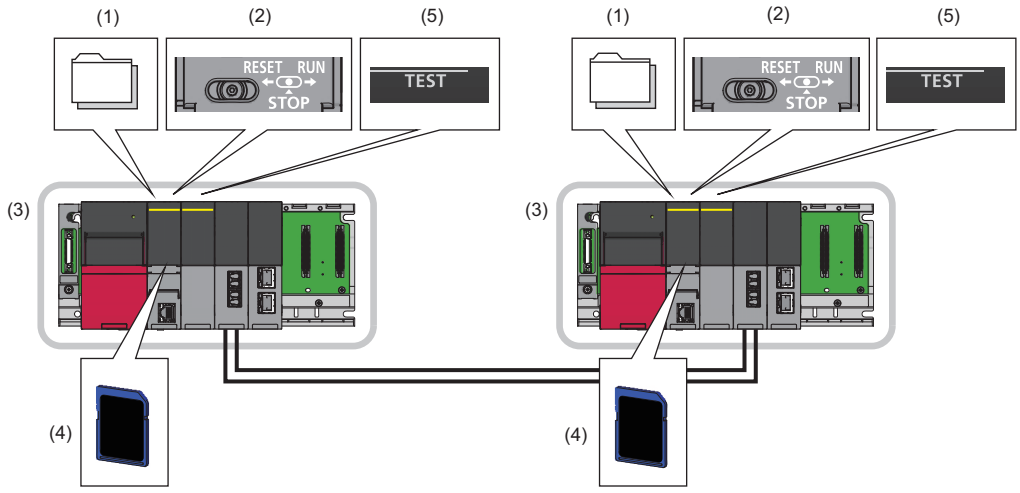
Displayed items

Item	Description	Setting range	Default
Safety Operation Mode Check	Set whether to perform the consistency check for the safety operation modes of both systems or not.	• Do Not Check • Check	Check

20.5 Program Execution in Both Systems

This function detects an error in the external device or network of the systems (control system and standby system) respectively by executing a program that diagnoses external devices and networks in both systems.
When "Both Systems Executions" is set to a program in "Both Systems Program Executions Setting", the program is executed by the CPU modules of both systems.

Ex.
When an error is detected in the external device connected to the standby system, the error is notified by generating a continuation error.



(1) A diagnostic program that is set as a program executed in both systems is executed.
(2) An error is detected in the external device connected to the standby system by the diagnostic program.
(3) The PALERT instruction is executed, and the detected error is displayed in the detailed information of the continuation error.

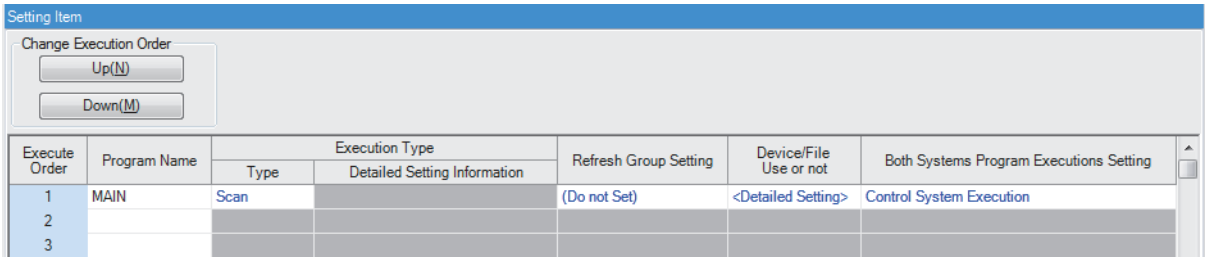
Programs whose execution type is the fixed scan execution type or event execution type and safety programs cannot be executed in both systems.

Both systems program executions setting

Set the parameter for each program to be executed in both systems.

[CPU Parameter] ⇨ [Program Setting]

Window



Displayed items

Item	Description	Setting range	Default
Both Systems Program Executions Setting	Set whether to execute a program only in the CPU module of the control system or in the CPU modules of both systems. • "Both Systems Executions" can be set for programs whose execution type is the initial execution type, scan execution type, and standby type. • Only "Control System Execution" can be set for programs whose execution type is the fixed scan execution type or event execution type and safety programs.	• Control System Execution • Both Systems Executions	Control System Execution

Point

To enable the output (Y) to external devices connected to the standby system in a program executed in both systems, set "Enable" in "Standby System Output Setting" of the CPU parameter. (☞ Page 258 Redundant System Operation Setting)

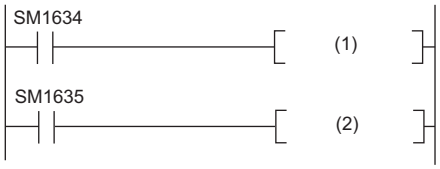
Operation of a program executed in both systems

The following table summarizes the operation of a program executed in both systems.

System	Operation
Control system	A program is executed according to its execution type.
Standby system	A program executed in both systems is executed according to its execution type. When an initial execution type program is set as a program executed in both systems, the program is executed in the first scan when the operating status of the CPU module is changed to RUN. When an initial execution type program is not set as a program executed in both systems, a scan execution type program that is set as a program executed in both systems is executed in the first scan when the operating status of the CPU module is changed to RUN.

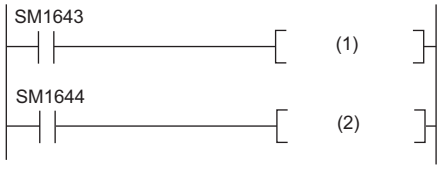
Point

- To perform different diagnostic processing in the control system and the standby system by using a program executed in both systems, use the special relay. SM1634 (Control system judgment flag) turns on only in the control system and SM1635 (Standby system judgment flag) turns on only in the standby system. The following is a program example that performs different processing by using the special relay. For the processing (1) and (2), use the SET instruction or another instruction to prevent duplication of a coil in the processing.



(1) Diagnostic processing in the control system, (2) Diagnostic processing in the standby system

- In a program executed in both systems, device/label values may differ between the control system and the standby system. If the systems are switched in this state, the program is executed using the different data and an unintended operation may be performed. To prevent this, the device/label values to be used can be initialized by using SM1643 (ON for only one scan after system switching (standby system to control system)) and SM1644 (ON for only one scan after system switching (control system to standby system)). For the processing (1) and (2), use the SET instruction or another instruction to prevent duplication of a coil in the processing.



(1) Initialization processing (standby system → control system), (2) Initialization processing (control system → standby system)

Operation at system switching

The operation of a program executed in both systems differs at system switching. The following table summarizes the operation at system switching.

Item		CPU module of the new control system	CPU module of the new standby system
Program execution	Initial execution type program	When an initial execution type program has not been completed in the old control system at system switching, it is executed again from the start.	When an initial execution type program has not been completed in the old standby system at system switching, it is not executed in the new standby system because the systems are switched after the program is completed in the old standby system.
	Scan execution type program	This type of program is executed from the step 0.	
Local device setting		This setting is in accordance with the parameter settings.	This setting is in accordance with the parameter settings.
File register setting		The file register setting before system switching is held.	The file register setting before system switching is held.
Direct access input (DX)		Data is input when an instruction using the direct access input (DX) is executed in the program after system switching.	
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 258 Redundant System Operation Setting)	
FROM/TO instructions		These instructions are executed when the instruction execution condition is satisfied in the program after system switching.	
Instruction that requires several scans		• The instruction is executed when the instruction execution condition is satisfied in the program after system switching. • When the systems are switched while the instruction is being executed, the instruction execution continues and the completion device turns on upon completion of the instruction execution.	

For the operation at system switching as for programs which are not set as those executed in both systems, refer to the following.

☞ Page 217 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 256 Program execution time
Constant scan		The constant scan function is invalid for the standby system.	—
Time required for system switching		If system switching is attempted to be performed while a program executed in both systems is being executed, the system is switched after the END instruction is completed. Thus, a time required for system switching may be extended.	—
Program execution type	Execution type change	The execution type of a program executed in both systems is not taken over at system switching. Even if the systems are switched after the execution type is changed by using a program control instruction (PSCAN(P), PSTOP(P), or POFF(P) instruction) on the old control system, the program is executed on the new control system with the execution type of the old standby system.	—
	Initial execution type	When the systems are switched while an initial execution type program set as a both systems execution program is being executed, the initial execution type program is executed twice in the CPU module of the new control system. The old standby system is switched to the new control system after the initial execution type program has been completed, and then the initial execution type program is executed again.	—
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 257 Tracking transfer
Device	Timer (T)	When the control system is switched to the standby system, the current value of the timer is not updated in the first scan of the new standby system and a timeout does not occur. As a result, an error of the time required for system switching + one scan is produced at system switching.	—
	Long timer (LT) and long retentive timer (LST)	When the long timer (LT) or long retentive timer (LST) is used in the standby system, time is not measured and a timeout does not occur. After the standby system has been switched to the control system, the long timer (LT) or long retentive timer (LST) is started up. To measure time in the standby system, use the timer (T).	—
	Interrupt pointer (I)	The interrupt pointer (I) cannot be used for the standby system.	—
Restricted instructions		Some instructions have restrictions when they are used in a program executed in both systems.	Page 257 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.

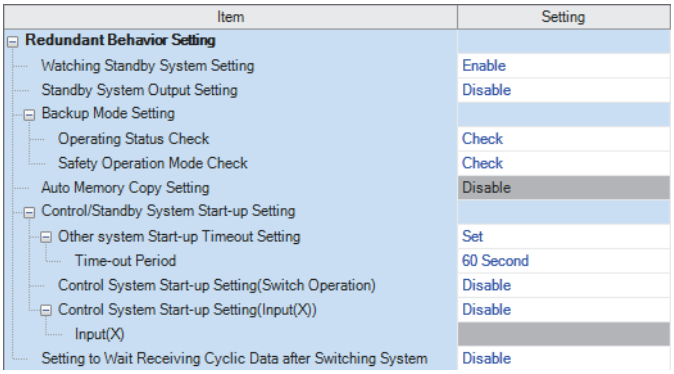
Classification	Notation	Description
Calling a subroutine program	CALL(P)	These instructions may not normally operate if a subroutine with any of a rising instruction, falling instruction, or SCJ instruction in a program which is not set as a both systems execution program is called in the standby system.
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)	

20.6 Redundant System Operation Setting

Set the redundant system operation of a system using the SIL2 Process CPU in the redundant settings of the CPU parameter.

[CPU Parameter] ⇨ [Redundant System Settings]

Window



Displayed items

Item		Description	Setting range	Default
Watching Standby System Setting		Set this item not to detect continuation errors when: • A Communication error with the other system has occurred or communications with the other system has been disabled. • The CPU module of the standby system is off, has been reset, or a stop error has occurred. This item is enabled only for the CPU module of the control system.	• Disable • Enable	Enable
Standby System Output Setting		Set this item to enable the output (Y) from the standby system for system inspection or adjustment in the standby system. • To diagnose devices that are connected to the standby system by using a both systems execution program, set "Enable". However, set the standby system output setting to "Disable" when common devices have been connected to both the control system and standby system. • When "Enable" is set, do not specify outputs (Y) to be used in the standby system as the data to be tracked. If such an operation is performed, the outputs (Y) are overwritten with the control system data and output from the standby system. (Even when the CPU module of the standby system is in the STOP state, the output (Y) received in tracking transfer is output.)	• Disable • Enable	Disable
Backup Mode Setting		☞ Page 250 Operating status, ☞ Page 251 Safety operation mode		
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 the own system is started up and the initial processing is completed. • When "Set" is selected, a stop error will occur in the own system unless communications are performed with the other system even though the timeout time has come. • When "Not Set" is set, the own system will wait until communications with the other system are established.	• Not Set • Set	Set
	Time-out Period		3 to 1800s (in increments of 1s)	60s
	Control System Start-up Setting (Switch Operation)	Set this item to enable the operation to be started in the control system with the switch operation (RUN→STOP→RUN) while the own system that has started is waiting for the start-up of the other system.	• Disable • Enable	Disable
	Control System Start-up Setting (Input (X))	Set this item to enable the operation to be started in the control system with the contact input (X) while the own system that has started is waiting for the start-up of the other system.	• Disable • Enable	Disable
	Input (X)		X0 to X2FFF	—
Setting to Wait Receiving Cyclic Data after Switching System*1		Set this item to wait for the execution of the sequence program until all CC-Link IE Field Network modules on the main base unit receive the latest cyclic data after system switching. ☞ Page 260 Setting to wait cyclic data receive after system switching	• Disable • Enable	Disable

*1 Before enabling the setting to wait cyclic data receive after system switching, check the versions of the CPU module, CC-Link IE Field Network module, and engineering tool used. (☞ Page 566 Added and Enhanced Functions)

Standby system output setting

Output timing

When "Standby System Output Setting" has been set to "Enable", the output timing from the standby system is at the completion of the END processing or depends on the refresh group setting or refresh settings of each module. (☞ Page 121 Group setting for refresh)

Therefore, when a program set in the refresh group setting is a program executed in the control system, the I/O refresh is not performed while the module is running because the program is not executed in the standby system. (When the CPU module is in the STOP/PAUSE state, the I/O refresh is performed at the timing of the END processing.)

To execute a program set in the refresh group setting in the standby system, set "Both Systems Program Executions Setting". (☞ Page 252 Program Execution in Both Systems)

Operation at system switching

The operation performed at system switching differs when "Standby System Output Setting" is set to "Enable". The following table shows the operation performed when "Standby System Output Setting" has been set to "Enable".

Item	CPU module of the new control system	CPU module of the new standby system
Output (Y)	The status of the old standby system is held and output refresh is performed.	The status of the old control system is held and output refresh is performed.
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 252 Program Execution in Both Systems

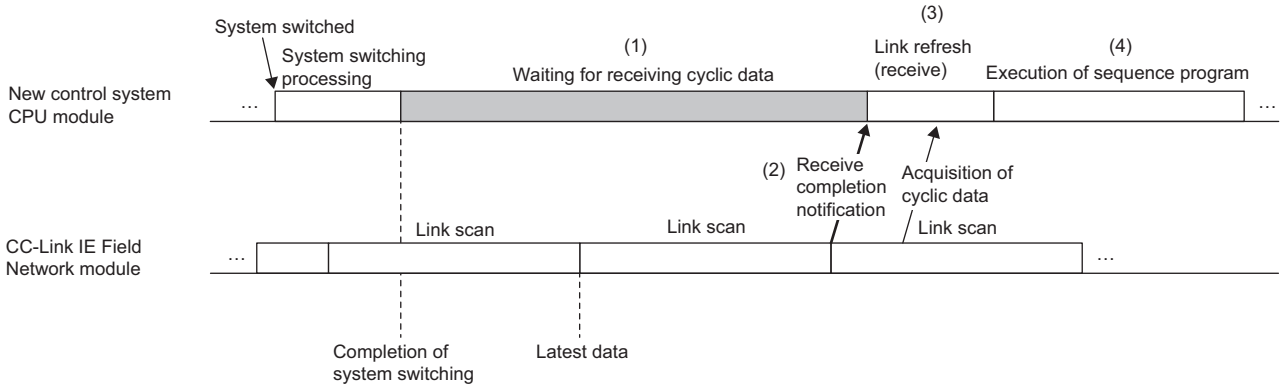
Setting to wait cyclic data receive after system switching

When system switching occurs in the redundant line structure of CC-Link IE Field Network, set this item to start the execution of the program in the new control system with new cyclic data obtained after system switching.

Restriction

Before enabling the setting to wait cyclic data receive after system switching, check the versions of the CPU module, CC-Link IE Field Network module, and engineering tool used. (☞ Page 566 Added and Enhanced Functions)

When this setting is enabled, the execution of the program is suspended until new cyclic data is received in the new control system after system switching.



(1) After system switching has been completed, the CPU module stands by until it is notified of the completion of the cyclic data receipt after system switching by all CC-Link IE Field Network modules on the main base unit.
(2) When the CC-Link IE Field Network modules receive cyclic data from the remote I/O station after system switching, the CC-Link IE Field Network modules notify the CPU module of the completion of the cyclic data receipt.
(3) When notified of the completion of the cyclic data receipt by the CC-Link IE Field Network modules, the CPU module executes link refresh (receive).
(4) The CPU module executes the sequence program using the new cyclic data.

- When this setting is enabled, the extended time from the completion of system switching to the first output is added to the cyclic data receipt waiting time. (☞ Page 556 Extended time until initial output after system switching (Tjo))
- This setting is executed in the new control system after system switching. This setting operates when the operating status of the CPU module in the new control system is RUN, STOP, or PAUSE. (This setting does not operate when a stop error has occurred in the CPU module in the new control system.)
- This setting operates when the link scan mode of the CC-Link IE Field Network is set to "Sequence Scan Asynchronous" or "Constant Link Scan". This setting does not operate when the link scan mode of the CC-Link IE Field Network is set to "Sequence Scan Synchronous Setting". (MELSEC iQ-R CC-Link IE Field Network User's Manual (Application))
- When cyclic data cannot be received within the timeout time due to disconnection of the network cable or other causes while cyclic data receipt is waited for, waiting for cyclic data receipt is stopped, and the sequence program is executed. Occurrence of a timeout can be checked in SM1756 (wait timeout for receiving cyclic data after system switching) and SD1756 (module information on wait timeout for receiving cyclic data after system switching).

Precautions

- Scan time monitoring with the watchdog timer is interrupted while cyclic data receipt is waited for. Thus, no error is detected even if the scan time monitoring time has elapsed while cyclic data receipt is waited for.
- Constant scan is disabled while cyclic data receipt is waited for. Thus, no error is detected even if the constant scan setting time has elapsed. Constant scan is enabled after waiting for cyclic data receipt has been completed, and the sequence program is executed.
- Because device/label access service processing is not accepted while the cyclic data receipt is waited for, set the communication timeout time with external devices in consideration of the cyclic data receipt waiting time. (☞ Page 556 Waiting time for cyclic data receive after system switching (Twcyc))

20.7 Redundant Function Module Communication Test

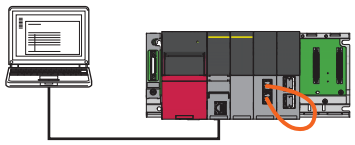
The hardware of the redundant function module is checked for an error when its communication is unstable.

The following table shows the test items included in the module communication test.

Test item	Description
Internal self-loopback test	Checks whether the communication function of the redundant function module normally operates.
External self-loopback test	Checks whether communications can be normally performed with a tracking cable that connects the IN connector and OUT connector of the redundant function modules.

Execution procedure of the module communication test

1. Connect the engineering tool directly to the CPU module of the standby system.
2. Set the CPU module operating status to the STOP state.
3. Connect the IN and OUT connectors of the standby system redundant function module with a tracking cable. (A cable disconnection error is detected in the control system.)



4. Open the "Redundant Operation" window of the engineering tool.

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

5. Click the [Execute Test] button of "Module Communication Test".

- LED status of the redundant function module during the module communication test

Status	RUN LED	ERR LED
Module communication test in execution	Flashing	Off
Completed successfully	On	Off
Completed with an error	On	On

6. When the test is completed with an error, take actions according to "Corrective Action" in the test result window.
7. When the test is completed successfully, connect the control system and standby system with the tracking cable. (☞ Page 49 Redundant function modules)
8. Click the [Close] button in the "Redundant Operation" window to exit the module communication test.
9. Set the CPU module operating status to the RUN state.

Point

A tracking communication error is detected when a tracking cable is disconnected.

Precautions

- Always connect the IN and OUT connectors of the redundant function module with the tracking cable before performing the module communication test.
- Ensure that the module communication test is performed on the CPU module of the standby system. Performing the test on the CPU module of the control system may cause an unintended operation.

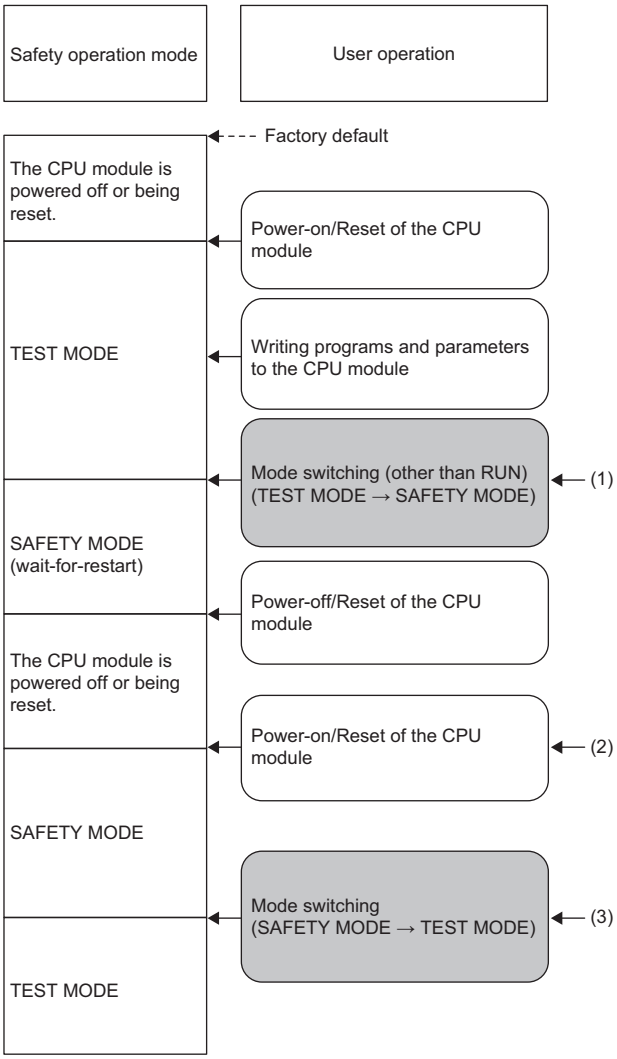
21 SWITCHING SAFETY OPERATION MODE

This section describes how to switch the safety operation mode.

Safety operation mode transition timing

The following table shows the timing at which safety operation mode switches.

Safety operation mode switching direction	Safety operation mode switch timing
TEST MODE → SAFETY MODE	When the CPU module is powered off and on or is reset after the safety operation mode switching operation
SAFETY MODE → TEST MODE	At the safety operation mode switching operation



(1) After switching from TEST MODE to SAFETY MODE, the safety operation mode will be SAFETY MODE (wait-for-restart). (The status will be wait-for-restart until the CPU module is powered off or is reset.) An error occurs if the CPU module is switched to the RUN state when in SAFETY MODE (wait-for-restart).

(2) After switching the safety operation mode (TEST MODE → SAFETY MODE), the mode will switch to SAFETY MODE when the CPU module is powered off and on or is reset.

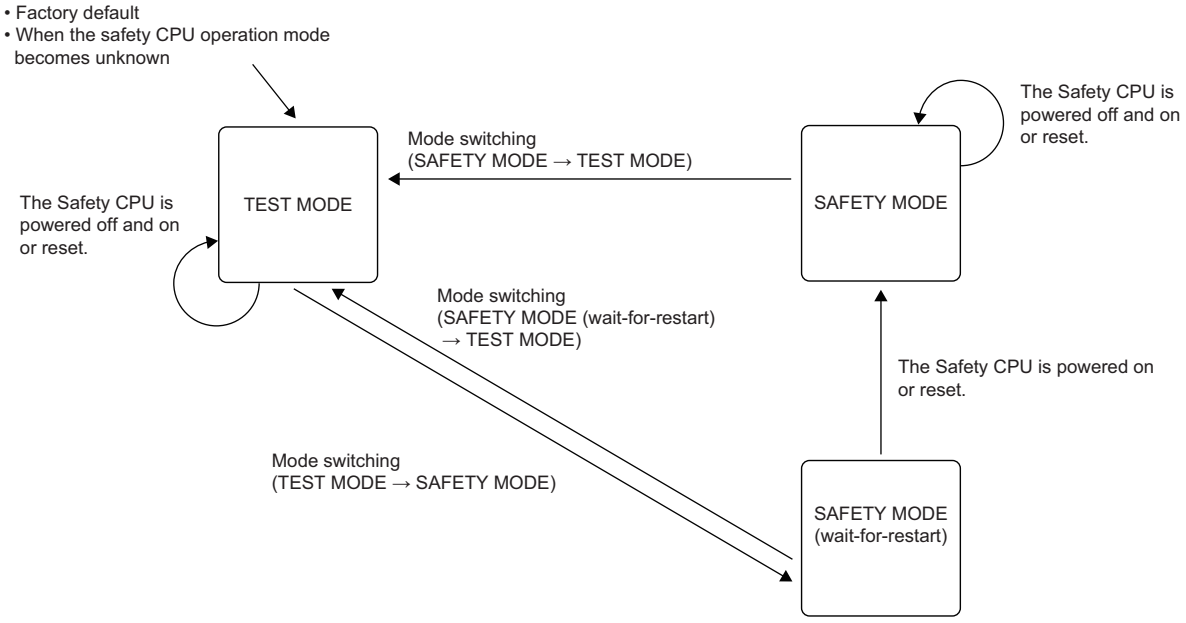
(3) After switching the safety operation mode (SAFETY MODE → TEST MODE), the mode will switch to TEST MODE.

Point

The CPU module retains the safety operation mode status even when the power is turned off, or a reset is performed.

Safety operation mode state transition

The CPU module retains the safety operation mode even in the event of a power failure, and therefore the current safety operation mode state will remain unchanged even if the CPU module is powered off and on or is reset. The SIL2 function module does not retain the safety operation mode in the event of a power failure, and the safety operation mode will be the same as that of the CPU module when the SIL2 function module is powered off and on or is reset.



Safety operation mode status	State transition by operation		
	Power off → on or reset	Safety operation mode switching operation	
		TEST MODE → SAFETY MODE	SAFETY MODE → TEST MODE
TEST MODE	→ TEST MODE	→ SAFETY MODE (wait-for-restart)	—
SAFETY MODE	→ SAFETY MODE	—	→ TEST MODE
SAFETY MODE (wait-for-restart)	→ SAFETY MODE	—	→ TEST MODE

Point

If [Online] ⇨ [User Authentication] ⇨ [Initialization of all PLC Data] is performed, the safety operation mode will revert to the default mode (TEST MODE). All programmable controller information can be initialized regardless of the safety operation mode.

Safety operation mode switching conditions

The following table lists the conditions under which the safety operation mode can be changed.

Safety operation mode switching direction	Condition
TEST MODE → SAFETY MODE	The user who operates the mode switching has "Developers" access level or higher, and is currently logged on.
	The safety operation mode is currently set to the TEST MODE.
	The CPU module operating status is other than RUN.
	The CPU module is not in the process of changing operating status.
	The following files held by the engineering tool, CPU module, and SIL2 function module match. • Safety CPU parameter • Safety module parameter • Safety program • Safety FB file • Safety global label setting file • Standard/safety shared label setting file
	The following files are not being written from another engineering tool. • Safety CPU parameter • Safety module parameter • Safety program • Safety FB file • Safety global label setting file • Standard/safety shared label setting file
SAFETY MODE → TEST MODE	The user who operates the mode switching has "Developers" access level or higher, and is currently logged on.
	The safety operation mode is currently set to SAFETY MODE or SAFETY MODE (wait-for-restart).
	The CPU module is not in the process of changing operating status.

How to switch the safety operation mode

The safety operation mode can be switched using the engineering tool.

[Online] ⇨ [Safety PLC Operation] ⇨ [Switch Safety Operation Mode]

The execution target of mode switching can be specified from the following options.

Execution target specification	Operation
Currently Specified Station	Switches the safety operation mode for the CPU module specified in the transfer setup.
Specify Both Systems	Switches the safety operation modes for the CPU modules of both systems.

For details, refer to the following.

GX Works3 Operating Manual

22 IDENTIFICATION CHECK FOR SAFETY DATA

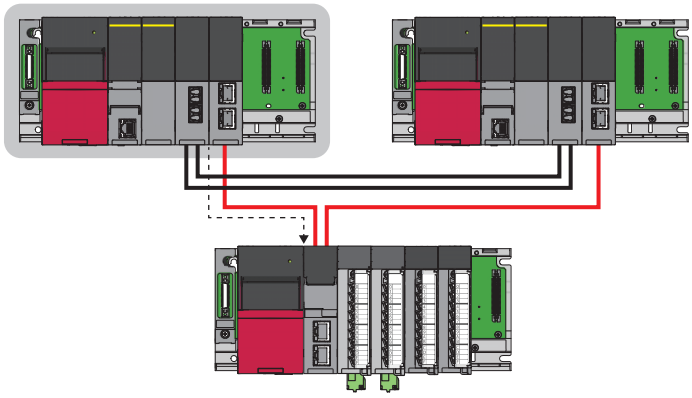
This function checks if the project data created using the engineering tool and the data in the CPU module are the same, and confirms that the program executed in SAFETY MODE is the one written by the user.
This function allows the user to inspect whether the files in the engineering tool and the CPU module match or not.

[Online] ⇨ [Safety PLC Operation] ⇨ [Check Safety Data Identity]
For details, refer to the following.
GX Works3 Operating Manual

MEMO

23 SAFETY COMMUNICATION FUNCTION

This function communicates data between the CPU module of the control system and modules supporting safety functions using safety protocols. Safety communication processing is not performed in the CPU module of the standby system.



Point

The SIL2 Process CPU performs safety communications between safety stations using the following network.

• CC-Link IE Field Network (safety communication function)

For details on the function and settings required for safety communications, refer to the following.

MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

23.1 Safety I/O Hold Time

The safety I/O hold time (Tioh) is the time until safety communications are disconnected (the safety output is turned off) after detection of a timeout for the safety refresh monitoring time of safety communications.

Safety I/O hold time setting

The safety I/O hold status of each safety connection can be checked with SA\SD1600 to SA\SD1663 (safety I/O hold status of each safety connection).

To prevent the safety communications from disconnecting at a system switching, configure the following setting.

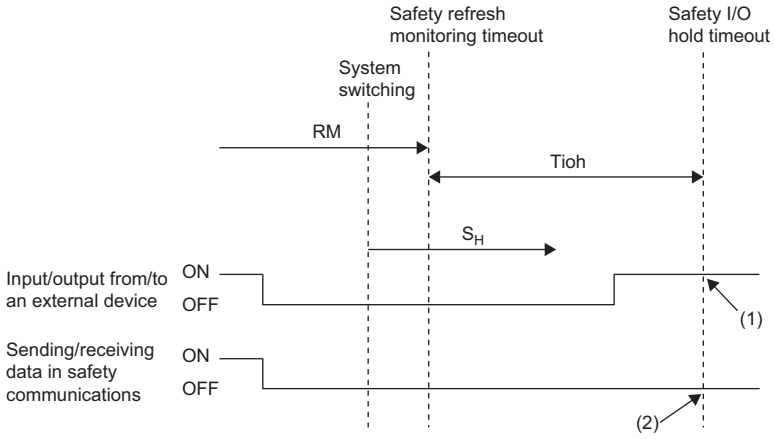
Safety I/O hold time (Tioh) > Safety data holding time at a system switching (S_H)

For the safety data holding time at a system switching, refer to the following.

MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

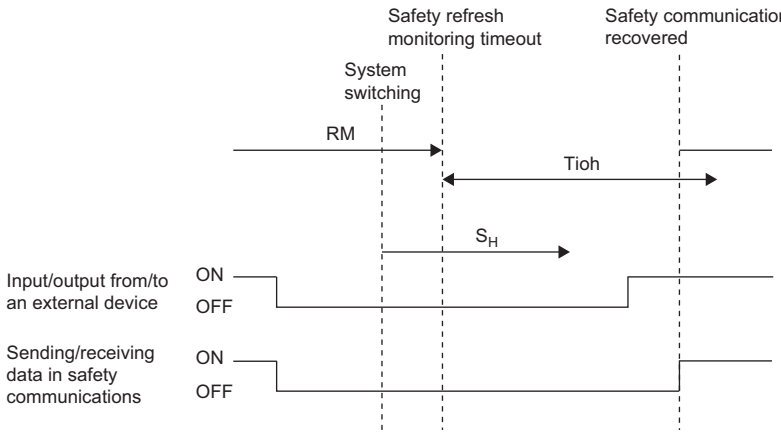
The following figures show the relationship between the input/output of the external device and sending/receiving data in safety communications.

- When safety communications are not recovered within the safety I/O hold time



RM: Safety refresh monitoring time
Tioh: Safety I/O hold time
S_H: Safety data holding time at a system switching
(1) The safety communications are disconnected. For the input values from the external device and output values to the external device, refer to the manual for each I/O module.
(2) A safety communication error occurs, and the safety station interlock is triggered.

- When the safety communications are recovered within the safety I/O hold time



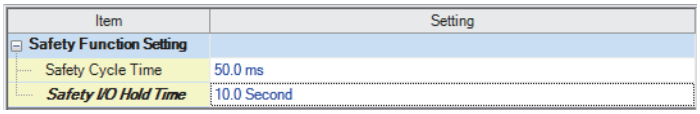
RM: Safety refresh monitoring time
Tioh: Safety I/O hold time
S_H: Safety data holding time at a system switching

Setting method

The following shows the setting method of the safety I/O hold time.

[CPU parameter] ⇨ [Safety Function Setting]

Window



Displayed items

Item	Description	Setting range	Default
Safety I/O Hold Time	Set the safety I/O hold time. If safety communications recovered successfully within the set safety I/O hold time, processing continues.	0.0 to 180.0s (in increments of 0.1s)	10.0s

Point

Note that the time until a timeout of safety communications increases in length by the amount of the setting value of the safety I/O hold time.

24 SLMP COMMUNICATIONS

This section describes the precautions on communications using the SLMP.

System switching

There are the following notes when the system IP address matching function is not used.

■Re-setting of the connection destination

When the relay CPU module is in the communication-disabled state (power-off, reset, or tracking cable disconnection) at system switching, the connection destination needs to be set again for communications using the SLMP.

■Re-execution of the write command

When "Control System" or "Standby System" has been selected in the transfer setup and the systems are switched, a target system mismatch is detected by a command issued during the system switching, and a communication error occurs. If a communication error occurs while a data write command is being issued, the data write command needs to be issued for the new control system.

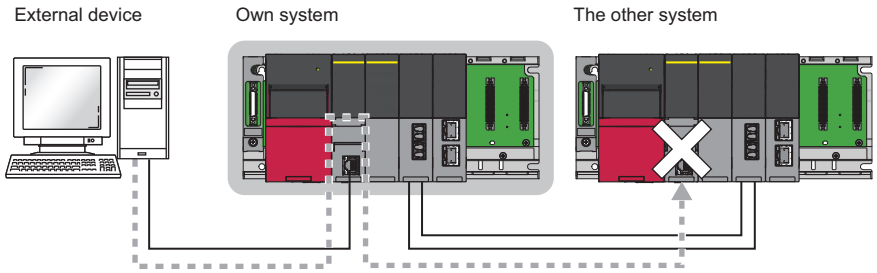
Remote operation

If a remote operation command is executed, the CPU modules enter different operating statuses and thus systems cannot be switched.

If the safety operation mode is SAFETY MODE, the safety global devices, safety global labels, safety local labels, and standard/safety shared labels are not cleared even when the clear mode is set to perform the clear processing using remote RUN.

Communicating with other systems

For SLMP communications via the built-in Ethernet port of the CPU module, when communications are performed to the other system that cannot respond (power-off, reset, or tracking cable disconnection), a timeout error may occur.



MEMO

25 PRECAUTIONS ON PROGRAMMING

This chapter describes precautions on programming.

25.1 Restricted Instructions

This section describes the instructions with restrictions.

Instructions that need to be executed again in a new control system

For an instruction that requires several scans for completing the processing, the instruction will be continuously executed when the system switching is performed during execution of the instruction. When a completion device has been used in an execution program of the control system, the completion device will not turn on even though the instruction is completed after the system switching from the control system to the standby system. However, the completion device will turn on after the system switching from the control system to the standby system and then to the control system again. Note that the completion status of the completion device is not reflected to the tracking device. When the system switching is performed during execution of an instruction, execute the instruction again as required.

Classification	Instruction symbol
Data processing instructions	SORTD(_U), DSORTD(_U)
Reading/writing data instructions	SP.DEVST, SP.FREAD, SP.FWRITE
Open/close processing instructions	SP.SOCOPEN, SP.SOCCLOSE
Socket communications instructions	SP.SOCRCV, S.SOCRCVS, SP.SOCSND, SP.SOCCINF, SP.SOCCSET, SP.SOCRMODE, S(P).SOCRDATA
Module dedicated instructions	Instructions that require several scans for completing processing

For some instructions, an error will occur if an instruction is executed during execution of the same instruction. For the operation for each instruction that is executed again during execution of the same instruction, refer to the following.

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

Re-execution of instruction when systems are switched during instruction execution

When the system switching is performed while an instruction that requires several scans is being executed, the instruction can be executed again in the new control system after the system switching by using the programs such as following.

■REMFR instruction

When the system switching is performed while the instruction is being executed (M201 = ON), SM1643 (ON for only one scan after system switching (standby system to control system)) will turn on for one scan in the new control system and the REMFR instruction will be executed again on station number 10 of network number 1.

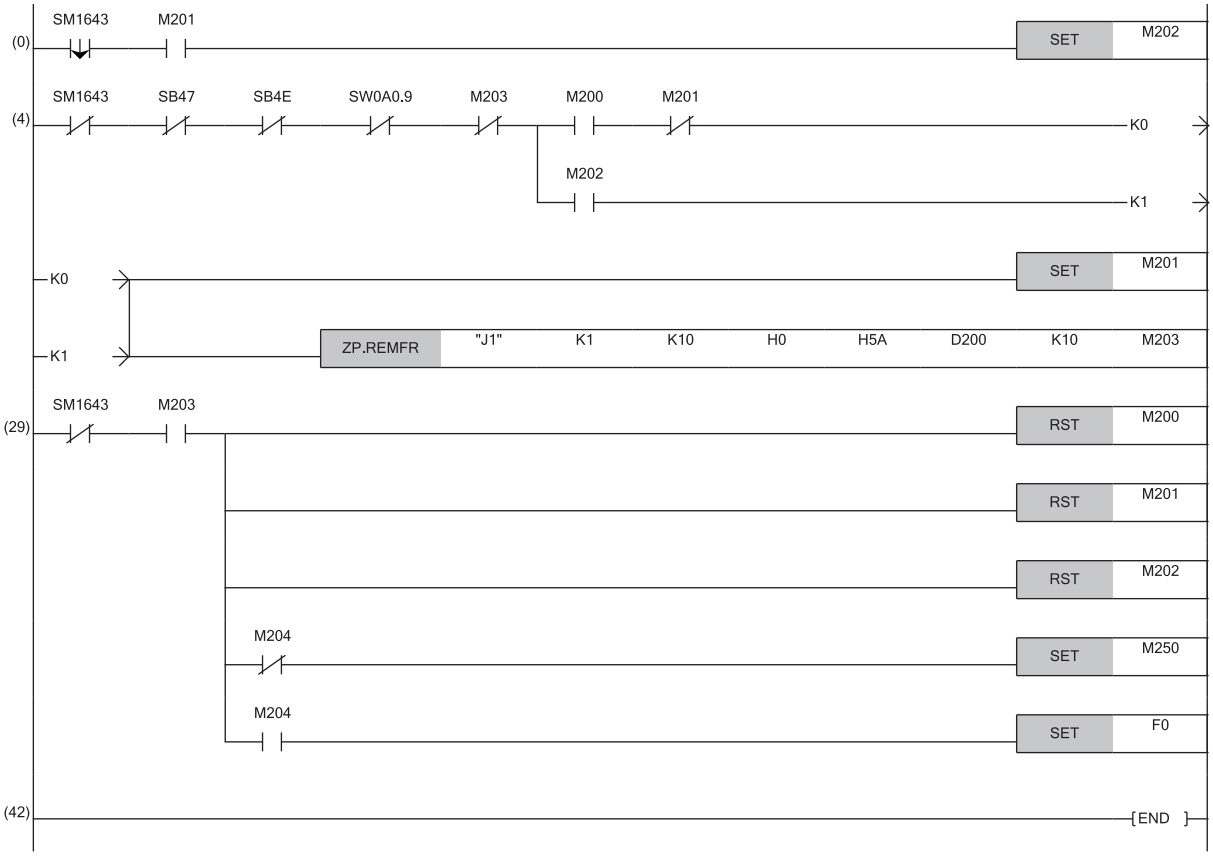
• Devices used

Device	Description
SM1643	ON for only one scan after system switching (standby system to control system)
SB47*1	Baton pass status of own station
SW0A0.9*1	Baton pass status of each station
M200*2	Reading request
M201*2	At instruction execution
M202*2	ON at instruction re-execution request due to system switching
M203*2	Completed without an error
M204*2	Completed with an error

*1 For details on the link special relay (SB) and link special register (SW), refer to the manuals for the network used.

*2 Change the device number according to the system.

• Program example



■REMTO instruction

When the system switching is performed while the instruction is being executed (M101 = ON), SM1643 (ON for only one scan after system switching (standby system to control system)) will turn on for one scan in the new control system and the REMTO instruction will be executed again on the station number 10 of the network number 1. When the system switching is performed while a write instruction such as the REMTO instruction is being executed, execution of the instruction may have been suspended before completion of writing data to the target module. Thus, insert an interlock in the new control system to read the X signals and buffer memory status of the target module and to determine whether or not to execute the instruction again.

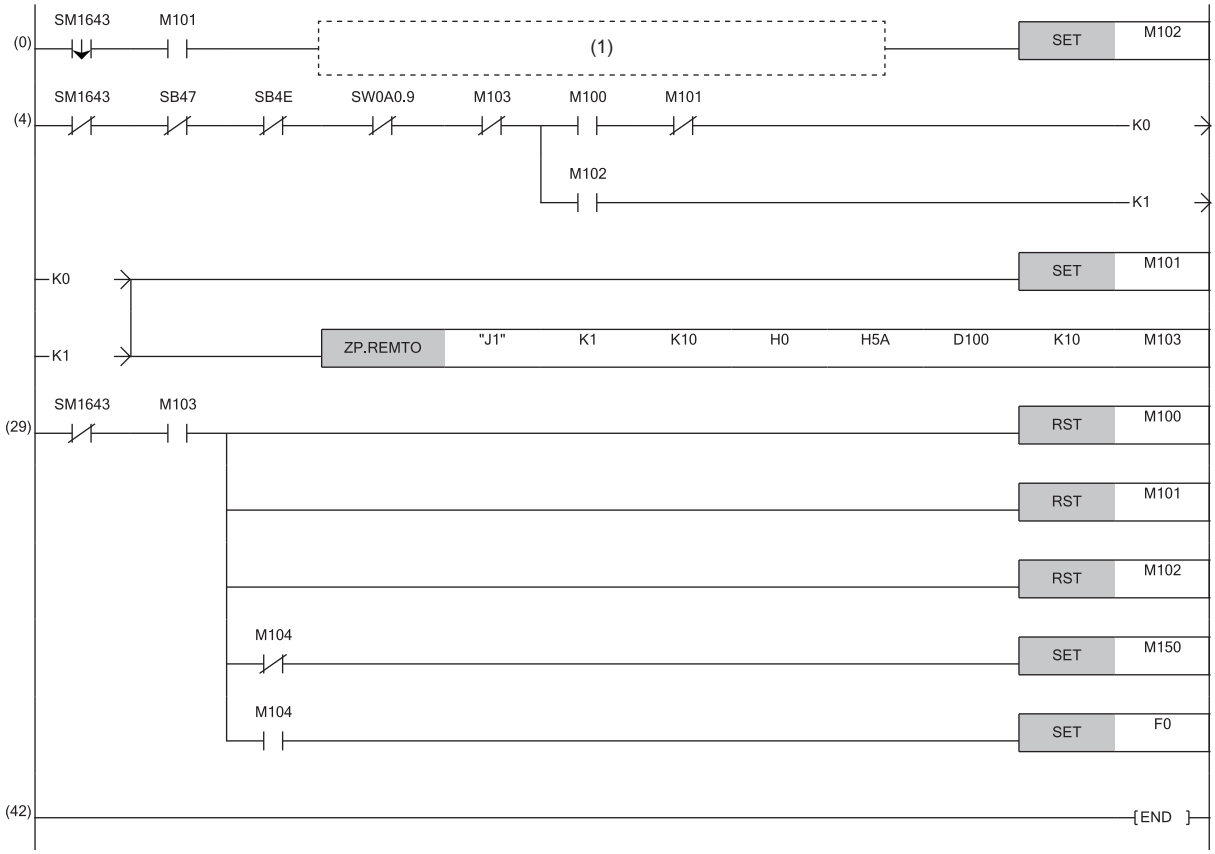
• Devices used

Device	Description
SM1643	ON for only one scan after system switching (standby system to control system)
SB47*1	Baton pass status of own station
SW0A0.9*1	Baton pass status of each station
M100*2	Write request
M101*2	At instruction execution
M102*2	ON at instruction re-execution request due to system switching
M103*2	Completed without an error
M104*2	Completed with an error

*1 For details on the link special relay (SB) and link special register (SW), refer to the manuals for the network used.

*2 Change the device number according to the system.

• Program example



(1) Add an interlock circuit for determining whether or not to execute the instruction again (according to the X signals and buffer memory status of the target module) as required.

Instructions whose operations vary depending on tracking of the signal flow memory

This section describes the instructions whose operations after the system switching vary depending on whether the signal flow memory is tracked or not. The operations vary when one of the following instructions is executed among program organization units that have the signal flow memory, memory to which tracking can be performed.

Classification/type of special relay	Instruction symbol
Rising instruction	LDP, ANDP, ORP, LDPI, ANDPI, ORPI, PLS, MEP, EGP, SET F, RST F, FF, LEDR, DUTY, LOGTRG, LOGTRGR, □P (including MOVP and INCP), SP.□, JP.□, GP.□, ZP.□
Falling instruction	LDF, ANDF, ORF, LDFI, ANDFI, ORFI, PLF, MEF, EGF
SCJ instruction	SCJ
Data processing instructions	SORTD(_U), DSORTD(_U)
TIMCHK instruction	TIMCHK
XCALL instruction	XCALL
Rising instruction using SM1643 as an execution condition	—

Rising instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, a rising instruction whose execution condition turned on during system switching will not be executed.

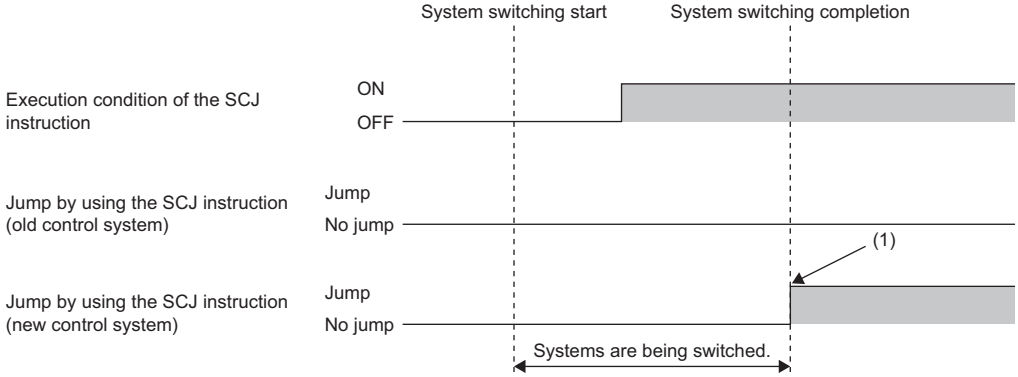
Falling instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, a falling instruction whose execution condition turned off before the system switching will be executed.

SCJ instruction

■When the signal flow memory is not tracked

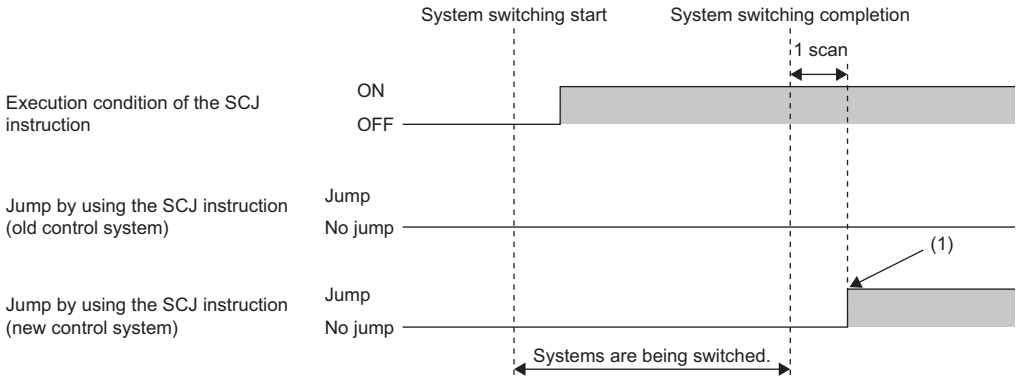
Once the system switching is performed, the execution condition of the SCJ instruction turns on after the signal flow memory of the new control system turns on. In the new control system, processing jumps to the pointer specified by the SCJ instruction in the first scan.



(1) After system switching, the processing jumps in the first and later scans.

■When the signal flow memory is tracked

When the system switching is performed, the execution condition of the SCJ instruction turns on while the signal flow memory remains off. In the new control system, the processing jumps to the pointer specified by the SCJ instruction in the second scan.



(1) After system switching, the processing jumps in the second and later scans.

Data processing instructions

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, the SORTD(_U)/DSORTD(_U) instruction in the first scan after the system switching will be executed not as the first execution but as continuous execution (continuous processing). When the instruction is executed for the first time, the instruction is executed without data to be stored in the devices used by the system, causing an unintended operation.

TIMCHK instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, the TIMCHK instruction in the first scan after the system switching will be executed not as the first execution but as continuous execution (continuous processing). At the first execution, the current value is not cleared and the device that turns on at timeout is not turned off. The instruction is executed with the status at the previous measurement.

XCALL instruction

If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on. Thus, the subroutine program will not be executed when the execution condition of the XCALL instruction remains off in the first scan after system switching.

Rising instruction using SM1643 as an execution condition

SM1643 is the special relay that turns on for one scan in the new control system after system switching. If the system switching is performed without the signal flow memory being tracked, the signal flow memory of the new control system turns on and a rising instruction cannot be executed. To execute a rising instruction where SM1643 has been set as an execution condition, use a falling edge pulse operation contact (LDF/AND/ORF instruction) and create a program in which the rising instruction is to be executed at the falling edge of SM1643, as shown below. However, when the falling edge of SM1643 has been set as the execution condition, the target instruction will be executed in the second scan after the system switching.



(1) The instruction will be executed in the second scan after the system switching.

Instructions that affect the status of another instruction when executed

When one of the following instructions is executed and the status of another instruction changes, the new status will not be tracked to the other system. When the system switching is performed during execution of an instruction, execute the instruction again as required.

Classification		Instruction symbol
Program execution control instructions	Disabling interrupt programs	DI
	Enabling interrupt programs	EI
	Disabling interrupt programs with specified priority or lower	DI
	Interrupt program mask	IMASK
	Disabling/enabling the specified interrupt pointer	SIMASK
File register operation instructions	Switching the file register block number	RSET(P)
	Changing the file register file name	QDRSET(P)
Timing check instruction	Generating timing pulses	DUTY

Instructions causing different operation results between both systems

This section describes the instruction that causes different operation results between the control system and standby system after system switching.

PID control instructions

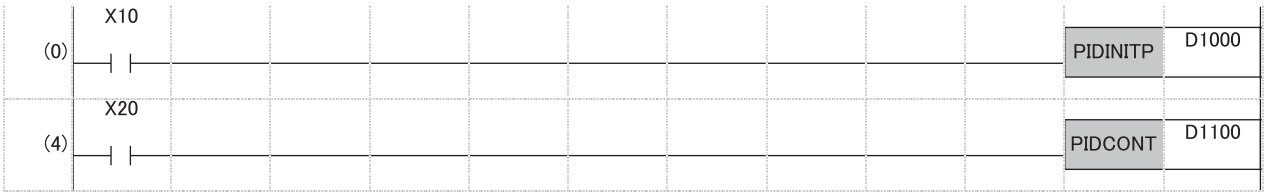
When using the following PID control instructions, include the number of device points used by the systems in tracking target data. Otherwise, the instructions cause different operation results between the control system and standby system after system switching.

n: Total number of loops

Classification		Instruction symbol	Number of device points used
PID control instructions (Inexact differential)	Registering the PID control data to the CPU module	S(P).PIDINIT	2 + n × 4
	PID operation	S(P).PIDCONT	10 + n × 23
PID control instructions (Exact differential)	Registering the PID control data to the CPU module	PIDINIT(P)	2 + n × 10
	PID operation	PIDCONT(P)	10 + n × 18

Ex.

When the number of loops is eight, the PIDINITP instruction needs 2 + 8 × 10 = 82 words and the PIDCONT instruction needs 10 + 8 × 18 = 154 words. Thus, include D1000 to D1081 and D1100 to D1253 in tracking target data.



Precautions for using the COM or ZCOM instruction

When the network module link refresh is performed by using the COM or ZCOM instruction, output from the remote I/O station may change after system switching. To prevent this, do not perform the network module link refresh with the COM or ZCOM instruction. For the COM instruction, whether or not to perform the network module link refresh can be set using SM775 (Selection of refresh processing during the COM instruction execution) and SD775 (Selection of refresh processing during the COM instruction execution). Set SM775 and SD775 and perform only the I/O refresh and device/label access service processing with the COM instruction.

○: Selectable, ×: Not selectable

Instruction	Refresh processing	In a system using the CPU module
COM instruction	I/O refresh	○
	Network module link refresh	○*1
	Intelligent function module refresh	×*2
	Refresh using the CPU buffer memory of the multiple CPU system (END)	×*3
	Device/label access service processing (communications with the engineering tool, GOT, or other external devices)	○
ZCOM instruction	Network module link refresh	○*1
	Intelligent function module refresh	×*2

*1 When the COM or ZCOM instruction is executed, tracking transfer is not performed. Thus, when the system switching is attempted after execution of the instruction and before completion of tracking transfer, the system switching is performed without tracking transfer. Thus, even though output to the network module is changed by the COM or ZCOM instruction with the CPU module in the control system, the change will not be reflected to the CPU module in the standby system. After the system switching, the CPU module in the new control system outputs old values to the network module, and output from the network module may vary.

*2 This processing is not selectable because no intelligent function modules can be mounted on main base units.

*3 This processing is not selectable because a multiple CPU system cannot be built.

Precautions for using the ADRSET instruction

Even though a file is written to both the control system and standby system, different addresses are assigned to the systems. To continue the processing in the new control system, use the ADRSET instruction to obtain indirect addresses again.

25.2 Interrupt from modules

This section describes the precautions for interrupts from modules.

When the old control system is switched to the new standby system

The old control system retains the interrupt factors that have occurred even after the system is switched to the new standby system by system switching before execution of an interrupt program. After the systems are switched again, the interrupt program of an interrupt factor that the old control system has retained will be executed.
Since the interrupt factor that has occurred on the old control system is not inherited to the new control system, the interrupt program of an interrupt factor that has occurred on the old control system will not be executed on the new control system.

When the old standby system is switched to the new control system

The old standby system retains the interrupt factors that have occurred. After the old standby system is switched to the new control system by system switching, the interrupt program of an interrupt factor that the old standby system has retained will be executed on the new control system.
When the old standby system has retained multiple interrupt factors, the scan time may greatly increase.

25.3 Precautions for using the annunciator (F)

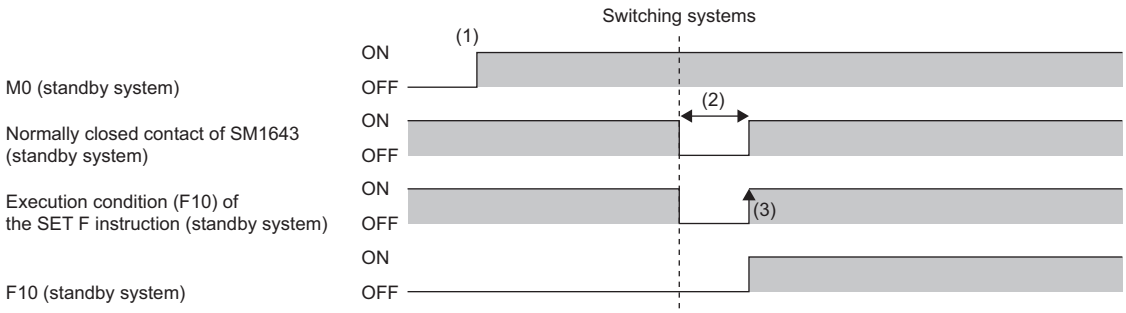
This section describes the precautions for using the annunciator (F).

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.

25.4 Precautions on timers and timer function blocks

This section describes the precautions on timers and timer function blocks at system switching.

Current values at system switching

For the timer (T), retentive timer (ST), and a timer function block TIMER_☐_M, the current values of the timers in the first scan of the CPU module of the new control system will not be updated after system switching.

Timeout before system switching

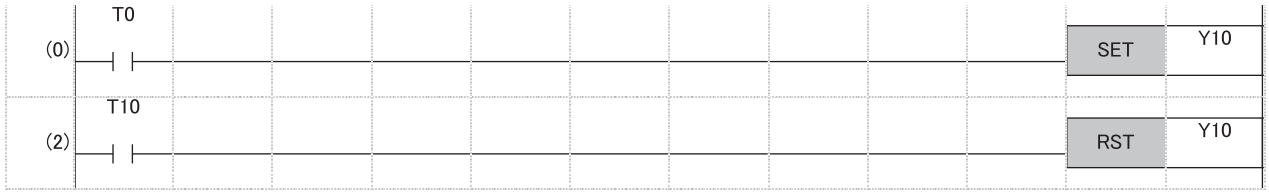
Depending on the timing to perform system switching such as power-off, tracking transfer processing is suspended and tracking data may not be reflected to the CPU module in the new control system. For the timer (T), retentive timer (ST), long timer (LT), long retentive timer (LST), and timer function blocks TIMER_☐_M, TP(_E), TON(_E), and TOF(_E), the timer whose time has been up before system switching may go into the state in which no timeout has occurred in the first scan after system switching.

When values are output (writing values to the buffer memory and the output (Y)) with a timer contact or an output variable, the timer statues may go into the state in which no timeout has occurred as described above, causing chattering of the output. To transfer data with modules or external devices using the output (Y) or buffer memory, a program may not properly function due to chattering of output after the system switching. To output values (writing data to the buffer memory and the output (Y)) to modules or external devices with a timer contact or an output variable, consider the time taken for data to be transferred from the CPU module in the control system to the CPU module in the standby system after the time is up.

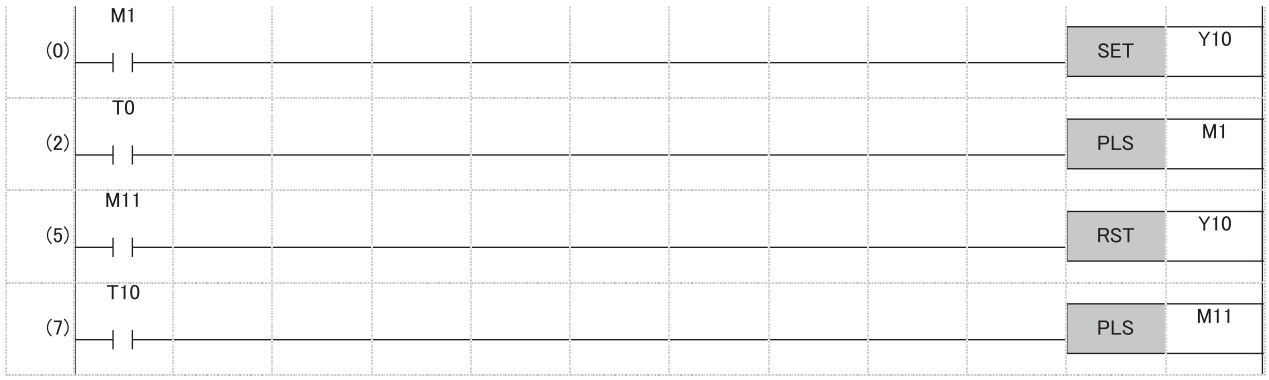
Ex.

Program that delays outputting values by one scan after the time of the timer (T) is up

[Without measures]



[With measures]



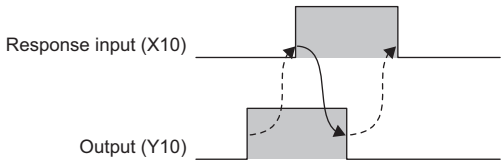
Point

In the CPU parameter, select "Transfer" (default setting) in "Signal Flow Memory Tracking Setting" of "Redundant System Settings". (☞ Page 232 Tracking transfer setting for the signal flow memory)

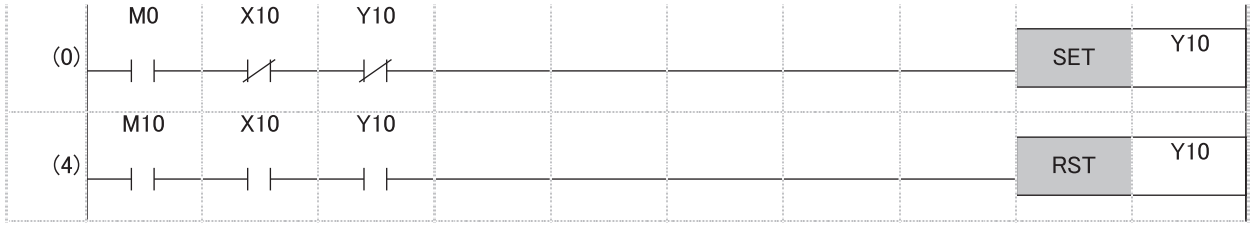
25.5 Precautions on access to intelligent function module or external devices

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.

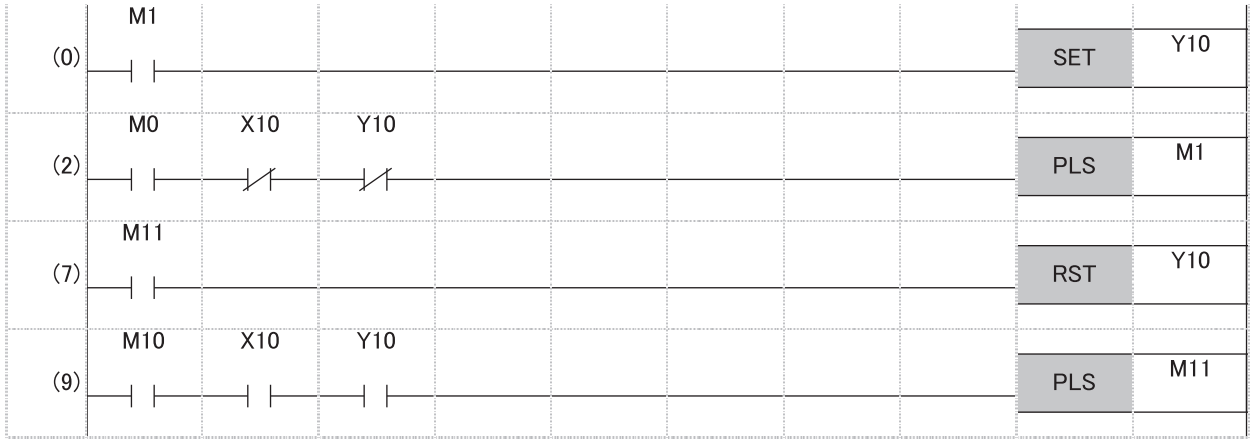
Ex.
When response input is returned to output



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]



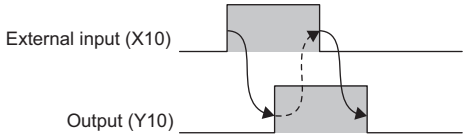
(2) PLS M1 delays SET Y0 by one scan.
(9) PLS M11 delays RST Y10 by one scan.

Point

In the CPU parameter, select "Transfer" (default setting) in "Signal Flow Memory Tracking Setting" of "Redundant System Settings". (☞ Page 232 Tracking transfer setting for the signal flow memory)

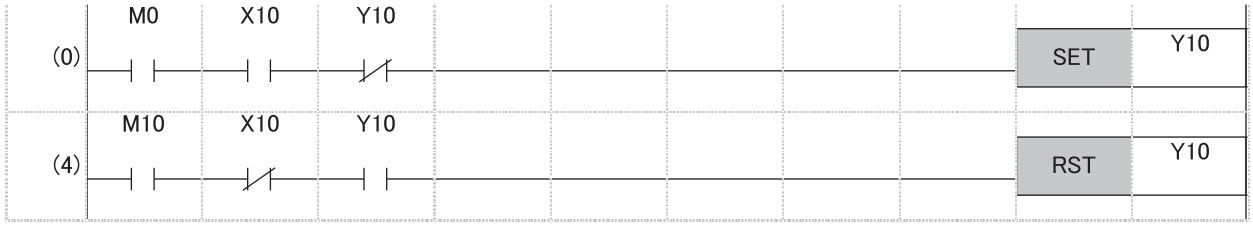
Ex.

When output is returned to external input

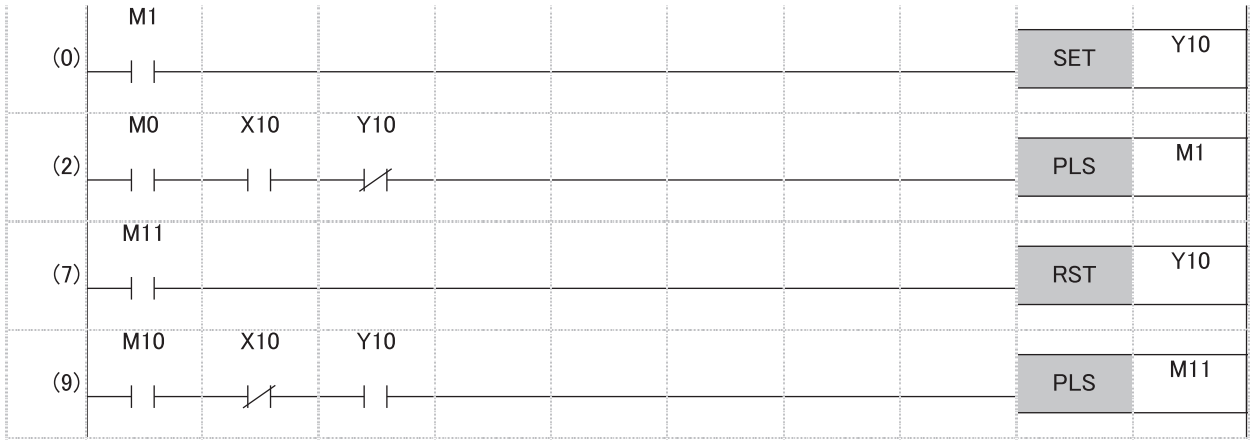


In the following program, turning on M0 turns on the output (Y10) and turning on M10 turns off the output (Y10).

[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 232 Tracking transfer setting for the signal flow memory)

25.6 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.

25.7 Precautions on outputting in the middle of the scan

When values are output during execution of a program with the following devices or setting, the program will be executed again from the step 0 in the new control system after system switching. Therefore, values may be output twice, before and after system switching. Output results may differ between the two outputs.

- Link direct device (Jn\Y)
- Direct access output (DY)
- Refresh at execution of a specified program ("Refresh Group Setting" of "Program Setting")

Before outputting values with one of the above devices or setting, check that the above operation does not cause any problem in the system design phase. If any problem occurs, do not perform the external output until the status of the received output signal becomes stable, or take other measures with external circuits.

Point

In the program of a system using the SIL2 Process CPU, performing the output refresh by the END processing is recommended instead of outputting with the link direct device (Jn\Y), direct access output (DY), or refresh at execution of a specified program ("Refresh Group Setting") in the middle of the scan.

PART 7 DEVICES, LABELS, AND CONSTANTS

This part consists of the following chapters.

26 STANDARD DEVICES

27 SAFETY DEVICES

28 STANDARD LABEL

29 SAFETY LABEL

30 LATCH FUNCTION

31 DEVICE/LABEL INITIAL VALUE SETTINGS

32 LABEL INITIALIZATION FUNCTION

33 CONSTANTS

26 STANDARD DEVICES

This chapter describes the standard devices.

26.1 Standard Device List

The following table lists the standard devices.

Classification	Type	Device name	Symbol	Number of points of Default	Parameter-set range	Notation
User device	Bit	Input	X	12K points*1	Unchangeable	Hexadecimal
	Bit	Output	Y	12K points*1		Hexadecimal
	Bit	Internal relay	M	12K points	Changeable (☞ Page 288 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/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 maximum*2	Unchangeable	Hexadecimal
	Bit	Link output	Jn\Y	16384 points maximum*2		Hexadecimal
	Bit	Link relay	Jn\B	32768 points maximum*2		Hexadecimal
	Bit	Link special relay	Jn\SB	512 points maximum*2		Hexadecimal
	Word	Link register	Jn\W	131072 points maximum*2		Hexadecimal
	Word	Link special register	Jn\SW	512 points maximum*2		Hexadecimal
Module access device	Word	Module access device	Un\G	268435456 points maximum*2	Unchangeable	Decimal
CPU buffer memory access device	Word	CPU buffer memory access device	U3En\G	268435456 points maximum*2	Unchangeable	Decimal
Index register	Word	Index register	Z	20 points	Changeable (☞ Page 313 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	Changeable (☞ Page 322 Pointer setting)	Decimal
	—	Interrupt pointer	I	1024 points	Unchangeable	Decimal

*1 X/Y0 to FFF are used as input/output points for I/O modules and intelligent function modules.

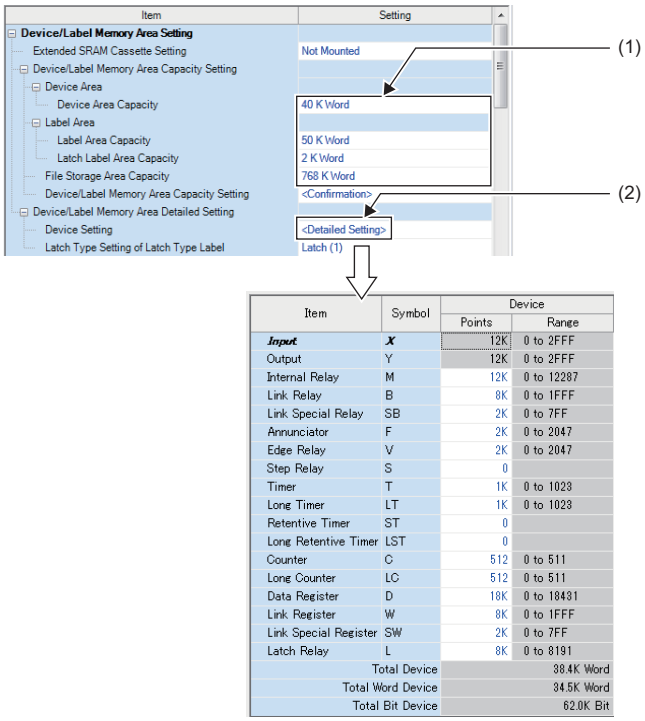
*2 These are the maximum points that can be handled in the CPU module. The number of points actually used differs depending on the module used.

26.2 Device Setting

The number of points of each user device can be changed (☞ Page 290 User Device)

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

Window



Item	Symbol	Device	
		Points	Range
Input	***X***	12K	0 to 2FFF
Output	Y	12K	0 to 2FFF
Internal Relay	M	12K	0 to 12287
Link Relay	B	8K	0 to 1FFF
Link Special Relay	SB	2K	0 to 7FF
Annunciator	F	2K	0 to 2047
Edge Relay	V	2K	0 to 2047
Step Relay	S	0	
Timer	T	1K	0 to 1023
Long Timer	LT	1K	0 to 1023
Retentive Timer	ST	0	
Long Retentive Timer	LST	0	
Counter	C	512	0 to 511
Long Counter	LC	512	0 to 511
Data Register	D	18K	0 to 18431
Link Register	W	8K	0 to 1FFF
Link Special Register	SW	2K	0 to 7FF
Latch Relay	L	8K	0 to 8191
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 148 Device/label memory area setting)
(2) The number of points of user devices can be changed.

Point

Specify each item so that the total number of points for each user device does not exceed the capacity of the device area. (☞ Page 148 Device/label memory area setting)

Range of use of device points

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

Type	Device name	Symbol	Range of use*1	Setting unit
Bit	Input	X	X0 to X2FFF	—
Bit	Output	Y	Y0 to Y2FFF	—
Bit	Internal relay	M	M0 to M94674943	64 points
Bit	Link relay	B	B0 to B5A49FFF	64 points
Bit	Annunciator	F	F0 to F32767	64 points
Bit	Link special relay	SB	SB0 to SB5A49FFF	64 points
Bit	Edge relay	V	V0 to V32767	64 points
Word	Timer	T	T0 to T5259711	32 points
Word	Retentive timer	ST	ST0 to ST5259711	32 points
Word	Long timer	LT	LT0 to LT1479295	1 point
Word	Long retentive timer	LST	LST0 to LST1479295	1 point
Word	Counter	C	C0 to C5259711	32 points
Word	Long counter	LC	LC0 to LC2784543	32 points
Word	Data register	D	D0 to D5917183	4 points
Word	Link register	W	W0 to W5A49FF	4 points
Word	Link special register	SW	SW to SW5A49FF	4 points
Bit	Latch relay	L	L0 to L32767	64 points

*1 This is the maximum number of points for the R120PSFCPU with an extended SRAM cassette (8MB) (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.

26.3 User Device

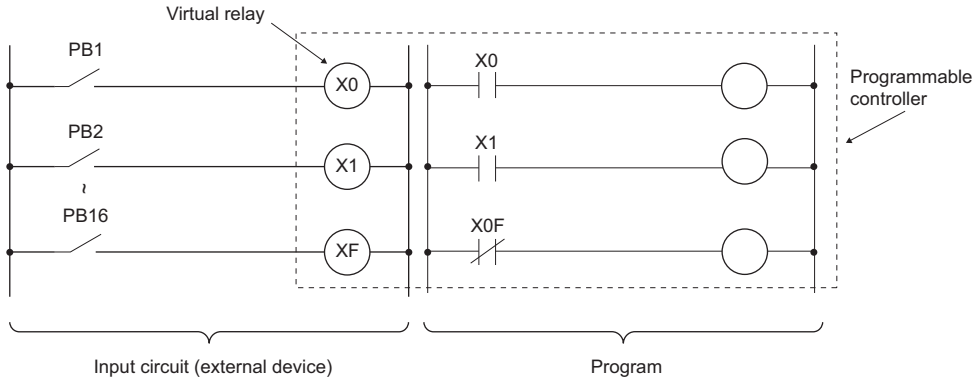
This section describes the user device.

Input (X)

This device provides the CPU module with commands and/or data using an external device, such as pushbutton, transfer switch, limit switch, and digital switch.

Concept of input

Assume that one virtual relay Xn is incorporated into the CPU module for each input point. In the program, a normally open contact and normally closed contact for the Xn are used.



Also, the input can be used as a target re-flesh (CPU module side) device of the remote input (RX), such as the CC-Link IE Field Network.

Output (Y)

This device outputs the control results of the program to various devices, such as external signal light/digital HMI/ electromagnetic switch (contactor)/solenoid.

Internal relay (M)

This device is used as an auxiliary relay within the CPU module. The following operations turn off all the internal relays.

- Powering off and on the CPU module
- Reset
- Latch clear

Latch relay (L)

This device is an auxiliary relay which enables latching (data retention during power failure). This device is used within the CPU module. This device latches operation results (ON/OFF information) even after the following operations.

- Powering off and on the CPU module
- Reset

Link relay (B)

This device is used as a CPU side device when refreshing bit data between the network module, such as the CC-Link IE Controller Network module and the CPU module.

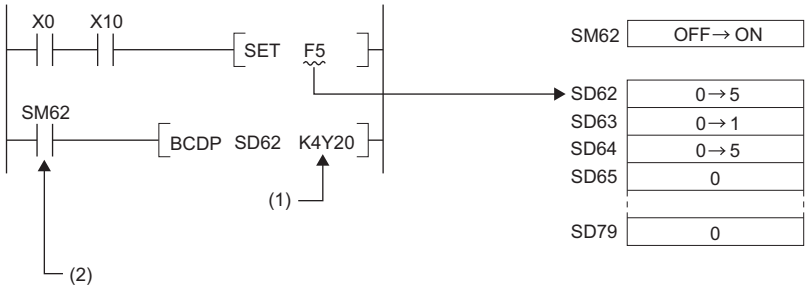
Refreshing network modules using link relay

Data are transferred/received between the link relay (B) within the CPU module and the link relay (LB) of the network module, such as the CC-Link IE Controller Network module. The refresh range is specified using parameters on the network module. The part which is not used for refreshing can be used for other applications.

Annunciator (F)

This device is an internal relay used for a customer-created program which detects malfunction/failure of the equipment. When the annunciators are turned on, SM62 (Annunciator) is turned on, the number of activated annunciators and their device numbers are stored in SD62 (Annunciator number) to SD79 (Table of detected annunciator numbers).

Ex.
Failure detection program



(1) Outputs the annunciator number of the annunciator that turned on.
(2) ON detection of the annunciator

Also the number of the annunciator that turned on first (the number stored in SD62) is registered in the event history.

Point

Only one annunciator number is registered in the event history while power is turned on.

On/off method for annunciator

Annunciators are turned on by either the SET F☐ instruction or the OUT F☐ instruction. Annunciators are turned off by the RST F☐ instruction or the LEDR instruction or the BKRST instruction.

Point

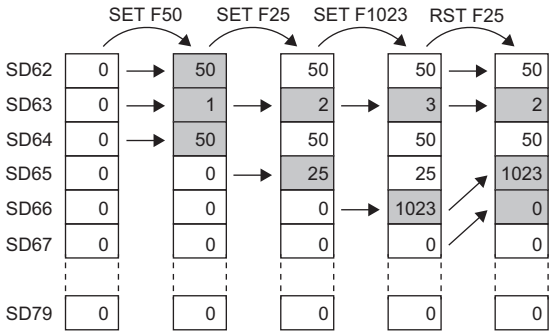
When the annunciators are turned on/off using any methods (e.g. the MOV instruction) other than shown above, the operation is the same as that of internal relays. As a result, SM62 is not turned on and annunciator numbers are not stored into SD62 and SD64 (Table of detected annunciator numbers) to SD79.

■Operations when annunciators are turned on

1. The annunciator numbers that turned on are stored sequentially into SD64 to SD79.
2. The annunciator number stored into SD64 is stored into SD62.
3. SD63 value (Number of annunciators) is incremented by one.

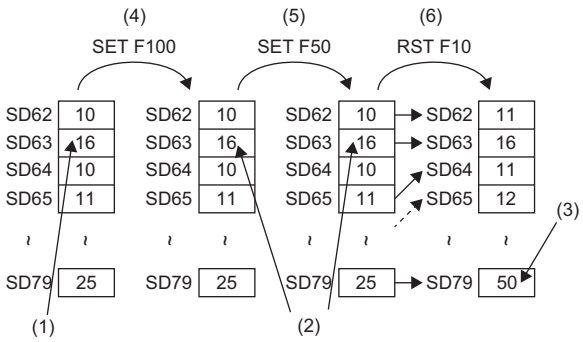
■Operations when annunciators are turned off

1. The number of the annunciator deactivated is removed, and the numbers of annunciators, which were lined up behind the removed one, move forward one by one.
2. When the annunciator number stored into SD64 is turned off, the annunciator number newly stored into SD64 is stored into SD62.
3. The value of SD63 is decremented by one. When the SD63 value gets decremented to 0, SM62 is turned off.



Point

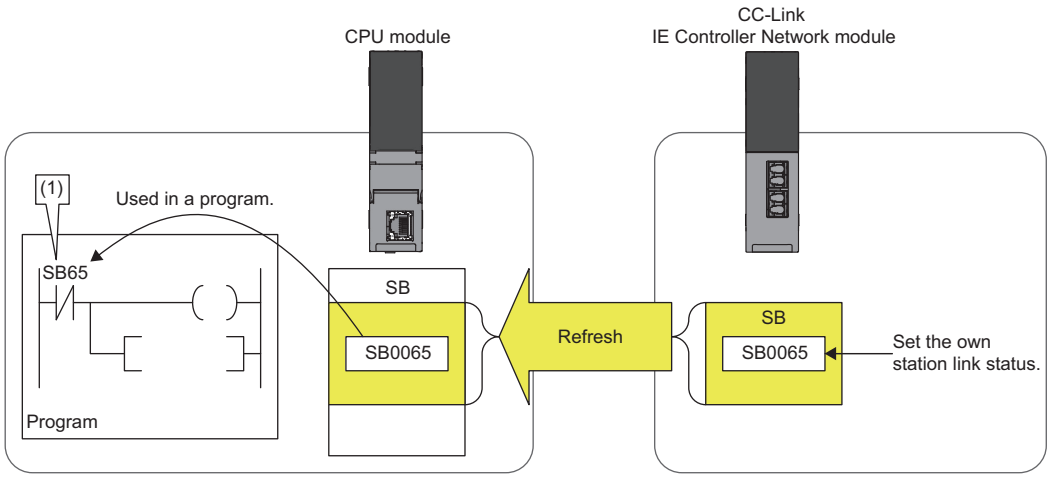
If more than 16 annunciators are turned on, the 17th annunciator onwards are not stored into SD64 to SD79. However, if the numbers of annunciators registered in SD64 to SD79 are turned off, the lowest numbers, which are not registered in SD62 to SD79, of the numbers of annunciators which were turned on for the 17th on and after, are stored into SD64 to SD79.



(1) Annunciators have been stored for the maximum number (16).
(2) Because annunciators have been stored for the maximum number, the value does not change.
(3) The smallest number is stored.
(4) Turn on the 17th device.
(5) Turn on the 18th device.
(6) Turn off the first device.

Link special relay (SB)

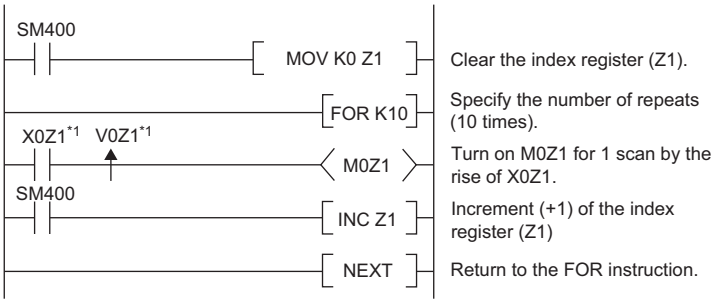
The communication status and error detection status of network modules, such as the CC-Link IE Controller Network module, are output to the link special relay (J□\SB□) on the network. The link special relay (SB) is a device for using as a refreshing target of link special relays in the network. The part which is not used for refreshing can be used for other applications.



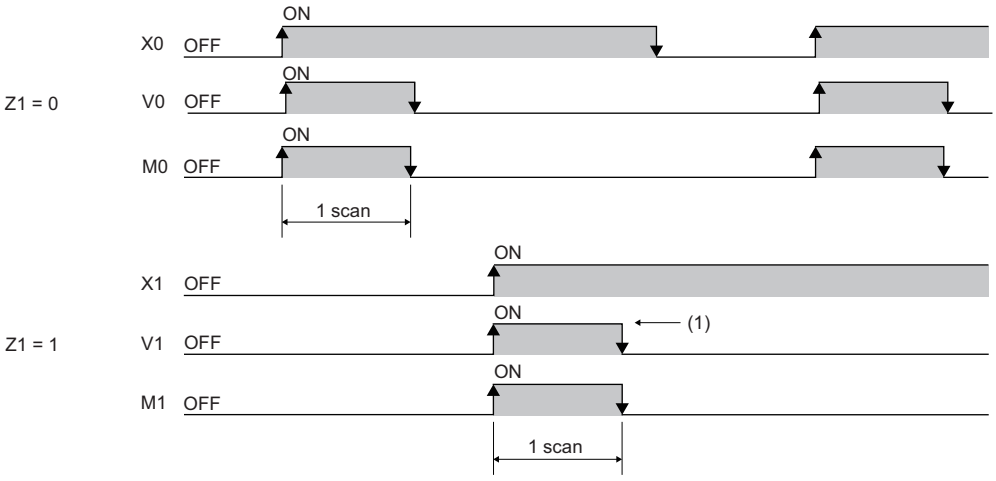
(1) The network status is checked.

Edge relay (V)

The edge relay is a device that memorizes operation results (on/off information) from the head of the ladder block, allowing its use only by the EGP/EGF instruction. This device is executed for various objectives such as the rising (from off to on) detection in the structured programs by the index modification.



*1 Edge relay V0Z1 memorizes on/off information of X0Z1.



(1) When X1 is rising, this device is turned on for one scan duration.

Timer

This device starts measurement when the timer coil is turned on. When the current value reaches a setting value, time is up and the contact is turned on. This timer is an up-timing type device and therefore the current value matches a setting value when the timer time is up.

Types of timers

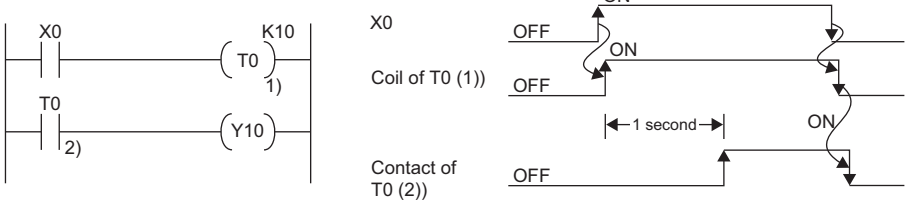
There are two types of timers: timer (T) which retains the current value in 16-bit units and long timer (LT) which retains it in 32-bit units. The timer (T) and the long timer (LT) are different devices and the number of device points can be set for each of them. In addition, there are the retentive timer (ST) and the long retentive timer (LST), both of which retain the current value even if the coil is turned off. *1

Types of timers			
Timer	Current value = 16 bits	Timer (T)	Low-speed timer
			High-speed timer
		Retentive timer (ST)	Low-speed retentive timer
			High-speed retentive timer
	Current value = 32 bits	Long timer (LT)	
		Long retentive timer (LST)	

*1 For the timer (T)/long timer (LT), the current value returns to 0 when the coil is turned off.

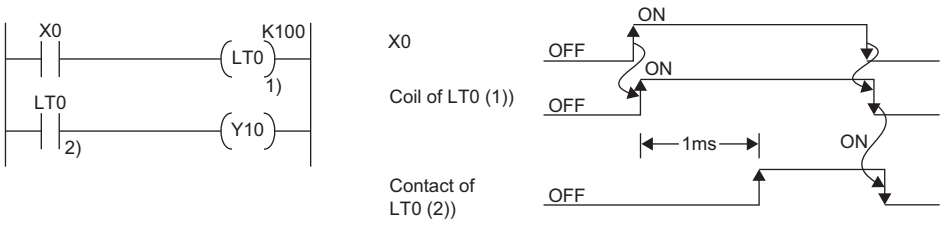
■Timer (T)

This device starts measurement when the coil of the timer is turned on. When the timer current value matches a setting value, time is up and the timer contact is turned on. When the timer coil is turned off, the current value returns to 0 and the timer contact is turned off.



■Long timer (LT)

This device can count from 0 to 4294967295 to measure the time. If the measuring unit is set to 0.01ms, the measurable time range of this device is from 0 to about 11.9 hours. The current value of the long timers is updated by adding a difference of the counter which is used in the system when the OUT LT☐ instruction is executed. Proper time can be measured even when the coil instruction of the long timer cannot be executed once per scanning because the counters used in the system count asynchronously with scanning. When the long timer coil is turned on, measurement starts and when the long timer current value matches a setting value, time is up and the long timer contact is turned on. When the long timer coil is turned off, the current value returns to 0 and the long timer contact is turned off.



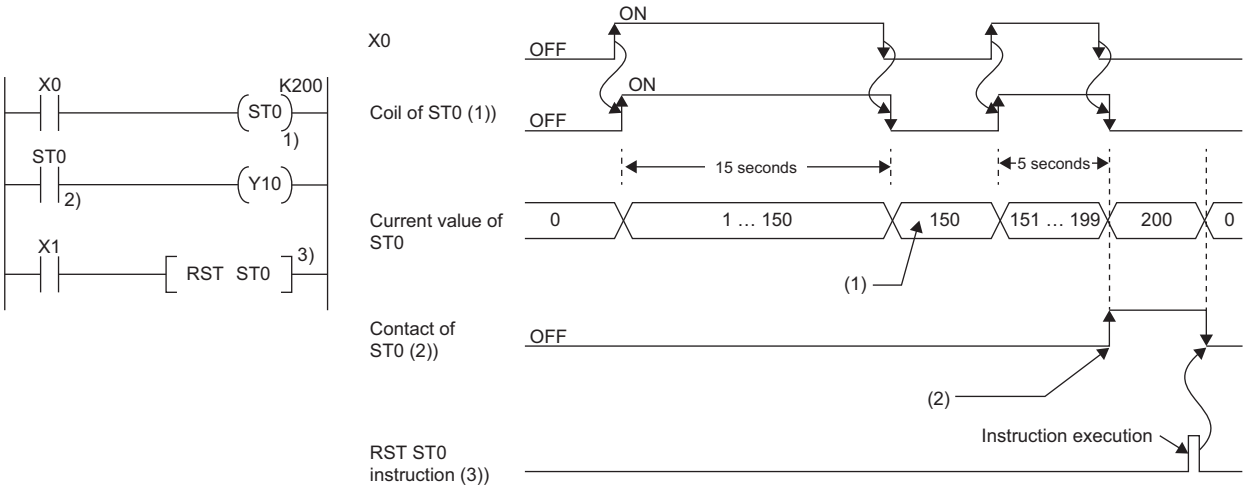
*1 This figure shows the example when the long timer time limit value setting is 0.01ms.

Point

- The long timer contact is turned on in the next or subsequent coil execution after the long timer coil is turned on. The long timer contact is not turned on simultaneously when the long timer coil is turned on.
- The long timer (LT) can be used in interrupt programs. (☞ Page 123 Interrupt Program)

■Retentive timer (ST)

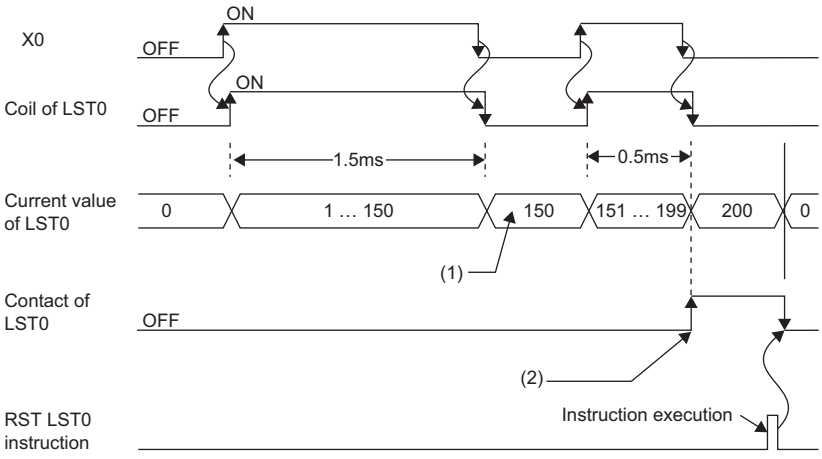
This device counts the sum of time duration in which the coil is turned on. When the retentive timer coil is turned on, measurement starts and when the timer current value matches a setting value (when time is up), the retentive timer contact is turned on. The current value and the contact on/off state 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 123 Interrupt Program)

■Low-speed/high-speed timer (T/ST)

The low-speed timer and high-speed timer are the same device which is set to a low speed or high speed timer by writing the instruction accordingly to specify it on the timer. For example, specifying OUT T0 generates a low-speed timer and specifying OUTH T0 produces a high-speed timer even when using the same T0 device. This also applies to the retentive timer.

Timer time limit value

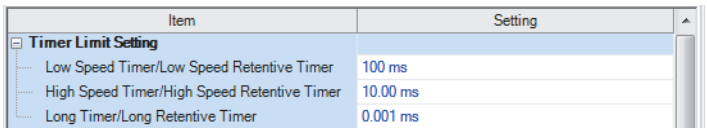
Although the low-speed timer and high-speed timer are the same device, timer limit value are different depending on how to specify the timer device (how to write the instruction). For example, specifying OUT T0 generates a low-speed timer and specifying OUT H T0 produces a high-speed timer even when using the same T0 device. This also applies to the retentive timer. The long timer cannot be set to a low-speed or high speed device. The time limit value for each timer is set in "Timer Limit Setting".

■Timer limit setting

The following window is to set the timer time limit values.

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Timer Limit Setting]

Window



Displayed items

Item	Description	Setting range	Default
Low Speed Timer/Low Speed Retentive Timer	Set the timer time limit value of T and ST used for the low-speed timer and low-speed retentive timer.	1 to 1000ms (unit: 1ms)	100ms
High Speed Timer/High Speed Retentive Timer	Set the timer time limit value of T and ST used for the high-speed timer and high-speed retentive timer.	0.01 to 100ms (unit: 0.01ms)	10.00ms
Long timer/Long retentive timer	Set the timer time limit value of LT and LST used for the long timer and long retentive timer.	0.001 to 1000ms (unit: 0.001ms)	0.001ms

Timer current value and the measurable range

This sections describes the timer current value and the measurable range.

■Timer (T/ST)

The current value range is 0 to 32767. The measurable time range is from 0 to (timer time limit value × 32767).

■Long timer (LT/LST)

The setting range of the current value is 0 to 4294967295, which is the same as the range of unsigned 32-bit integers. The measurable time range is from long timer time limit value to (timer time limit value × 4294967295).

Handling timers

When executing the timer coil (the OUT T☐ instruction), the timer coil is turned on/off, the current value is updated, and the contact is turned on/off.

Accuracy of timers

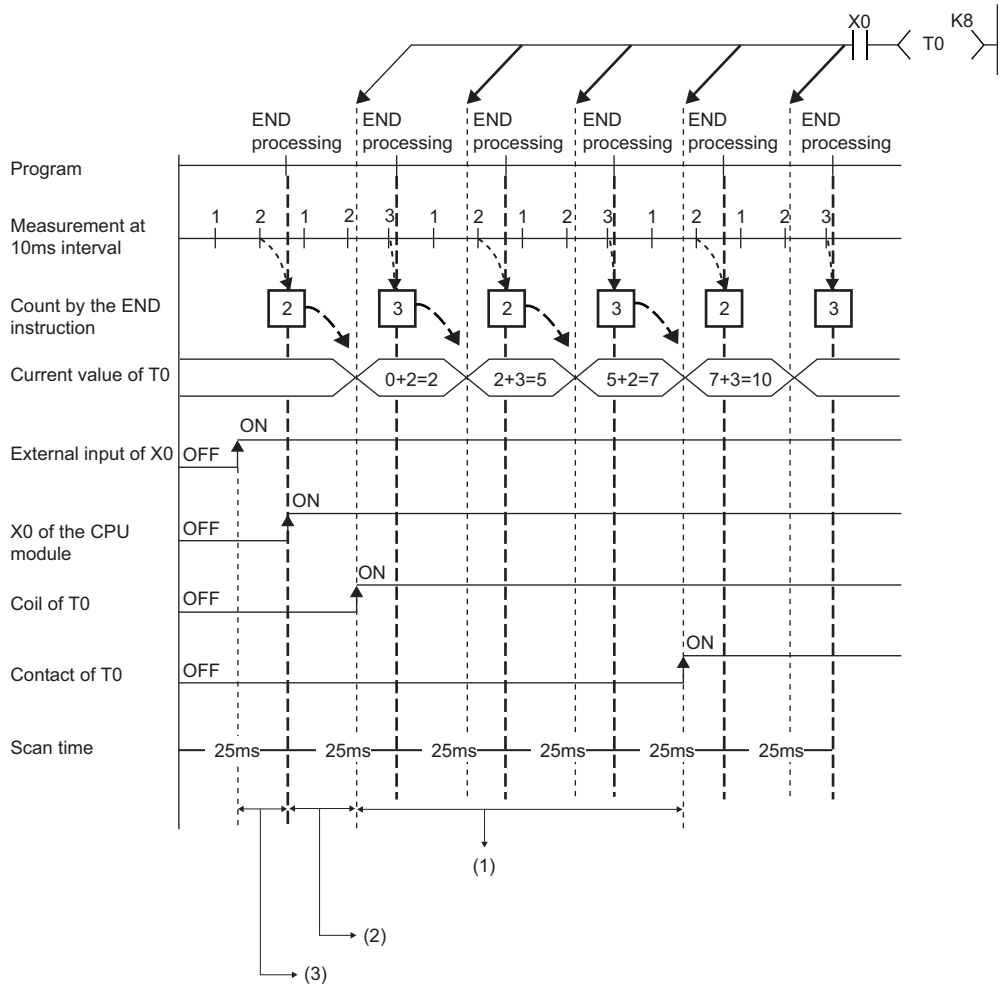
This sections describes the accuracy of timers.

■Timer (T/ST)

The scan time value measured by the END instruction is added to the current value when the OUT T☐ instruction is executed. If the timer coil is turned off when the OUT T☐ instruction is executed, the current value is not updated. The maximum response accuracy of the timer (the time duration from capture of an input (X) to output of it) is "2 scan time + timer time limit setting".

Ex.

Timer limit setting = 10ms, setting value of T0 = 8 (10ms × 8 = 80ms), scan time = 25ms



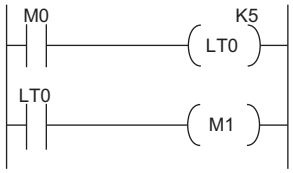
(1) Accuracy from when the coil of the timer turns on until when the contact of the timer turns on
- (1 scan time + timer limit setting) to (1 scan time)

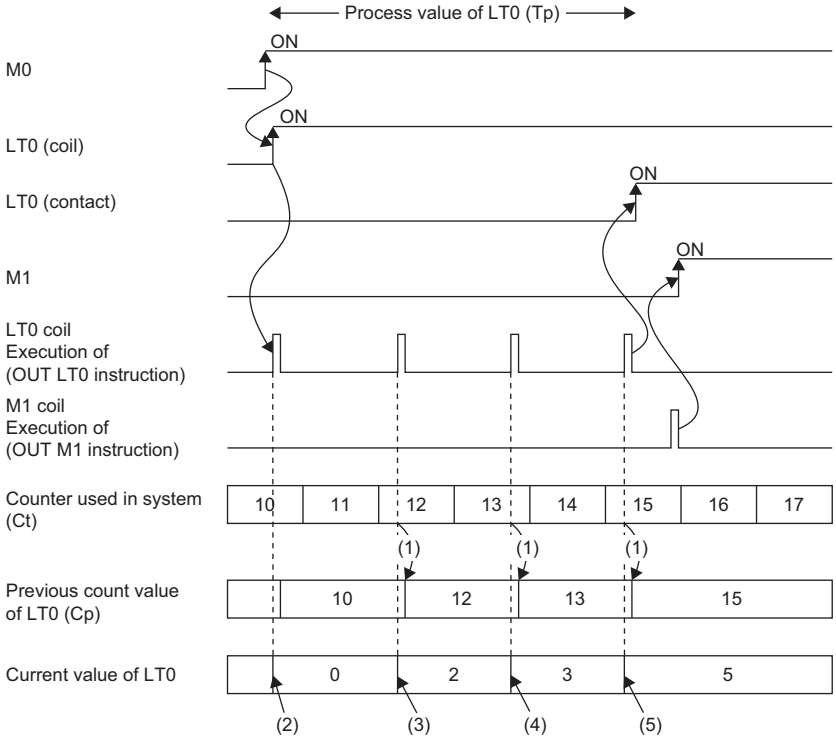
(2)Timing when the coil of the timer turns on

(3)Input fetching timing

■Long timer (LT/LST)

In the following program, the accuracy of Tp (the time duration from the long timer coil activation to long timer contact activation) is (Ts-Tu) ≤ Tp < (Ts + Tu).

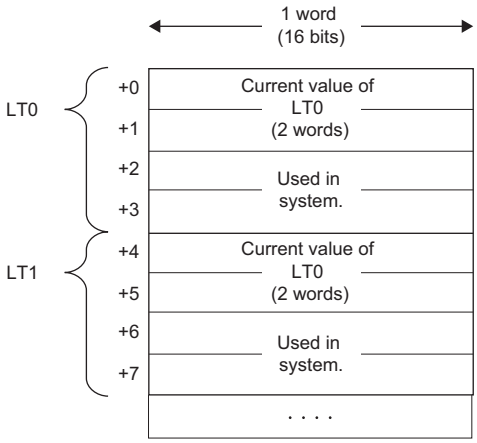




(1) Cp is updated.
(2) LT0 is initialized to 0.
(3) The result of (Ct - Cp) is added.: 0 + (12 - 10) = 2
(4) The result of (Ct - Cp) is added.: 2 + (13 - 12) = 3
(5) The result of (Ct - Cp) is added.: 3 + (15 - 13) = 5
Tp: time duration from the long timer coil activation to long timer contact activation
Ts: setting value of the long timer
Tu: time limit value of the long timer

Data configuration of long timer (LT/LST)

The long timer (LT) and the long retentive timer (LST) use four words (64 bits) for each point. If the most significant two words are changed in a program, it is impossible to measure the time properly, because they are used by the system.



Point

The current value of the long timer (LT) and the long retentive timer (LST) is 32-bit data. It can be specified by the instructions which can specify signed or unsigned 32 bit data. (It cannot be specified with the BK + Instruction.)

Precautions

This section describes the precautions when using the timer and long timer.

■Precautions about timer usage

- Do not describe more than one coil (the OUT T☐ instruction) on the same timer during a single scanning. Doing so results in improper measurement because the timer current value is updated when the coil for each timer is executed.
- When timer is not used for data collection for each scan: While the coil of a timer (e.g. T1) is turned on, the timer coil (the OUT T☐ instruction) cannot be skipped by the instructions such as the CJ. When the timer coil is skipped, proper measurement is impossible because the timer current value is not updated. In addition, when the timer exists in a subroutine program, be sure to execute a subroutine call including T1 coil only once for each scanning operation while the coil of the timer (e.g. T1) is turned on. Otherwise proper measurement is impossible.
- The timer cannot be used in the initial execution type program, the fixed scan execution type program, or the event execution type program where the occurrence of an interrupt is set to be a trigger. The timer can be used in standby type programs if the coil of timer (OUT T☐ instruction) is executed one time for one scan using a subroutine program.
- The timer cannot be used in interrupt programs. The timer can be used in subroutine programs or FB programs if the coil of timer (OUT T☐ instruction) is executed one time for one scan.
- When setting value is 0: The contact is turned on when the OUT T☐ instruction is executed.
- Even when the setting value is increased after the timer time is up, the timer status does not change (time continues to be up) and the timer does not operate.
- Do not set the timer setting value to 32768 or above. If used when set to 32768 or above, the timer contact may not turn on.

■Precautions about long timer usage

This section describes the precautions when using long timers (LT/LST).

- The long timer cannot be used in initial execution type programs.
- Even when the setting value is increased after the long timer time is up, the long timer status does not change (time continues to be up) and the long timer does not operate.

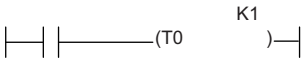
■Timer setting value and timer limit setting

When the condition is "Timer setting value < Scan time + Timer limit setting", the coil and the contact may be turned on at the same time depending on the timing when the coil turns on. When the condition is not satisfied, reduce the timer limit setting value to satisfy the condition.

Ex.

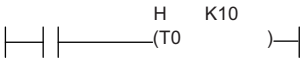
When changing the low-speed timer to high-speed timer and reducing the timer limit setting value (scan time: 20ms)

Before change (low-speed timer)



■Timer limit setting
Low Speed Timer/Low Speed Retentive Timer: 100ms
Timer setting value (100ms × 1 = 100ms) < Scan time (20ms) + Timer limit setting (100ms)

After change (high-speed timer)

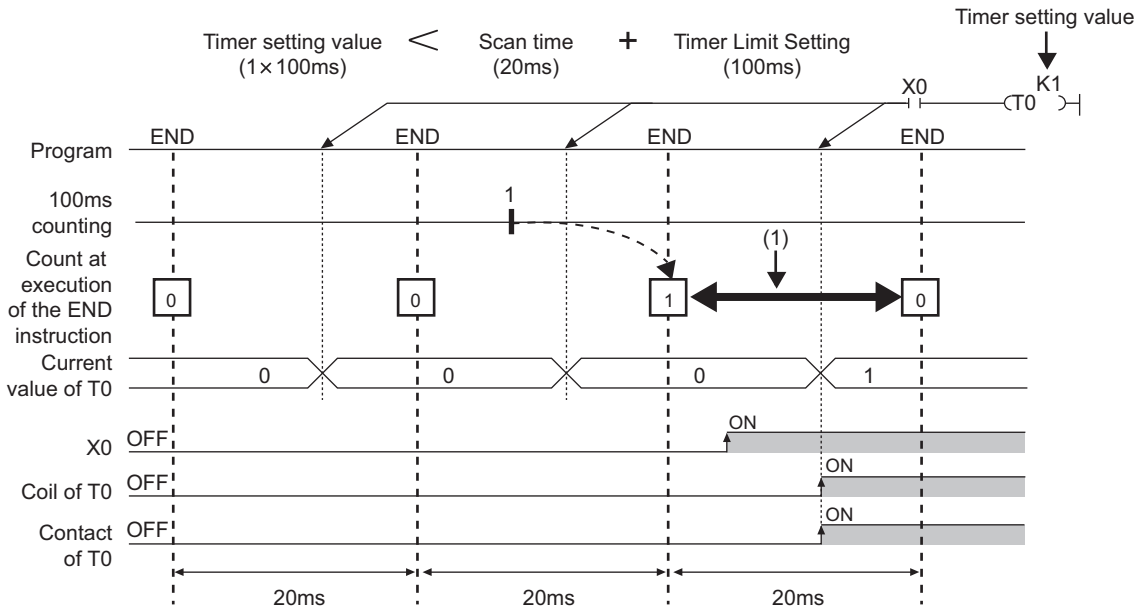


■Timer limit setting
High Speed Timer/High Speed Retentive Timer: 10.00ms
Timer setting value (10.00ms × 10 = 100ms) < Scan time (20ms) + Timer limit setting (10ms)

The following shows an example of when the coil and the contact are turned on at the same time when the condition is "Timer setting value < Scan time + Timer limit setting".

Ex.

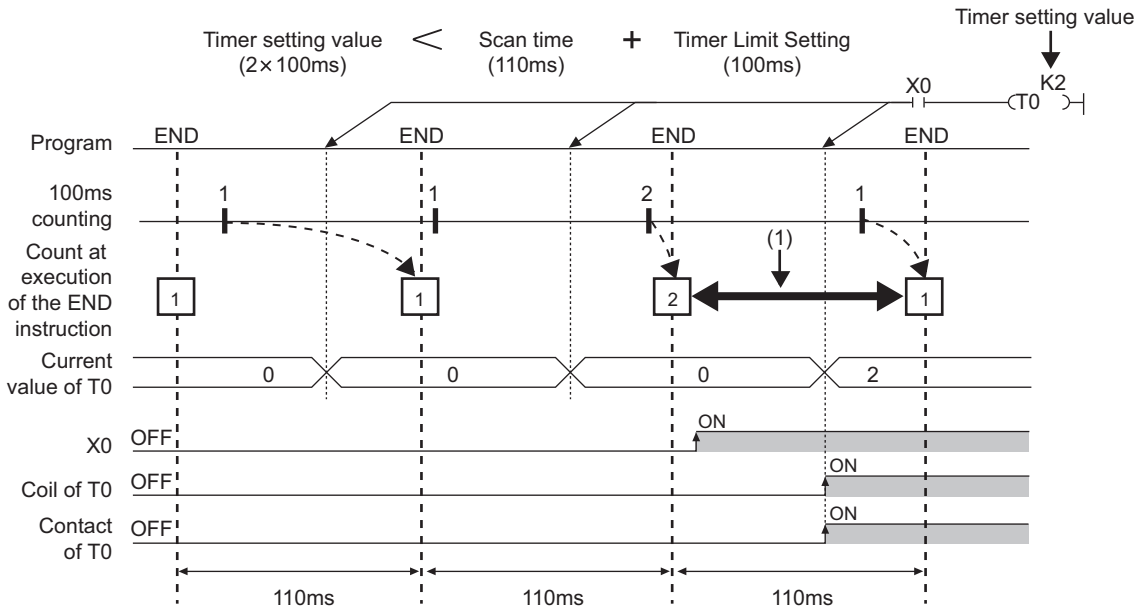
In the case where the values are set as follows (timer setting value: 1 (1 × 100ms), scan time: 20ms, and timer limit setting: 100ms), when the coil of the timer (T0) turns on in the next scan after the coefficient of the END instruction becomes equal to or higher than the timer setting value, the coil and the contact turn on at the same time because the timer current value becomes equal to the timer setting value at startup of the timer.



(1) The coefficient of the END instruction becomes equal to or higher than the timer setting value. Therefore, when the coil turns on during this period, the contact also turns on at the same time.

Ex.

In the case where the values are set as follows (timer setting value: 2 (2 × 100ms), scan time: 110ms, and timer limit setting: 100ms), when the coil of the timer (T0) turns on in the next scan after the coefficient of the END instruction becomes equal to or higher than the timer setting value, the coil and the contact turn on at the same time because the timer current value becomes equal to the timer setting value at startup of the timer.



(1) The coefficient of the END instruction becomes equal to or higher than the timer setting value. Therefore, when the coil turns on during this period, the contact also turns on at the same time.

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 123 Interrupt Program)

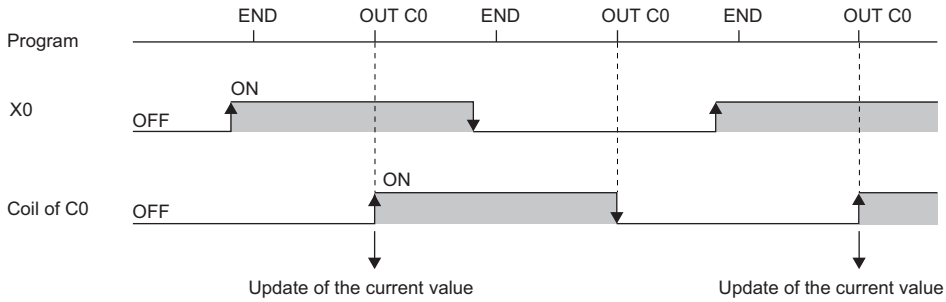
Counting process

When executing the counter coil (OUT C☐ instruction / OUT LC☐ instruction), the counter coil is turned on/off, the current value is updated (count value +1), and the contact is turned on/off. The current value is updated (count value +1), when the counter coil input is rising (from off to on). The current value is not updated when the coil input is off, remains on, and is turned off.

[Ladder example]



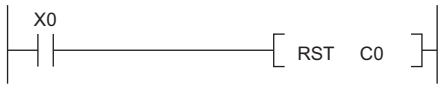
[Update timing of the current value]



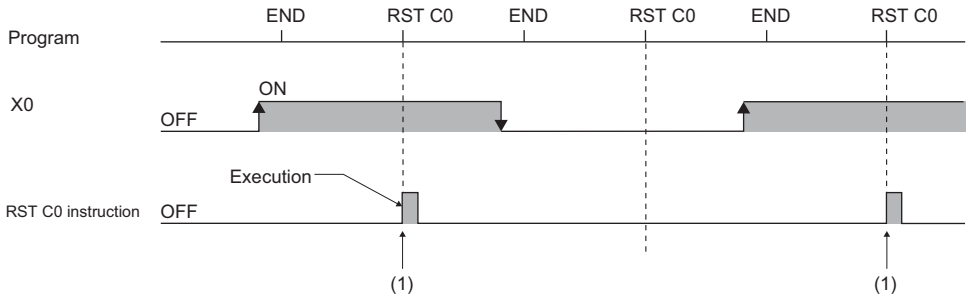
Resetting counters

The counter current value is not cleared even when the counter coil input is turned off. To clear the counter current value (resetting) and turn off the contact, issue the RST C☐/RST LC☐ instruction. When executing the RST C☐ instruction, the counter value is cleared and the contact is turned off.

[Ladder example]



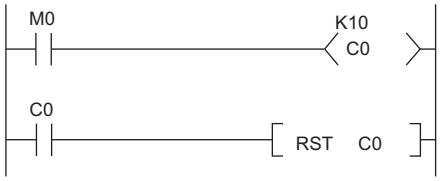
[Reset timing of the counter]



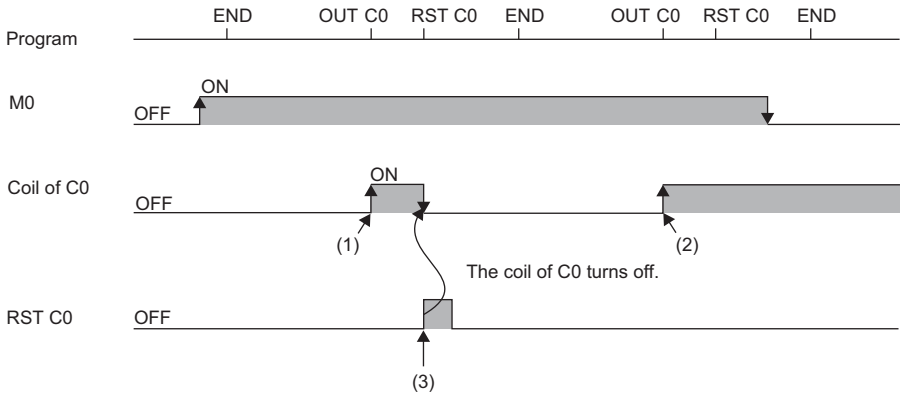
(1) The count value is cleared, and the contact turns off.

■Precautions about counter reset

When executing the RST C☐ instruction, the coil for C☐ is also turned off. If the execution condition for the OUT C☐ instruction is turned on after the RST C☐ instruction is executed, the coil of C☐ is turned on and the current value is updated (count value +1) when the OUT C☐ instruction is executed.

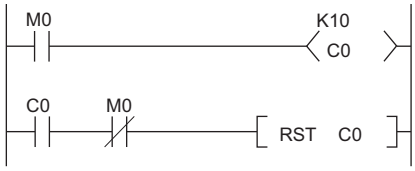


In the above ladder example, the coil of C0 is turned on by turning on M0 and as a result the current value is updated. When C0 count reaches its upper limit, C0 contact is turned on and C0 current value is cleared by execution of the RST C0 instruction. At this time C0 coil is also turned off. When M0 is turned on at the next scanning, the current value is updated because C0 coil is turned on during the OUT C0 instruction execution (the current value is changed to 1).



(1) The current value update contact turns on.
(2) The current value is updated because the coil of C0 turns on.
(3) The count value is cleared, and the contact turns off.

To address the above problem, insert a normally closed contact of the execution condition for the OUT C0 instruction into the execution condition for the RST C0 instruction to prevent C0 coil from being turned off while the execution condition (M0) of the OUT C0 instruction is turned on as shown in the following ladder example.

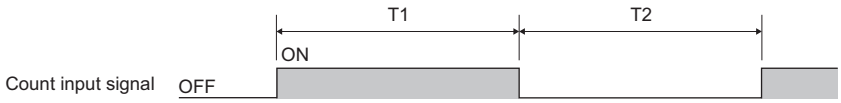


Maximum counting speed for counters

Counting is possible only when on/off time of the input condition is longer than the execution interval of the same OUT C☐ instruction. Maximum counting speed for counters can be obtained by the following equation:

$$\text{Maximum counting speed Cmax} = \frac{n}{100} \times \frac{1}{T} \text{ [times/s]}$$

*1 Duty (n) is a value which expresses the ratio of on/off time of the count input signal as a percent (%) value.



- When $T1 \geq T2$, $n = \frac{T2}{T1+T2} \times 100\%$
- When $T1 < T2$, $n = \frac{T1}{T1+T2} \times 100\%$

Data register (D)

This device can store numerical values.

Link register (W)

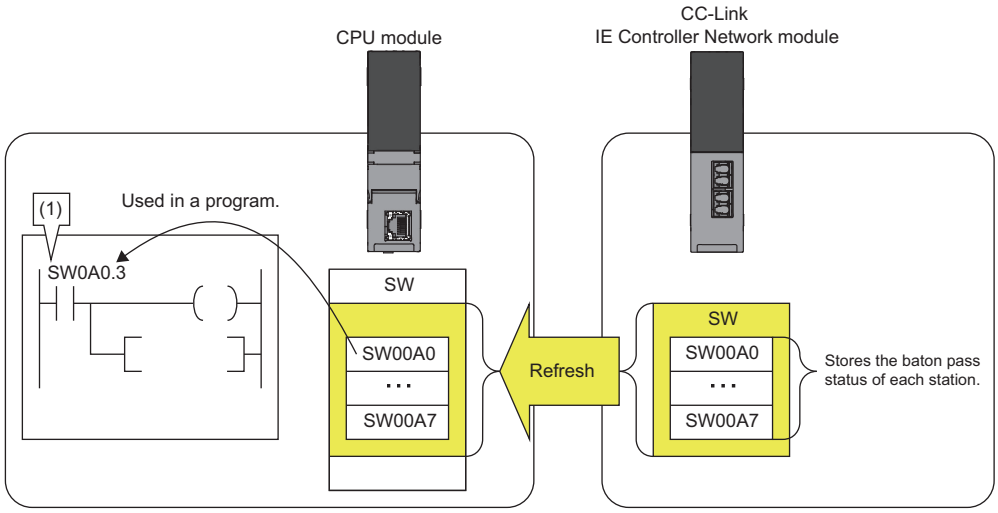
This device is used as a CPU module side device when refreshing word data between the network module, such as the CC-Link IE Controller Network module and the CPU module.

Refreshing network modules using link register

Data are transferred/received between the link register (W) within the CPU module and the link register (LW) of the network module, such as the CC-Link IE Controller Network module. The refresh range is specified using parameters on the network module. The part which is not used for refreshing can be used for other applications.

Link special register (SW)

Word data information on the communication status and error detection status of a network, such as CC-Link IE Controller Network, are output into the link special register (J□\SW□) on the network. The link special register (SW) is a device for using as a refreshing target of link special registers in the network. The part which is not used for refreshing can be used for other applications.



(1) The network status is checked.

26.4 System Device

The system device is used by the system. Assignment/capacity is fixed and cannot optionally be altered.

Function device (FX/FY/FD)

This device is used for the subroutine programs with argument passing. Data is written/read between the subroutine call sources with argument passing and the subroutine programs with argument passing. When using the function device in a subroutine program, the device used in each subroutine program call source can be determined. As a result, when using the same subroutine program, it can be used without being aware of the call source of other subroutine programs.

Function input (FX)

This device is used when passing on/off data to a subroutine program. In the subroutine program, bit data specified in a subroutine call instruction with argument passing are captured and used for operation. All the bit data specification devices of the CPU module are available.

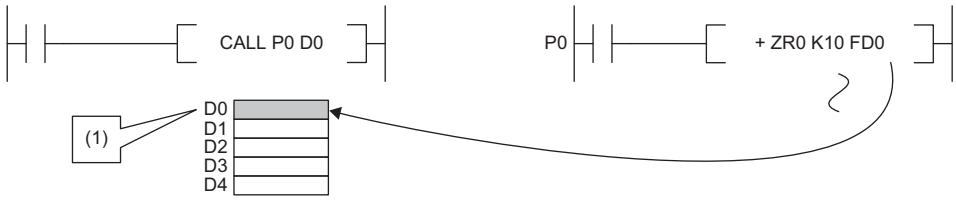
Function output (FY)

This device is used when passing operation results (on/off data) in a subroutine program to a subroutine program call source. Operation results are stored into the device specified in the subroutine program with argument passing. The bit data specification device other than the CPU module input (X) are available.

Function register (FD)

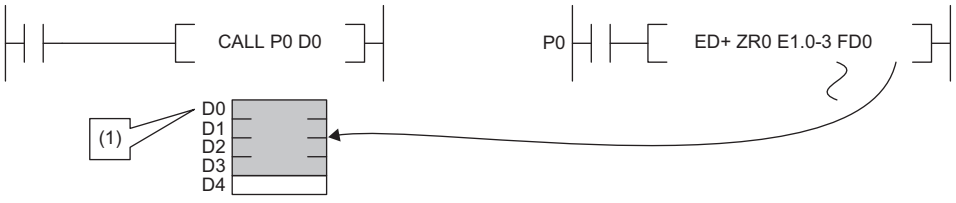
This device is used for writing/reading between the subroutine call source and the subroutine program. The input/output condition of the function register is automatically identified by the CPU module. "Source data" in a subroutine program means data inputted into the subroutine program. "Destination data" in a subroutine program means data outputted from the subroutine program. One point of a function register occupies maximum of four words and can store 16-bit data, 32-bit data, 64-bit data, single-precision real number, and double-precision real number. However the number of words to be used depends on the instruction in the subroutine program.

For example, for the destination of addition instruction (+instruction) of 16-bit signed integer, one word is used.



(1) The data is stored in one point of D0.

Besides, for the destination of addition instruction (ED+instruction) of double-precision real number, four words are used.



(1) The data is stored in four points of D0 to D3.

Special relay (SM)

This is the internal relay for which the specification is defined in the CPU module, where the status of the CPU module is stored. (☞ Page 479 List of Special Relay Areas)

Special register (SD)

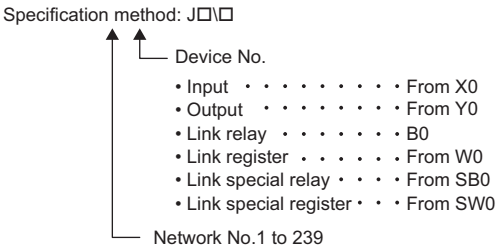
This is the internal register for which the specification is defined in the CPU module, where the status (diagnostics information, system information, etc) of the CPU module is stored. (☞ Page 493 List of Special Register Areas)

26.5 Link Direct Device

This device directly accesses link relays and/or link registers of the network module in the CC-Link IE Controller Network and/or CC-Link IE Field Network.

Specification method

Specify a link direct device as shown below: The link register 10 (W10) of the network number 2 can be specified as "J2\W10".



Point

For bit devices, digit specification is allowed. (Example: J1\K1X0, J10\K4B0)

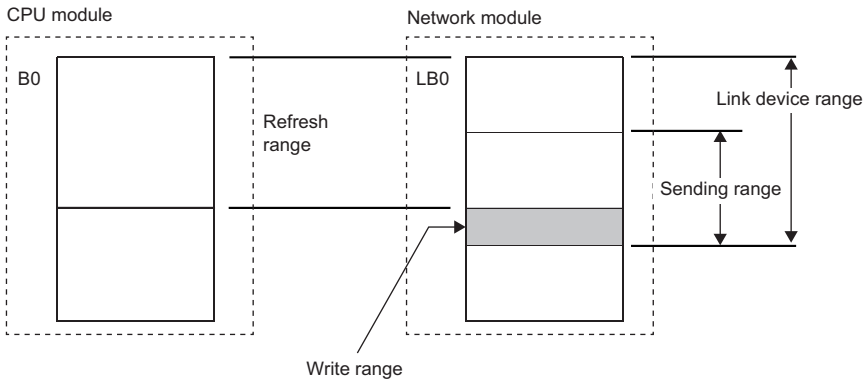
Specification range

All the link devices of the network module can be specified. The link devices which fall outside the range specified with "Refresh Setting" can also be specified.

Specification range for writing

Writing should be done in the range which is within the link device range specified as a send range of network parameters, and outside the range specified as the refresh range for "Refresh Setting". *1

Note that writing in the range specified as the refresh range overwrites link device data of the network module during refresh process. When writing data to a writing range of another station using the link direct device, data is overwritten with received data during data reception from another station.



*1 There is only one network module to which the link direct device can write data for each network number. When more than one network module with the same network number is mounted, the network module with the lowest slot number is the target for writing by the link direct device.

Specification range for reading

Data can be read from all the range of link devices of the network module. *1

*1 There is only one network module for which reading is allowed with the link direct device for each network number. When more than one network module with the same network number is mounted, the network module with the lowest slot number is the target for reading by the link direct device.

Difference from link refresh

The following table shows the difference between the link direct device and link refresh.

Item		Link direct device	Link refresh
Description method in program	Input	Jn\K4X0...	X0...
	Output	Jn\K4Y0...	Y0...
	Link relay	Jn\K4B0...	B0...
	Link register	Jn\W0...	W0...
	Link special relay	Jn\K4SB0...	SB0... Or module label
	Link special register	Jn\SW0...	SW0... Or module label
Access range in relation to network module		All the link devices for each network module	The range specified in "Refresh Setting"
Guarantee range of access data		In word (16-bit) units	In word (16-bit) units

26.6 Module Access Device

This device directly accesses from the CPU module to the buffer memory of the intelligent function module mounted on the main base unit.

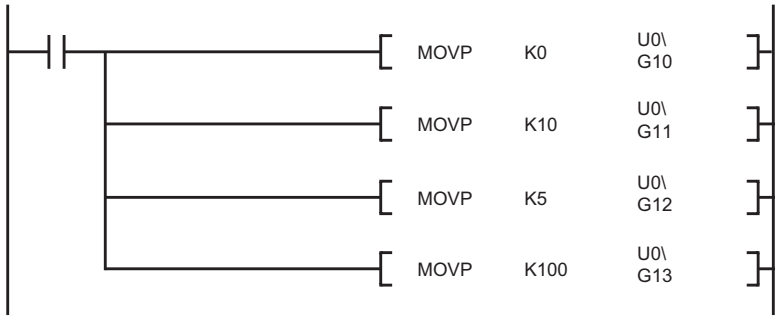
Specify this device with 'Un\Gn'. (Example: U5\G11)

Specified item		Value to be specified
Un	Start I/O number of intelligent function modules	Upper two digits when a start I/O number is described in three digits (00H to FFH) Example: 1F stands for X/Y1F0
Gn	Buffer memory address	0 to 268435455 (decimal)

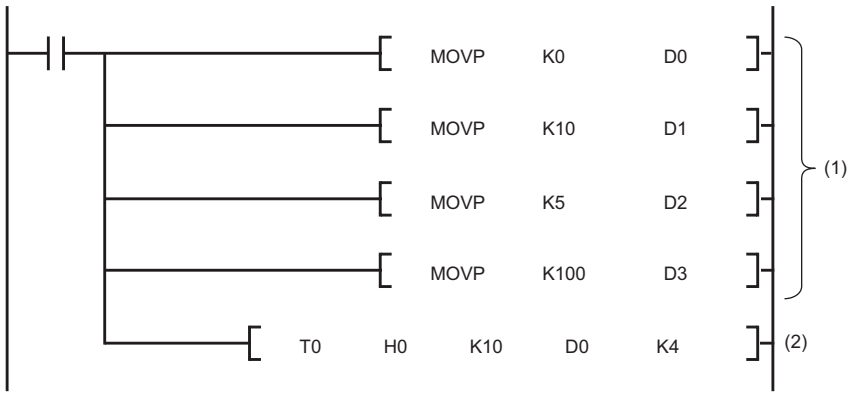
Point

When reading/writing the buffer memory data using the module access device more than twice within a program, the processing speed can be increased by conducting read/write operations at a single point of the program using the FROM/TO instruction.

• When data is written using the module access device more than twice:



• When data is written at a single point of the program using the TO instruction:



(1) Store data to devices such as the data register (D).

(2) Write data to only one point in the program.

Precautions

The following describes the precautions for when the module access device is used.

• If data is written to the refresh-target memory using a program while the refresh function is being used, the CPU module overwrites the data in the target memory at the execution of the refresh function. Thus, the expected operation may not be acquired. When the refresh function is used, do not write the data directly to the refresh-target memory but write it to the refresh-source memory.

*1 When the data is transferred from the CPU module to the module, the target memory is the buffer memory or link device. When the data is transferred from the module to the CPU module, the target memory is the specified device of the CPU module.

26.7 CPU Buffer Memory Access Device

This device accesses memory used by built-in functions of the CPU module, such as the Ethernet function.

Specification method

Specify this device with 'Un\Gn'. (Example: U1\G4095)

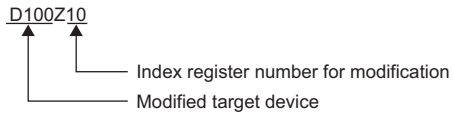
Specified item		Value to be specified
G (CPU buffer memory area)	CPU buffer memory	G
	Fixed scan communication area	HG
n (CPU buffer memory address)		0 to 268435455 (decimal)

26.8 Index Register (Z/LZ)

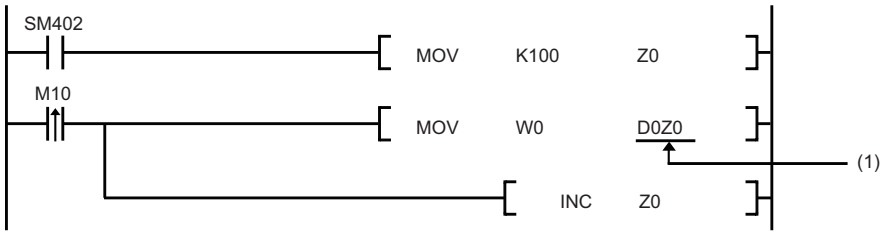
This device is used for the index modification of the device. The index modification is the indirect specification using the index register.
Specify the device with the number obtained from "Device number of device targeted for modification" + "Contents of index register".

16-bit index modification

The device number is modified using the index register (Z). The modification range for the device in the case of the 16-bit index modification is -32768 to 32767.



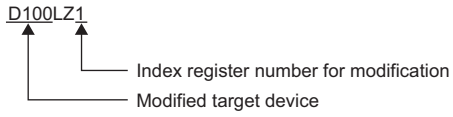
Ex.
Modifying D0 with Z0



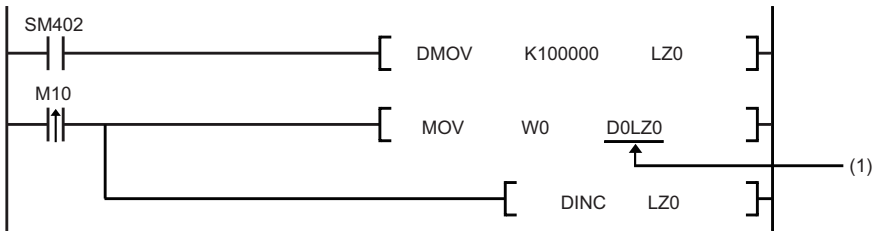
(1) Access D0Z0 = D100.

32-bit index modification

The device number is modified using the long index register (LZ). The modification range for the device in the case of the 32-bit index modification is -2147483648 to 2147483647.



Ex.
Modifying D0 with LZ0



(1) Access D0LZ0 = D100000.

In addition, 32-bit index modification with ZZ expression using two index registers is also available.

Device for which index modification can be performed

The following table lists the devices that can be targeted for index modification.

Item	Description
16-bit index modification	X, DX, Y, DY, M, L, B, F, SB, V, 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, R, ZR, RD, P*3, I*3, 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, R, ZR, RD, K, H

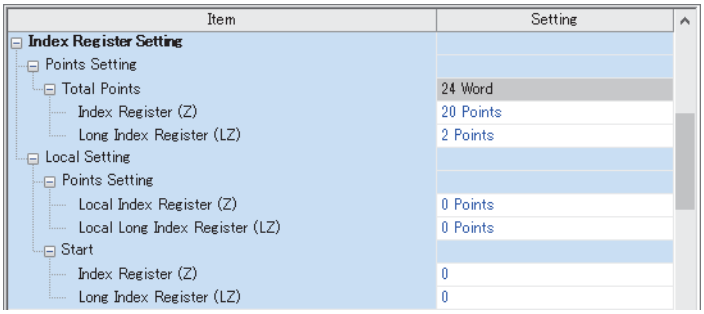
*1 The devices can be used for the contact, coil and current value.
*2 For network numbers and the specification source of I/O numbers, 32-bit index modification cannot be performed.
*3 When the devices are used as an interrupt pointer, index modification cannot be performed.

Index register setting

The following window allows to specify the number of points for the index register (Z) and long index register (LZ) and the range where they are used as a local device. The total number of points of the index register (Z) and the long index register (LZ) must be set to 24 words.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Index Register Setting]

Window



Displayed items

Item			Description	Setting range	Default
Number of points setting	Total Points		Check the total number of points for index register and long index register.	—	—
	Index register (Z)		Set the number of points for the index registers.	0 to 24 points (in two-point increments)	20 points
	Long index register (LZ)		Set the number of points for the long index registers.	0 to 12 points (in one-point increments)	2 points
Local setting	Number of points setting	Local Index register (Z)	Set the number of points for index registers used as a local device. Set within the range of the index register.	0 to 24 points (in one-point increments)	0 points
		Local long index register (LZ)	Set the number of points for long index registers used as a local device. Set within the range of the long index register.	0 to 12 points (in one-point increments)	0 points
	Head	Index register (Z)	Set the start number for the local index registers. Set within the range of the index register.	0 to 23	0
		Long index register (LZ)	Set the start number for the local long index registers. Set within the range of the long index register.	0 to 11	0

Combination of index modification

This section describes the combination of index modification.

Modification order for the device specification and index modification

According to the priority order shown below, the device specification (digit specification, bit specification, indirect specification) and index modification can be applied. However, some word devices may not follow the priority order shown below.

Order of priority	When the device targeted for the device specification and index modification is the bit device	When the device targeted for the device specification and index modification is the word device
High ↑ ↓ Low	1: Index modification 2: Digit specification	1: Index modification 2: Indirect specification 3: Bit specification

Specification method combined with device specification

The device targeted for specification is modified in order of: 1st modification, 2nd modification and then 3rd modification. Besides, the following contents can be used only for the device for which the 1st modification can be applied. (For example, index modification + digit specification is impossible for the function input (FX).)

Device targeted for specification	1st modification	2nd modification	3rd modification	Example
Bit device	Index modification	Digit specification	—	K4M100Z2
Word device	Index modification	Bit specification	—	D10Z2.0
	Index modification	Indirect specification	—	@D10Z2
	Bit specification	Index modification	—	D10.8Z2
	Indirect specification	Bit specification	—	@D10.8
	Index modification	Indirect specification	Bit specification	@D10Z2.8
	Indirect specification	Bit specification	Index modification	@D10.8Z2

Precautions

This section describes the precautions on using index modification.

Index modification between the FOR and NEXT instructions

Between the FOR instruction and the NEXT instruction, pulse output is provided through the edge relay (V). However, pulse output by the PLS, PLF, or pulse conversion (□P) instruction is not available (☞ Page 293 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 293 Edge relay (V))

Device range check for index modification

For details on the device range check when index modification is performed, refer to the following.

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

Change of the index modification range due to switching from 16-bit to 32-bit

To change the index modification range for switching from 16-bit to 32-bit, the user must:

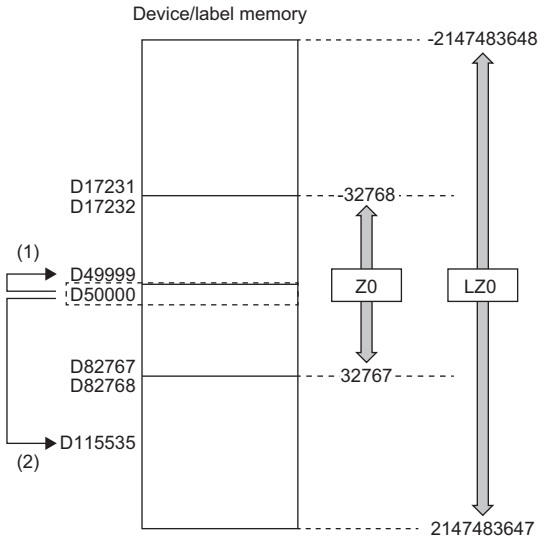
- Review the index modification block(s) within the program.
- To perform the 32-bit index modification specification with ZZ expression, review the range of the index register (Z). Note that the range within the LZ cannot be specified.
- For 32-bit index modification with ZZ expression, because the specified index register (Zn) and the immediately following index register (Zn+1) are used, caution must be taken to prevent duplicated index registers from being used.
- Review the number of points of the index register (Z) and that of the long index register (LZ), which are specified in "Index Register Setting" (☞ Page 313 Index register setting)

When values are stored in the index registers

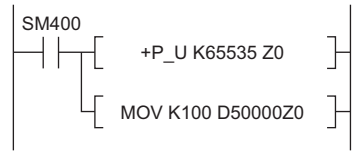
For 16-bit index modification using the index register (Z), the range is -32768 to 32767. Therefore, when values within the range from 32768 to 65535 are stored in the index register (Z) for an instruction which processes unsigned data, the instruction does not work in design because the range of the index modification will be -32768 to 32767. For the range of values larger than or equal to 32768, the long index register (LZ) must be used so that 32-bit-based index modification can be applied.

Ex.

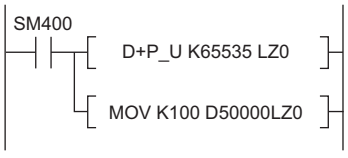
Operation for Index modification



(1) When unexpected operation is executed



(2) When proper operation is executed



(1) When the value 65535 is stored in the index register (Z), D50000(-1) to D49999 are accessed because the value is turned into -1 when an index modification is performed.

(2) When a value larger than or equal to 32768 is used for an index modification, the value must be stored in the long index register (LZ). In doing so, the value 65535 is used as such for an index modification using the long index register (LZ) and D50000 (65535) to D115535 become accessible.

26.9 File Register (R/ZR)

This device is a word device for extension. This device is specifically a file register file which exists in the file storage area on the device/label memory.

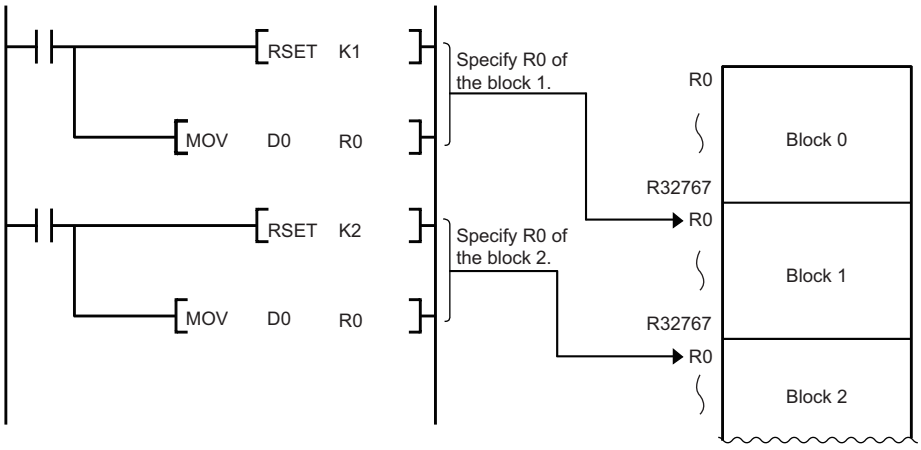
Specification method

There are two types of the specification methods for the file register: block switching and serial number methods.

Block switching method

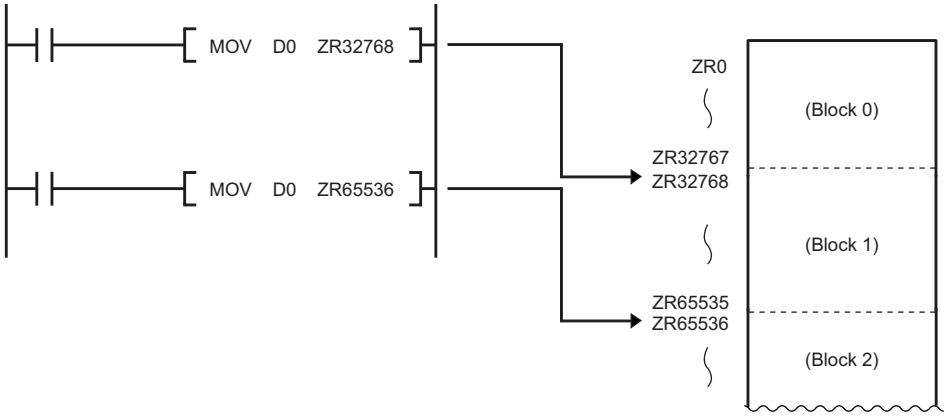
In this method the number of points of file register being used is specified by being divided in increments of 32K point (R0 to R32767). When using more than one block, specification is conducted by switching to the block number used by the RSET instruction. "R" is used as the device symbol. The range of "R" is from R0 to R32767. However in the following cases the upper limit of the device number is "block size (unit: word) -1".

- The file register file size is smaller than 64K bytes.
- The file register file size is not a multiple of 64K bytes and the end block is specified in the RSET instruction.



Serial number method

In this method file registers having more than 32K points are specified using serial device numbers. File registers of the multiple blocks can be used as consecutive file registers. "ZR" is used as the device symbol. The range of ZR is from ZR0 to (file register file size (unit: word) -1).



Setting file registers

This section describes the settings required to use the file registers.

Configuration procedure

This section describes the procedure to use the file registers.

1. Set the file register usage with [CPU Parameter].
2. To use the file register for each program, previously create the device memory which will become the file register file. (GX Works3 Operating Manual)
3. When using the file registers, which are common for all programs, a file register file with the name and capacity set in the file register setting is created. *1
4. Write parameters and file register files into the CPU module.

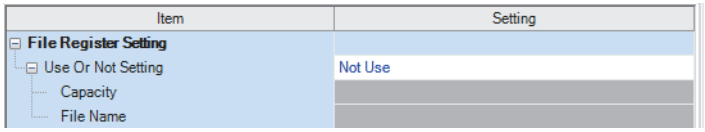
*1 If the capacity is not set, it must be set when creating a file register file and writing it to the programmable controller in the same manner as the procedure 2.

File register setting

This setting must be completed before using the file registers.

[CPU Parameter] ⇨ [File Setting] ⇨ [File Register Setting]

Window



Displayed items

Item	Description	Setting range	Default
Use Or Not Setting	Specify whether or not file registers should be used.	• Not Use • Use File Register of Each Program • Use Common File Register in All Programs	Not Use
Capacity	Specify the capacity of the file register in increments of 1K words when "Use Common File Register in All Programs" is selected.	This value depends on whether or not the extended SRAM cassette is mounted and its capacity. (☞ Page 148 Setting range of capacity of each area)	—
File name	Assign a file name to the file register when "Use Common File Register in All Programs" is selected.	1 to 60 characters	—

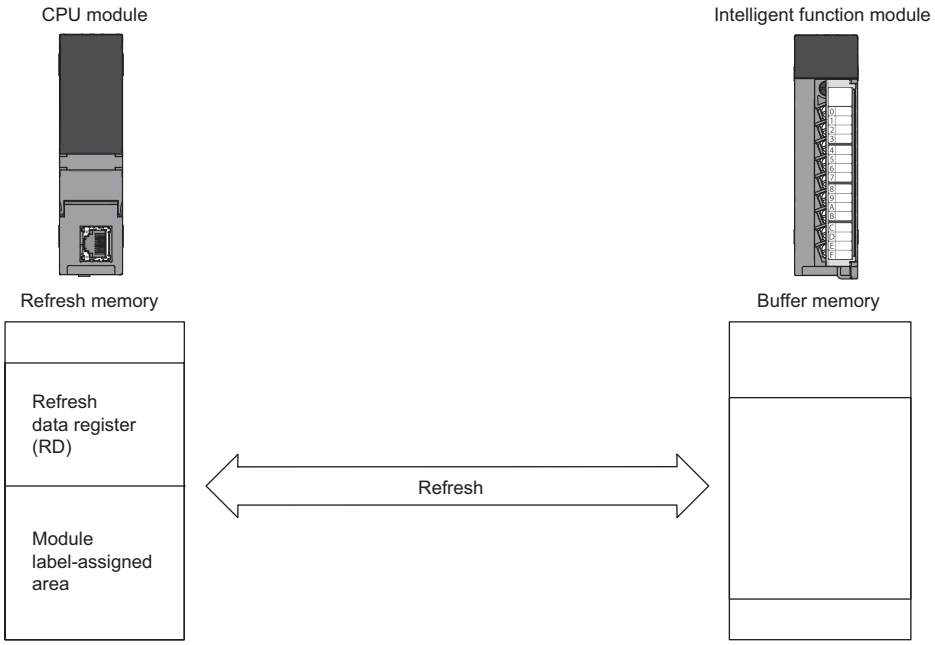
Clearing file registers

To clear the file registers, use the following methods (☞ Page 157 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)

26.10 Refresh Data Register (RD)

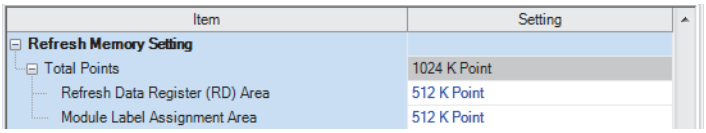
This device is provided for using as a refreshing target of buffer memory on the various devices, such as an intelligent function module. Refresh Data Register (RD) is assigned into the refresh memory area. (☞ Page 152 Refresh memory)



Refresh memory setting

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Refresh Memory Setting]

Window

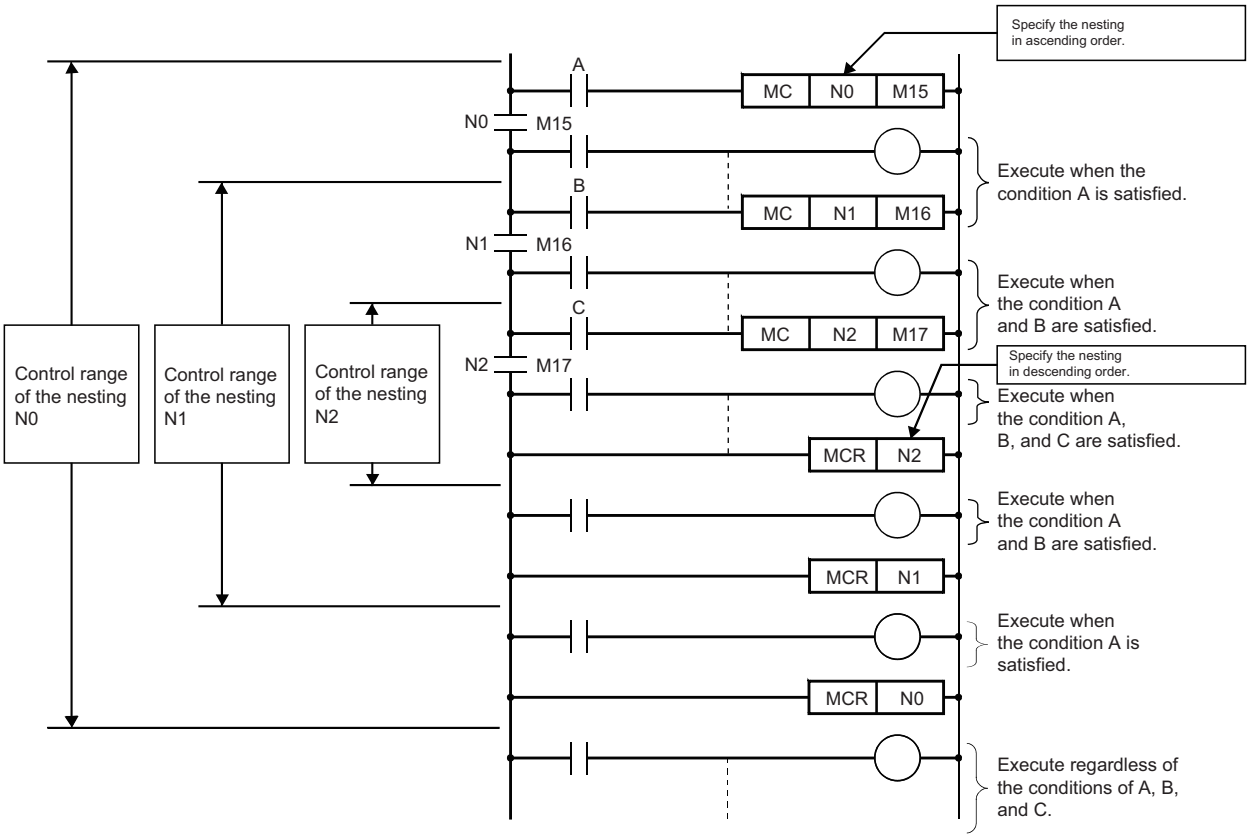


Displayed items

Item	Description	Setting range	Default
Total Points	Check the total number of points for the refresh data register and the assigned area.	—	1024K Point
Refresh Data Register (RD) area	Set the number of points of refresh data register.	0 to 1024K points (in one-point increments)	512K Point
Module Label Assignment Area	Set the number of points for the module label assigned area.	0 to 1024K points (in one-point increments)	512K Point

26.11 Nesting (N)

This device is used in the master control instructions (the MC/MCR instruction)*1 and enables the programming of operation conditions in a nesting structure. Specify this device from outside the nesting structure starting with the lowest number (in ascending order from N0 to N14).



*1 This instruction creates an effective ladder-switching program using opening/closing the common rail of the ladders.

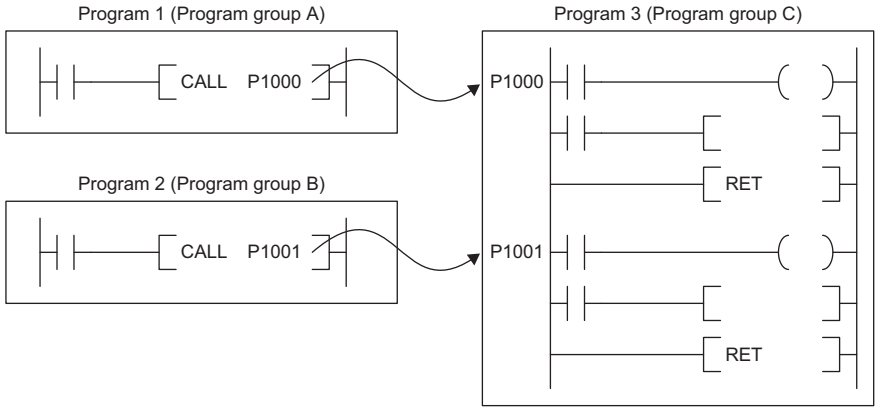
26.12 Pointer (P)

This device is used in the jump instructions (the CJ/SCJ/JMP instruction) and/or subroutine program call instructions (such as the CALL instruction). There are two types of pointer: the global pointer and the local pointer. Use the pointer when:

- Specifying the jump destination and label of the jump instructions (the CJ/SCJ/JMP instruction).
- Specifying the call destination and label (the head of subroutine program) of the subroutine call instructions (such as the CALL instruction).

Global pointer

This is the pointer which enables calling by the subroutine call instruction from all the program being executed.



Effective use of number of points

The number of points for global pointers can be obtained with the following formula: "pointer number of points specified by the parameter" - "the total point number of local pointer being used for each program".

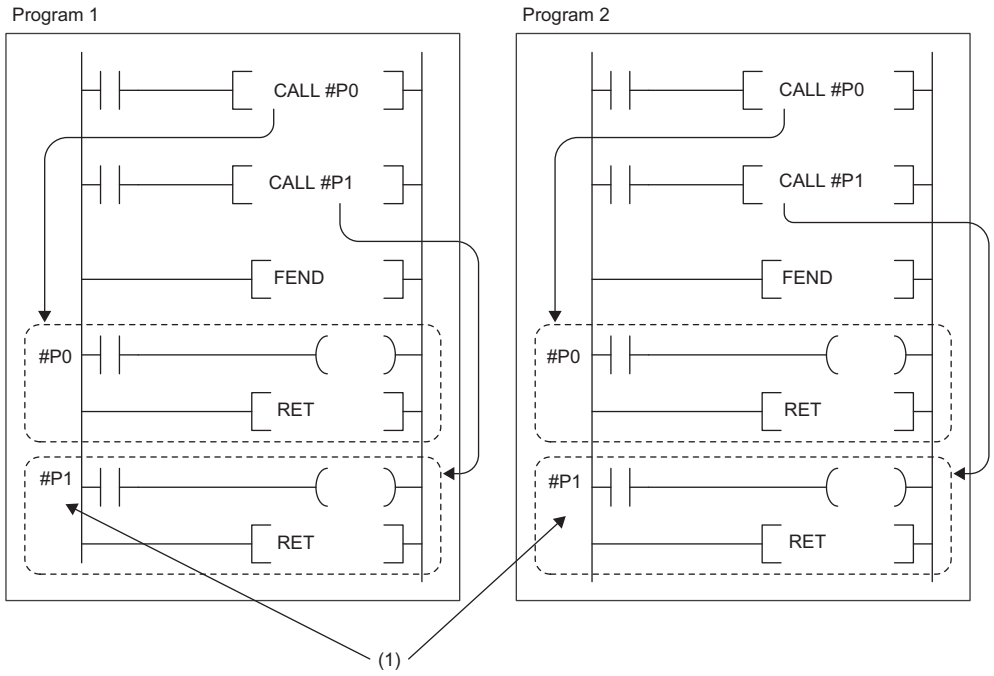
Precautions

This section describes the precautions when the global pointer is used.

- A global pointer with the same pointer number cannot be set as a label in multiple points.

Local pointer

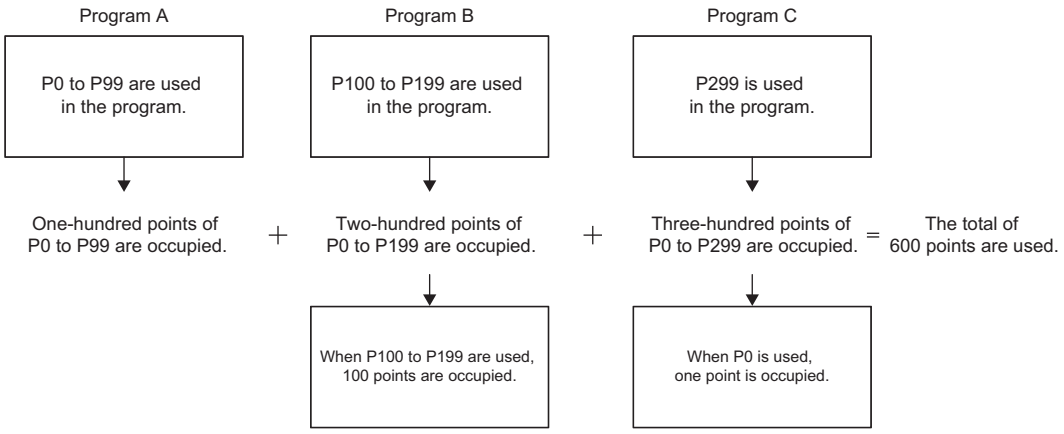
This is the pointer to be independently used in each program where the same pointer number can be used. This pointer is specified in the following format: # (pointer number) (Example: #P0) (☞ Page 328 Specification method for standard local devices).



(1) The local pointers with the same number can be used between different programs.

Effective use of number of points

Local pointer number of points are shared among all the programs. The range of the local pointer number of points used by each program is from P0 to the maximum value of the local pointer being used in that program. For example, even when a program actually uses only P99, 100 points (P0 to P99) are considered to be used. When using local pointers in multiple programs, they can be effectively used by using them in ascending order from P0 in each program group.

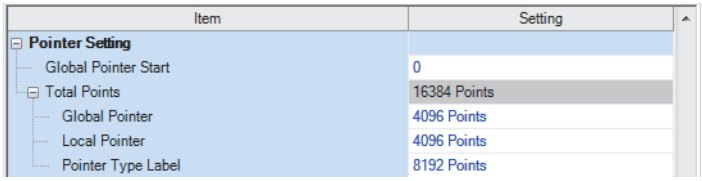


Pointer setting

The following menu item is to set pointers.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Pointer Setting]

Window



Displayed items

Item	Description	Setting range	Default
Global Pointer Start	Set the start number of the global pointer.	P0 and over*1	0
Total Points	Check the total number of points for the pointer.	—	16384 points*2
Global Pointer	Set the number of points of the global pointer.	0 to 32768 points (in one-point increments)	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

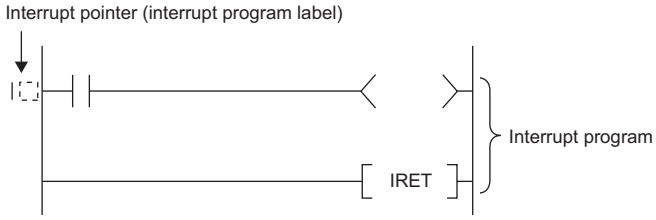
*1 Up to the number of "Total points of pointer device area" - "Number of points of global pointer".
*2 The default number of points is 32768 for the R120PSFCPU.
*3 The default number of points is 8192 for the R120PSFCPU.
*4 The default number of points is 16384 for the R120PSFCPU.

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)".

26.13 Interrupt Pointer (I)

This device is used as a label located at the head of the interrupt program. This pointer can be used in all the programs being executed.



Point

Setting the execution type of program to the event execution type eliminates the need to write (I☐) the interrupt pointer. (☞ Page 114 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.
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
I50 to I1023	Interrupt from module		5 to 8	25 to 998

Point

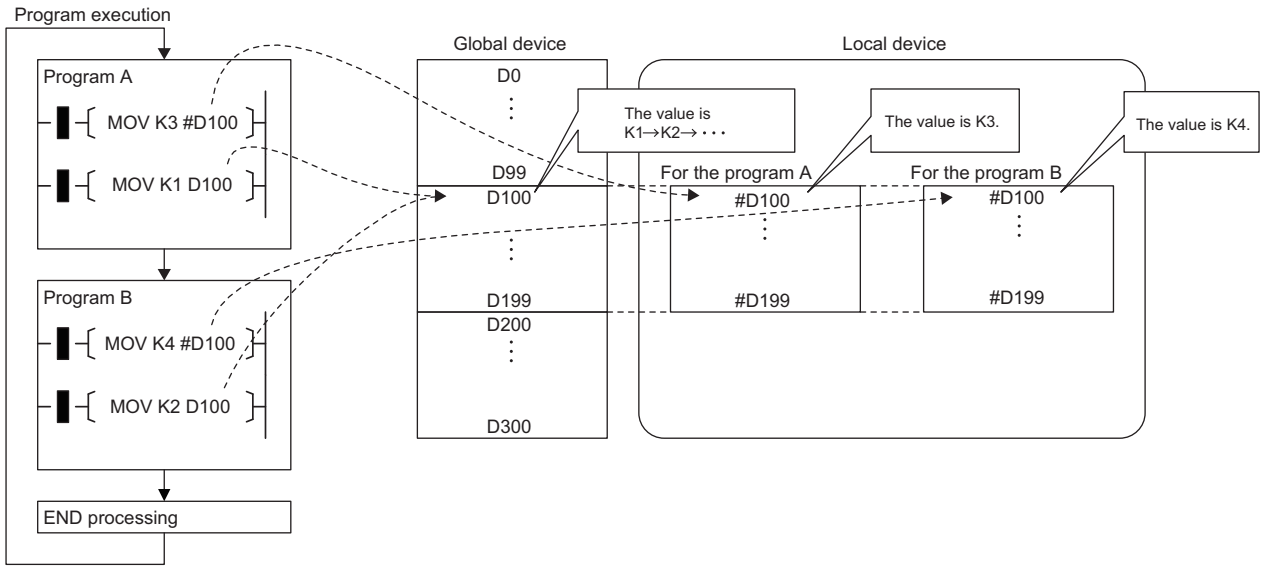
- The interrupt priority is the order which is executed at the time of the multiple interrupt. (☞ Page 133 Interrupt priority)
- The interrupt priority order is the order which is executed when the interrupt factor with the same interrupt priority is generated. (☞ Page 136 Multiple interrupt execution sequence)

26.14 Standard Global Device

This device can be shared by all standard programs. All devices that are not set as standard local devices are handled as standard global devices.

26.15 Standard Local Device

This standard device can be used independently in each program. When creating multiple standard programs, programming can be completed without being aware of devices used in other standard programs.



Point

The standard local device area must be reserved independently from the standard global device area. Therefore a standard local device and a standard global device with the same device number can exist.

Devices available as standard local device

The following devices are available as standard local devices.

- Internal relay (M)
- Edge relay (V)
- Timer (T, LT, ST, LST)
- Counter (C, LC)
- Data register (D)
- Pointer (P)

Point

Because the index register (Z, LZ) saves/returns during program execution, it should be regarded separately as the local index register unlike other local devices. (☞ Page 312 Index Register (Z/LZ))

Standard local device area

The CPU module reserves the standard local device area on the device/label memory based on the number of points setting of the standard local device when:

- CPU module is powered off and on or is reset.
- Operating status of the CPU module is changed from STOP to RUN.

When local device is used in subroutine program

Local devices to be used vary depending on whether SM776 (Local device setting at CALL) is turned on or off. Local index register to be used is also determined according to the SM776 setting.

SM776	Local device to be used
Off	Uses local devices of the program file from which subroutine program is called.
On	Uses local devices of the program file into which the subroutine program is stored.

Restriction

- In terms of on/off setting for SM776, the value (on or off) used when the subroutine call occurs is considered to be effective. Therefore when on/off setting for SM776 is switched in the subroutine program, the modified value (on or off) is not effective until the next subroutine call occurs.
- On/off setting for SM776 cannot be specified for each program file because it is enabled for each CPU module.

When a standard local device is used in interrupt and other programs

When a standard local device is used for an interrupt program / a fixed scan execution type program / an event execution type program triggered by occurrence of an interrupt, it works regardless of the SM777 setting. (☞ Page 335 SM777 setting)

Clearing standard local devices

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 standard local devices

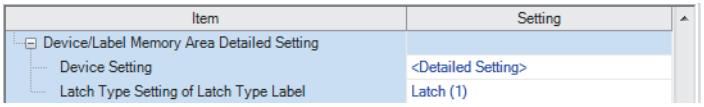
Set the range where each device will be used as a standard local device and also set whether or not it should be used.

■Range setting

The range setting for standard local devices is common to all the programs. Therefore, the range for standard local devices cannot be set for each program.

Operating procedure

"Device/Label Memory Area Detailed Setting" window



Item	Setting
Device/Label Memory Area Detailed Setting	
Device Setting	<Detailed Setting>
Latch Type Setting of Latch Type Label	Latch (1)

"Device Setting" window

Item	Symbol	Local Device		
		Start	End	Points
Input	X			
Output	Y			
Internal Relay	M			
Link Relay	B			
Link Special Relay	SB			
Annunciator	F			
Edge Relay	V			
Step Relay	S			
Timer	T			
Long Timer	LT			
Retentive Timer	ST			
Long Retentive Time	LST			
Counter	C			
Long Counter	LC			
Data Register	D			
Link Register	W			
Link Special Registe	SW			
Latch Relay	L			
Total Device				0.0K Word
Total Word Device				0.0K Word
Total Bit Device				0.0K Bit

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

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

2. Set the range where each device will be used as a standard local device.

Point

Configure the setting range of the standard local device within the range which has been set for the number of device points. The number of standard local devices used is calculated by the following calculation formula. Set the number of standard local devices used so that the number is equal to or less than the capacity of the standard local device area.

Total number of standard local devices used = ((A ÷ 16) + B + (C × 2) + (D × 4) + ((E × 2) ÷ 16)) × F

- A: Number of points of the standard local devices M and V
- B: Number of points of the standard local devices D, T (current value), ST (current value), and C (current value)
- C: Number of points of the standard local device LC (current value)
- D: Number of points of the standard local devices LT and LST
- E: Number of points of the standard local devices T (contact/coil), ST (contact/coil), C (contact/coil), and LC (contact/coil)
- F: Number of programs using the standard local device

■Setting unit

The increment of setting of a standard local device is the same as that of a standard global device. (☞ Page 289 Range of use of device points)

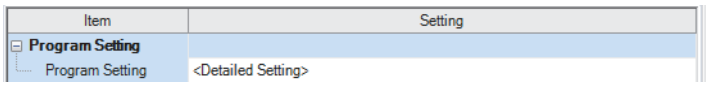
Setting whether or not it should be used

Whether or not standard local devices should be used can be set for each program. Since the standard 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.

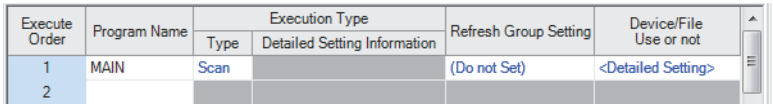
[CPU Parameter] ⇨ [Program Setting]

Operating procedure

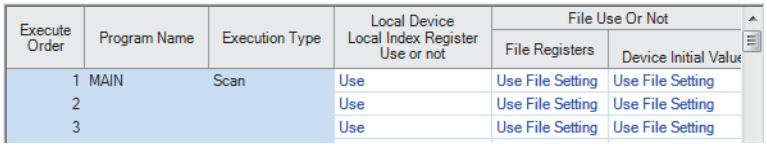
"Program Setting" window



"Detailed Setting" window



"Setting of Device/File Use Or Not" window



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 standard local devices in a program which is configured not to use standard local devices.

Specification method for standard local devices

To specify standard local devices in the program, add "#".

Ex.

For example, #D100, K4#M0, and @#D0 can be used.

Point

Standard local devices are listed with a preceding # symbol in the program. This symbol is helpful to differentiate standard local devices from standard global devices.

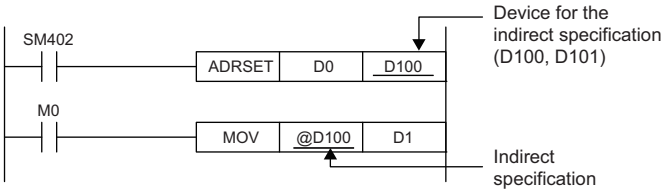
Precautions

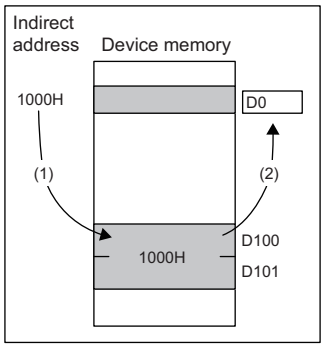
This section describes the precautions when using standard local devices.

- Like standard global devices, the timer (T, LT, ST, LST) and counter (C, LC) specified as a standard 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 standard global devices and standard local devices by the index modification is not allowed.
- When the range of the 32-bit index modification is across the setting ranges of standard local devices of the index register, proper index modification is impossible.
- Standard local devices are not latched.

26.16 Indirect Specification

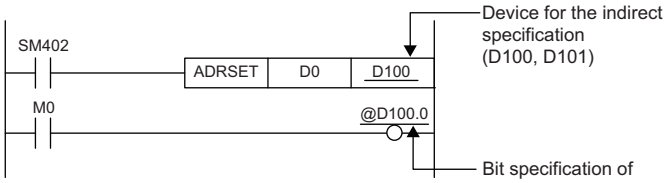
Specify the device using the indirect address of device. Store the indirect address of device to be specified into the device for indirect specification, and write as "@ + Device for indirect specification".

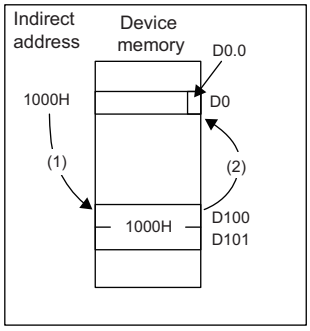




(1) The indirect address of D0 is read into D100, D101.
(2) The indirect address is used to indirectly specify D0.

Besides, specifying a bit of a word device allows the indirect specification for the instruction that specifies bits.





(1) The indirect address of D0 is read into D100, D101.
(2) The indirect address is used to output at the 0th bit of D0 which was indirectly specified using the indirect address.

The indirect specification can be used in the device/label memory or refresh memory.

Indirect address of device

To specify, use the 32-bit data, and to hold the value, use the device of two words. The indirect address of the device can be obtained with the ADRSET instruction. The ADRSET instruction specifies the indirect address of the device using instructions that handle 32-bit data. For the ADRSET instruction, refer to the following.

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

Point

When the block or the file of the file register is switched through the RSET or QDRSET instruction, the indirect address refers to the one of the block or the file before they are switched. To allow the indirect address in the device for indirect specification to specify the block or file after the file register is switched, specify the ADRSET instruction to obtain the indirect address again after block or file are switched.

Devices that can allow indirect specification

This section lists devices that can allow indirect specification.

Type	Device*3
Devices that can allow indirect specification where @ is added*1	T, ST, C, D, W, SW, FD, SD, Un\G, Jn\W, Jn\SW, U3En\G, 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 They can also be used for the standard local device. (e.g.: @#D0)
*2 The indirect address of device can be obtained for the standard 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.

27 SAFETY DEVICES

This chapter describes the safety devices.

27.1 Safety Devices

A safety device is a device used in safety programs. The safety devices can be used only in the safety programs.

Restriction

- The safety devices cannot be used in the standard programs.
- An index modification and indirect specification cannot be performed in safety programs.

List of safety devices

The following table lists the safety devices.

Classification	Type	Device name	Symbol	Number of points (default)	Changeability with parameter settings	Notation
Safety user devices	Bit	Safety input	SA\X	8K points	Selectable either 8K or 12K points	Hexadecimal
	Bit	Safety output	SA\Y	8K points		Hexadecimal
	Bit	Safety internal relay	SA\M	6K points	Changeable (☞ Page 148 Device/label memory area setting)	Decimal
	Bit	Safety link relay	SA\B	4K points		Hexadecimal
	Bit/word	Safety timer	SA\T	512 points		Decimal
	Bit/word	Safety retentive timer	SA\ST	0 points		Decimal
	Bit/word	Safety counter	SA\C	512 points		Decimal
	Word	Safety data register	SA\D	12K points		Decimal
	Word	Safety link register	SA\W	4K points		Hexadecimal
Safety system devices	Bit	Safety special relay	SA\SM	4K points	Unchangeable	Decimal
	Word	Safety special register	SA\SD	4K points		Decimal

Point

For details on standard devices, refer to the following.

☞ Page 286 STANDARD DEVICES

Safety user device

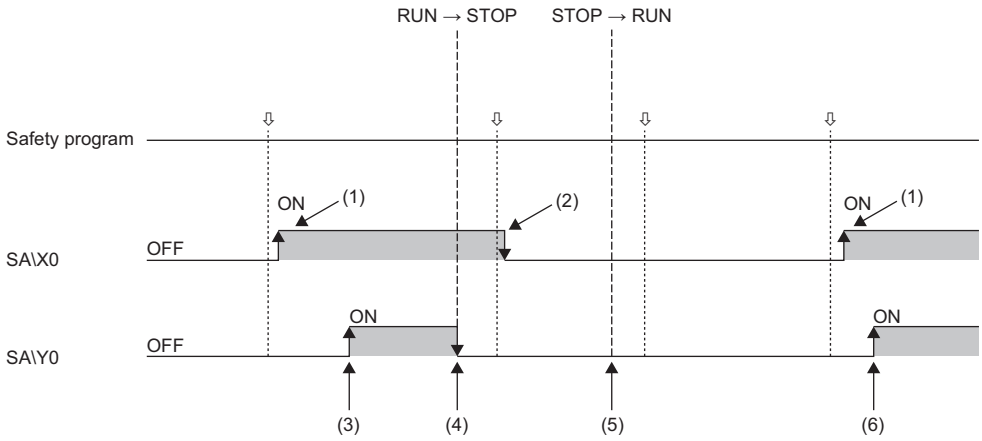
This section describes the safety user devices.

Safety input (SA\X)

Safety inputs are used to give instructions or data to the CPU module with external devices such as emergency stop buttons, safety plugs, door switches, and light curtains.

Safety output (SA\Y)

Safety outputs are used to output safety program control results to devices such as external relays and contactors.
When the status changes from STOP to RUN, the value immediately before the RUN state is retained, and the value is updated at the moment safety cycle processing is executed for the first time after the status is changed to the RUN state.



⇩: Safety cycle processing start
(1) Turns on when safety communications are received.
(2) Turns off when safety communications are received.
(3) SA\Y0 turns on when SA\X0 turns on in the safety program.
(4) SA\Y0 turns off when the status of the CPU module is switched to STOP.
(5) The value is maintained when the status changes from STOP → RUN.
(6) Turns on when a safety program is executed.

Safety internal relay (SA\M)

This device is used as an auxiliary relay within the CPU module. The following operations for the CPU module turn off all safety internal relays.

- Powering off and on
- Reset

Safety link relay (SA\B)

This device is used as a device when sending and receiving safety data between CPU modules over CC-Link IE Field Network.

■Refreshing network modules using safety link relay

Data is sent and received between CPU modules. 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 294 Timer)

■Safety timer types

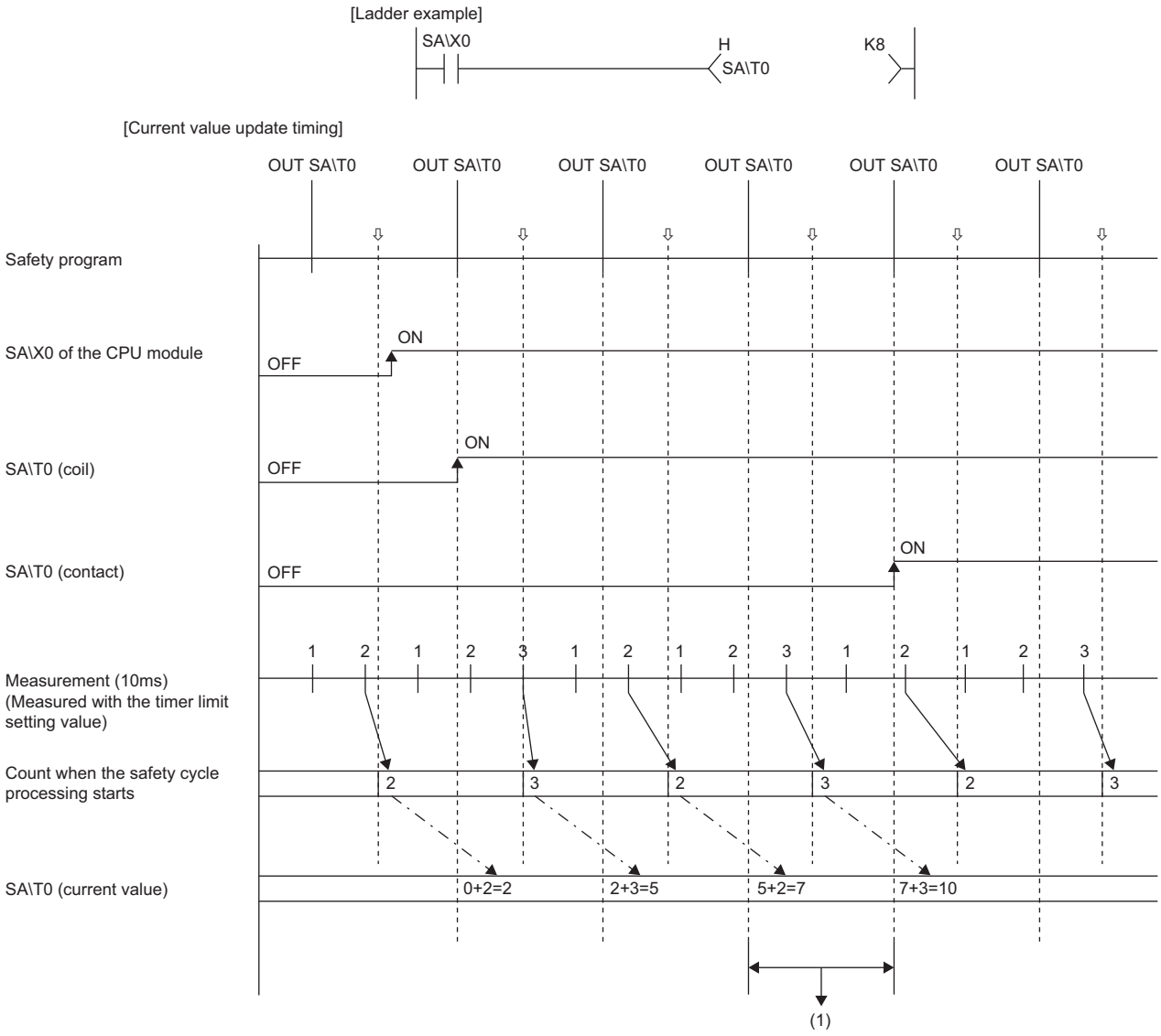
There is a safety timer (SA\T) that retains the current value in 16-bit units. In addition, there is a safety retentive timer (SA\ST) that retains the current value even if the coil turns off.

■Safety timer accuracy

The current value is measured when safety cycle processing starts. The value for the elapsed time since the previous safety cycle processing was started until the present is added to the current value when the OUT SA\T☐ instruction is executed. If the safety timer coil is off at the execution of the OUT SA\T☐ instruction, the current value is not updated. The maximum response accuracy of the timer is the "elapsed time since previous safety cycle processing was started until the present + timer limit setting".

Ex.

Timer limit setting = 10ms, SA\T0 setting value = 8



⇩: Safety cycle processing start

(1) Time accuracy - (Elapsed time since the previous safety cycle processing was started until the present + Timer limit setting) to (Elapsed time since the previous safety cycle processing was started until the present)

(2) Sets the coefficient when starting safety cycle processing.

Safety counter (SA\C)

This device counts the number of rising operation of the input condition in the program. The safety counter is an up-timing type device and therefore when the count value matches a setting value, the count reaches its upper limit and the contact is turned on. Operations other than the following are the same as those for the counter. (☞ Page 302 Counter)

■Safety counter types

There is a safety counter (SA\C) that retains the counter value in 16-bit units.

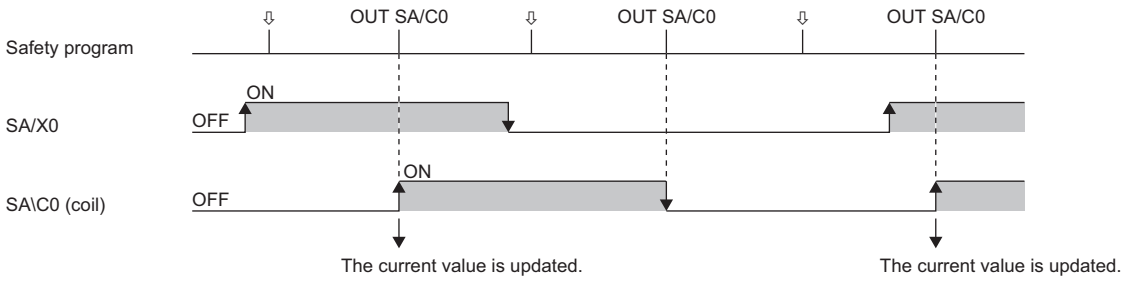
■Counting process

When the safety counter coil is executed, the safety counter coil is turned on/off, the current value is updated (count value +1), and the contact is turned on/off. The current value is updated (count value +1) when the safety counter coil input is rising (off to on). The current value is not updated when the coil input is off, on to on, and on to off.

[Ladder example]



[Current value update timing]



⇩: Safety cycle processing start

■Resetting the safety counter

The safety counter current value is not cleared even when the counter coil input is turned off. To clear (reset) the safety counter current value and turn off the contact, issue the RST SA\C☐ instruction. When the RST SA\C☐ instruction is executed, the counter value is cleared and the contact is turned off.

Safety data register (SA\D)

This device can store numerical values.

Safety link register (SA\W)

This device is used as a device when sending and receiving safety data between CPU modules over CC-Link IE Field Network.

■Refreshing network modules using safety link register

Data is sent and received between CPU modules. The safety communications send/receive device range is set in the CC-Link IE Field Network safety communication settings. Locations that are not used for safety communications send/receive devices can be used for other applications.

Safety system devices

This section describes the safety system devices.

Safety special relay (SA\SM)

This relay stores the CPU module status relating to safety control. (☞ Page 524 List of Safety Special Relay Areas)

Safety special register (SA\SD)

This register stores the CPU module status relating to safety control. (☞ Page 527 List of Safety Special Register Areas)

27.2 Safety Global Devices

A safety global device is a device that can be shared by all the safety programs.

27.3 Safety Local Devices

A safety local device is a device used individually by each safety program.

Devices that can be used as a safety local device

The following devices can be used as a safety local device.

- Safety internal relay (SA\M)
- Safety timer (SA\T)
- Safety retentive timer (SA\ST)
- Safety counter (SA\C)
- Safety data register (SA\D)

How to set safety local devices

The range and usability of the safety local devices can be set in the same way as the standard local devices. (☞ Page 327 Setting method for standard local devices)

Point

Set safety local devices within the number of device points set in the safety device point quantity setting. The number of safety local device points used is calculated by the following calculation formula. Set the number of safety local device points used so that the number is equal to or less than the capacity of the safety local device area.

Total number of safety local device points used = ((A ÷ 16) + B + (C × 2) ÷ 16)) × D

- A: Number of points of the safety local device (SA\M)
- B: Number of points of the safety local devices (SA\D, SA\T (current value), SA\ST (current value), and SA\C (current value))
- C: Number of points of the safety local devices (SA\T (contact/coil), SA\ST (contact/coil), and SA\C (contact/coil))
- D: Number of programs using the safety local device

How to specify safety local devices

To specify safety local devices in a safety program, add "#".

Ex.

SA\#D100, SA\K4#M0

Point

In the program, safety local devices are displayed with "#" in front of the device name. This helps users to distinguish local devices from global devices.

SM777 setting

Regardless of the SM777 (Local device setting in interrupt programs) setting, local devices/safety local devices of a program file in the storage location are always used in the standard program/safety program.

28 STANDARD LABEL

A label is a variable consisting of a specified string used in I/O data or internal processing.
A label used in standard programs is called a standard label. (Labels are called standard labels to distinguish them from safety labels.)
Programs can be created without considering the size of devices and buffer memory by using standard labels. For this reason, a program using standard labels can be reused easily even in a system having a different module configuration.
When standard labels are used, there are some precautions on programming and functions used. For details, refer to the following.
☞ Page 348 Precautions

Point

There are two types of labels described in this manual.
- Standard global label
- Standard local label
- Safety label (☞ Page 350 SAFETY LABEL)

There are other types of labels available in addition to the global labels and local labels.
[System labels]
A system label is a label that provides the same data in all projects compatible with iQ Works. It can be referenced from the GOT and the CPU modules and Motion controllers on other stations, and used for monitoring and accessing data.
For details, refer to the following.
📖Let's start iQ Works Version 2
[Module labels]
A module label is a label defined uniquely by each module. A module label is automatically generated by the engineering tool from the module used, and can be used as a global label.
For details, refer to the following.
📖 Function Block Reference for the module used

28.1 Standard Global Label

A global label is a label that provides the same data within a single project. It can be used in all programs in the project.
A global label can be used in program blocks and function blocks.
The settings of a standard global label include a label name, class, and data type.
By opening global labels, they can be referenced from GOT and other stations, and can be used for monitoring and accessing data.

Device assignment

Devices can be assigned to standard global labels.

Item	Description
Label to which no device is assigned	• Programming without being aware of devices is possible. • Defined labels are allocated to the label area or latch label area in the device/label memory.
Label to which a device is assigned	• If a device is to be programmed as a label against a device that is being used for input or output, the device can be assigned directly. • Defined labels are allocated to the device area in the device/label memory.

28.2 Standard Local Label

A local label is a label that can be used only in the declared POU. Standard local labels outside the declared POU cannot be used.

The settings of a standard local label include a label name, class, and data type.

28.3 Class

The standard label class indicates from which POU and how a label can be used.

Different classes can be selected depending on the type of POU.

Global label				
Class	**Description**	**Applicable POU**		
		Program block	**Function block (FB)**	**Function (FUN)**
VAR_GLOBAL	A common label that can be used in both program blocks and function blocks	○	○	×
VAR_GLOBAL_CONSTANT	A common constant that can be used in both program blocks and function blocks	○	○	×
VAR_GLOBAL_RETAIN	A latch type label that can be used in both program blocks and function blocks	○	○	×

Local label				
Class	**Description**	**Applicable POU**		
		Program block	**Function block (FB)**	**Function (FUN)**
VAR	A label that can be used within the range of a declared POU. This label cannot be used in other POUs.	○	○	○
VAR_CONSTANT	A constant that can be used within the range of a declared POU. This label cannot be used in other POUs.	○	○	○
VAR_RETAIN	A latch type label that can be used within the range of a declared POU. This label cannot be used in other POUs.	○	○	×
VAR_INPUT	A label that inputs a value into a function or function block. This label receives a value, and the received value cannot be changed in a POU.	×	○	○
VAR_OUTPUT	A label that outputs a value from a function or function block	×	○	○
VAR_OUTPUT_RETAIN	A latch type label that outputs a value from a function block	×	○	×
VAR_IN_OUT	A local label that receives a value and outputs the value from a POU. The value can be changed in a POU.	×	○	×
VAR_PUBLIC	A label that can be accessed as the public variable from other POUs	×	○	×
VAR_PUBLIC_RETAIN	A latch type label that can be accessed as the public variable from other POUs	×	○	×

28.4 Data Types

The data types of a standard label are classified according to the bit length, processing method, and value range.

There are two data types.

- Primitive data type
- Generic data type (ANY type)

Primitive data type

The following table lists the data types included in the primitive data type.

Data type		Description	Value range	Bit length
Bit	BOOL	Represents the alternative status, such as on or off.	0 (FALSE), 1 (TRUE)	1 bit
Word [unsigned]/bit string [16 bits]	WORD	16-bit array	0 to 65535	16 bits
Double word [unsigned]/bit string [32 bits]	DWORD	32-bit array	0 to 4294967295	32 bits
Word [signed]	INT	Positive and negative integer values	-32768 to 32767	16 bits
Double word [signed]	DINT	Positive and negative double-precision integer values	-2147483648 to 2147483647	32 bits
Single-precision real number*1	REAL	Numerical values of decimal places (single-precision real number values)	-2^{128} to -2^{-126}, 0, 2^{-126} to 2^{128} E-3.402823+38 to E-1.175495-38, 0, E1.175495-38 to E3.402823+38	32 bits
Double-precision real number*1	LREAL	Numerical values of decimal places (double-precision real number values)	-2^{1024} to -2^{-1022}, 0, 2^{-1022} to 2^{1024} E-1.79769313486231+308 to E-2.22507385850721-308, 0, E2.22507385850721-308 to E1.79769313486231+308	64 bits
Time*2	TIME	Numerical values as day, hour, minute, second, and millisecond	T#-24d20h31m23s648ms to T#24d20h31m23s647ms*3	32 bits
String	STRING	Characters represented by ASCII code or Shift JIS code	255 one-byte characters maximum	Variable
String [Unicode]	WSTRING	Characters represented by Unicode	255 characters maximum	Variable
Timer	TIMER	Structure corresponding to the device, timer (T)	☞ Page 339 Timer and counter data types	
Retentive timer	RETENTIVETIMER	Structure corresponding to the device, retentive timer (ST)		
Long timer	LTIMER	Structure corresponding to the device, long timer (LT)		
Long retentive timer	LRETENTIVETIMER	Structure corresponding to the device, timer (LST)		
Counter	COUNTER	Structure corresponding to the device, counter (C)		
Long counter	LCOUNTER	Structure corresponding to the device, counter (LC)		
Pointer	POINTER	Type corresponding to the device, pointer (P) (☞ Page 320 Pointer (P))		

*1 For the number of significant digits and input range of real number data input by the engineering tool, refer to the following.
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

*2 The time type is used in a time data type function of standard functions. For standard functions, refer to the following.
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)

*3 For the notation of time, refer to the following.
☞ Page 372 Notation of time

Point

- The bit data in the word type label can be used by specifying a bit number.
- The bit type array label can be used as 16-bit or 32-bit data by specifying the number of digits.

For the bit specification and digit specification methods, refer to the following.

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

■Timer and counter data types

The data types of the timer, counter, long counter, retentive timer, long retentive timer, and long timer are the structures having a contact, coil, or current value.

Data type		Member name	Member data type	Description	Value range
Timer	TIMER	S	Bit	Indicates a contact. The operation is the same as the contact (TS) of a timer device.	0 (FALSE), 1 (TRUE)
		C	Bit	Indicates a coil. The operation is the same as the coil (TC) of a timer device.	0 (FALSE), 1 (TRUE)
		N	Word [unsigned]/bit string [16 bits]	Indicates the current value. The operation is the same as the current value (TN) of a timer device.	0 to 65535*1
Retentive timer	RETENTIVETIMER	S	Bit	Indicates a contact. The operation is the same as the contact (STS) of a retentive timer device.	0 (FALSE), 1 (TRUE)
		C	Bit	Indicates a coil. The operation is the same as the coil (STC) of a retentive timer device.	0 (FALSE), 1 (TRUE)
		N	Word [unsigned]/bit string [16 bits]	Indicates the current value. The operation is the same as the current value (STN) of a retentive timer device.	0 to 65535*1
Long timer	LTIMER	S	Bit	Indicates a contact. The operation is the same as the contact (LTS) of a long timer device.	0 (FALSE), 1 (TRUE)
		C	Bit	Indicates a coil. The operation is the same as the coil (LTC) of a long timer device.	0 (FALSE), 1 (TRUE)
		N	Double word [unsigned]/bit string [32 bits]	Indicates the current value. The operation is the same as the current value (LTN) of a long timer device.	0 to 4294967295*1
Long retentive timer	LRETENTIVETIMER	S	Bit	Indicates a contact. The operation is the same as the contact (LSTS) of a long retentive timer device.	0 (FALSE), 1 (TRUE)
		C	Bit	Indicates a coil. The operation is the same as the coil (LSTC) of a long retentive timer device.	0 (FALSE), 1 (TRUE)
		N	Double word [unsigned]/bit string [32 bits]	Indicates the current value. The operation is the same as the current value (LSTN) of a long retentive timer device.	0 to 4294967295*1
Counter	COUNTER	S	Bit	Indicates a contact. The operation is the same as the contact (CS) of a counter device.	0 (FALSE), 1 (TRUE)
		C	Bit	Indicates a coil. The operation is the same as the coil (CC) of a counter device.	0 (FALSE), 1 (TRUE)
		N	Word [unsigned]/bit string [16 bits]	Indicates the current value. The operation is the same as the current value (CN) of a counter device.	0 to 65535
Long counter	LCOUNTER	S	Bit	Indicates a contact. The operation is the same as the contact (LCS) of a long counter device.	0 (FALSE), 1 (TRUE)
		C	Bit	Indicates a coil. The operation is the same as the coil (LCC) of a long counter device.	0 (FALSE), 1 (TRUE)
		N	Double word [unsigned]/bit string [32 bits]	Indicates the current value. The operation is the same as the current value (LCN) of a long counter device.	0 to 4294967295

*1 The unit of the current value is set in CPU parameters ("Timer Limit Setting").

For details on the operation of each device, refer to the following.

☞ Page 286 STANDARD DEVICES

The specification method of each member is the same as that of the structure data type. (☞ Page 345 Structures)

Generic data type (ANY type)

The generic data type is the data type of the labels which summarize several primitive data types.

Generic data types are used when multiple data types are allowed for function and function block arguments and return values.

Labels defined in generic data types can be used in any sub-level data type.

For the types of generic data types and the primitive data types, refer to the following.

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

Definable data types and initial values

The following tables list the definable data types and initial value setting possibilities for each label class.

Global label		
Class	**Definable data type**	**Initial value setting possibility**
VAR_GLOBAL	Primitive data type, array, structure, function block	○
VAR_GLOBAL_CONSTANT	Primitive data type*1*2	×
VAR_GLOBAL_RETAIN	Primitive data type*1, array, structure	○

Local label (program block)		
Class	**Definable data type**	**Initial value setting possibility**
VAR	Primitive data type, array, structure, function block	○
VAR_CONSTANT	Primitive data type*1*2	×
VAR_RETAIN	Primitive data type*1, array, structure	○

Local label (function)		
Class	**Definable data type**	**Initial value setting possibility**
VAR	Primitive data type*2, array, structure	×
VAR_CONSTANT	Primitive data type*1*2	×
VAR_INPUT	Primitive data type*1*2, array, structure	×
VAR_OUTPUT		×
Return value		×

Local label (function block)		
Class	**Definable data type**	**Initial value setting possibility**
VAR	Primitive data type, array, structure, function block	○
VAR_CONSTANT	Primitive data type*1*2	×
VAR_RETAIN	Primitive data type*1, array, structure	○
VAR_INPUT		○
VAR_OUTPUT		○
VAR_OUTPUT_RETAIN		○
VAR_IN_OUT		×
VAR_PUBLIC		○
VAR_PUBLIC_RETAIN		○

*1 The pointer type cannot be defined.

*2 None of the timer, retentive timer, long timer, long retentive timer, counter, and long counter types can be defined.

Point

- The initial value of the global label where the device has been assigned follows that in the device.
- The initial value of the function block follows the local label setting within the function block.
- The initial value of the structure type follows that of the structure definition.

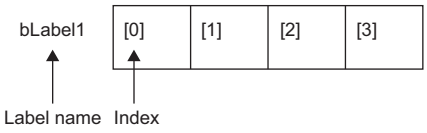
28.5 Arrays

An array represents a consecutive aggregation of same data type labels as a single name.
Primitive data types and structures can be defined as arrays.

Ex.

Array image and setting in engineering tool

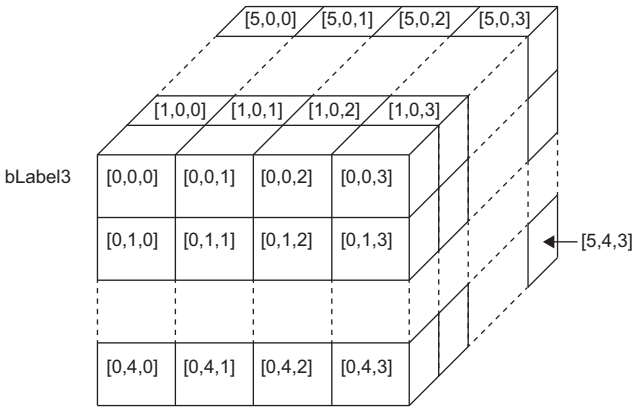
- One-dimensional array (The number of elements is 4.)



- Two-dimensional array (The number of elements is 5 × 4.)

bLabel2				
	[0,0]	[0,1]	[0,2]	[0,3]
	[1,0]	[1,1]	[1,2]	[1,3]
	[4,0]	[4,1]	[4,2]	[4,3]

- Three-dimensional array (The number of elements is 6 × 5 × 4.)



- Settings in the engineering tool

	Label Name	Data Type		Class
1	bLabel1	Bit(0..3)	...	VAR
2	bLabel2	Bit(0..4,0..3)	...	VAR
3	bLabel3	Bit(0..5,0..4,0..3)	...	VAR

Defining arrays

■Array elements

When an array is defined, the number of elements, or the length of array, must be determined. For the range of the number of elements, refer to the following.

☞ Page 344 Range of the number of array elements

■Dimension number of multidimensional array

Up to three-dimensional array can be defined.

■Definition format

The following table lists definition format.

The range from the array start value to the array end value is the number of elements.

Number of array dimensions	Format	Remarks
One dimension	Array of primitive data type/structure name (array start value..array end value)	• For the primitive data type: ☞ Page 338 Primitive data type • For the structure name: ☞ Page 345 Structures
	[Definition example] Bit (0..15)	
Two dimensions	Array of primitive data type/structure name (array start value..array end value, array start value..array end value)	
	[Definition example] Bit (0..1, 0..15)	
Three dimensions	Array of primitive data type/structure name (array start value..array end value, array start value..array end value, array start value..array end value)	
	[Definition example] Bit (0..2, 0..1, 0..15)	

■Initial value

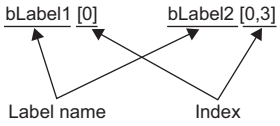
One initial value can be set for a single array definition. (Different initial values cannot be set for each element.)

The same initial value is stored in all the array elements.

How to use arrays

To use an array, add an index enclosed by '[ ]' after each label name to identify individual labels.

An array with two or more dimensions should be represented with indexes delimited by a comma (,) in '[ ]'.

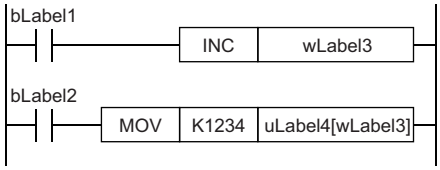


The following table lists the types of indexes that can be specified for arrays.

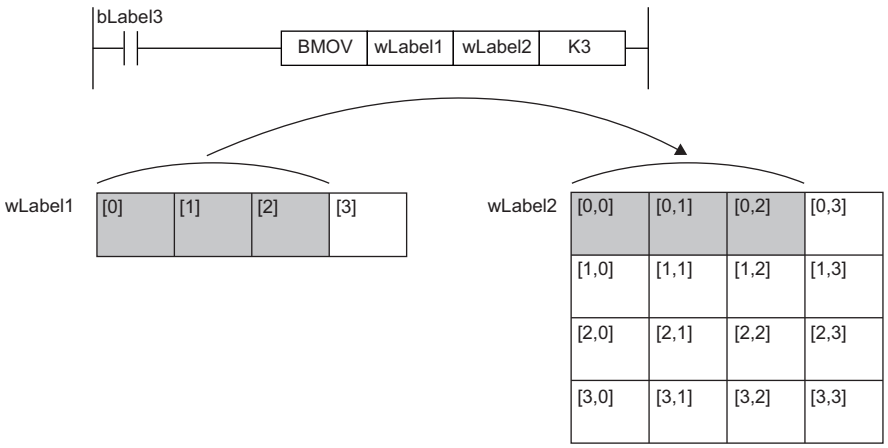
Type	Specification example	Remarks
Constant	bLabel1[0]	An integer can be specified.
Device	bLabel1[D0]	A word device, double-word device, decimal constant, or hexadecimal constant can be specified. (ST, LST, G, and HG cannot be specified.)
Label	bLabel1[uLabel2]	The following data types can be specified. • Word [unsigned]/bit string [16 bits] • Double word [unsigned]/bit string [32 bits] • Word [signed] • Double word [signed]
Expression	bLabel1[5+4]	Expressions can be specified only in ST language.

Point

- The data storage location becomes dynamic by specifying a label for the array index. This enables arrays to be used in a program that executes loop processing. The following is a program example that consecutively stores "1234" in the "uLabel4" array.



- The element number of the array can be omitted in ladder diagram. If the element number is omitted, it is regarded as the start number and converted. For example, when the defined label name is "boolAry" and the data type is "Bit (0..2, 0..2)", the operation of "boolAry[0,0]" is the same as that of "boolAry".
- When a multidimensional array is specified as setting data of instructions, functions, and function blocks that use arrays, the rightmost element is regarded as a one-dimensional array.



Range of the number of array elements

The maxim number of array elements varies depending on the data type.

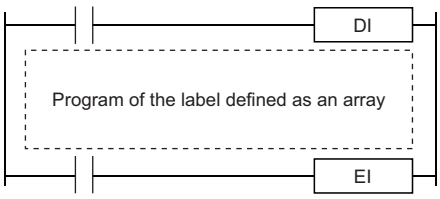
Data type	Setting range
Bit Word [unsigned]/bit string [16 bits] Word [signed]	1 to 2147483648
Double word [unsigned]/bit string [32 bits] Double word [signed] Single-precision real number Time	1 to 1073741824
Timer Counter Retentive timer Long counter Long retentive timer Long timer	1 to 32768
Double-precision real number	1 to 536870912
String	1 to 67108864
String [Unicode]	1 to 33554432
Structure type Function block	1 to 32768

Precautions

■When an interrupt program is used

When a label or device is specified for the array index, the operation is performed with a combination of multiple instructions. For this reason, if an interrupt occurs during operation of the label defined as an array, data inconsistency may occur producing an unintended operation result.

To prevent data inconsistency, create a program using the DI/EI instructions that disables/enables interrupt programs as shown below.



For the DI/EI instructions, refer to the following.

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

■Array elements

When accessing the element defined in an array, access it within the range of the number of elements.

If a constant out of the range defined for the array index is specified, a compile error will occur.

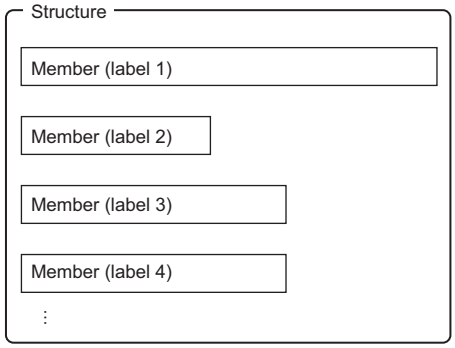
If the array index is specified with data other than a constant, a compile error will not occur. The processing will be performed by accessing another label area or latch label area.

28.6 Structures

A structure is a data type containing one or more labels and can be used in all POUs.
Members (labels) included in a structure can be defined even when their data types are different.

Creating structures

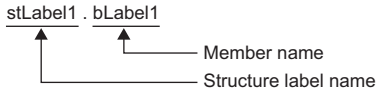
To create a structure, first define the structure, and then define members in the structure.



How to use structures

To use a structure, register a label using the defined structure as the data type.
To specify each member in a structure, add the member name after the structure label name with a period '.' as a delimiter in between.

Ex.
Specifying a member in the structure



Point

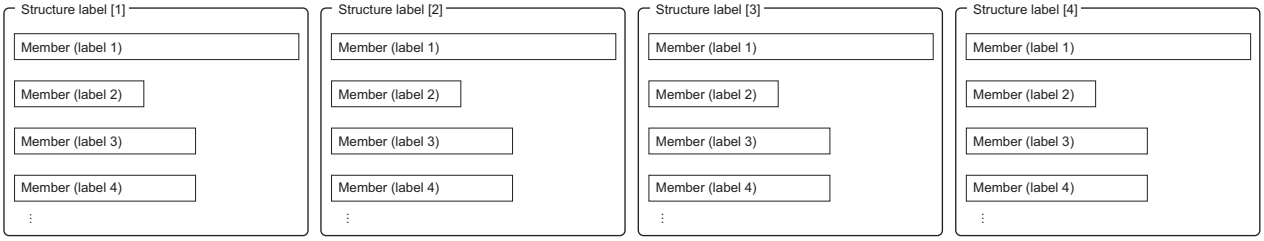
- When labels are registered by defining multiple data types in a structure and used in a program, the order the data is stored after operation is not the order the data types were defined. When programs are converted using the engineering tool, labels are classified into type and data type, and then assigned to the memory (memory assignment by packing blocks).

GX Works3 Operating Manual

- If the label of a structure is specified for an instruction that uses control data (a group of operands that determines operation of the instruction), the labels are not assigned in the order defined by packing blocks.

Structure arrays

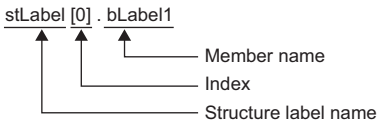
A structure can also be used as an array.



When a structure is declared as an array, add an index enclosed by '[ ]' after the structure label name.
A structure array can also be specified as an argument of a function or function block.

Ex.

Specifying an element of a structure declared as an array



Data types that can be specified

The following data types can be specified as structure members.

- Primitive data type
- Pointer type
- Array
- Other structures

Types of structures

Each of the following labels is predefined as a structure.

Type	Reference
Module label	🕮 Function Block Reference for the module used
Timer type	☞ Page 338 Data Types
Retentive timer type	
Counter type	
Long timer type	
Long retentive timer type	
Long counter type	

28.7 Label Access Setting from External Device

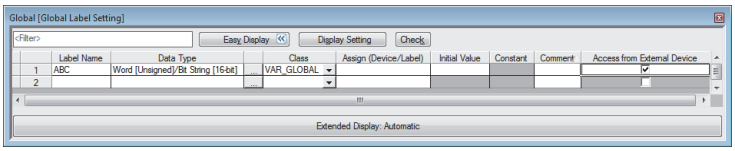
In the transfer setup of external devices that communicate with the CPU module through GOT, SLMP, or other methods, select "Control System/Standby System", "System A/System B", or "Not Specified". Then, configure the communication setting with a global label name specified.

Configuration procedure

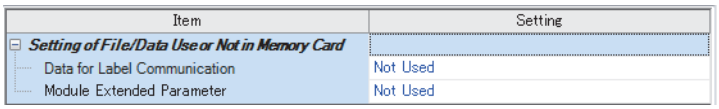
This section describes the configuration procedure to enable access by specifying the global label from external devices.

Operating procedure

"Global Label Setting" window



"Setting of File/Data Use or Not in Memory Card windows



1. Set the label in "Global Label Setting" and select the "Access from External Device" checkbox. *1
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.

*1 Safety global labels or standard/safety shared labels cannot be specified in the global label settings.

Point

Data storage to the data memory, not using the SD memory card, does not need the setting of procedure 3.

Label communication data storage location and operation at the time of each setting operation

This section describes the storage location for label communication data and the accessibility according to the setting of whether to use the file/data in a memory card.

○: Access from external device possible (Communication memory indicated in parentheses) ×: Access from external device not possible (Generates communication error)

Label communication data storage location	When "Enable" is set in the setting for the usage status of the file/data in a memory card	When "Disable" is set in the setting for the usage status of the file/data in a memory card
Data memory	×*1	○ (Data memory)
SD memory card	○ (SD memory card)	×
Data memory and SD memory card	○ (SD memory card)	○ (Data memory)

*1 An error occurs in the CPU module.

File operation

The label communication data can be written from an engineering tool. Reading and deleting*1 are not possible.

*1 Deleting an entire folder is possible.

28.8 Precautions

Functions with restrictions

The following functions have restrictions on the use of labels.

Item		Description
CPU parameter	• Trigger of an event execution type program	Use devices because global labels nor local labels cannot be specified for these functions.*1
Module parameter	• Predefined protocol support function	
	• Refresh setting of intelligent function module • Refresh setting of network module (SB/SW only)	Use module labels for these functions. Use devices if module labels are not used.*1
	• Refresh setting of network module (other than SB/SW)	Use devices because global labels nor local labels cannot be specified for these functions.*1

*1 Global labels can be used as devices by assigning a device.

■Defining and using a global label with a device assigned

Define a global label following the procedure below, and use it when the functions having restriction on the use of labels are executed.

Since the device area in the device/label memory is used, secure the device area capacity.

1. Secure the device area to be used.

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

2. Define a label as a global label, and assign a device manually.
3. Use the label defined in step 2 for the functions having no restrictions on the use of labels. Use the device assigned to the label for the function having restrictions on the use of labels.

■Copying the value of a label used to a different device

If a device cannot be assigned to a global label, copy the value of the label to a different device and use the device for the function having restrictions on the use of labels.

Since the device area in the device/label memory is used, secure the device area capacity.

1. Secure the device area to be used.

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

2. Create a program using the label.
3. Use the device where the data has been transferred in step 2 for the function having restrictions on the use of labels.

Point

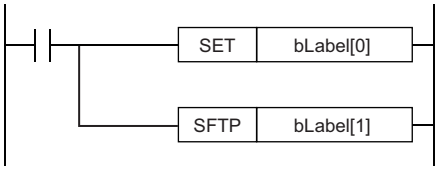
- The number of steps increases because of the transfer instruction. (The scan time increases.)
- Decide the transfer instruction position considering the timing of writing data to the label and executing the function.

Precautions for creating programs

When specifying a label as an operand used in instructions, match the data type of the label with that of the operand. In addition, when specifying a label as an operand used in instructions that control continuous data, specify the data range used in instructions within the data range of the label.

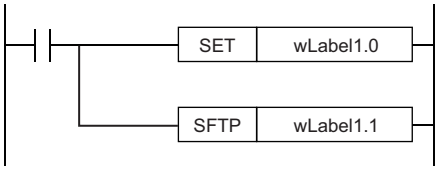
Ex.

SFT(P) instruction



To shift the bits correctly, specify the array of the bit type label.

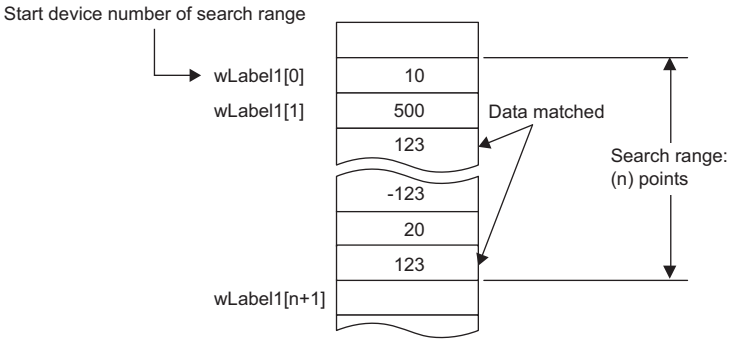
or



Specify the bit number of the word type label.

Ex.

SER(P) instruction



Specify a label which has a larger data range than the search range (n) points.

Restrictions on naming labels

The following restrictions apply when naming labels.

- Start the name with a character or underline (_). Numbers cannot be used at the beginning of label names.
- Reserved words cannot be used.

For details on the reserved words, refer to the following.

GX Works3 Operating Manual

Program of external devices

When the global label which is set to the "Access from External Device" is deleted or the online change including label name change is executed, change the program of external devices that refer to the global label deleted/changed.

29 SAFETY LABEL

This chapter describes the safety labels.

29.1 Safety Label

A label used in safety programs is called a safety label. Information not described in this section is same as that of standard labels. (☞ Page 336 STANDARD LABEL)

Safety label types

There are three safety label types. Only the following labels can be used in safety programs.

- Safety global label*1
- Standard/safety shared label*2
- Safety local label

*1 Safety devices can be assigned.

*2 This label can be used in both standard programs and standard function blocks.

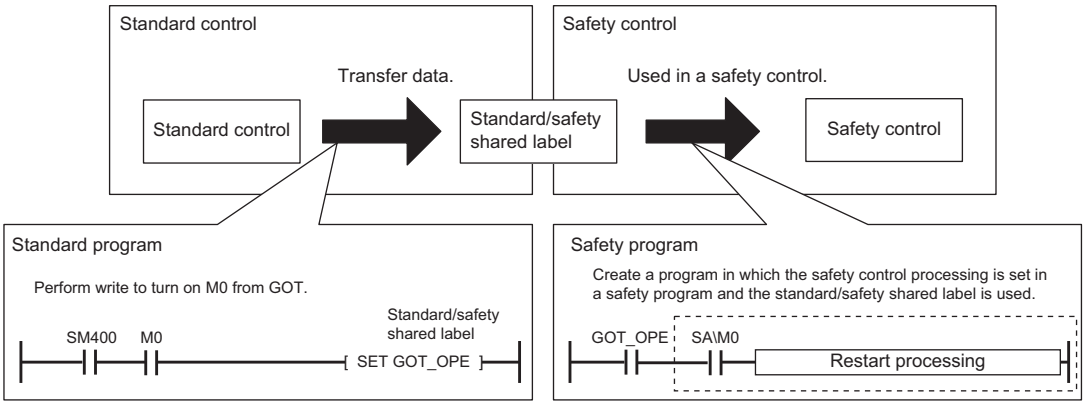
Restriction

- An initial value cannot be set to safety labels and standard/safety shared labels in the CPU parameter.
- To use aliases, a safety label can assign only safety labels as its original data location. Likewise, a standard/safety shared label can assign only standard/safety shared labels. A standard label cannot assign safety labels and standard/safety shared labels as its original data location. (GX Works3 Operating Manual)

How to use standard/safety shared labels

A standard/safety shared label is used to pass device data from a safety program to a standard program, and vice versa. When a standard/safety shared label is used in a safety program as shown in the examples below, the program needs to be created so that the safety state is secured.

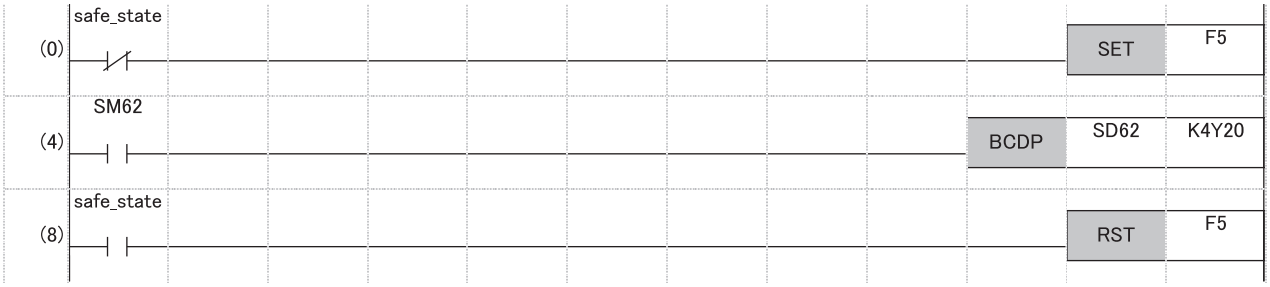
■To restart safety control by the command from the GOT



■To use the annunciator (F)

The safe state signal status can be controlled using the annunciator (F) in the standard program. The safe state signal status is passed from the safety program to the standard program via the standard/safety shared label (safe_state), and the status is controlled with the annunciator No.5. If an error is detected with the annunciator, the corresponding annunciator number is output to Y20.

• Standard program



(0) When the safe state signal turns off, the annunciator No.5 turns on.
(4) The annunciator number detected by SM62 (Annunciator) is output to Y20.
(8) When the safe state signal turns on, the annunciator No.5 turns off.

Classes

The following table lists the availability of the classes of safety global labels and standard/safety shared labels.

○: Applicable, ×: Not applicable

Class	Availability	
	Safety global label	Standard/safety shared label
VAR_GLOBAL	○	○
VAR_GLOBAL_CONSTANT	○	○
VAR_GLOBAL_RETAIN	×	×

The following table lists the availability of the classes of safety local labels.

○: Applicable, ×: Not applicable

Class	Availability		
	Safety program	Safety function	Safety function block
VAR	○	○	○
VAR_CONSTANT	○	○	○
VAR_RETAIN	×	×	×
VAR_INPUT	×	○	○
VAR_OUTPUT	×	○	○
VAR_OUTPUT_RETAIN	×	×	×
VAR_IN_OUT	×	×	○
VAR_PUBLIC	×	×	○
VAR_PUBLIC_RETAIN	×	×	×

Data types

Primitive data type

The following table lists the availability of primitive data types.

○: Applicable, ×: Not applicable

Data type		Availability
Bit	BOOL	○
Word [unsigned]/bit string [16 bits]	WORD	○
Double word [unsigned]/bit string [32 bits]	DWORD	○
Word [signed]	INT	○
Double word [signed]	DINT	○
Single-precision real number	REAL	×
Double-precision real number	LREAL	×
Time	TIME	○
String	STRING	×
String [Unicode]	WSTRING	×
Timer	TIMER	○
Retentive timer	RETENTIVETIMER	○
Long timer	LTIMER	×
Long retentive timer	LRETENTIVETIMER	×
Counter	COUNTER	○
Long counter	LCOUNTER	×
Pointer	POINTER	×

Structures

The structure definition is shared by standard programs and safety programs. However, it cannot be used in the following cases.

- A member of the primitive data type which cannot be used in safety programs exists.
- An initial value is set in the structure definition.

30 LATCH FUNCTION

30.1 Latch with Battery

The data in each device/label of the CPU module is cleared to its default value in the following cases:

- When the CPU module is powered off and on
- When the CPU module is reset
- When a power failure exceeding the allowable momentary power failure time occurs

The data in each device/label with latch setting is retained by using the battery on the CPU module unit even under the above described conditions. For this reason, control can be continued with the retained data even if the power of the CPU module is turned off or a power failure exceeding the allowable momentary power failure time occurs while the data is managed by a sequential control.

Types of latch

The following types of latch are available:

- Latch (1): Data in this range can be cleared by a latch clear operation
- Latch (2): Data in this range cannot be cleared by a latch clear operation

Applicable devices and labels

This section describes the devices/labels that can be latched.

Applicable devices

The devices below can be latched. However, local devices cannot.

Device	Specification method	Applicable latch type
Internal relay (M)	Specify the latch range.	Latch (1) or Latch (2)
Link relay (B)	Specify the latch range.	Latch (1) or Latch (2)
Annunciator (F)	Specify the latch range.	Latch (1) or Latch (2)
Edge relay (V)	Specify the latch range.	Latch (1) or Latch (2)
Timer (T)/Long timer (LT)/Retentive timer (ST)/ Long retentive timer (LST)	Specify the latch range.	Latch (1) or Latch (2)
Counter (C) and Long counter (LC)	Specify the latch range.	Latch (1) or Latch (2)
Data register (D)	Specify the latch range.	Latch (1) or Latch (2)
Link register (W)	Specify the latch range.	Latch (1) or Latch (2)
Latch relay (L)	Specify the number of points (latch is performed the same number of times as the specified number of points).	Latch (2) only
File register (R, ZR)	Specify the number of points (latch is performed the same number of times as the specified number of points). Specify the latch range.*1	Latch (2) only

*1 When "Use Common File Register in All Programs" is enabled, any of the number of points is registered within the range for Latch (2). Also, the range can be specified in "Latch (2)" of "Latch Range Setting" (values out of the specified range are cleared to 0).

Point

- When "Use File Register of Each Program" is enabled, the latch range for file registers cannot be specified (the whole range for file registers is retained).
- When the file register file which should be used is changed by the QDRSET instruction, the latch range setting for file registers is disabled. After the change, the whole range for file registers is retained regardless of the latch range setting.

Applicable labels

The labels below can be latched.

Label type	Class
Global label	VAR_GLOBAL_RETAIN
Local label of a program block	VAR_RETAIN
Local label of a function block	VAR_RETAIN
	VAR_OUTPUT_RETAIN
	VAR_PUBLIC_RETAIN

Setting latch on devices

Multiple latch ranges can be set for a device type. A total of 32 latch ranges between latch (1) and latch (2) can be set. However, the ranges of latch (1) and latch (2) must not overlap.

Setting a latch range

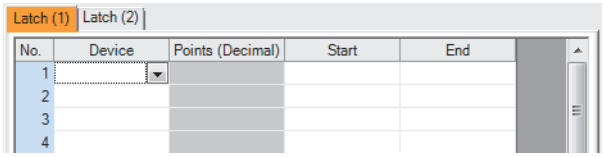
Set the device to latch, its range, and the latch type.

Operating procedure

"Device Setting" window

Item	Symbol	Latch (1)	Latch (2)
Input	*X*		
Output	Y		
Internal Relay	M	No Setting	No Setting
Link Relay	B	No Setting	No Setting
Link Special Relay	SB		
Annunciator	F	No Setting	No Setting
Edge Relay	V	No Setting	No Setting
Step Relay	S		
Timer	T	No Setting	No Setting
Long Timer	LT	No Setting	No Setting
Retentive Timer	ST	No Setting	No Setting
Long Retentive 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

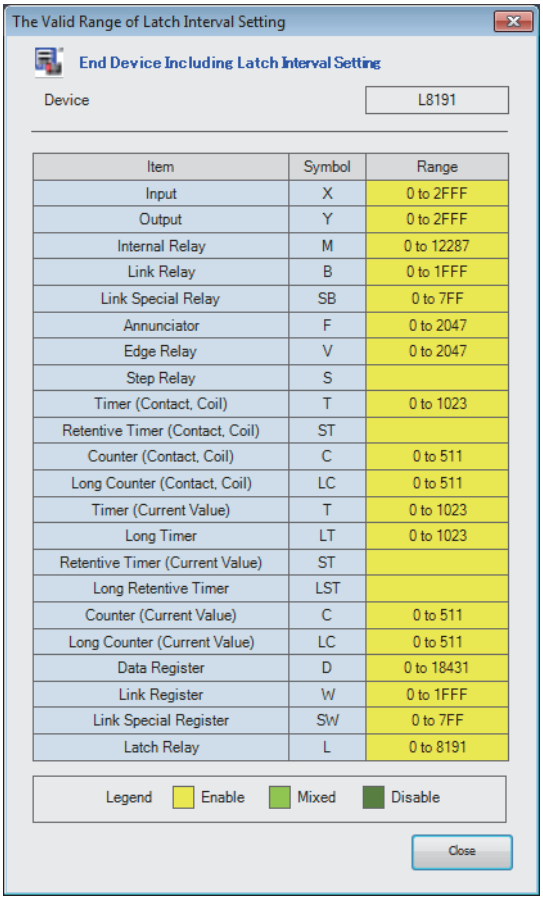
The user can specify the operation which should be performed at a latch interval (☞ Page 358 Timing of the latch processing) within the effective range of the latch interval setting[*1].

*1 The effective range of the latch interval means the range of devices which is enabled on the "The Valid Range of Latch Interval Setting" window. For other ranges (ranges of devices which are not enabled on the "The Valid Range of Latch Interval Setting" window), this setting is not applied, the END processing is accelerated, and latching is performed in real time.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device Latch Interval Setting]

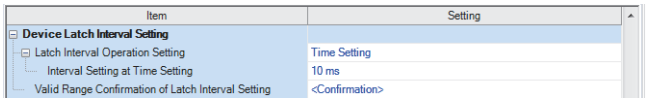
Operating procedure

"The Valid Range of Latch Interval Setting" window



1. On the "Valid Range Confirmation of Latch Interval Setting" window, check the effective range for the latch timing setting. Specify the latch interval operation within the effective range.

"Device Latch Interval Setting" window



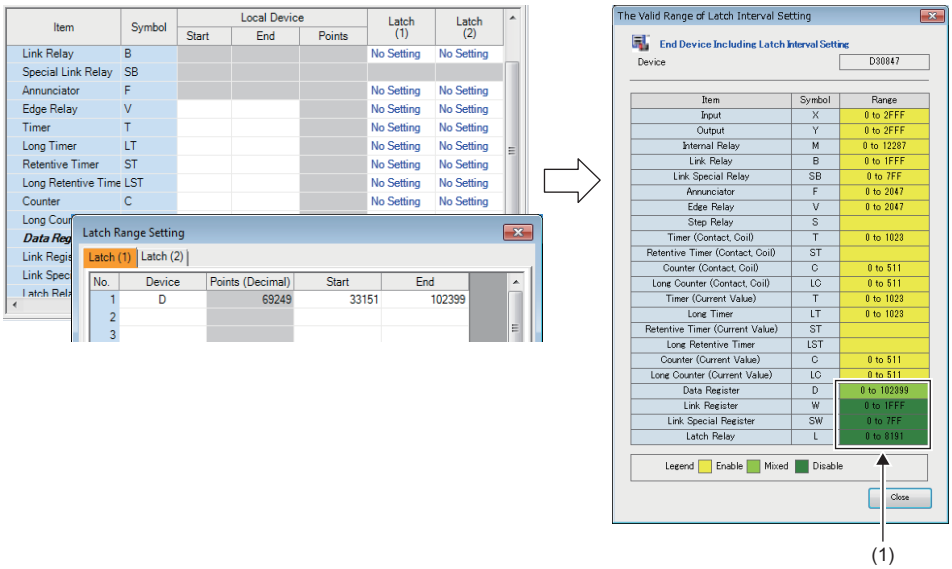
2. In "Latch Interval Operation Setting", select the timing of the latch processing. When "Time Setting" is selected, specify the time interval of latching.

Displayed items

Item	Description	Setting range	Default
Latch Interval Operation Setting	Sets the timing of the latch processing.	• Per Scan • Time Setting	Time Setting
Interval Setting at Time Setting	Sets the time interval of latching (when "Time Setting" is selected).	1 to 2000ms (in units of 1ms)	10ms
Valid Range Confirmation of Latch Interval Setting	Shows the effective range for the latch interval setting.	—	—

Point

For device latching, increasing the device range in the device setting of CPU parameters eliminates the latch processing from the END processing for the devices and enables real-time latching. For example, assume the following configuration for R08PSFCPU: (1) 0K word is specified for the file storage area and 168K words for the device area in "Device/Label Memory Area Capacity Setting", (2) 100K points is specified for the data register (D) on the "Device Setting" window, and (3) D41088 and later are specified as the latch target devices. In this configuration, D41089 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.



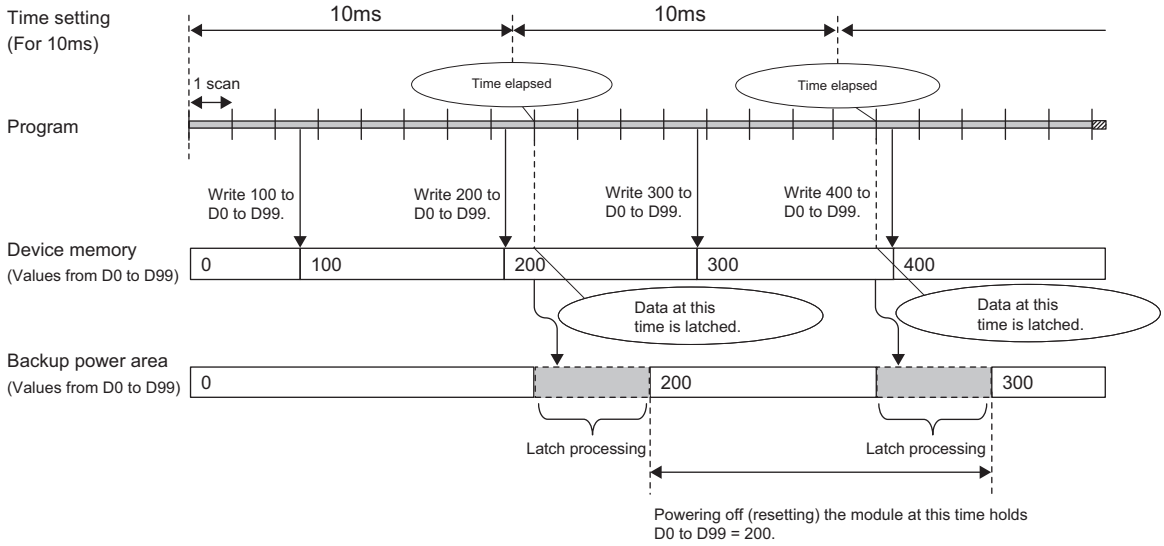
(1) D41089 or later areas of the data register (D) and areas of the devices (W, SW, and L) in disabled ranges can be latched in real time.

■Timing of the latch processing

The timing of the latch processing is determined based on the effective range of the latch interval setting and the operation setting for the specified latch interval (☞ Page 356 Setting the latch interval)

• When set to "Time Setting"

Latch processing is started in the END processing executed after the set time.

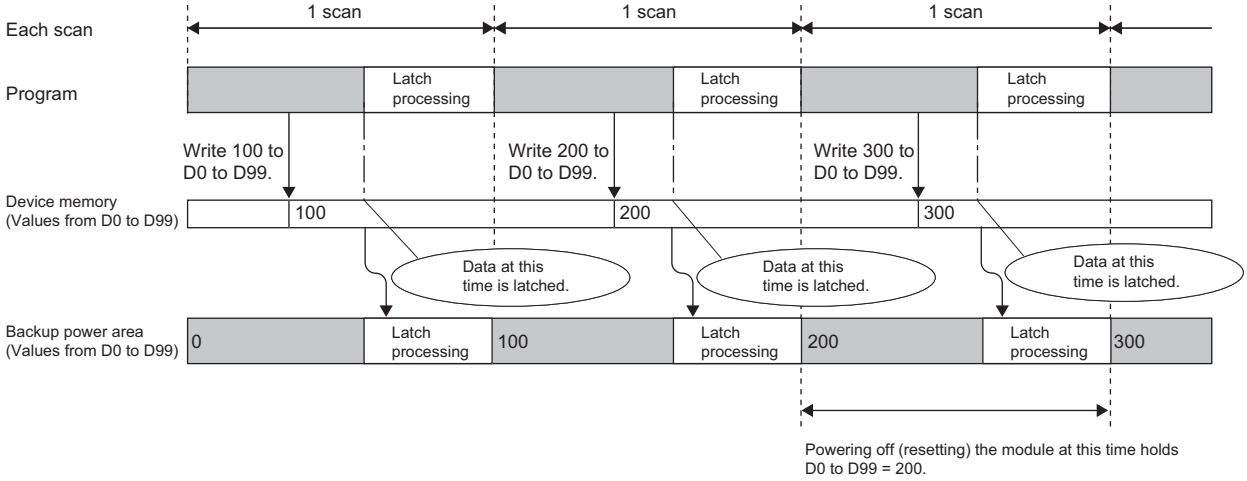


Point

- The more the latch points, the more it takes to complete the latch processing. If the set time elapses while latch processing is still executing, the next one is executed in the END processing that follows the completion of the previous latch processing. Set a latch interval longer than the latch processing time so that the specified time does not elapse during latch processing.
- If the latch interval is shorter than the scan time, latch timing occurs more than once within one scan. However, latch processing is executed once during the END processing.

• When set to "Per Scan"

Latch processing is executed in the END processing of every scan. This setting allows to always retain the device of the previous scan because the devices are latched at every scan. However, it may increase the scan time.

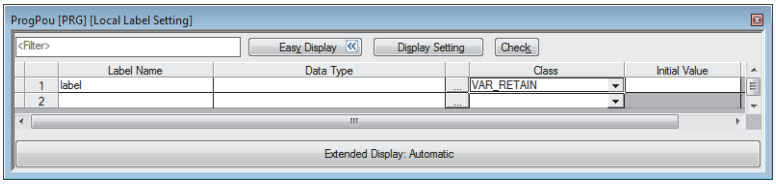


Setting latch on labels

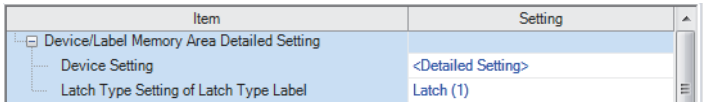
This section describes latch setting on labels.

Operating procedure

Label edit window



"Device/Label Memory Area Detailed Setting" window



1. In the label edit window, specify "RETAIN" for label attribute.

2. There are two types of latch for labels: latch (1) and latch (2). Select one. The selected latch type is applied to labels of all latch attributes.

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Device/Label Memory Area Detailed Setting] ⇨ [Latch Type Setting of Latch Type Label]

Clearing latch range data

Latch range data is cleared by either of the operations below. (☞ Page 157 Memory Operation)

- Latch clear: Performed from the engineering tool. (GX Works3 Operating Manual)
- Latch clear by program: Execute the RST instruction for latched devices, or clear by transferring K0 by using the MOV or FMOV instruction.

Precautions

This section describes the precautions when using the latch function.

- If the latch range and number of device points are modified by using parameters, latch is performed on the modified latch range. However, if the value of the parameter setting the latch range is different between the previous and current operations because the CPU module was powered off and on or is reset, and the latch range has been increased, the device range in the increased portion is not latched.

31 DEVICE/LABEL INITIAL VALUE SETTINGS

This function sets the initial values of devices and labels used in the program directly (not via the program) to the devices, labels, and buffer memory areas of intelligent function modules.

31.1 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.

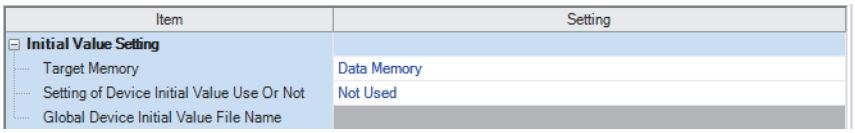
1. First, the user must create an initial device value file. To set initial values to a local device, create an initial device value file with the same name as the program which sets these values, and specify the range of the values. To set initial values to a global device, create an initial device 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 361 Initial value setting)
5. Write the set initial device value file and the CPU parameters to the CPU module.
(GX Works3 Operating Manual)
6. The data in the specified initial device value file is automatically set to the specified device or buffer memory of the intelligent function module when the CPU module is powered off and on, reset, or the status changes from STOP to RUN.

Initial value setting

Configure the initial value setting.

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

Window



Displayed items

Item	Description	Setting range	Default
Target Memory	Sets the storage memory for the initial device value file, initial global label value file, and initial local label value file.	• Data Memory • Memory Card	Data Memory
Setting of Device Initial Value Use Or Not	Selects whether or not to use initial device values.*2	• Not Used • Use	Not Used
Global Device Initial Value File Name	Sets the name of the initial global device value file.*1	60 characters or less	—

*1 If nothing is specified, initial global device values are not applied.

*2 If "Setting of Device Initial Value Use Or Not" is set to "Use" and the specified initial global device value file does not exist, an error occurs. If the other file (initial local device value file, initial global label value file, or initial local label value file) does not exist, an initial value is not applied, but an error does not occur.

Applicable range of initial device value files

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

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

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

Up to 1000 ranges can be set in one initial device value file. Up to 8000 data points can be set in one range.

Setting initial label values

This section describes the settings of initial label values. For labels assigned to a device, the initial value setting follows the initial device value.

Setting procedure

The procedure for using initial label values is as follows.

1. Set up initial label values using a global label and local label.
2. Configure CPU parameters (☞ Page 361 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.

31.2 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

31.3 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 label initial value are also set when the CPU module status changes from STOP to RUN.*1 Therefore, the initial device value and initial label value cannot be used for areas that are not desirable to be set when the CPU module status changes from STOP to RUN (because they should be set when the power is turned off and on, and changed by the program). By using the MOV instruction, for example, create a program to set an initial value to the specified device or label. In addition, if it is a module access device, make sure to write the initial value in the buffer memory by using the TO instruction.

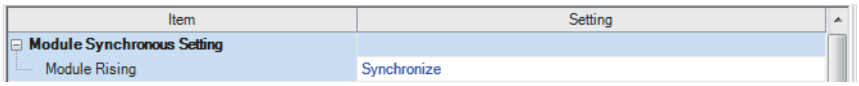
*1 For details on the initial label value setting, refer to the following.

Page 364 LABEL INITIALIZATION FUNCTION

- To make the CPU module hold a device only when its values fall into the range of device initial setting (such as module access device and link direct device), the start-up time between the device and the CPU module must be synchronized using the CPU parameter, "Module Synchronous Setting".

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Module Synchronous Setting]

Window



32 LABEL INITIALIZATION FUNCTION

The labels assigned to label areas will be initialized (Initial values are set if the values have been set, or the labels are cleared to zero if not) when the CPU module is powered off and on or the operating status of the CPU module is switched from STOP to RUN after data is rebuilt (reassigned) and then written to the programmable controller.

32.1 Initialization of Labels After Rebuilt All (Reassignment)

All the labels assigned to the label area will be initialized (initial values are set if the values have been set, or the labels are cleared to zero if not) when the CPU module is powered off and on or the operating status of the CPU module is switched from STOP to RUN after data is rebuilt (reassigned) with the engineering tool and then written to the programmable controller.

Point

Labels are initialized only when the data is written to the programmable controller for the first time after data is rebuilt (reassigned). Thus, to write the same project to the other CPU module, rebuild (reassign) the data again.

Label initialization operation

The following table lists the label initialization operations.

Target label		Label initial value setting	Label initialization operation*3	
			When the CPU module is powered on	When the operating status of the CPU module is switched from STOP to RUN*2
Standard label	Labels outside the latch range	Set	Initial values are set.	Initial values are set.
		Not set	Zero clear*1	Zero clear*1
	Latch type labels	Set	Initial values are set.	Initial values are set.
		Not set	Zero clear*1	Zero clear*1
Safety label and standard/safety shared label		Set	— (No setting)	— (No setting)
		Not set	Zero clear*1	Zero clear*1

*1 The labels (whose attributes are other than CONSTANT) to be assigned to label areas are cleared to zero.
*2 This operation is also performed when the operating status of the CPU module is switched from PAUSE to RUN.
*3 Since the labels assigned to devices are assigned to device areas, the labels are not initialized by this function.

Operating procedure

The following describes the operating procedure for this function.

1. Rebuild (reassign) all data.

[Convert] ⇨ [Rebuild All]

2. Set the CPU module to the STOP state.
3. Write the new program files. When initial values have been set in the labels used in the program, write the label initial value file together with the program files.

[Online] ⇨ [Write to PLC]

4. Set the CPU module to the RUN state. (Labels are initialized only when the operating state is changed from STOP to RUN for the first time.)
5. Labels are not initialized at the second state change or later. Label initial values can be set at the second state change or later depending on the parameter settings. (☞ Page 366 Label Initial Value Reflection Setting)

Point

Since this function automatically initializes labels (zero clear), the reset operation after writing is unnecessary.

Restriction

Labels are initialized only when the data is written to the programmable controller for the first time after data is rebuilt (reassigned). To write the same project to other programmable controller, rebuild (reassign) the data again before writing.

Precautions

The following describes precautions on this function.

- Even though SM326 (SFC device/label clear mode) is on (device/label values are kept), only labels are initialized when data is rebuilt and then written to the programmable controller. (Devices are kept.)
- Before boot operation, clear the value of the latch label. Even if the boot file after rebuilt all (reassignment) is written to the SD memory card and boot operation is executed, label initialization function does not operate.
- Once all data is rebuilt (reassigned) and written, the label initialization cannot be canceled. For example, when all data is written to the programmable controller without being rebuilt (reassigned) again after the data is rebuilt (reassigned) and written to the programmable controller, labels will be initialized after the CPU module is powered on or the operating status of the CPU module is switched from STOP to RUN.

32.2 Label Initial Value Reflection Setting

With the default settings, initial label values are not set in labels when the operating status of CPU module is switched from STOP to RUN even though the label initial values have been set for the labels.
This function can set whether or not to set label initial values when the operating status of the CPU module is switched from STOP to RUN.

Label initialization operation

The following table lists the label initialization operations.

Target label		Label initial value setting	Label initialization operation*4			
			When "Label Initial Value Reflection Setting" has been set to "Disable"		When "Label Initial Value Reflection Setting" has been set to "Enable"	
			When the CPU module is powered on	When the operating status of the CPU module is switched from STOP to RUN*3	When the CPU module is powered on	When the operating status of the CPU module is switched from STOP to RUN*3
Standard label	Labels outside the latch range	Set	Initial values are set.*1	Initial values are not set. (Values do not change.)	Initial values are set.	Initial values are set.
		Not set	Zero clear*2	Values do not change.	Zero clear*2	Values do not change.
	Latch type labels	Set	Initial values are not set. (Values do not change.)	Initial values are not set. (Values do not change.)	Initial values are set.	Initial values are set.
		Not set	Values do not change.	Values do not change.	Values do not change.	Values do not change.
Safety label and standard/safety shared label		Set	— (No setting)	— (No setting)	— (No setting)	— (No setting)
		Not set	Zero clear*2	Values do not change.	Zero clear*2	Values do not change.

*1 Since values in labels outside the latch range are cleared when the CPU module is powered off, initial values are set in the labels when the CPU module is powered on even though "Label Initial Value Reflection Setting" has been set to "Disable".
*2 The labels (whose attributes are other than CONSTANT) to be assigned to label areas are cleared to zero.
*3 This operation is also performed when the operating status of the CPU module is switched from PAUSE to RUN.
*4 Since the labels assigned to devices are assigned to device areas, the labels are not initialized by this function.

Setting procedure

The following describes the procedure of disabling label initial values to be set when the operating status of the CPU module is switched from STOP to RUN.

1. Check that "Label Initial Value Reflection Setting" has been set to "Disable". (☞ Page 367 Setting method)
2. Write data to the programmable controller.
3. Set the CPU module to the RUN state. (Label initial values are not set. *1)

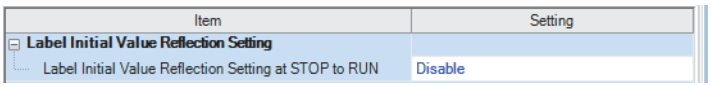
*1 Initial values are set when the CPU module is powered on or the operating status of the CPU module is switched from STOP to RUN for the first time after all data is rebuilt (reassigned) and written to the programmable controller.

Setting method

The following describes how to configure the label initial value reflection setting.

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

Window



Displayed items

Item	Description	Setting range	Default
Label Initial Value Reflection Setting at STOP to RUN	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".

33 CONSTANTS

This section describes constants.

33.1 Decimal Constant (K)

Use this type of constants when specifying decimal data in a program. Specify the decimal constant using K☐ character (e.g. K1234). The specification range depends on the argument data type of the instruction using the decimal constant as shown in the following table:

Argument data type of the instruction		Specification range for decimal constant
Data size	Data type name	
16 bit	Word (signed)	K-32768 to K32767
	Word (unsigned)/bit string (16 bit)	K0 to K65535
	16-bit data	K-32768 to K65535
32 bit	Double word (signed)	K-2147483648 to K2147483647
	Double word (unsigned)/bit string (32 bit)	K0 to K4294967295
	32-bit data	K-2147483648 to K4294967295

33.2 Hexadecimal Constant (H)

Use this type of constants when specifying hexadecimal data in a program. Specify the hexadecimal constant using H☐ character (e.g. H1234). To specify data in BCD, specify each digit of hexadecimal value with numerical characters 0 to 9. The specification range depends on the argument data type of the instruction using the hexadecimal constant. The range is H0 to HFFFF for 16-bit data size, and H0 to HFFFFFFFF for 32-bit data size.

33.3 Real Constant (E)

Use this type of constants when specifying a real number in a program. There are two types of real numbers: single-precision real number and double-precision real number. Specify it using E☐ character. (e.g. E1.234).

Setting range for real numbers

The setting ranges are different between the single-precision real number and double-precision real number.

Specification range for single-precision real numbers

The following is the specification range for single-precision real numbers.

$-2^{128} < \text{device} \leq -2^{-126}$, 0, $2^{-126} \leq \text{device} < 2^{128}$

(E-3.40282347+38 to E-1.17549435-38, 0, E1.17549435-38 to E3.40282347+38)

Specification range for double-precision real number

The following is the specification range for single-precision real numbers.

$-2^{1024} < \text{device} \leq -2^{-1022}$, 0, $2^{-1022} \leq \text{device} < 2^{1024}$

(E-1.7976931348623157+308 to E-2.2250738585072014-308, 0, E2.2250738585072014-308 to E1.7976931348623157+308)

Point

For the number of significant digits and input range of real number data input by the engineering tool, refer to the following.

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

Processing when operation is in progress

This section describes the processing when operation is in progress.

When overflow and/or underflow occurs:

If an overflow and/or underflow occurs when the operation is in progress, the following process is performed:

- When overflow occurs: An error is returned.
- When underflow occurs: 0 is returned (no error occurs).

When a special value is input:

When a special value*1 is used for the input data operation, an error is returned. Note that when "-0" is generated during the operation is in progress, it is handled as "+0", and therefore "-0" is not returned as an operation result.

*1 The special values include -0, denormalized number, nonnumeric number, and ±∞.

33.4 Character String Constant

The character string can be specified by enclosing it with single quotation marks (' ') or double quotation marks (" "). (Example: "ABCDE") Note that the NULL character (00H) becomes the termination character.

Point

For details on character string data, refer to the following.

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

33.5 Notation of Constants

This section describes the notation of constants.

Type		Notation	Example	Applicable data type
Boolean value		Set "FALSE" or "TRUE".	TRUE, FALSE	Bit
		Add "K" or "H" before "0" or "1".	K0, K1, H0, H1	
Integral	Binary	Add "2#" before a binary number.	2#0010, 2#01101010, 2#1111_1111	• Word [unsigned]/bit string [16 bits] • Double word [unsigned]/bit string [32 bits] • Word [signed] • Double word [signed]
	Octal	Add "8#" before an octal number.	8#0, 8#337, 8#1_1	
	Decimal	Directly enter a signed or unsigned decimal number.	123, +123, -123, 12_3	
		Add "K" before a signed or unsigned decimal number.	K123, K-123	
	Hexadecimal	Add "16#" before a hexadecimal number.	16#FF, 16#1_1	
		Add "H" before a hexadecimal number.	HFF, HEAD	
Real number	Decimal notation	Directly enter a signed or unsigned real decimal number with a decimal point.	2.34, +2.34, -2.34, 3.14_15	• Single-precision real number • Double-precision real number
		Add "E" before a signed or unsigned real decimal number with a decimal point.	E2.34, E-2.34	
	Exponential notation	Add "E" and a signed or unsigned exponent (decimal) at the end of the real number.	1.0E6, 1.0E-6	
		Add a signed exponent (decimal) at the end of the real number with "E".	E1.001+5, E1.001-6	
String	String	Enclose a string in single quotation marks (' ').	'ABC'	String
	String [Unicode]	Enclose a string in double quotation marks (" ").	"ABC"	String [Unicode]
Time		Add "T#" or "TIME#" at the beginning. (☞ Page 372 Notation of time)	T#1h, T#1d2h3m4s5ms, TIME#1h	Time

Point

In the notation of binary, octal, decimal, hexadecimal, and real numbers, the numbers can be separated using an underscore (_) to make programs easy to see. For example, the double word [unsigned] in the notation of binary is as follows.

2#1100_1100_1100_1100

The separations by underscores (_) are ignored in program processing.

Notation of time

In the notation of time, add "T#" or "TIME#" at the beginning of the value specified in units of time; d (day), h (hour), m (minute), s (second), and ms (millisecond). The following table lists the effective range for each unit of time.

Item	Effective range
d (day)	0 to 24
h (hour)	0 to 23
m (minute)	0 to 59
s (second)	0 to 59
ms (millisecond)	0 to 999

- Each unit of time can be omitted sequentially from the beginning to the end or in reverse order.
  T#31m23s, T#31m0s648ms
- A sign can be added only before the first unit of time.
  T#-31m23s
- An unsigned real decimal number with a decimal point can be used for the last unit of time. The values after the decimal point of ms (millisecond) are rounded down.
  T#-24d20h31m23s648.123ms (The time is treated as T#-24d20h31m23s648ms.)
  T#1.2345ms (The time is treated as T#1ms.)
- Only for the first unit of time other than d, the following values within the effective range can be input.
  h: 0 to 596
  m: 0 to 35791
  s: 0 to 2147483
  ms: 0 to 2147483647

Special character (escape sequence)

"$" is used as an escape sequence in a string. The following characters can be input as escape sequences with "$".

Symbol used in a string or printer code	Escape sequence
$	$$
'	$'
"	$"
Line feed	$L or $l
Line break	$N or $n
Page feed	$P or $p
Return	$R or $r
Tab	$T or $t
Two hexadecimal digits and characters that correspond to the ASCII code	Two hexadecimal digits following $

If the two hexadecimal digits following "$" do not correspond to the ASCII code, a conversion error occurs.

PART 8 TROUBLESHOOTING

This part consists of the following chapters.

34 TROUBLESHOOTING PROCEDURE

35 TROUBLESHOOTING BY SYMPTOM

36 MAINTENANCE AND INSPECTION

37 ERROR CODES

38 EVENT LIST

34 TROUBLESHOOTING PROCEDURE

This section describes errors that may occur during system operation, error causes, and actions to be taken.
For the troubleshooting specific to each module, refer to the manual for the module used.

Point

Saving the program and devices at the time of an error helps to analyze the error cause.
For reading data from the programmable controller, refer to the following.
GX Works3 Operating Manual

When the system has any trouble, perform troubleshooting in the following order.

1. Check the LED status of the power supply module.
(MELSEC iQ-R Module Configuration Manual)
2. Check the LED status of the CPU module. (Page 375 LED status of the CPU module)
3. Check the LED status of each I/O module and intelligent function module. (User's Manual (Application) for the module used)
4. Connect an engineering tool, and execute the system monitor function. The error module can be identified. (Page 376 System monitor)
5. Select the error module, and execute the module diagnostics function. The error cause and the action to be taken can be displayed. (Page 376 Module diagnostics)
6. If the error cause cannot be identified by the module diagnostics, check the executed operations and detected error logs on the event history window. (Page 376 Event history)
7. If the error cause cannot be identified in steps 1 to 6, perform troubleshooting by symptom. (User's Manual (Application) for the module used)

34.1 Troubleshooting with LED Indicators

Check the LED status of each module as primary diagnostics.

LED status of the CPU module

The following table lists the check points on the LED status of the CPU module.
The error status can be checked visually using the READY LED and ERROR LED.

LED indicator		Error status	Program execution status	Description	Action
READY	ERROR				
On	Off	No error	Continued	Normal operation	—
	On	Minor error		A minor error or warning has been detected. Program execution and data communications with network stations are continued.	Identify the error cause using the engineering tool, and take an action.
	Flashing	Moderate error	Stopped	A programming error, parameter setting error, or temporal noise has been detected. Program execution and data communications with network stations are stopped.	Identify the error cause using the engineering tool, and take an action.
Off	On/ flashing	Major error		A hardware failure has been detected. No operation cannot be performed.	Perform troubleshooting for the error module. If the problem still exists after troubleshooting, replace the error module. (☞ Page 377 When the READY LED of the CPU Module Turns Off)
	Off	Hardware failure		A hardware failure has been detected. No operation cannot be performed.	Check that the power is supplied to the system. If the power is supplied properly, the possible cause is a hardware failure. Perform troubleshooting for the error module. If the problem still exists after troubleshooting, replace the error module. (☞ Page 377 When the READY LED of the CPU Module Turns Off)
		No error		Power is not supplied or power went out.	—
Flashing	On	Minor error		A minor error has been detected during online module change.	Identify the error cause using the engineering tool, and take an action.
	Off	No error		• The READY LED is flashing every 400ms: Online module change is in progress. • The READY LED is flashing every 2s: Initial processing is being performed.	—
Flashing	Off	No error		The READY LED is flashing every 2s: Initial processing is being performed.	—

Point

The LED status can also be checked on the module diagnostics window using the engineering tool. (GX Works3 Operating Manual)

34.2 Troubleshooting Using the Engineering Tool

Check the error or history using the engineering tool, and identify the error cause. More detailed information on the error as well as the error cause and action to be taken can be checked by using the engineering tool.
The engineering tool has the following functions for troubleshooting.

Function	Description
System monitor	Displays the module configuration, and detailed information and error status of each module. (☞ Page 376 System monitor)
Module diagnostics	Diagnoses the module. (The current error and its details can be checked.) (☞ Page 376 Module diagnostics)
Event history	Displays the event information collected by the CPU module, such as errors occurred in each module, executed operations, and network errors. (☞ Page 376 Event history)

For details on each function, refer to the following.
📖GX Works3 Operating Manual

System monitor

This function displays the module configuration, and detailed information and error status of each module.
For the error module identified, the module diagnostics function can be executed from this window.

[Diagnostics] ⇨ [System Monitor]

Point

The systems A and B are both displayed on the main base information part (on the left side of the window). The display of system A and system B can be switched in the main base information.

Module diagnostics

This function diagnoses the module. (The current error and its details can be checked.)
Information required for troubleshooting, such as the current error, details and cause of the error, and action to be taken, are displayed. The incorrect parameter setting and program error location can also be identified by selecting the error and clicking the [Error Jump] button.
On the [Module Information List] tab, the LED status and the switch status of the target module can be checked.

[Diagnostics] ⇨ [System Monitor] ⇨ Double-click the error module.

Event history

This function displays the event information, such as errors occurred in the module(s) mounted, executed operations, and network errors.
Since information collected before the CPU module is powered off or reset can also be displayed, the error cause can be identified based on the past operations and errors.
The displayed information can also be saved in CSV file format.

[Diagnostics] ⇨ [System Monitor] ⇨ [Event History] button

Point

Use this function for the following purposes:

- To check the error status and identify the cause of error occurred in machinery or equipment
- To check when and how the program and parameters of the programmable controller have been changed
- To check for unauthorized access

For details on this function and collected information, refer to the following.
☞ Page 194 Event History Function

35 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.

35.1 When the POWER LED of the Power Supply Module Turns Off

For troubleshooting, refer to the following:

MELSEC iQ-R Module Configuration Manual

35.2 When the READY LED of the CPU Module Turns Off

Check the following:

Check item	Action
Check if the CPU module is mounted on the main base unit properly.	Remove the CPU module from the main base unit, and mount it back on the main base unit.
Check if the READY LED of another module is on.	If the READY LED of another module is on, the CPU module detected a major error. Replace the CPU module.
Check if the READY LED turns on when the power supply module is replaced and the power is restored to the system. (Check the LED status after the power supply module on the extension base unit is also replaced.)	If the READY LED turns on, an error has occurred in the power supply module(s) before the replacement. Replace the power supply module.
Check if the READY LED does not turn on even after the power supply module is replaced and the power is restored to the system. (Check the LED status after the power supply module on the extension base unit is also replaced.)	If the READY LED does not turn on, an error has occurred in a module other than the power supply module. Repeatedly supply power to the system, returning the modules back to the system one by one. The last module mounted immediately before the READY LED turned off has been failed. Replace the module.
Check if the system was powered on immediately after power-off.	Wait for five seconds or longer after power-off and power on the system again.

If the READY LED of the CPU module does not turn on even after the items above are checked and the actions are taken, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.

35.3 When an Error Has Occurred in a Redundant Function Module

When the RUN LED turns off

When the RUN LED turns off after the redundant function module is powered on, check the following.

Check item	Action
Check if the redundant function module has been properly mounted.	If not, properly mount the module on the base unit.

If the RUN LED of the redundant function module still does not turn on even after the above action is taken, the possible cause is a hardware failure of the module. Please consult your local Mitsubishi representative.

When the ERR LED turns on or flashes

Check the following:

Check item	Action
Check if an error has occurred by performing the module diagnostics.	Take actions proposed by the module diagnostics.

If the ERR LED still turns on or flashes even after the above action is taken, perform a module communication test and check the module for a hardware failure. (☞ Page 261 Redundant Function Module Communication Test)

When the L ERR LED turns on

Check the following:

Check item	Action
Check if the tracking cables used are proper.	• Check if the tracking cables satisfy the standards. (☞ Page 69 Redundant Function Module) • Check that the length of each tracking cable is within the range of the specifications. (☞ Page 69 Redundant Function Module) • Check if no disconnection is found on each tracking cable.
Check if each tracking cable has been properly connected.	Check the connection status of each tracking cable by performing the module diagnostics. If a tracking cable has not been properly connected, connect it again.

If the L ERR LED still turns on even after the above actions are taken, perform a module communication test and check the redundant function module and tracking cables for failures. (☞ Page 261 Redundant Function Module Communication Test)

35.4 When an Error Has Occurred in a SIL2 Function Module

When the READY LED turns off

When the READY LED of the SIL2 function module turns off, check the following.

Check item	Action
Check if the SIL2 function module has been properly mounted.	If not, properly mount the module on the base unit.
Check if the system was powered on immediately after power-off.	Wait for five seconds or longer after power-off and power on the system again.

If the READY LED of the SIL2 function module still does not turn on even after the above actions are taken, the possible cause is a hardware failure of the module. Please consult your local Mitsubishi representative.

35.5 When Data Cannot Be Written to the Programmable Controller

Check the following:

Check item	Action
Check if password is not registered.	Authenticate a password using the engineering tool.
Check if the SD memory card is write-protected while the SD memory card is being targeted for writing.	Clear the write-protect for the SD memory card.
When the target of the write is an SD memory card, check that the SD memory card has been formatted.	Carry out formatting for the SD memory card.
Check if the data to be written exceeds the capacity of CPU built-in memory or SD memory card.	Check the CPU built-in memory or SD memory card for their free space.
Check if a password has been set for the programmable controller.	Write user information to the programmable controller from the engineering tool.
Check if the user has logged on to the programmable controller.	Log on to the programmable controller from the engineering tool.

If data cannot be written to the programmable controller even after the items above are checked and the actions are taken, the possible cause is a hardware failure. Please consult your local Mitsubishi representative.

35.6 When Data Cannot Be Read from the Programmable Controller

Check the following:

Check item	Action
Check if password is not registered.	Authenticate a password using the engineering tool.
Check if the target memory to be read is correctly specified.	Check the read target memory (CPU built-in memory, SD memory card, or intelligent function module).
Check if a password has been set for the programmable controller.	Write user information to the programmable controller from the engineering tool.
Check if the user has logged on to the programmable controller.	Log on to the programmable controller from the engineering tool.
Check if program restoration information has not been written.	When program restoration information has not been written, data cannot be read from the programmable controller. When writing data to the programmable controller, write program restoration information.

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.

35.7 When the Operating Status of the CPU Module Cannot Be Changed

Check the following:

Check item	Action
Check if a stop error has been occurred.	Check the error cause using an engineering tool and eliminate the error cause.
Check if the online change processing is left suspended.	Execute the online change again.

If the CPU module operating status cannot be changed even after the items above are checked and the actions are taken, the possible cause is a hardware failure. Please consult your local Mitsubishi representative.

35.8 When the Ethernet Function Cannot Be Used

For troubleshooting, refer to the following:

MELSEC iQ-R Ethernet User's Manual (Application)

35.9 When Cyclic Data Is Turned Off at the Time of System Switching

When cyclic data is turned off or temporarily turned off at the time of system switching, check the following.

Check item	Action
Check if the own station send range is the tracking range.	Check that the cyclic data in the send range is set to be transferred. (☞ Page 231 Tracking transfer setting)
Check if the interlock used in the program has no errors.	Take one of the following measures. ■Modify the program so that the interlock is triggered using the following labels. (MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)) • 'Own station data link error status' (SB0049) • 'Own station master/sub master function operating status' (SB004E) • 'Each station redundant system data link error status' (SB01B0) • 'Redundant system master station data link error status' (SB01B1) • 'Redundant system station number 0 data link error status' (SB01BF) • 'Each station redundant system data link status' (SW01B0 to SW01B7) ■Check that "Setting to Wait Receiving Cyclic Data after Switching System" of the CPU parameter is enabled. If the setting is disabled, enable it. (☞ Page 260 Setting to wait cyclic data receive after system switching)

36 MAINTENANCE AND INSPECTION

This chapter describes the maintenance and inspection.

36.1 Module Replacement

This section describes the methods of replacing modules while the system is operating.

- Replacing modules in a control system with the online module change function
- Replacing modules in a standby system (To replace modules in a standby system, power off the standby system or use the online module change function.)

○: Replaceable, ×: Not replaceable

Replacement target	Replacement		Reference
	Control system	Standby system	
CPU module and SIL2 function module*4	×	○	Page 382 Replacing a CPU module and SIL2 function module
Power supply module	×	○	Page 382 Replacing a power supply module
Redundant power supply module	○*1	○	Page 383 Replacing a redundant power supply module
Redundant function module	○*3	○*2	Page 383 Replacing a redundant function module
I/O module	○	○	Page 383 Replacing I/O modules
Intelligent function module	○*5*6	○*5	Page 384 Replacing an intelligent function module
Main base unit	×	○	Page 385 Replacing a main base unit

*1 This module in a redundant power supply system can be replaced.

*2 To replace this module without powering off the system, use the online module change function. (MELSEC iQ-R Online Module Change Manual)

*3 When an error has been detected on a redundant function module in the control system, a continuation error occurs on the CPU module, and the control system and the standby system continue operating without being switched. Perform online module change if the redundant function module has failed. (MELSEC iQ-R Online Module Change Manual)

*4 Since the CPU module and SIL2 function module are used as a pair, replace them at the same time.

*5 A module supporting the online module change function can be replaced by online module change. (MELSEC iQ-R Online Module Change Manual)

*6 When replacing a module not supporting the online module change function, the module can be replaced by turning on SM1646 (System switching by a user) to execute system switching and powering off the main base unit in the standby system after system switching.

Point

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 190 Error Clear)

The following describes how to replace modules in the standby system (by powering off the standby system).

Replacing a CPU module and SIL2 function module

This section describes the procedure for replacing a CPU module and SIL2 function module.

1. Check the system of the replacement target CPU module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target CPU module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED is on on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes.

3. Replace the CPU module and SIL2 function module in the standby system.

Replace the CPU module and SIL2 function module in the standby system with a new CPU module and SIL2 function module with the same models as those of the CPU module and SIL2 function module in the control system. The pair versions of the CPU module and SIL2 function must be the same. If an SD memory card or an extended SRAM cassette has been inserted to the replacement target CPU module, insert it to the new CPU module.

4. Change the switch status of the CPU module in the standby system.

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

5. Power on the standby system.

Power on the standby system.

6. Execute the memory copy from the control system to the standby system.

Execute the memory copy using the engineering tool. (☞ Page 242 Memory Copy from Control System to Standby System)
When the memory copy is completed, the MEMORY COPY LED of the redundant function module in the control system turns off and that in the standby system turns on.

7. Restart the standby system.

Restart the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on.

8. Check the safety operation mode.

Change the safety operation mode of the CPU module in the standby system to SAFETY MODE. (☞ Page 262 SWITCHING SAFETY OPERATION MODE)

Replacing a power supply module

This section describes the procedure for replacing the power supply module.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED is on on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes.

3. Replace the power supply module in the standby system.

Replace the power supply module in the standby system.

4. Power on the standby system.

Check if wiring to the power supply module and the power supply voltage are correct, and power on the standby system.
When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on.

Replacing a redundant power supply module

Select the standby system or control system and power off and replace the redundant power supply module in the system. After that, power off and replace the module in the other system. Since the redundant power supply module that is not the replacement target supplies power to the modules on the base unit, controls can be continuously performed during replacement of the other redundant power supply module.
For the replacement procedure, refer to the following.
MELSEC iQ-R Module Configuration Manual

Replacing a redundant function module

This section describes the procedure for replacing the redundant function module.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED is on on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes.

3. Replace the redundant function module in the standby system.

Replace the redundant function module in the standby system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on.

5. Check the modules for errors.

Check that no error has occurred on the CPU module or redundant function module in the standby system. If the ERROR LED of either of the modules is on or flashes, check the cause of an error and eliminate the error cause.

Replacing I/O modules

The following describes the procedure for replacing the I/O module on the main base unit of the standby system by turning the power off while the redundant system is operating. When the replacement target is in the control system, turn on SM1646 (System switching by a user) in advance to execute system switching before switching to the standby system. To replace this module without turning the power off while the system is operating, use the online module change function. (MELSEC iQ-R Online Module Change Manual)

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace the I/O modules in the standby system.

Replace the I/O modules in the standby system with new I/O modules whose models are the same as the ones of the I/O modules in the control system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

Replacing an intelligent function module

The following describes the procedure for replacing the intelligent function module on the main base unit of the standby system by turning the power off while the redundant system is operating. When the replacement target is in the control system, turn on SM1646 (System switching by a user) in advance to execute system switching before switching to the standby system.

To replace this module without turning the power off while the system is operating, use the online module change function. (📖 MELSEC iQ-R Online Module Change Manual)

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED flashes on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes. (The control system will wait for the start-up of the other system.)

3. Replace an intelligent function module in the standby system.

Disconnect the cables such as the communication cable connected to the intelligent function module in the standby system. Replace the intelligent function module with an intelligent function module whose model is the same as that of the intelligent function module in the control system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on. (The control system will not wait for the start-up of the other system.)

5. Perform module diagnostics on the standby system.

Check with error information in the module diagnostics that no error has occurred. In addition, when the network module is replaced, check the network status with the network diagnostics.

Procedure for remote head module replacement

For details on the procedure to replace the remote head modules, refer to the following.

📖 MELSEC iQ-R CC-Link IE Field Network Remote Head Module User's Manual (Application)

Replacing a main base unit

This section describes the procedure for replacing the main base unit.

1. Check the system of the replacement target power supply module.

Check the CTRL LED and SBY LED of the redundant function module in the system of the replacement target module for the following. The target module is in the standby system in this case.

- The CTRL LED is off and the SBY LED is on on the redundant function module of the system of the replacement target CPU module.

2. Power off the standby system.

Power off the standby system. When the standby system is powered off, the BACKUP LED of the redundant function module in the control system flashes.

3. Replace the main base unit in the standby system.

Replace the main base unit in the standby system with a new main base unit whose model is the same as that of the main base unit in the control system. When mounting modules to the new base unit, mount each module to the slots of the unit in the same order as that of the base unit of the control system.

4. Power on the standby system.

Power on the standby system. When the standby system is powered on, the BACKUP LED of the redundant function module in the control system turns on.

5. Check the modules for errors.

Check that no error has occurred on the CPU module or redundant function module in the standby system. If the ERROR LED of either of the modules is on or flashes, check the cause of an error and eliminate the error cause.

37 ERROR CODES

The CPU module stores the corresponding error code in the special register (SD) upon detection of an error by the self-diagnostic function. If an error occurs when the data communications are requested from the engineering tool, intelligent function module, or network system connected, the CPU module returns the corresponding error code to the request source. The error details and cause can be identified by checking the error code. The error code can be checked in either of the following ways:

- Module diagnostics of the engineering tool (GX Works3 Operating Manual)
- Special register (SD0 (Latest self-diagnostics error code), SD10 to SD25 (Self-diagnostic error number)) (☞ Page 493 List of Special Register Areas)

This section describes errors that may occur in the CPU module and actions to be taken for the errors.

37.1 Error Code System

All error codes are given in hexadecimal format (4 digits) (16-bit unsigned integer). The type of error includes the error, which is detected through the self-diagnostic function of each module, and the common error, which is detected during data communications between modules. The following table lists the error detection type and the error code ranges.

Error detection type		Range	Description
By the self-diagnostic function of each module	Minor error	1000H to 1FFFH	Error code specific to each module, such as self-diagnostic errors
	Moderate error	2000H to 3BFFH	
	Major error	3C00H to 3FFFH	
During data communications between modules		4000H to 4FFFH	Error in the CPU module
		6F00H to 6FFFH	Error in the redundant function module
		7000H to 7FFFH	Error in the serial communication module
		B000H to BFFFH	Error in the CC-Link module
		C000H to CFBFH	Error in the Ethernet-equipped module
		D000H to DFFFH	Error in the CC-Link IE Field Network module
		E000H to EFFFH	Error in the CC-Link IE Controller Network module

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 493 List of Special Register Areas)

Detailed information	Item	Description
Detailed information 1	Error location information*1	Information on the location in a program, such as step numbers
	Drive/file information	Information on the corresponding drive name and file name
	Parameter information	Information on the parameter, such as parameter storage location and parameter type
	System configuration information	Information on the system configuration, such as I/O numbers and power supply numbers
	Safety station system configuration information	Information on the own station or other stations at a safety communication error
	Frequency information	Information on the frequency, such as the number of writes to memory
	Time information	Information on time
	TEST MODE continuous RUN prevention setting	Continuous RUN allowable time in TEST MODE
	Failure information	Information on failures
	System switching information	Information on the system switching cause or system switching failure cause
	Data type (tracking transfer) information	Information on tracking transfer setting data
	Tracking transfer trigger information	Information on tracking block numbers when data are transferred
Detailed information 2	Drive/file information	Information on the corresponding drive name and file name
	Annunciator information	Information on the annunciator areas
	Parameter information	Information on the parameter, such as parameter storage location and parameter type
	System configuration information	Information on the system configuration, such as I/O numbers and power supply numbers
	Process control instruction processing information	Information on processing blocks of the process control instructions
	Program error information	Program error code

*1 The step number which is displayed in the error location information, is the one that is counted from the start of the file. It may differ from the step number of the program which is displayed in error jump of the engineering tool.

*2 For details on error classifications and error codes, refer to the manual of each device connected. Note that if 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.

37.2 Operation When an Error Occurs

There are two types of errors: continuation errors and stop errors.

Stop error

If a stop error occurs, the CPU module stops its operation and the operating status changes to STOP. Modules can communicate with the CPU module even after the stop error occurs in the CPU module. The external output of each module is controlled in accordance with the output mode setting in error. (☞ Page 186 CPU module operation upon error detection setting)

Continuation error

If a continuation error occurs, the CPU module continues its operation. (The operating status remains the same.)

37.3 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.

37.4 How to Clear Errors

Continuation errors can be cleared. For the error clear method, refer to the following.

☞ Page 190 Error Clear

Also, errors in the standby system can be cleared from the CPU module of the control system by using SM1679 (Error reset (the other system)) in a program or external device. (☞ Page 193 Clearing Errors in the CPU Module of the Standby System from the CPU Module of the Control System)

37.5 List of Error Codes

Codes of errors detected by the self-diagnostic function (1000H to 3FFFH)

The following table lists error codes detected by the self-diagnostic function.

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1000H	Power shutoff	• A momentary power failure has occurred. • The power supply has been shut off.	Continue	Check the power supply status.	—	Always
1010H	Power shutoff (either of the redundant power supply modules)	Power has been shut off or the power supply voltage has dropped in either of the redundant power supply modules on the redundant power supply base unit. Or, only one redundant power supply module is mounted.	Continue	• Check the power supplied to the power supply modules on the base unit. • Check that two power supply modules are mounted on the base unit. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.	System configuration information	Always
1020H	Failure (either of the redundant power supply modules)	A failure has been detected in either of the redundant power supply modules on the redundant power supply base unit.	Continue	The possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.	System configuration information	Always
1030H	Invalid power supply module	An invalid power supply module has been mounted on the redundant power supply base unit.	Continue	Mount only applicable power supply modules. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.	System configuration information	Always
1031H	Power supply module configuration error	A power supply module other than the redundant power supply module has been mounted on the redundant power supply base unit.	Continue	Mount only applicable power supply modules. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.	System configuration information	Always
1080H	ROM write count error	■CPU module • The number of writes to the flash ROM (data memory, program memory, and system memory (memory used by the system when the CPU module executes its function)) exceeded 100000 times. • Files are frequently written by the event history function. ■SIL2 function module The number of writes to the flash ROM exceeded 100000 times. (Number of writes > 100000)	Continue	■CPU module Replace the CPU module. ■SIL2 function module Replace the SIL2 function module.	Frequency information	At power-on, at RESET, at write
1090H	Battery error	• The voltage of the battery built in the CPU module has dropped below the specified value. • The connector of the battery built in the CPU module is disconnected. • The connector of the battery built in the CPU module is not securely connected.	Continue	• Replace the battery. • Connect the battery connector to use the file storage area in the device/label memory or the latch function. • Check the connection status of the battery connector. If it is loose, securely connect the connector.	—	Always
1100H	Memory card access error	Data cannot be written to the memory card because the write protect switch is locked.	Continue	Unlock the write protect switch of the memory card.	—	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1120H	SNTP clock setting error	Time setting has failed when the CPU module is powered on or reset.	Continue	• Check if the time is correctly set in parameter. • Check if the specified SNTP server is operating normally and there is no failure on the network accessing the SNTP server computer.	—	At power-on, at RESET
1124H	Default gateway/ gateway IP address error	• The default gateway is not set correctly. • The gateway IP address is not set correctly. • The default gateway/gateway IP address (network address after the subnet mask) is different from that of the IP address of the own node.	Continue	• Correct the default gateway IP address. • Set the same network address as that of the IP address.	Parameter information	Always
1128H	Own node port number error	The port number is incorrect.	Continue	Correct the port number.	—	Always
1129H	Open specification port number error	The port number setting of the external device is incorrect.	Continue	Correct the port number of the external device.	—	Always
112DH	Specified IP address error	The IP address setting of the external device for the open processing is incorrect.	Continue	• Correct the IP addresses. • Check if the class of the IP address is set to A/B/C.	—	Always
112EH	Connection establishment failed	A connection could not be established in the open processing.	Continue	• Check the operation of the external device. • Check if the open processing has been performed in the external device. • Check and correct the port number of the module, IP address/port number of the external device, and opening method. • When the firewall is set in the external device, check if the access is permitted. • Check if the Ethernet cable is disconnected.	—	Always
1133H	Socket communications response send error	The response send failed during socket communications.	Continue	• Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the Ethernet communication load is high because of the Ethernet functions being executed. Use the Ethernet functions so that the communication load is reduced. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub.	—	Always
1134H	TCP connection timeout	A TCP ULP timeout error has occurred in the TCP/IP communication. (The external device does not send an ACK response.)	Continue	• Check the operation of the external device. • Check and correct the TCP ULP timeout value. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the Ethernet communication load is high because of the Ethernet functions being executed. Use the Ethernet functions so that the communication load is reduced. • Check if the connection cable is disconnected.	—	Always
1152H	IP address error	The IP address is not set correctly.	Continue	Correct the IP addresses.	Parameter information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1155H	Connection number acquisition error	• The specified connection was already closed in TCP/IP communications. • Open processing is not performed.	Continue	• Perform the open processing for the specified connection. • Check if the open processing has been performed in the external device.	—	Always
1157H	Receive buffer securement error	• The specified connection was already closed in UDP/IP communications. • Open processing is not performed.	Continue	• Perform the open processing for the specified connection. • Check if the open processing has been performed in the external device.	—	Always
1165H	UDP/IP send failed	Data was not sent correctly with UDP/IP.	Continue	• Check the settings for connection with the external device. • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the Ethernet communication load is high because of the Ethernet functions being executed. Use the Ethernet functions so that the communication load is reduced. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute a PING test. If the test completes with an error, take an action to correct the error.	—	Always
1166H	TCP/IP send failed	Data was not sent correctly with TCP/IP.	Continue	• Check the settings for connection with the external device. • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the Ethernet communication load is high because of the Ethernet functions being executed. Use the Ethernet functions so that the communication load is reduced. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute a PING test. If the test completes with an error, take an action to correct the error.	—	Always
1167H	Unsend data send error	Unsent data found, but could not be sent.	Continue	• Check the settings for connection with the external device. • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the Ethernet communication load is high because of the Ethernet functions being executed. Use the Ethernet functions so that the communication load is reduced. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute a PING test. If the test completes with an error, take an action to correct the error.	—	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1180H	Redundant IP address error	• The same IP address has been set as the system A IP address, system B IP address, and/or control system IP address • Network addresses of the system A IP address, system B IP address, and control system IP address are different.	Continue	• Set different IP addresses for the system A IP address, system B IP address, and control system IP address. • Set the same network address for the system A IP address, system B IP address, and control system IP address.	Parameter information	At power-on, at RESET
1200H	Module moderate error	A moderate error has been notified from the intelligent function module connected.	Stop/ continue	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, identify the error module, and eliminate the error cause. • 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
1220H	Another CPU module moderate error	A moderate error has been notified from another CPU module.	Stop/ continue		System configuration information	Always
1800H	Annunciator ON	The annunciator (F) on status has been detected.	Continue	Check the detailed error information (annunciator information) by executing module diagnostics using the engineering tool, and correct the program corresponding to the displayed annunciator number.	Error location information and annunciator information	At instruction execution
1810H	Operation continuation error	The PALERT(P) instruction was executed.	Continue	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool and check the corresponding program - one that outputs the program error code.	Error location information and program error information	At instruction execution
1830H	Receive queue full	The number of transient receive request exceeded the upper limit of simultaneously processable requests.	Continue	• Reduce the frequency of transient transmission, and perform transmissions again. • Increase the frequency of transient transmission by using the COM instruction.	—	Always
1832H	Transient data error	Too much transient transmission processing exists and transient transmission cannot be performed.	Continue	Correct the number of transient transmission executions.	—	Always
1900H	Constant scan time error	The scan time exceeded the constant scan time set in the CPU parameters.	Continue	Check and correct the constant scan time setting.	Time information	At END instruction execution
1A00H	Safety cycle time exceeded	Processing of the safety program, from safety input to safety output, did not complete within the safety cycle time.	Continue	Check and correct the safety cycle time.	Time information	At interrupt occurrence
1A01H	Safety cycle processing error	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.			—	

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1A40H	Destination station error in safety communications	• Error information has been received from a safety communication destination station. • The safety protocol version of the device actually connected to perform safety communications and the version set in the safety communication setting do not match.	Continue	• Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check "Station No.", "Slot No. (another station)", and "Base No. (another station)". If the destination station is an intelligent device station, execute the system monitor for the remote head module in the station and check the error details. • When the destination station does not detect an error, the safety protocol version that is not supported by the station is specified in the safety communication setting. Specify the safety protocol version that is supported by the safety communication destination station. Write parameters to the CPU module.	Safety station system configuration information, error information of other stations (CC-Link IE TSN/CC-Link IE Field)	At interrupt occurrence
1A50H	Parameter mismatch in safety communication destination station	• The safety approvals codes written to the CPU module and to the safety communication destination station do not match. • The parameters written to the CPU module and to the safety communication destination station do not match.	Continue	Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check the displayed station number. Check the safety communication setting method and procedure referring to the manual for the safety communication destination station, and write parameters to the safety communication destination station and the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the module performing safety communications. Please consult your local Mitsubishi representative.	Safety station system configuration information	At interrupt occurrence
1A51H	Device mismatch in safety communication destination station	• The model or production information on the safety communication destination station is incorrect. • The device actually connected to perform safety communications and the device set in the network configuration settings do not match.	Continue	• Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check the displayed station number. Then, match the connected device and the device set in the network configuration settings. • Write parameters to the safety communication destination station and the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the module performing safety communications. Please consult your local Mitsubishi representative.	Safety station system configuration information	At interrupt occurrence

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1A52H	Device version mismatch in safety communication destination station	The version of the device actually connected to perform safety communications and the version of the device set in the network configuration settings do not match.	Continue	• Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check the displayed station number. Then, match the version of the connected device and the version set in the network configuration settings or the profile data. • Write parameters to the safety communication destination station and the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the module performing safety communications. Please consult your local Mitsubishi representative.	Safety station system configuration information	At interrupt occurrence
1A53H	Safety protocol version mismatch in safety communication destination station	The safety protocol version of the device actually connected to perform safety communications and the version set in the safety communication setting do not match.	Continue	• Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check the displayed station number. Then, match the safety protocol versions between the safety communication setting with the error and the safety communication setting of the connected device. • Write parameters 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 or the module performing safety communications. Please consult your local Mitsubishi representative.	Safety station system configuration information	At interrupt occurrence
1A60H, 1A61H	Timeout in safety communications	A timeout error occurred during safety communications.	Continue	• Check and correct the safety refresh monitoring time and the transmission interval monitoring time setting values. • The possible cause is malfunction due to noise or cable failure. Check that there is no error in a transmission path by executing CC-Link IE Field diagnostics. Check the distance of wires and cables and the grounding status of each device as well. • Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check "Station No.", "Slot No. (another station)", and "Base No. (another station)". If the destination station is an intelligent device station, execute the system monitor for the remote head module in the station and check the error details. • Check that no online operation is being performed from a peripheral such as the engineering tool, or no program or parameter is being written to the safety communication destination station. • The possible cause is a hardware failure of the module performing safety communications. Please consult your local Mitsubishi representative.	Safety station system configuration information	At interrupt occurrence

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1A62H	Timeout in safety communications	• A timeout error occurred during safety communications. • The safety protocol version of the device actually connected to perform safety communications and the version set in the safety communication setting do not match.	Continue	• Check and correct the safety refresh monitoring time and the transmission interval monitoring time setting values. • The possible cause is malfunction due to noise or cable failure. Check that there is no error in a transmission path by executing CC-Link IE Field diagnostics. Check the distance of wires and cables and the grounding status of each device as well. • Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check "Station No.", "Slot No. (another station)", and "Base No. (another station)". If the destination station is an intelligent device station, execute the system monitor for the remote head module in the station and check the error details. • Check that no online operation is being performed from a peripheral such as the engineering tool, or no program or parameter is being written to the safety communication destination station. • Check whether the safety protocol version of the device and the version set in the safety communication setting match. • The possible cause is a hardware failure of the module performing safety communications. Please consult your local Mitsubishi representative.	Safety station system configuration information	At interrupt occurrence
1A63H to 1A66H	Timeout in safety communications	A timeout error occurred during safety communications.	Continue	• Check and correct the safety refresh monitoring time and the transmission interval monitoring time setting values. • The possible cause is malfunction due to noise or cable failure. Check that there is no error in a transmission path by executing CC-Link IE Field diagnostics. Check the distance of wires and cables and the grounding status of each device as well. • Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check "Station No.", "Slot No. (another station)", and "Base No. (another station)". If the destination station is an intelligent device station, execute the system monitor for the remote head module in the station and check the error details. • Check that no online operation is being performed from a peripheral such as the engineering tool, or no program or parameter is being written to the safety communication destination station. • The possible cause is a hardware failure of the module performing safety communications. Please consult your local Mitsubishi representative.	Safety station system configuration information	At interrupt occurrence

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1A70H to 1A73H	Safety communications receipt data error	The received data is abnormal.	Continue	• Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check "Station No.", "Slot No. (another station)", and "Base No. (another station)". If the destination station is an intelligent device station, execute the system monitor for the remote head module in the station and check the error details. • The possible cause is a hardware failure of the module performing safety communications. Please consult your local Mitsubishi representative.	Safety station system configuration information	At interrupt occurrence
1A80H	Timeout in safety communications	Safety communications with a destination station could not be resumed within a safety I/O hold time.	Continue	• Check and correct the safety I/O hold time. • The possible cause is malfunction due to noise or cable failure. Check that there is no error in a transmission path by executing CC-Link IE Field diagnostics. Check the distance of wires and cables and the grounding status of each device as well. • Check the detailed information (safety station system configuration information) of the error by executing module diagnostics using the engineering tool and check "Station No.", "Slot No. (another station)", and "Base No. (another station)". If the destination station is an intelligent device station, execute the system monitor for the remote head module in the station and check the error details. • The possible cause is a hardware failure of the module performing safety communications. Please consult your local Mitsubishi representative.	Safety station system configuration information	At interrupt occurrence
1B00H	Network configuration mismatch	The CC-Link IE Field Network configuration differs between the systems A and B. (The error is detected in the standby system.)	Continue	• Check that all network cables are connected correctly. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the network cable. Please consult your local Mitsubishi representative.	System configuration information	Always
1B20H	System consistency check error (operating status)	The operating status of the CPU module differs between the systems A and B. (The error is detected in the standby system.)	Continue	Set the same operating status to the CPU modules in both systems.	—	Always
1B21H	System consistency check error (safety operation mode)	• The safety operation mode differs between the systems A and B. • Switching of the safety operation mode is executed for the systems A and B. (The error is detected in the standby system.)	Continue	Set the same safety operation mode to the CPU modules in both systems.	—	Always
1B40H	Redundant system error	Both systems were found to be set as system A or system B when tracking cables were connected. (The error is detected in the control system or the standby system.)	Continue	Set one system as system A and the other system as system B by performing online operation. Then, restart the CPU modules in both systems.	—	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1B42H	Redundant system error	The CPU module operated in redundant mode was connected to the CPU module set to process mode with tracking cables. (The error is detected in the control system or the standby system.)	Continue	Read the project stored in the CPU module operated in redundant mode, and write it to the CPU module set to process mode. Then, reset the CPU module set to process mode, and run it again.	—	Always
1B43H	Redundant system error	• The firmware version of the CPU module connected is not compatible with the other one. • The Process CPU and the SIL2 Process CPU are connected. • In a redundant system with redundant extension base unit, the extension cable is not connected between the main base unit of one system and the extension base unit. Or, the extension base unit cannot be recognized by the other system. (The error is detected in the control system or the standby system.)	Continue	• Replace one CPU module with the one having a firmware version that can be used with the other CPU module together, referring to the MELSEC iQ-R Module Configuration Manual. Then, restart the system. • Replace one CPU module in either of two systems so that the same CPU module models are used in both systems. Then, write a project to the replaced CPU module and restart the system. • Check the system configurations of both systems and correct them. • Connect the extension cable securely and properly to the connector of the redundant extension base unit, and then restart the CPU module in which the error has been detected. If the same error code is displayed again, the possible cause is a failure of the extension cable. Replace the extension cable.	—	Always
1B60H	Standby system CPU module error	In backup mode, the standby system CPU module has not started up. (The error is detected in the control system.)	Continue	• If the standby system is powered off, power on the system. • If the standby system CPU module is in the RESET state, clear the RESET state. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	—	Always
1B61H	Standby system CPU module error	In backup mode, a stop error of the standby system CPU module was detected. (The error is detected in the control system.)	Continue	Eliminate the error cause, and restart the system. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	—	Always
1B70H	Tracking communications disabled	Data communications with the other system cannot be performed even in backup mode. (The error is detected in the control system or the standby system.)	Continue	• If the standby system is powered off, power on the system. • If a WDT error has occurred in the other system CPU module, eliminate the error cause, and restart the system. • Connect tracking cables securely and properly to the connectors of the redundant function modules in both systems. (One cable shall be connected between the IN connector of the module in system A and the OUT connector of the module in system B. The other cable shall be connected between the OUT connector of the module in system A and the IN connector of the module in system B.) • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.	—	Always

Error code	Error name	Error details and cause	Stop/ continue	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.)	Continue	• Connect tracking cables securely and properly to the connectors of the redundant function modules in both systems. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module or tracking cable. Please consult your local Mitsubishi representative.	—	Always
1B78H	Tracking communication error	During tracking transfer, a tracking communication error occurred. (The error is detected in the control system or the standby system. Note that if the error is detected in the standby system, the detailed information is not stored.)	Continue	• If the other system is powered off, power on the system. • If a WDT error has occurred in the other system CPU module, eliminate the error cause, and restart the system. • Connect tracking cables securely and properly to the connectors of the redundant function modules in both systems. • Set the program execution time of the standby system shorter than that of the control system. If the shorter value cannot be set, correct the SD1662 value. • If the load on the communications with the external devices via the other system is high, perform the communications without going through the other system or lighten the load on the communications. • Check the power supply status. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.	Data type (tracking transfer) information	At END instruction execution
1B80H	Tracking transfer error	The tracking data set in the CPU parameters exceeds the allowable tracking transfer range. (The error is detected in the control system.)	Continue	Set the device/label data within the allowable range in the CPU parameters ("Tracking Device/Label Setting").	Tracking transfer trigger information	At END instruction execution
1B81H	Tracking transfer error	The file register capacity has been set in the CPU parameters of the control system CPU module is less than the file register capacity set for tracking transfer. (The error is detected in the control system.)	Continue	• Check and correct the tracking transfer settings in the CPU parameters so that they will be within the device range or the file register capacity. • Check and correct the device settings or the file register capacity setting in the CPU parameters.	—	At END instruction execution
1B82H		The file register data is transferred from the control system CPU module to the standby system CPU module exceeding the capacity of the receive side. (The error is detected in the standby system.)				
1BA0H	Redundant function module error	An error has been detected in the redundant function module.	Continue	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the status of the redundant function module. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
1BD0H	System switching error	The systems were not switched. There was a cause of system switching failure. (The error is detected in the control system.)	Continue	Check the cause of system switching failure in the detailed information (system switching information) of the error by executing module diagnostics using the engineering tool, eliminate the error cause, and switch the systems from the engineering tool again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or tracking cable. Please consult your local Mitsubishi representative.	System switching information	At system switching execution
1BD1H	System switching error	The systems were not switched by using the SP.CONTSW instruction because SM1646 (System switching by a user) was off. (The error is detected in the control system.)	Continue	To switch the systems by using the SP.CONTSW instruction, turn on SM1646 and then execute the instruction.	Error location information	At instruction execution
2000H	Module configuration error	The module type set in the system parameters ("I/O Assignment Setting") differs from that of the module actually mounted.	Stop	Re-set the module type in the system parameters in accordance with the CPU module or intelligent function module actually mounted.	System configuration information	At power-on, at RESET
2001H	Module configuration error	The I/O numbers set in the system parameters ("I/O Assignment Setting") are overlapping between modules.	Stop	Re-set the I/O numbers in the system parameters in accordance with the intelligent function module or I/O module actually mounted.	System configuration information	At power-on, at RESET
2002H	Module configuration error	The number of points assigned to the intelligent function module in the system parameters ("I/O Assignment Setting") is smaller than that of the module actually mounted.	Stop	Re-set the number of points in the system parameters in accordance with the intelligent function module actually mounted.	System configuration information	At power-on, at RESET
2004H	Module configuration error	• Nine or more CC-Link IE Controller Network modules and/or MELSECNET/H network modules are mounted in the entire system. (The CC-Link IE built-in Ethernet interface module is included if the module is used as a CC-Link IE Controller Network module.) • Five or more MELSECNET/H network modules are mounted in the entire system.	Stop	• Reduce the number of CC-Link IE Controller Network modules and/or MELSECNET/H network modules to eight or less in the entire system. (The CC-Link IE built-in Ethernet interface module is included if the module is used as a CC-Link IE Controller Network module.) • Reduce the number of MELSECNET/H network modules to four or less in the entire system.	System configuration information	At power-on, at RESET
2005H	Module configuration error	• Two or more interrupt modules (QI60) with no interrupt pointer setting are mounted. • The interrupt pointer numbers are overlapping in the interrupt module (QI60) with no interrupt pointer setting and a module with an interrupt pointer setting.	Stop	• Mount only one QI60 in the entire system. • Configure the interrupt pointer setting for the QI60. • Correct the interrupt pointer setting.	System configuration information	At power-on, at RESET, at STOP → RUN state
2006H	Module configuration error	A module is mounted on the 65th slot or later.	Stop	Remove the module mounted on the 65th slot or later.	System configuration information	At power-on, at RESET
2007H	Module configuration error	A module is mounted on the slot whose number is later than that specified in the system parameters ("I/O Assignment Setting").	Stop	Remove the module mounted on the slot whose number is later than that specified in the system parameters.	System configuration information	At power-on, at RESET
2008H	Module configuration error	• A module is mounted over or across the maximum number of I/O points (4096).	Stop	• Remove the module mounted over or across the maximum number of I/O points (4096). • Replace the module mounted on the last slot with the one that does not exceed the maximum number of I/O points (4096).	System configuration information	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2009H	Module configuration error	There is no response from the I/O module or intelligent function module accessed.	Stop	• Check and correct the I/O assignment setting in the system parameters. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the I/O module or intelligent function module. Please consult your local Mitsubishi representative.	System configuration information	Always
2020H	Module configuration error	There is a mounted module that is not supported, or there is a mounted module that is not supported by the network type (module model name) set in system parameters ("I/O Assignment Setting").	Stop	• Remove the unsupported module if any. • Check whether modules are supported by the network type (module model name) set in the system parameters. • If all the modules are supported, the possible cause is a hardware failure of the CPU module, base unit, I/O module, or intelligent function module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2021H	Module configuration error	In a multiple CPU system, the control CPU of the Q series intelligent function module incompatible with the multiple CPU system is set to other than CPU No.1.	Stop	• Replace the Q series intelligent function module with the one (function version B) compatible with the multiple CPU system. • Set the control CPU of the Q series intelligent function module incompatible with the multiple CPU system to CPU No.1.	System configuration information	At power-on, at RESET
2022H	Module configuration error	A power supply module other than the redundant power supply module has been mounted on the redundant power supply base unit.	Stop	Mount only the redundant power supply module. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module, CPU module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2040H	CPU module configuration error	• The number of CPU modules set in the system parameters ("I/O Assignment Setting") differs from the number of CPU modules actually mounted. • The CPU module is mounted on the slot different from the one specified in the system parameters (I/O assignment setting). • Two or more Safety CPUs are mounted.	Stop	• Correctly set the number of CPU modules (including the empty setting) in the system parameters in accordance with the number of CPU modules actually mounted. • Correctly set the system parameters so that the setting and actual CPU module mounting status will be the same. • Mount only one Safety CPU.	System configuration information	At power-on, at RESET
2041H	CPU module configuration error	• The CPU module is not mounted on the slot that is set for the CPU module in the system parameters ("I/O Assignment Setting"). • The CPU module is mounted on the slot that is set for empty in the system parameters ("I/O Assignment Setting"). • An I/O module or intelligent function module is mounted between the CPU modules. • Arrangement of CPU modules mounted is not appropriate to configure a multiple CPU system.	Stop	• Check and correct the I/O assignment setting in the system parameters. • Correctly set the number of CPU modules (including the empty setting) in the system parameters in accordance with the number of CPU modules actually mounted. • Remove the I/O module or intelligent function module mounted between the CPU modules.	System configuration information	At power-on, at RESET
2043H	CPU module configuration error	The CPU module is mounted on the inapplicable slot.	Stop	• Mount the CPU module on the applicable slot (CPU slot or I/O slot 0 to 6). • Remove the CPU module from the inapplicable slot.	System configuration information	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2044H	CPU module configuration error	The host CPU No. set in the system parameters ("I/O Assignment Setting") differs from the one determined by the mounting position of the CPU module.	Stop	Re-set the host CPU No. in the system parameters in accordance with the mounting position of the CPU module.	System configuration information	At power-on, at RESET, at STOP → RUN state
2046H	CPU module configuration error	Another CPU module was accessed, but there was no response.	Stop	• Check and correct the I/O assignment setting in the system parameters. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of another CPU module. Please consult your local Mitsubishi representative.	System configuration information	Always
2050H	CPU module configuration error	An unsupported CPU module is mounted.	Stop	Remove the unsupported CPU module. If all the CPU modules are supported, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2051H	CPU module configuration error	A CPU module that does not support the online module change (direct change) function is used in a multiple CPU system.	Stop	• Refer to the MELSEC iQ-R Online Module Change Manual, and correct the system configuration. • To disable the online module change (direct change) function, set "Direct change setting" to "Disable" in the CPU parameters. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2052H	CPU module configuration error	A multiple CPU system was configured with CPU modules incompatible with multiple CPU systems.	Stop	Check and correct the system configuration.	System configuration information	At power-on, at RESET
2060H	Base unit configuration error	Eight or more extension base units are connected.	Stop	Reduce the number of extension base units to seven or less.	System configuration information	At power-on, at RESET
2061H	Base unit configuration error	Any of the following base units is connected: QA1S3□B, QA1S5□B/QA1S6□B, QA6□B, QA6ADP+A5□B/A6□B, or QA1S6ADP-S1+A1S5□B/ A1S6□B.	Stop	Remove the inapplicable base unit: QA1S3□B, QA1S5□B/QA1S6□B, QA6□B, QA6ADP+A5□B/A6□B, and QA1S6ADP-S1+A1S5□B/A1S6□B.	System configuration information	At power-on, at RESET
2063H	Base unit configuration error	Extension base unit levels are overlapping.	Stop	Check and correct the level setting of the extension base units.	System configuration information	At power-on, at RESET
2070H	Base unit configuration error	• An unsupported base unit is connected. • A GOT is bus-connected to the Q series extension base unit.	Stop	• Check the firmware version of the CPU module, and replace the CPU module with the one supporting the use of redundant extension base units. • Disconnect the unsupported base unit. If all the base units are supported, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative. • Disconnect the GOT bus-connected to the Q series extension base unit.	System configuration information	At power-on, at RESET
2080H	Inter-module synchronization configuration error	An inter-module synchronization signal error has been detected.	Stop	The possible cause is a hardware failure of the CPU module, base unit, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
20E0H	Module unrecognized	■CPU module • A module that the CPU module cannot recognize is mounted. • In a multiple CPU system, the module cannot be recognized because the control CPU setting in the system parameters is different from that of other CPU modules. ■Redundant function module Data communications with the CPU module cannot be performed.	Stop	■CPU module • Mount only applicable modules. • Correct the system parameter settings for the CPU No.2 and later in accordance with those of the CPU No.1. • The possible cause is a hardware failure of the I/O module or intelligent function module accessed. Please consult your local Mitsubishi representative. ■Redundant function module The possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	System configuration information	Always
2100H	Memory error	• No extended SRAM cassette is inserted. • The capacity of extended SRAM cassette set in does not match "Extended SRAM Cassette Setting" in the CPU parameter with the actual capacity of the cassette inserted.	Stop	Check that an extended SRAM cassette is inserted. Or, correct the capacity set in "Extended SRAM Cassette Setting" in the CPU parameter so that it matches with the actual capacity of the cassette inserted. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or extended SRAM cassette. Please consult your local Mitsubishi representative.	Drive/file information, parameter information	At power-on, at RESET
2101H	Memory error	An extended SRAM cassette is inserted or removed while the programmable controller is powered on.	Stop	• Do not insert or remove an extended SRAM cassette during operation. • Check the mounting status of the CPU module of the extended SRAM cassette. If the same error code is displayed again, the possible cause is a hardware failure of the extended SRAM cassette. Please consult your local Mitsubishi representative.	Drive/file information	Always
2102H	Memory error	An error has been detected in the inserted extended SRAM cassette.	Stop	Check the mounting status of the CPU module of the extended SRAM cassette. If the same error code is displayed again, the possible cause is a hardware failure of the extended SRAM cassette or CPU module. Please consult your local Mitsubishi representative.	Drive/file information	Always
2103H	Memory error	An unsupported extended SRAM cassette is mounted.	Stop	Replace the extended SRAM cassette with the one supported by the CPU module.	Drive/file information	Always
2120H	Memory card error	The memory card was removed without the card being disabled.	Stop/ continue	Disable the memory card, and then remove it.	Drive/file information	Always
2121H	Memory card error	An error has been detected in the memory card.	Stop/ continue	Format the memory card, re-insert the memory card, or replace the memory card. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information	Always
2122H	Memory card error	The CPU module did not start because the memory card was not restored during the startup processing.	Stop	Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the memory card. Replace the memory card.	Drive/file information	At power-on, at RESET
2160H	IP address duplication error	Overlapping IP addresses have been detected.	Stop	Check and correct the IP addresses.	—	Always
2180H	Invalid file	An invalid file has been detected.	Stop	Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, select the correct file name, and write the specified file to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 function module. Please consult your local Mitsubishi representative.	Drive/file information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
21A0H	File specification error	• The file specified in the CPU parameters does not exist. • The memory card is disabled by tuning on SM606 (SD memory card forced disable instruction). • The file register file does not exist in the specified memory when the file register setting is set to "Use Common File Register in All Programs" and the file capacity is not set in the CPU parameters ("File Setting"). • The file specified in the memory card parameters ("Boot Setting") does not exist in the memory card.	Stop	• Turn off SM606. (Cancel the disabled state.) • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, select the correct file name, and write the specified file to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the device/label memory in the CPU module, memory card, or SIL2 function module. Please consult your local Mitsubishi representative.	Drive/file information, parameter information	At instruction execution, at interrupt occurrence, at power-on, at RESET, at STOP → RUN state, at END instruction execution
21A1H	File specification error	The file specified in parameter cannot be created.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the name and size of the file corresponding to the displayed parameter number. • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and take the following actions: (1) Format the corresponding drive. (2) Delete unnecessary files on the corresponding drive to increase free space. (3) Unlock the corresponding drive if it is locked.	Drive/file information, parameter information	At write, at power-on, at RESET, at STOP → RUN state
21A2H	File specification error	The CPU module model set to the file using the engineering tool differs from that of the CPU module actually mounted.	Stop	Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and correct the CPU module model set to the file in accordance with that of the CPU module actually mounted.	Drive/file information	At write, at power-on, at RESET, at STOP → RUN state
21A3H	File specification error	The files names are overlapping.	Stop	Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and take the following actions: (1) Format the corresponding drive. Write the program(s) and FB program(s) to the CPU module again. Then, reset the CPU module and run it again. (2) Correct the file names so that they do not overlap.	Drive/file information	At write, at power-on, at RESET, at STOP → RUN state
2200H	Parameter error	• The system parameter file and CPU parameter file do not exist. • The memory card parameter file or module extension parameter file stored in the memory card cannot be accessed because the memory card is disabled by turning on SM606 (SD memory card forced disable instruction).	Stop	• Write the system parameter file and CPU parameter file to the CPU module. • Turn off SM606. (Cancel the disabled state.)	Parameter information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2220H	Parameter error	• The parameter setting is corrupted. • Parameters that the firmware version of the corresponding CPU does not support are written.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and write the displayed parameter setting to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, memory card, SIL2 function module, I/O module, or intelligent function module. Please consult your local Mitsubishi representative. • Check the firmware version of the CPU module and use a supported product. Then, write parameters again.	Parameter information	At power-on, at RESET, at STOP → RUN state, at write
2221H	Parameter error	The set value is out of range.	Stop	Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, memory card, SIL2 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. • For the simple CPU communication function, "Initial Communication Setting" is enabled.	Stop	Check the firmware version of the CPU module and use a supported product. Use the product in a supported category. • Set the number of occupied stations so that the total number is 16 stations or less, and set 1 to the group number. • Disable "Initial Communication Setting".	Parameter information	At power-on, at RESET, at STOP → RUN state, at END instruction execution, at instruction execution, at module access
2222H	Parameter error	• Use of the function that is not supported by the module is enabled in parameter. • The module is non-operational. • Parameters that the firmware version of the module does not support were written.	Stop	• Remove the unsupported module if any. • Use functions supported by the module. • Check the status of the module. • Check the firmware version of the module and use a supported product. • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, memory card, SIL2 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.	Stop	Reset the CPU module and run it again.	Parameter information	Always

Error code	Error name	Error details and cause	Stop/ continue	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.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, display the error-detected area by clicking the [Error Jump] button, and increase the capacity of the area. (If the capacity of the area cannot be increased, decrease the capacity of other areas.) • Reduce the number of labels or local devices used. • If no global label is used, delete the global label setting file.	Parameter information	At write, at power-on, at RESET, at STOP → RUN state
2225H	Parameter error	• The CPU module model set to the project using the engineering tool differs from that of the CPU module actually mounted. • The operation set in the memory card parameters cannot be performed. (The boot function cannot be executed.)	Stop	• Correct the CPU module model set to the project in accordance with the CPU module actually mounted. • Delete the memory card parameter settings. • Remove the memory card so that the operation set in the memory card parameters will not be performed. (Do not execute the boot operation.)	Parameter information	At write, at power-on, at RESET, at STOP → RUN state
2226H	Parameter error	The SFC setting in the CPU parameters is incorrect. (Block 0 was set to start automatically, however, block 0 does not exist.)	Stop	Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number.	Parameter information	At power-on, at RESET, at STOP → RUN state, at SFC program execution
2227H	Parameter error	The execution type of the SFC program set in the CPU parameter program settings is other than the scan execution type.	Stop	Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number.	Parameter information	At power-on, at RESET, at STOP → RUN state
2240H	Parameter error (module)	In a multiple CPU system, the I/O module or intelligent function module controlled by another CPU module is specified in the module parameters.	Stop	Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state
2241H	Parameter error (module)	• The I/O numbers set in the system parameters differ from those of the module actually mounted. • The target module is not mounted on the slot where the system parameters and module parameters are set. • The module type set in parameter differs from that of the module actually mounted.	Stop	• Check if the system configuration displayed on the system monitor window of the engineering tool match the actual system configuration. • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state, at END instruction execution, at instruction execution, at module access

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2242H	Parameter error (module)	The intelligent function module has detected a module parameter error.	Stop	Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed I/O number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the intelligent function module connected. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET, at STOP → RUN state
2260H	Parameter error (network)	Network numbers are overlapping.	Stop	Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the intelligent function module connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET
2261H	Parameter error (network)	Different network types (CC IE Control extended mode/normal mode) are set between the control station and the normal station.	Stop	Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the intelligent function module connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state
2262H	Parameter error (network)	• When the station number of the MELSECNET/H network module is 0, parameters of PLC to PLC network are set. • The station type set in the module parameters differs from that of the module actually mounted. • In the CPU module in redundant mode, "RJ71GF11-T2" (the model name cannot be set for the redundant system) is selected to the model name in the I/O assignment setting of the system parameter.	Stop	• Correct the station number of the MELSECNET/H network module. • Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. • In the CPU module in redundant mode, select "RJ71GF11-T2(MR)", "RJ71GF11-T2(SR)", or "RJ71GF11-T2(LR)" (model names can be set for the redundant system) to the model name in the I/O assignment setting of the system parameter. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the intelligent function module connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state
2263H	Parameter error (network)	Even though the CC-Link IE module or MELSECNET/H network module is mounted, a different CC-Link IE module or MELSECNET/H network module is set in the system parameters ("I/O Assignment Setting"), or CC-Link IE module or MELSECNET/H network module parameters have not been set.	Stop	Set the system parameters and module parameters. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or the intelligent function module connected. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2280H	Parameter error (refresh)	• The refresh setting is not set correctly. (Data were refreshed exceeding the file register capacity.) • The refresh settings (number of points) are different from those of other numbered CPU modules.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number so that the data are refreshed within the specified device range. (Take the following actions: increase the number of file register points, create a file register file having a capacity for all of the target data to be refreshed, or reduce the refresh device range.) • Rewrite the refresh settings (number of points) in the CPU parameters for all the CPU modules. (Use the same number of points in the refresh settings for all the CPU modules.)	Parameter information	At power-on, at RESET, at STOP → RUN state, at END instruction execution, at instruction execution, at module access
2281H	Parameter error (refresh)	A device that cannot be used as a refresh device is specified.	Stop	Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number.	Parameter information	At power-on, at RESET, at STOP → RUN state
2282H		The number of specified refresh points is invalid.				
2283H		The total number of refresh points exceeded the maximum limit.				
22E0H	Parameter verification error	• In a multiple CPU system, the system parameter settings of the host CPU module differ from those of other CPU modules. • In a multiple CPU system, the system parameter settings are overwritten only to the host CPU module, and the settings differ from those of other CPU modules.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the system parameter settings corresponding to the displayed number for the CPU No.2 and later. The settings need to be the same among all the CPU modules. (The module synchronization setting and fixed scan communication setting need to be the same between the CPU modules that use these functions.) • When the system parameter settings are changed, update the settings of all the CPU modules connected. (The system parameter settings must be same in all the CPU modules.)	Parameter information, system configuration information	At write, at power-on, at RESET, at STOP → RUN state
2300H	Security key authentication error	The security key set to the program does not match the one registered to the CPU module (or cassette).	Stop	Check and correct the security key setting.	Drive/file information	At power-on, at RESET, at STOP → RUN state
2301H		The security key is set to the program, but it is not registered to the CPU module (or cassette).				
2302H	Security key authentication error	• The security key set to the file is corrupted and does not match the one registered to the CPU module. • The security key registered to the CPU module is corrupted and does not match the one set to the file.	Stop	Write the file to the CPU module again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	—	At power-on, at RESET, at STOP → RUN state
2303H	Security key authentication error	The security key is registered to the CPU module and cassette.	Stop	Check and correct the security key setting.	—	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2320H	Remote password setting error	• The start I/O number of the remote password target module is set to other than 0H to 0FF0H. • There is a problem on the slot specified by the start I/O number of the remote password setting. (1) No module is mounted. (2) The mounted intelligent function module does not support the remote password setting.	Stop	• Set the start I/O number of the remote password target module within the range 0H to 0FF0H. • On the specified slot, mount an intelligent function module that supports the remote password setting.	System configuration information	At power-on, at RESET, at STOP → RUN state
2321H	Remote password setting error	In a multiple CPU system, the module controlled by another CPU module is specified by the start I/O number of the remote password setting.	Stop	Check and correct the remote password setting.	System configuration information	At power-on, at RESET, at STOP → RUN state
2400H	Module verification error	• The module information at power-on differs from the information of modules actually mounted. • The I/O module or intelligent function module is not mounted properly or was removed during operation.	Stop/ continue	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and check the module corresponding to the displayed slot number. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	System configuration information	Always
2401H	Module verification error	A CPU module, I/O module, or intelligent function module was mounted on the base unit during operation.	Stop/ continue	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and check the module corresponding to the displayed slot number. • Do not mount a CPU module, I/O module, nor intelligent function module during operation. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	System configuration information	Always
2420H	Fuse blown error	The output module with a blown fuse has been detected.	Stop/ continue	• Check the FUSE LED of each output module, and replace the one with the FUSE LED on. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and replace the module corresponding to the displayed slot number.	System configuration information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2440H	Module major error	• In a multiple CPU system, the control CPU setting in the system parameters is different from that of other CPU modules. • In a multiple CPU system, other CPU modules (SIL2 Process CPU or Safety CPU) have detected a parameter verification error. • An error has been detected in the I/O module or intelligent function module during the initial processing.	Stop	• Correct the system parameter settings for the CPU No.2 and later in accordance with those of the CPU No.1. • Eliminate the error cause of another CPU module (Safety CPU or SIL2 Process CPU). • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2441H	Module major error	An error has been detected in the I/O module or intelligent function module when the instruction was executed.	Stop/ continue	• Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	Error location information, system configuration information	At instruction execution
2442H		An error has been detected in the I/O module or intelligent function module during the END processing.	Stop/ continue		System configuration information	At module access
2443H		An error has been detected in the I/O module or intelligent function module.	Stop			
2450H	Module major error	• A major error has been notified from the intelligent function module connected. • The I/O module or intelligent function module is not mounted properly or was removed during operation.	Stop/ continue	• Take measures to reduce noise. • Check the connection status of the extension cable. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and check the module corresponding to the displayed slot number. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	System configuration information	Always
2460H	Another CPU module major error	An error has been detected in another CPU module during the initial processing.	Stop	• Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the host CPU module or another CPU module where the error has been detected. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
2461H		An error has been detected in another CPU module when the instruction was executed.	Stop/ continue		Error location information, system configuration information	At instruction execution
2462H		An error has been detected in another CPU module during the END processing.	Stop/ continue		System configuration information	At END instruction execution
2463H		An error has been detected in another CPU module.	Stop		System configuration information	At power-on, at RESET
2470H		A major error has been notified from another CPU module.	Stop/ continue		System configuration information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2480H	Multiple CPU error	• In a multiple CPU system, an error has been detected in the CPU module where "Stop" is set in the operation mode setting parameter. • Any CPU module other than CPU No.1 is mounted in the inapplicable slot. (An error occurs in the CPU module mounted in the inapplicable slot.)	Stop	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, identify the error module, and eliminate the error cause. • Remove the CPU module from the inapplicable slot.	System configuration information	Always
2481H	Multiple CPU error	In a multiple CPU system, any CPU module other than CPU No.1 was disconnected from the base unit during operation. Or, any CPU module other than CPU No.1 was reset.	Stop	Check the mounting status and reset status of the CPU modules other than CPU No.1.	System configuration information	Always
24C0H 24C1H	System bus error	An error was detected on the system bus.	Stop	■CPU module • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C2H	System bus error	• The I/O module or intelligent function module is not mounted properly or was removed during operation. • An error was detected on the system bus.	Stop	■CPU module • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and check the module corresponding to the displayed slot number. • Check the connection status of the extension cable. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	System configuration information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
24C3H	System bus error	An error was detected on the system bus.	Stop	■CPU module • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C4H	System bus error	An error was detected on the system bus.	Stop	• Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the base unit, extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C5H					—	
24C6H	System bus error	An error was detected on the system bus.	Stop	■CPU module • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or extension cable. Please consult your local Mitsubishi representative. ■Redundant function module • The possible cause is malfunction due to noise. Take measures to reduce noise by checking the distance of wires and cables, and the grounding status of each device. • Execute a module communication test. If the same error code is displayed again, the possible cause is a hardware failure of the redundant function module. Please consult your local Mitsubishi representative.	—	At module access
24C8H	System bus error	An error was detected on the system bus.	Stop	• Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the extension cable, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	—	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
24D0H	System bus error	• The extension level setting of the Q series extension base unit is overlapping with that of any other extension base units. • An unsupported base unit is connected. • Incorrect connections of the extension cables have been detected in the redundant extension base unit. • An error was detected on the system bus.	Stop	• Check and correct the level setting of the Q series extension base unit. • Check the connection status of the extension cable. • Check that the 10m mark is printed on the base unit connected when using the ten-meter extended cables (RC100B). (MELSEC iQ-R Module Configuration Manual) • Disconnect the unsupported base unit. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and identify the extension cables incorrectly connected. Connect the OUT1 connector to the IN1 connector of the next level, and connect the OUT2 connector to the IN2 connector of the next level. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or extension cable. Please consult your local Mitsubishi representative.	System configuration information	Always
24E0H	System bus error	An error was detected on the system bus.	Stop	• Take measures to reduce noise. • Check the mounting status and reset status of the CPU modules other than CPU No.1. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	System configuration information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
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. ■SIL2 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.	Stop	■CPU module • Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take the following actions: (1) Check and correct the program so that it can be executed within the 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. ■SIL2 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 CPU module. Please consult your local Mitsubishi representative.	Time information	Always
2501H		■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. ■SIL2 function module • The scan time exceeded the execution monitoring time set in parameter.				
2520H	Invalid interrupt	Even though an interrupt was requested, there is no interrupt factor.	Stop	• Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.	System configuration information	At interrupt occurrence
2521H					—	
2522H	Invalid interrupt	An interrupt was requested from the module with no interrupt pointer setting.	Stop	• Check and correct the interrupt pointer setting in the module parameters. • Take measures so that no interrupt is requested from the module with no interrupt pointer setting. • Check and correct the interrupt setting in the buffer memory of the intelligent function module. • Correct the BASIC program executed in the QD51.	System configuration information	At interrupt occurrence
2800H	I/O number or network number specification error	The specified I/O number is out of range (other than 0 to FFH, 3E0 to 3E3H).	Stop/ continue	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
2801H		The I/O number of the module that does not exist was specified.	Stop/ continue		Error location information, system configuration information	At instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2802H	I/O number or network number specification error	• The I/O number of the module that does not support the instruction was specified. • The dedicated instruction specified in the program cannot be executed in the specified module or mode.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the execution propriety (including support status and execution mode) of the dedicated instruction, referring to the manual for the target module.	Error location information, system configuration information	At instruction execution
2803H	I/O number or network number specification error	The I/O number of the module that cannot be specified in the instruction was specified.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information, system configuration information	At instruction execution
2804H		The specified network number is out of range (other than 1 to 239).				
2805H		The network number that does not exist was specified.				
2806H	I/O number or network number specification error	An I/O module or intelligent function module controlled by another CPU module was specified.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Delete the link direct device that specifies a network module controlled by another CPU module from the program. • Specify a network module controlled by host CPU module by using the link direct device.	Error location information, system configuration information	At instruction execution
2807H	I/O number or network number specification error	The module cannot be identified in the instruction that requires a specification of the I/O module or intelligent function module. (There is a mistake in the string specifying the module.)	Stop/ continue	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
2810H	I/O number or network number specification error	The I/O module or intelligent function module specified in the instruction cannot execute the instruction.	Stop/ continue	The possible cause is a hardware failure of the I/O module or intelligent function module specified in the instruction. Please consult your local Mitsubishi representative.	Error location information, system configuration information	At instruction execution
2820H	Device, label, or buffer memory specification error	• The device or label area used in the instruction exceeded the specified range. • The file register file is not set or was accessed without setting it in the CPU parameters (file setting).	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Set the file register file in parameter, and access the file.	Error location information, process control instruction processing information	At instruction execution, at END instruction execution
2822H	Device, label, or buffer memory specification error	The device or label that cannot be used in the instruction was specified.	Stop/ continue	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
2823H		• The buffer memory area of the module specified in the instruction exceeded the specified range. • The module specified in the instruction does not have buffer memory.				
2824H		The access prohibited area in the buffer memory was accessed.				

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
2840H	File name specification error	The file specified in the instruction does not exist.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, create the specified file, and write it to the CPU module. Or, set all the required files in the CPU parameters (file setting).	Error location information, drive/file information	At instruction execution
2841H	File name specification error	The program file specified in the instruction is not set in the CPU parameters (program setting).	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and set the specified program file in the CPU parameter.	Error location information, drive/file information	At instruction execution
2842H	File name specification error	A file that cannot be specified with the instruction was specified.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and check the specified file.	Error location information, drive/file information	At instruction execution
3070H	Operation stop error	The PABORT instruction was executed.	Stop	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool and check the corresponding program - one that outputs the program error code.	Error location information and program error information	At instruction execution
3100H	Program error	The program includes any instruction that cannot be used or decoded in the CPU module.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the program to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 function module. Please consult your local Mitsubishi representative.	Error location information	At power-on, at RESET, at STOP → RUN state, at instruction execution
3120H	Program error	The CPU module does not support the dedicated instruction executed.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check the firmware version of the CPU module and use a supported product.	Error location information	At power-on, at RESET, at STOP → RUN state, at instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3121H	Program error	The number of devices used in the dedicated instruction specified in the program is incorrect.	Stop	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
3122H	Program error	The function block or function specified in the program does not exist.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 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.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 function module. Please consult your local Mitsubishi representative.	Error location information	At power-on, at RESET, at STOP → RUN state
3141H	FB/FUN program error	The structure of FB/FUN program is incorrect.	Stop	• Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 function module. Please consult your local Mitsubishi representative.	Error location information	At instruction execution
3142H	Temporary area error	The temporary area was used incorrectly.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and check the program. • Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 function module. Please consult your local Mitsubishi representative.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3143H	FB/FUN program error	The FB/FUN program file is incorrect.	Stop	• Take measures to reduce noise. • Write the sequence program(s) and FB program(s) to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 function module. Please consult your local Mitsubishi representative.	Error location information	At instruction execution
3200H	Program execution error	■CPU module 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.) ■SIL2 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 memory/device settings in the CPU parameter. • Parameters that require resetting after being rewritten at the CPU module were rewritten.	Stop	■CPU module • 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. ■SIL2 function module • Batch write the CPU parameters, safety CPU parameters, safety program, safety FB file, safety global label setting file, and standard/safety shared label setting file. • If not using safety global labels or standard/safety shared labels, delete the safety global label setting file and standard/safety shared label setting file. • Reset the CPU module and run it again.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3201H	Program execution error	Even though no program is set in the CPU parameters, multiple program files exist.	Stop	• Set a program in the CPU parameter. • Delete unnecessary program files.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3202H	Program execution error	• The program file is incorrect. Or, the program file is not written properly. • For the subroutine type FB, "Use MC/MCR to Control EN" of "Inherent Property" is set to "Yes". • In the option settings of the engineering tool, "Use Phase Processing Instructions" is set to "Yes".	Stop	• Write the program file to the CPU built-in memory again. • Set "Use MC/MCR in EN Control" in "Inherent Property" of subroutine-type FB to "No" and write the program file to the CPU built-in memory again. • Replace the CPU module with the one that supports the use of the subroutine-type FB for which "Use MC/MCR in EN Control" of "Inherent Property" is set to "Yes". • Set "Use Phase Processing Instructions" to "No" in the option settings of the engineering tool and write the program file to the CPU built-in memory again. • Replace the CPU module with the one that supports the use of the phase processing instructions.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3203H	Program execution error	No program file exists.	Stop	• Check if the system parameter file, CPU parameter file, and program file exist. • Write the system parameter file, CPU parameter file, and program file to the CPU built-in memory.	Drive/file information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3205H	Program execution error	• After the global label setting file was modified, only the modified file was written to the programmable controller. Or, without writing the modified global label setting file, only the sequence program file(s) and FB file(s) were written to the programmable controller. (The global label setting file was not written to the programmable controller.) • After an FB file was modified, only the modified FB file was written to the programmable controller. Or, without writing the modified FB file, only the program file(s) and global label setting file were written. (The modified FB file was not written to the programmable controller.) • After the "Access from External Device" setting of the global label setting was changed, only the global label assignment information was written to the programmable controller. Or, after the "Access from External Device" setting was changed, only the sequence program file(s) and FB file(s) were written to the programmable controller. (The global label setting file was not written.) • When the "Access from External Device" setting is not selected, the sequence program file(s), FB file(s), and global label setting file are written to the programmable controller without initializing the memory where the global label assignment information file is stored.	Stop	• Write all the sequence program file(s), FB file(s), global label setting file, and global label assignment information file to the programmable controller. • If no global label is used, delete the global label setting file. • If the "Access from External Device" setting is not selected, initialize the memory where the global label assignment information file is stored, and write the sequence program file(s), FB file(s), and global label setting file to the programmable controller.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3206H	Program execution error	• After the sequence program was modified, only the modified sequence program file was written to the programmable controller. Or, without writing the modified sequence program file, only the initial local label value file was written to the programmable controller. • After the global label setting file was modified, only the modified file was written to the programmable controller. Or, without writing the modified global label setting file, only the initial global label value file was written.	Stop	• Write both the sequence program file(s) and initial local label value file to the programmable controller. • Write both the global label setting file and initial global label value file to the programmable controller. • If no initial global label value is used, delete the initial global label value file. • If no initial local label value is used, delete the initial local label value file.	Drive/file information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3207H	Program execution error	The signal flow area to be used in the FB program cannot be secured.	Stop	Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and identify the error program file. Then, reduce the number of instructions that use the signal flow area of the FB program used in the sequence program file.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3208H	Program execution error	The label file is incorrect.	Stop	Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, select the correct file name, and write the specified file to the CPU module.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3300H	Pointer setting error	The total number of points of local or global pointers used in the program exceeded the points set in the CPU parameters (pointer device area).	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check and correct the pointer device area point setting in the CPU parameters.	Error location information	At power-on, at RESET, at STOP → RUN state
3301H	Pointer setting error	The total number of points of pointer-type labels used in the program exceeded the points set in the CPU parameters (pointer-type label assignment area).	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Check and correct the pointer-type label assignment area point setting in the CPU parameters.	Error location information	At power-on, at RESET, at STOP → RUN state
3302H	Pointer setting error	Multiple global pointers with the same number or pointer-type global labels are used in the program. (The pointer numbers or the labels are overlapping.)	Stop	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At power-on, at RESET, at STOP → RUN state
3303H	Pointer setting error	Multiple local pointers with the same number or pointer-type local labels are used in the program. (The pointer numbers or the labels are overlapping.)				
3320H	Interrupt pointer setting error	The interrupt pointer numbers used in the files are overlapping.				
3340H	FOR-NEXT instruction error	The NEXT instruction was not executed even though the FOR instruction was executed. Or, there are more FOR instructions than NEXT instructions.				At END instruction execution
3341H	FOR-NEXT instruction error	The NEXT instruction was executed even though the FOR instruction was not executed. Or, there are more NEXT instructions than FOR instructions.				At instruction execution
3342H	FOR-NEXT instruction error	The BREAK instruction was executed even though the FOR instruction was not executed.				
3360H	Nesting depth error	The number of nesting levels in the subroutine program exceeded its limit (16).	Stop	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the number of nesting levels (16 or less).	Error location information	At instruction execution
3361H		The number of nesting levels in the FOR instruction exceeded its limit (16).				
3362H		The number of nesting levels in the DI instruction exceeded its limit (16).				

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3363H	Nesting depth error	The number of nesting levels in the function block or function exceeded its limit (32).	Stop	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the number of nesting levels (32 or less).	Error location information	At instruction execution
3380H	Pointer execution error	The pointer specified in the instruction does not exist.	Stop	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
3381H		The RET instruction does not exist in the executed subroutine program.				At END instruction execution
3382H		The RET instruction exists before the FEND instruction in the main program.				At instruction execution
33A0H	Interrupt pointer execution error	The interrupt pointer corresponding to the interrupt input does not exist.	Stop	Check if the program corresponding to the interrupt pointer number set in the module parameters exists.	—	At instruction execution
33A1H	Interrupt pointer execution error	• The IRET instruction does not exist in the executed interrupt program. • The STOP instruction has been executed in an interrupt program.	Stop	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information	At instruction execution
33A2H		The IRET instruction exists before the FEND instruction in the main program.				
33A3H		The IRET instruction or STOP instruction was executed in the fixed scan execution type program.				
33A4H		The IRET instruction or STOP instruction has been executed in an event execution type program.				
33C0H	FB/FUN execution error	Before the FB/FUN program ends, the call source program ended.	Stop	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program. • Take measures to reduce noise. • Write the program to the CPU module again. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 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.	Stop	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the number of nesting levels in the function.	Error location information	At instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3400H	Operation error	Division where the divisor is zero was performed.	Stop/ continue	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the program.	Error location information, process control instruction processing information	At instruction execution
3401H		Data that cannot be converted by using the data conversion instruction was input.				
3402H		The operation was performed with the invalid data (-0, denormalized number, NaN (not a number), or ±∞).				
3403H		An overflow has occurred during the operation.				
3404H		A string that is not supported in the instruction was specified.				
3405H		The input data was out of range.				
3406H		The operation result is out of the output range. (The operation result of the instruction that concatenate character strings exceeded the allowable number of characters.)				
3420H		The link direct device, module access device, or CPU buffer memory access device is specified for both (s) and (d) used in the BMOV instruction.			Error location information	
3421H	Operation error	• When writing data to the data memory by using the SP.DEVST instruction, the number of writes per day exceeded the number set in SD771. • The value set in SD771 is out of range.	Stop/ continue	• Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and check if the SP.DEVST instruction is used correctly in the program. • Execute the SP.DEVST instruction again on another day, or change the value in SD771. • Set the value in SD771 within the settable range.	Error location information	At instruction execution
3422H	Operation error	The structure of the PID control instruction is incorrect.	Stop/ continue	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and correct the PID control instruction structure.	Error location information	At instruction execution
3423H	Operation error	The size of data to be sent/ received by the socket communications instruction exceeds the allowable range.	Stop/ continue	• Check and change the send data size of the CPU module or the external device. • If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Error location information	At instruction execution
3440H	Operation error	In a multiple CPU system, the multiple CPU dedicated instruction (the one whose symbol starts with D(P)) was executed when "Do Not Use" was set to "Fixed scan communication function" in the system parameters ("Multiple CPU Setting").	Stop/ continue	• Change "Fixed scan communication function" to "Use". • Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and change the multiple CPU dedicated instruction to the one whose symbol starts with M(P).	Error location information	At instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3441H	Operation error	In a multiple CPU system, the number of data points was specified exceeding the multiple CPU dedicated instruction areas applicable for each CPU module.	Stop/ continue	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and change the number of data points for the multiple CPU dedicated instruction.	Error location information	At instruction execution
34A0H	Operation error	Response data of the socket communications instruction cannot be created.	Stop/ continue	• Increase the request interval. • Decrease the number of request nodes. • Wait for a response to the previous request before sending the next request. • Correct the timeout value.	Error location information	At END instruction execution
3600H	Safety system configuration error	• The SIL2 function module controlled by the host CPU module is not mounted to the right of the CPU module. • The slot right of the CPU module is set as "Empty" in the system parameters (I/O Assignment Setting).	Stop	• Mount the SIL2 function module controlled by the host CPU module to the right of the CPU module. • Re-set the system parameters (I/O Assignment Setting) so that the SIL2 function module is right of the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, the SIL2 function module, or the base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
3608H	Safety system configuration error	Two or more SIL2 function modules are mounted.	Stop	Mount only one SIL2 function module.	System configuration information	At power-on, at RESET
3610H	Safety system configuration error	The SIL2 function module with a pair version that differs from the one of the SIL2 Process CPU is mounted.	Stop	Mount the safety function module with the same pair version as the one of the SIL2 function module.	System configuration information	At power-on, at RESET
3621H	SIL2 system configuration error	• A module with a firmware version that does not support the safety functions that are available to the system using the SIL2 Process CPU is mounted. • A module that does not support the system using the SIL2 Process CPU is mounted.	Stop	• Check the firmware version of the corresponding module. If the module that does not support the safety functions that are available to the system using the SIL2 Process CPU is mounted, remove the module from the base unit. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool and check the module corresponding to the displayed slot number. If the module does not support the system using the SIL2 Process CPU, remove the module from the base unit.	System configuration information	At power-on, at RESET
3622H	SIL2 system configuration error	• The redundant function module is not mounted to the right of the SIL2 function module. • The slot right of the SIL2 function module is set as "Empty" in the system parameters (I/O Assignment Setting).	Stop	• Mount the redundant function module to the right of the SIL2 function module. • Re-set the system parameters (I/O Assignment Setting) so that the redundant function module is right of the SIL2 function module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, the SIL2 function module, the redundant function module, or the base unit. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
3640H	Safety mutual monitoring error	A file mismatch has been detected in the mutual monitoring between the SIL2 Process CPU and SIL2 function module.	Stop	Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again	—	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3641H	Safety mutual monitoring error	• An error has been detected in the SIL2 Process CPU or SIL2 function module in the mutual monitoring between the SIL2 Process CPU and SIL2 function module. • The SIL2 Process CPU or the SIL2 function module is not mounted properly or was removed during operation.	Stop	• 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 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, the SIL2 function module, or the base unit. Please consult your local Mitsubishi representative.	System configuration information	Always
3680H	Parameter error (safety function)	Because a SIL2 function module controlled by the host CPU module is not mounted, the host CPU module cannot communicate parameters.	Stop	Mount a SIL2 function module controlled by the host CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 function module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET
3681H	Parameter error (safety function)	Connection between modules controlled by the host CPU module is set in the safety communication settings.	Stop	Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, SIL2 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 transmission interval monitoring time in the safety communication setting is shorter than the safety cycle time, or is set to the same value.	Stop	Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter settings corresponding to the displayed number so that the transmission interval monitoring time in the safety communication settings will not be shorter than or equal to the safety cycle time.	Parameter information	At power-on, at RESET
3683H	Parameter error (safety function)	The safety refresh monitoring time in the safety communication setting is shorter than the transmission interval monitoring time, or is set to the same value.	Stop	Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter settings corresponding to the displayed number so that the safety refresh monitoring time in the safety communication settings will not be shorter than or equal to the transmission interval monitoring time.	Parameter information	At power-on, at RESET
3684H	Parameter error (safety function)	The safety CPU parameters do not exist.	Stop	Write the safety CPU parameters.	Parameter information	At power-on, at RESET, at STOP → RUN state
3685H	Parameter error (safety function)	Passive is selected for the open system in the safety communication setting when the operation mode of the CPU module is redundant mode.	Stop	Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter settings corresponding to the displayed parameter number. Then, select Active for the open system of the safety communication setting.	Parameter information	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3686H	Parameter error (safety function)	Safety communication setting parameters that the firmware version of the network module does not support are written.	Stop	Match the safety protocol version that is set in the safety communication setting and the safety protocol version that is supported by the network module. Or, check the firmware version of the network module and use the supported product. Then, write parameters again.	Parameter information	At power-on, at RESET
36A0H	Parameter error (safety function)	A value exceeding the maximum number of safety programs to be executed is set in the program settings.	Stop	Correct the number of safety programs so that it is within the maximum number to be executed.	Parameter information	At power-on, at RESET
36A1H	Parameter error (safety function)	A value exceeding the maximum number of safety FB files to be executed is set in the FB/FUN file settings.	Stop	Correct the number of safety FB files so that it is within the maximum number to be executed.	Parameter information	At power-on, at RESET
36C0H	Safety operation mode switching error	An attempt was made to change the CPU operating status to RUN when the safety operation mode is SAFETY MODE (wait-for-restart).	Stop	Reset the CPU module and run it again.	—	Always
36E1H	File error	• Safety programs and safety FB files, that the data writing to the programmable controller did not complete successfully, exist. • Some safety programs and safety FBs are not in the normal state.	Stop	Write safety programs and safety FBs to the programmable controller again and check that the data writing has completed successfully. Then, reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information	At power-on, at RESET, at STOP → RUN state
3700H	System consistency check error (system configuration)	• The module mounting status of each slot or the module model names differ between the systems A and B. • The module model names set in the system parameters ("I/O Assignment Setting") differ between the systems A and B. (The error is detected in the control system or the standby system.)	Stop	• Match the module mounting status of each slot and the model of the modules used between the systems, and restart the CPU module. • Set the same module model name in the system parameters ("I/O Assignment Setting") for both systems.	System configuration information	At power-on, at RESET, at system switching execution, at tracking cable connection
3701H	System consistency check error (CPU module)	• The CPU module model names differ between the systems A and B. • The safety protocol versions set in the safety communication setting differ between the systems A and B. Or the set safety protocol version is not supported by the CPU module. (The error is detected in the control system or the standby system.)	Stop	• Mount the same CPU module, and restart the CPU module. • Correct the safety communication setting to match the safety protocol versions between the systems A and B. Or replace the CPU module with the one whose firmware supports the safety protocol version, then restart the CPU module.	—	At power-on, at RESET, at system switching execution, at tracking cable connection
3710H	System consistency check error (SD memory card mounting status)	The memory card mounting status differs between the systems A and B. (The error is detected in the control system or the standby system.)	Stop	Match the memory card mounting status between the systems, and restart the CPU module.	—	At power-on, at RESET
3711H	System consistency check error (memory card write protect switch status)	The write protect switch status of the memory card differs between the systems A and B. (The error is detected in the control system or the standby system.)	Stop	Match the write protect switch status between the systems, and restart the CPU module.	—	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	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.)	Stop	• Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, format the corresponding drive of the standby system CPU module, write all files to the CPU module, and restart the standby system CPU module. (Set the same number of steps as that of the control system CPU module in "Allocate Memory for Online Program Change" on the "Program Detail Setting" window.) • Execute the memory copy to match files in the control system and the standby system. Then, reset the CPU module of the standby system and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive/file information	At write, at power-on, at RESET, at STOP → RUN, at system switching execution, at tracking cable connection
3730H	Redundant function module communication error	Data communications with the redundant function module has failed.	Stop	• Check that there is no error in the CPU module, base unit, or redundant function module. • Take measures to reduce noise. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	Always
3740H	Redundant system error	The operation mode differs between the systems A and B. (The error is detected in the standby system.)	Stop	Restart the standby system CPU module.	—	At power-on, at RESET, at tracking cable connection
3741H	Redundant system error	• Both systems were set as system A or system B at start-up. • The CPU module waiting for a start-up of the other system detected that the system A/B setting is not set to the CPU module of the other system.	Stop	• Set one system as system A and the other system as system B by performing online operation. Then, restart the CPU modules in both systems. • Set system A or B to the CPU module with no system setting, and restart the CPU modules in both systems. • Restore the backup data including the system A/B settings to set one system as system A and the other system as system B.	—	At power-on, at RESET, at tracking cable connection
3742H	Redundant system error	The systems were started without system A/B setting.	Stop	Set one system as system A and the other system as system B by performing online operation. Then, restart the CPU modules in both systems. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, or tracking cable. Please consult your local Mitsubishi representative.	—	At power-on, at RESET
3743H	Redundant system error	The system setting (system A or B) was not determined.	Stop	Set one system as system A and the other system as system B by performing online operation. Then, restart the CPU modules in both systems.	—	At power-on, at RESET
3744H	Redundant system error	Both systems are set as control systems. (The error is detected in the system B (control system).)	Stop	Restart the CPU module in the system B.	—	At tracking cable connection

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3745H	Redundant system error	• The systems were started with one CPU module set to redundant mode and the other CPU module set to process mode. • The systems were started in a state where the CPU modules with firmware versions that cannot be used together are mounted. • The systems were started with the Process CPU and the SIL2 Process CPU.	Stop	When using the CPU modules with firmware versions that cannot be used together, replace either CPU module with the one that can be used together, referring to the MELSEC iQ-R Module Configuration Manual. When using the Process CPU and the SIL2 Process CPU together, replace either CPU module so that the same CPU module models are used in both systems. Then, write a project to the CPU module in process mode or to the one after replacement and restart the CPU modules in both systems.	—	At power-on, at RESET
3746H	Tracking communications disabled	Data communications with the other system could not be performed within the period of time set in the CPU parameters ("Other System Start-up Timeout Setting").	Stop	• If the other system CPU module is powered off, restart the CPU modules in both systems. • If a WDT error has occurred in the other system CPU module, eliminate the error cause, and restart the system. • Connect tracking cables securely and properly to the connectors of the redundant function modules in both systems, and restart the CPU module(s) with the error. (One cable shall be connected between the IN connector of the module in system A and the OUT connector of the module in system B. The other cable shall be connected between the OUT connector of the module in system A and the IN connector of the module in system B.) • Set a longer time in the CPU parameters ("Other System Start-up Timeout Setting"). • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.	—	At power-on, at RESET
3747H	Redundant system error	The system cannot be started as a control system because the CPU module of the other system has been stopped due to a stop error.	Stop	Eliminate the error cause, and restart the CPU modules in both systems.	—	At power-on, at RESET
3748H	Redundant system error	• The systems were started in a state where the CPU modules with firmware versions that cannot be used together are mounted. • The systems were started with the Process CPU and the SIL2 Process CPU. • In a redundant system with redundant extension base unit, the extension cable is not connected between the main base unit of one system and the extension base unit. Or, the extension base unit cannot be recognized by the other system. (The error is detected in the control system or the standby system.)	Stop	• Replace one CPU module with the one having a firmware version that can be used with the other CPU module together, referring to the MELSEC iQ-R Module Configuration Manual. Then, restart the system. • Mount the same CPU module, and restart the CPU module. • Check the system configurations of both systems and correct them. • Connect the extension cable securely and properly to the connector of the redundant extension base unit, and then restart the CPU module in which the error has been detected. If the same error code is displayed again, the possible cause is a failure of the extension cable. Replace the extension cable.	—	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3749H	Redundant system error	Both systems are set as control systems. (The error is detected in both systems.)	Stop	• Take measures to reduce noise. • Reset the CPU modules in both systems and run them again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module of the own or other system. Please consult your local Mitsubishi representative.	—	At END instruction execution
3760H	Control system CPU module error	In backup mode, a stop error of the control system CPU module was detected. (The error is detected in the standby system.)	Stop	Eliminate the error cause, and restart the CPU modules in both systems.	—	Always
3780H	Redundant system configuration error	• A network module that does not support the redundant system function is mounted on the main base unit. • A network module with a firmware version that does not support the redundant system function is mounted on the main base unit. • An Ethernet interface module whose network type of the port 1 and/or port 2 is set to an option other than "Ethernet" is mounted on the main base unit.	Stop	• Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed slot number. If the module does not support the redundant system function, remove the module from the base unit. • Check the firmware version of the corresponding module. If the module does not support the redundant system function, remove the module from the base unit. • Set the network type of both port 1 and port 2 to "Ethernet" in the system parameters ("I/O Assignment Setting"). If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or network module. Please consult your local Mitsubishi representative.	System configuration information	At power-on, at RESET
3781H	Redundant system configuration error	Two or more redundant function modules are mounted.	Stop	Mount only one redundant function module.	System configuration information	At power-on, at RESET
3782H	Redundant system configuration error	An extension base unit is connected.	Stop	Remove all extension base units.	System configuration information	At power-on, at RESET
3783H	Redundant system configuration error	In redundant mode, nine or more CC-Link IE Field Network modules or CC-Link modules are mounted.	Stop	Reduce the number of CC-Link IE Field Network modules and CC-Link modules to eight or less for each.	System configuration information	At power-on, at RESET
3784H	Redundant system configuration error	Although the control CPU is operating in redundant mode, the redundant function module is not mounted on the main base unit.	Stop	Mount the redundant function module on the main base unit, and restart the system. Or, set the CPU module to process mode, and restart the system. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	—	At power-on, at RESET, at STOP → RUN state
37A0H	Parameter error (redundant function)	The range set for tracking transfer in the CPU parameters exceeds the device range or the file register capacity set in the CPU parameters.	Stop	Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, correct the parameter setting corresponding to the displayed number, and take the following actions: (1) Check and correct the tracking transfer settings in the CPU parameters so that they will be within the device range or the file register capacity. (2) Check and correct the device settings or the file register capacity setting in the CPU parameters.	Parameter information	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
37A1H 37A2H	Parameter error (redundant function)	A device that cannot be used is specified for tracking transfer in the CPU parameters.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, correct the parameter setting corresponding to the displayed number, and write parameters to the CPU module again. • Take measures to reduce noise. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or redundant function module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET
37A3H	Parameter error (redundant function)	Total amount of data set for tracking transfer in the CPU parameters exceeds the maximum limit.	Stop	Set the tracking capacity of each block within the allowable range in the CPU parameters.	Parameter information	At power-on, at RESET
37A4H	Parameter error (redundant function)	In the CPU module which operates in redundant mode, parameters for the unsupported function have been set or parameters related to a redundant system have not been set.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, correct the parameter setting corresponding to the displayed number, and write parameters to the CPU module again. • Correct the operation mode set in the engineering tool. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Parameter information	At power-on, at RESET, at write
37A7H	Parameter error (redundant function)	Use of the redundant functions that are not supported by the modules is enabled in the parameters.	Stop	• Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool and correct the parameter setting corresponding to the displayed number. • Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, and check the module corresponding to the displayed slot number. If the module does not support the redundant function, remove the module from the base unit.	Parameter information, system configuration information	At power-on, at RESET
37C0H	Program error	The program includes an instruction that cannot be used in the operation mode set in the engineering tool.	Stop	Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) by clicking the [Error Jump] button, and delete the program. Then, write the program to the CPU module again.	Error location information	At write, at power-on, at RESET, at STOP → RUN state
37E0H	Stopped operation of the standby system due to the memory copy function	The operation of the standby system has been suspended because the memory copy function has been executed.	Stop	Power off and on or reset the CPU module after the memory data has been copied from the control system to the standby system.	—	At memory copy

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3C00H	Hardware failure	A hardware failure has been detected.	Stop	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module and SIL2 function module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 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
3C01H						
3C02H						At power-on, at RESET, at END instruction execution, at interrupt occurrence
3C03H						Always
3C0FH				■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module and SIL2 function module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, extended SRAM cassette, or SIL2 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.		
3C10H	Hardware failure	A hardware failure has been detected.	Stop	■CPU module • Take measures to reduce noise. • Check the mounting status of the CPU module and SIL2 function module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 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						At END instruction execution, at instruction execution

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3C12H	Hardware failure	• The waveform of the voltage out of the specified range has been detected in the power supply module. • A hardware failure has been detected in the power supply module, CPU module, base unit, or extension cable. (In a redundant power supply system, the error is detected when two power supply modules failed.)	Stop	• Check the waveform of the voltage applied to the power supply module. • Check the mounting status of the CPU module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module, CPU module, base unit, or extension cable. Please consult your local Mitsubishi representative.	Failure information	Always
3C13H	Hardware failure	A hardware failure has been detected.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3C14H	Hardware failure	A hardware failure has been detected.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3C15H 3C16H	Hardware failure	A hardware failure has been detected.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module or SIL2 function module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 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.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module or SIL2 function module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 function module. Please consult your local Mitsubishi representative.	Failure information	Always

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3C20H	Memory error	An error has been detected in the memory.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module or SIL2 function module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 function module. Please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET
3C21H						At END instruction execution, at power-on, at RESET
3C22H						
3C2FH				■For CPU module or SIL2 function module • Take measures to reduce noise. • Check the mounting status of the CPU module or SIL2 function module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 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.		Always
3C30H	Memory error	An error has been detected in the memory.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module or SIL2 function module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 function module. Please consult your local Mitsubishi representative.	Failure information	At instruction execution
3C31H						Always
3C32H						
3C33H						At power-on, at RESET
3C34H	Memory error	An error has been detected in the memory.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET

Error code	Error name	Error details and cause	Stop/ continue	Action	Detailed information	Diagnostic timing
3E00H	Operation circuit error	■For CPU module or SIL2 function module An error has been detected in the CPU module or SIL2 function module. ■Redundant function module An error has been detected in the redundant function module.	Stop	■For CPU module or SIL2 function module • Take measures to reduce noise. • Check the mounting status of the CPU module or SIL2 function module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 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
3E01H	Operation circuit error	An error has been detected in the CPU module or SIL2 function module.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module or SIL2 function module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 function module. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution, at power-on, at RESET
3E10H 3E11H	Circuit error	An error has been detected in the CPU module or SIL2 function module.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module or SIL2 function module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 function module. Please consult your local Mitsubishi representative.	Failure information	At END instruction execution, at power-on, at RESET
3E12H						At power-on, at RESET
3E20H	Program execution error	The entire program was executed without executing the END instruction.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module or SIL2 function module. • Format the memory. Write all files to the CPU module. Then, reset the CPU module and run it again If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 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.				At instruction execution
3E60H	Safety mutual monitoring error	A hardware failure has been detected in the mutual monitoring between the SIL2 Process CPU and SIL2 function module.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module or SIL2 function module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or SIL2 function module. Please consult your local Mitsubishi representative.	Failure information	At power-on, at RESET, at interrupt occurrence
3E61H						Always
3E68H	Hardware failure	A hardware failure has been detected.	Stop	• Take measures to reduce noise. • Check the mounting status of the CPU module. • Reset the CPU module and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always

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

These error codes are not stored in SD0 because they are not detected by the self-diagnostic function of the CPU module.

Error code	Error name	Error details and cause	Action
4000H	Common error	Serial communication sum check error	• Connect the serial communication cable correctly. • Take measures to reduce noise.
4001H	Common error	An unsupported request was executed. (The request was executed to the CPU module that does not support the request.)	• Check the command data of the SLMP/MC protocol. • Check the CPU module model name selected in the engineering tool. • Check the target CPU module model name. • Check that the target network number is not duplicated.
4002H	Common error	An unsupported request was executed.	• Check the command data of the SLMP/MC protocol. • Check the CPU module model name selected in the engineering tool. • Execute the request again. • If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.
4003H	Common error	Command for which a global request cannot be performed was executed.	Check the command data of the SLMP/MC protocol.
4004H	Common error	All the operations to the CPU module are disabled because of the following reason: • The CPU module is starting up.	Perform operations to the CPU module again after the start-up processing ends.
4005H	Common error	The amount of data handled according to a specified request is out of range.	Check the command data of the SLMP/MC protocol.
4006H	Common error	Initial communication has failed.	• When using serial communication, inquire of the external device manufacturer for support conditions. • When using serial communication, check the CPU module model name selected in the engineering tool. • When using Ethernet communication, shift the communication start timing.
4008H	Common error	The CPU module is BUSY. (The buffer is not vacant.)	Execute the request again after the specified period of time has elapsed.
400BH	Common error	A set value of data handled according to a specified request is invalid.	• Check the request data of a dedicated instruction. • Check the command data of the SLMP/MC protocol.
4010H	CPU module operation error	The request cannot be executed because the CPU module is running.	Set the operating status of the CPU module to STOP, and execute the request again.
4013H	CPU module operation error	The request cannot be executed because the CPU module is not in the STOP state.	Set the operating status of the CPU module to STOP, and execute the request again.
4021H	File related error	The specified drive (memory) does not exist or there is an error.	• Check the specified drive (memory) status. • Back up data in the CPU module, and then initialize the memory.
4022H	File related error	• The file with the specified file name or file No. does not exist. • The specified program block does not exist. ■When CPU Module Logging Configuration Tool is used The data logging was started in the following state. • A program name (program No.) that does not exist was specified.	• Check the specified file name and file No. • Check the specified program block name. ■When CPU Module Logging Configuration Tool is used • Check the specified program name.
4023H	File related error	The file name and file No. of the specified file do not match.	Delete the file and then recreate the file.
4024H	File related error	The specified file cannot be handled by a user.	Do not access the specified file.
4025H	File related error	The specified file is processing the request from another engineering tool.	Forcibly execute the request. Or, execute the request again after the processing being performed ends.

Error code	Error name	Error details and cause	Action
4026H	File related error	The file password set in advance to the target drive (memory) must be specified.	Specify the file password set in advance, and then access to the drive (memory).
4027H	File related error	The specified range is larger than the file size.	Check the specified range and access within that range.
4028H	File related error	The same file already exists.	Forcibly execute the request. Or, change the file name and execute the request again.
4029H	File related error	The specified file capacity cannot be obtained.	Review the specified file capacity, and execute the request again.
402AH	File related error	The specified file is abnormal.	Back up data in the CPU module, and then initialize the memory.
402BH	File related error	The request cannot be executed in the specified drive (memory).	Set the operating status of the CPU module to STOP, and execute the request again.
402CH	File related error	The request cannot be executed currently.	Execute the request again after a while.
402FH	File related error	Writing of a file did not complete.	• Back up data internally in the CPU module, and delete the corresponding file or initialize the drive 4. Then, write the file to the programmable controller again. • Write the program restoration information. Then, read the program files from the programmable controller again.
4030H	Device specification error	• The specified device is not supported. ■When CPU Module Logging Configuration Tool is used • The data logging specifying a device that is not supported was started.	Check the specified device.
4031H	Device specification error	• The specified device number is out of range. • The CPU module does not support the specified device. ■When CPU Module Logging Configuration Tool is used • The data logging specifying a device number that does not exist was started.	• Check the specified device number. • Check the device assignment of the CPU module. • Check the specified device.
4032H	Device specification error	• The device modification was incorrectly specified. Or, the unusable device (TS, TC, SS, SC, CS, or CC) was specified in any of the following SLMP/MC protocol commands; Read random, Write random (in units of words), Entry monitor device, or Execute monitor. ■When CPU Module Logging Configuration Tool is used • The data logging specifying a device modification that is not supported was started.	• Check the device modification method. • Check the specified device.
4033H	Device specification error	Writing cannot be done because the specified device is for system use.	Do not write the data in the specified device, and do not turn on or off.
4034H	Device specification error	The dedicated instruction cannot be executed since the completion device for the dedicated instruction does not turn on.	Since the completion device for the SREAD or SWRITE instruction does not turn on in the CPU module on the target station, execute the instruction again after setting the operating status of the CPU module on the target station to the RUN status.
403FH	Device specification error	The link direct device specified cannot be executed.	Set the link direct device setting of the CPU parameter to the "Extended Mode (iQ-R Series Mode)".
4040H	Intelligent function module specification error	The request cannot be executed to the specified intelligent function module.	Check whether the specified module is the intelligent function module having the buffer memory.
4041H	Intelligent function module specification error	The access range exceeds the buffer memory range of the specified intelligent function module.	Check the start address and access number of points and access using a range that exists in the intelligent function module.
4042H	Intelligent function module specification error	The specified intelligent function module cannot be accessed.	• Check that the specified intelligent function module is operating normally. • Check the specified module for a hardware fault.
4043H	Intelligent function module specification error	• The intelligent function module does not exist in the specified position. ■When CPU Module Logging Configuration Tool is used • The data logging specifying a device that does not exist or cannot be accessed was started.	Check the I/O number of the specified intelligent function module.
4044H	Intelligent function module specification error	A control bus error occurred during access to the intelligent function module.	Check the specified intelligent function module and other modules and base units for a hardware fault.

Error code	Error name	Error details and cause	Action
4049H	Intelligent function module specification error	A request cannot be processed because the module extension parameter of the positioning module is being used for the positioning control.	Turn off module ready (Yn+0) of the positioning module or execute the processing again after removing the extension parameter of the positioning module from the target data.
4050H	Protect error	The request cannot be executed because the write protect switch of the SD memory card is on.	Turn off the write protect switch.
4051H	Protect error	The specified drive (memory) cannot be accessed.	Check the following and take action. • Is the drive (memory) the one that can be used? • Is the specified drive (memory) correctly installed?
4052H	Protect error	The specified file attribute is read only, so the data cannot be written.	Do not write data in the specified file. Or, change the file attribute.
4053H	Protect error	An error occurred when writing data to the specified drive (memory).	Check the specified drive (memory). Or, write data again after changing the corresponding drive (memory).
4054H	Protect error	An error occurred when deleting the data in the specified drive (memory).	Check the specified drive (memory). Or, delete data again after replacing the specified drive (memory).
4060H	Online registration error	• The online debug function (such as online change) and the data logging function are being executed with another engineering tool. • Data is being written to the flash ROM (data memory, program memory, and system memory) and the SD memory card. • The global label assignment information is being written to the programmable controller (data memory). ■When CPU Module Logging Configuration Tool is used • An attempt was made to write or delete data logging settings or to execute data logging to the setting registered by another request source.	• Execute the function again after the processing of the function executed from another engineering tool ends. • If the operation of another engineering tool is on hold, resume and finish the operation of another engineering tool, and then execute the function again. • Execute the function again after the completion of writing to the flash ROM and the SD memory card. If the same error code is displayed again, reset the CPU module.
4061H	Online registration error	• Settings for the online debug function (such as online change) are incorrect. • The CPU module is powered off or reset during the monitoring.	• Register an online debug function (such as online change and external input/output forced on/off), and then execute the function. • Execute again after checking the communication route such as the communication cable. • Power on or reset the CPU module and execute the monitoring again.
4063H	Online registration error	The registered number of locked files exceeded the maximum value.	Execute the request again after the file access from another engineering tool ends.
4064H	Online registration error	• The specified contents of the online debug function (such as 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. • The cassette set in the cassette setting in the CPU parameter differs from the one actually inserted. • An unsupported extended SRAM cassette is mounted. • In the CPU module which operates in redundant mode, the number of device points of the step relay (S) is set to other than 0 in the device setting of the CPU parameters.	• Check the device assignment of the CPU module or the device assignment of the request data. • Correct the cassette setting in the CPU parameters so that it matches with the cassette actually inserted. • Replace the extended SRAM cassette with the one supported by the CPU module. • Set the number of device points of the step relay (S) to 0 in the device setting of the CPU parameters.
4066H	Online registration error	The specified file password is incorrect.	Check and specify the correct file password.
4067H	Online registration error	Monitor communication has failed.	Execute again after checking the communication route such as the communication cable.
4068H	Online registration error	Operation is disabled because it is being performed with another engineering tool.	Execute the request again after the processing of the function executed from another engineering tool ends.
406AH	Online registration error	The drive (memory) number other than 0 to 4 was specified.	Check the drive and specify the drive number correctly.
406BH	Online registration error	Online operation was interrupted due to a CPU module error.	Check the status of the CPU module by executing the module diagnostics. Identify the error, and take a corrective action referring to the troubleshooting section.
406CH	Online registration error	The number of functions that can be executed simultaneously exceeds the limit.	Execute again after stopping the functions that are executed in another engineering tool.

Error code	Error name	Error details and cause	Action
406DH	Online registration error	The operation cannot be performed because the operation is performed from the same activation source.	Execute again after the operation from the same activation source has completed.
406EH	Online registration error	The specified operation cannot be maintained.	Check and correct the operation.
4070H	Verification error	• The program not yet corrected and the one corrected by online program change are different. • The execution program that was written to the programmable controller (including online change) or the execution program that was written by using online change operation differs from the program restoration information to be written.	• Read the program from the CPU module to match it with that of the engineering tool, and then execute the online change again. • Write the program including the program restoration information to the programmable controller (including online change), or execute online change.
4071H	Verification error	After the realtime monitor started, the CPU parameters in the CPU module has been changed or deleted.	• Start the realtime monitor again. • Write the project of GX Works3 at the time of the start of the realtime monitor to the CPU module.
4072H	Verification error	After the realtime monitor started, the global label setting file in the CPU module or the sequence program file having a monitoring target program name has been changed or deleted.	• Read the project of GX Works3 from the CPU module, save it, and load it into GX LogViewer. Then, start the realtime monitor again. • Write the project of GX Works3 at the time of the start of the realtime monitor to the CPU module. • Check and correct the SD940 setting.
4080H	Any other error	• Request data error ■When CPU Module Logging Configuration Tool is used • Request or setting data error	• Check the request data that has been specified. ■When CPU Module Logging Configuration Tool is used • Check the specified data, and write it to the CPU module again.
4081H	Any other error	The search target data cannot be detected.	Check the data to be searched.
4082H	Any other error	The specified command cannot be executed because it is being executed.	Execute the command again after the processing of the command from another engineering tool ends.
4083H	Any other error	• An attempt was made to perform operation to the program not registered in parameter. ■When CPU Module Logging Configuration Tool is used • The data logging specifying the program that is stored in the CPU module but not registered in the CPU parameters was started.	Register the program in parameter.
4084H	Any other error	The specified pointer (P or I) does not exist.	Check if the pointer (P or I) exist in the data.
4085H	Any other error	The pointer (P or I) cannot be specified because the program is not specified in parameter.	Register the program to be executed in parameter first, and specify the pointer (P or I).
4086H	Any other error	The specified pointer (P or I) has already been added.	Check and correct the pointer number to be added.
4087H	Any other error	The number of pointers (P or I) exceeds its limit.	Check and correct the specified pointer (P or I).
4088H	Any other error	• The specified step number is not at the head of the instruction. • The program differs from that stored in the CPU module. ■When CPU Module Logging Configuration Tool is used • The data logging specifying the step number that does not exist or is not specified as the start number of the instruction was started.	• Check and correct the specified step number. • Read the program from the CPU module to match it with that of the engineering tool, and then execute the online change again.
4089H	Any other error	An attempt was made to insert/delete the END instruction by online program change.	• Check the specified program file contents. • Set the operating status of the CPU module to STOP, and write the program.
408AH	Any other error	The file capacity exceeded after the online change was executed.	• Check the capacity of the specified program file. • Set the operating status of the CPU module to STOP, and write the program.
408BH	Any other error	The remote request cannot be executed.	• Change the status of the CPU module so that the remote request can be executed, and execute the request again. • For remote operation, set the parameter to "Enable remote reset".
408DH	Any other error	The instruction code that cannot be handled exists.	• Check whether the model of the used CPU module is correct or not. • The program where online change was attempted includes the instruction that cannot be handled by the CPU module specified for the project. Check the program and delete the instruction.

Error code	Error name	Error details and cause	Action
408EH	Any other error	• The write step is illegal. • The program differs from that stored in the CPU module.	• Set the operating status of the CPU module to STOP, and write the program. • The starting position of online program change is not specified with the correct program step number. Check whether the engineering tool supports the model and version of the CPU module that is specified for the project. • Read the program from the CPU module to match it with that of the engineering tool, and then execute the online change again.
40A0H	SFC device specification error	A block No. outside the range was specified.	Check the setting and correct it.
40A1H	SFC device specification error	The number of blocks exceeds the range.	Check the set quantity and correct it.
40A2H	SFC device specification error	A step No. outside the range was specified.	Check the setting and correct it.
40A3H	SFC device specification error	The number of steps exceeds the range.	Check the set quantity and correct it.
40A4H	SFC device specification error	A sequence step No. outside the range was specified.	Check the setting and correct it.
40A5H	SFC device specification error	The specified device is outside the range.	Check the set quantity and correct it.
40A6H	SFC device specification error	The block specification pattern and step specification pattern are incorrect.	Check the setting and correct it.
40A7H	SFC device specification error	■R00CPU, R01CPU, R02CPU: • A block No. that does not exist in the 0 to 127 range was specified. ■CPU module other than above models: • A block No. that does not exist in the 0 to 319 range was specified.	Check the setting and correct it.
40A8H	SFC device specification error	■R00CPU, R01CPU, R02CPU: • A step No. that does not exist in the 0 to 127 range was specified. ■CPU module other than above models: • A step No. that does not exist in the 0 to 511 range was specified.	Check the setting and correct it.
40B0H	SFC file related error	The drive (memory) specified with the SFC program file operation is incorrect.	Check the setting and correct it.
40B1H	SFC file related error	The SFC program specified with the SFC program file operation does not exist.	Check the specified file name and correct it.
40B2H	SFC file related error	The program specified with the SFC program file operation is not an SFC program.	Check the specified file name and correct it.
40B3H	SFC file related error	An attempt was made to rewrite a dedicated SFC instruction by changing an SFC program online.	Set the operating status of the CPU module to STOP, and write the SFC program.
40B4H	SFC file related error	An attempt was made to change or delete the active block.	Inactivate the target block.
40B5H	SFC file related error	The number of SFC steps after the program change exceeded the maximum.	Reduce the number of SFC steps to be added and execute the online change again.
40B6H	SFC file related error	The specified block does not exist.	Read the program from the CPU module to match it with that of the engineering tool, and then execute the online change again.
40B7H	SFC file related error	The online change cannot be executed to the standby type SFC program.	After setting the operating status of the CPU module to STOP, write the SFC program to the programmable controller.
40B8H	SFC file related error	The device number of SFC information device is outside the range.	Check and correct the setting of the block information.
40B9H	SFC file related error	The changed SFC program is incorrect.	Execute again after checking the communication route such as the communication cable.
40BAH	SFC file related error	The online change for each block cannot be executed to the SFC block whose number of sequential steps exceeds 32K steps.	• To execute the online change for each block, the number of sequential steps of the target SFC block must be within 32K steps. • After setting the operating status of the CPU module to STOP, write the SFC program to the programmable controller.

Error code	Error name	Error details and cause	Action
40BBH	SFC file related error	The online change cannot be executed because writing to the programmable controller in the STOP state has just completed or a program execution error has occurred.	• After the operating status of the CPU module is changed from STOP to RUN, execute the online SFC block change or the online program change. • After preventing the program execution error from occurring, execute the online SFC block change or the online program change.
40BDH	SFC file related error	Online change (SFC block) execution error	• Read the program from the CPU module to match it with that of the engineering tool, and then execute the online change again. • After setting the operating status of the CPU module to STOP, write the SFC program to the programmable controller.
40BEH	SFC file related error	The program cannot be changed online because the part to be changed has an active step (a step holding operations).	• Omit the step from the part. • Deactivate the step.
40C0H	Label communication error	The specified label name does not exist.	• Check the label setting. If the "Access from External Device" checkbox is not selected, check the checkbox.
40C1H	Label communication error	Label access with a label name has failed because the specified array element number is outside the range.	Specify the array element number within the set range.
40C2H	Label communication error	Label access with a label name has failed because the bit array type label is not specified by bit specification.	Specify the label by bit specification, and access again.
40C3H	Label communication error	Label access with a label name has failed because the word array type label is not specified by word specification.	Specify the label by word specification, and access again.
40C4H	Label communication error	Label access with a label name has failed because the number of labels used in the program exceeded its limit.	Reduce the number of labels in the program, and access the labels multiple times.
40C5H	Label communication error	• Label access with a label name has failed because the global label setting file and global label assignment information do not match. • The request execution has failed because the global label is being modified by the CPU module specified.	• Check the access from an external device of the specified global label setting, and write the global label setting file and the global label assignment information to the programmable controller together. • Execute the label access again after a while.
40C6H	Label communication error	The request execution has failed because the global label is being modified by the CPU module specified.	After a while, write the data to the programmable controller/execute online change or execute the label access again.
40C7H	Label communication error	• Although changes of the global label were written to the programmable controller, the reflection operation (STOP → RUN or Power-on and RESET) of the global label setting file is not performed. • The request cannot be executed because the data in the global label setting file being processed and the specified consistency check data are not the same.	• Perform the reflection operation (STOP → RUN or Power-on and RESET) of the global label setting file. • Correct the details of the global label setting, and write the file to the programmable controller again.
40C8H	Label communication error	Registration of a label definition has failed because the number of registered labels reached its limit.	Reduce the number of registered labels with the "Access from External Device" checkbox selected.
40C9H	Label communication error	A label definition could not be registered because the registration target memory capacity has been exceeded its limit.	• Reduce the number of registered labels with the "Access from External Device" checkbox selected. • If the data memory is specified as the registration target memory, change the memory to the SD memory card. • Change the settings of the functions that use the registration target memory.
40CAH	Label communication error	• A label definition could not be changed, added, or deleted because the label communication data does not exist in the CPU module. ■When CPU Module Logging Configuration Tool is used • An attempt was made to change, add, or delete labels without creating the label communication data.	Write the label communication data to the programmable controller.
40CBH	Label communication error	Data are not written to the CPU module because the data type of the specified label does not match the size of the write data.	Change the size of data written from the external device (SLMP/MC protocol device) so that it matches the data type of the label in the specified CPU module.
40CCH	Label communication error	Online program change has failed because the global label setting file before modification and the global label assignment information do not match.	Write the global label setting file and the global label assignment information to the programmable controller together after modification.

Error code	Error name	Error details and cause	Action
40CEH	Label communication error	An attempt was made to access a label that cannot be accessed with a label name.	• Change the data type of the specified label to the one other than "Function Block" or "Pointer". • Change the class of the specified label to the one other than "VAR_GLOBAL_CONSTANT". • Specify a device that is not being modified. (Bit-specified word devices and digit-specified bit devices can be specified.) • Change the data type of the specified label to the one other than "Bit and two-dimensional array" or "Bit and three-dimensional array".
40D0H	Label communication error	The target CPU module does not support "Access from External Device" of "Global Label Setting".	Disable "Access from External Device", and then write the data to the programmable controller again.
40D1H	Label communication error	When the byte is specified for the unit for reading/writing data, an odd-numbered value is specified for the read/write array data length.	Specify an even-numbered value for the read/write array data length.
4100H	Any other error	Hardware failure of the CPU module.	Replace the CPU module.
4101H	Any other error	Serial communication connection was executed for a different CPU module series.	Check the CPU module series.
4103H	Any other error	The instruction written by online program change is incorrect or invalid.	Execute online program change again. Or, set the operating status of the CPU module to STOP, and write the program.
4104H	Any other error	An instruction(s) that cannot be executed by the CPU module set to process mode or redundant mode is included in the instructions targeted for online program change.	Delete the instruction(s) that cannot be executed, and execute the online program change function again. Or, set the operating status of the CPU module to STOP, and write the program.
4105H	Any other error	Hardware failure of the CPU module internal memory	The possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.
4108H	Any other error	The device monitor/test cannot be performed normally.	Execute the function again. Check that the access prohibited area is not accessed, and execute the function again.
410AH	Any other error	• The specified command cannot be executed because the online program change is being executed. ■When CPU Module Logging Configuration Tool is used • The data logging where a step No. is specified as the collection start condition or trigger condition was started during the online program change. • The data logging where a label (global label or local label) is specified as the collection start condition, data collection target, or trigger condition was started during the online program change. • The data logging setting file where a label (global label or local label) is specified as the collection start condition, data collection target, or trigger condition was written during the online program change.	Execute the request again after the online program change.
410BH	Any other error	The monitor condition registration was cleared after the online program change was executed.	Execute the registration of monitoring condition again after the online program change.
410CH	Any other error	Writing to the specified data is not supported.	• Check that the version of the engineering tool used is correct. • Check the setting and correct it.
410EH	Any other error	When the execution status of the online program change is in error, the online program change command was issued.	Issue the online program change cancel command, and execute the function again.
410FH	Any other error	During the execution of the online program change function, the cancel command was issued by the same request source.	Issue the command again after the currently-performed processing ends.
4110H	CPU module error	The request cannot be executed because the CPU module is in a stop error state.	Reset the CPU module and execute the request again.
4111H	CPU module error	The request cannot be executed because the other CPU modules, except the host CPU module, have not yet started in a multiple CPU system.	Execute the request again after all the other CPU modules have started.
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.

Error code	Error name	Error details and cause	Action
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).
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.

Error code	Error name	Error details and cause	Action
4150H	File related error	An attempt was made to initialize the drive (memory) protected by the system.	Do not initialize the target drive (memory) as it cannot be initialized.
4151H	File related error	An attempt was made to delete the file/folder protected by the system.	Do not delete the target file as it cannot be deleted.
4160H	Online registration error	The registered number of I/O devices of the forced on/off target exceeded the maximum.	Cancel the registration of I/O devices of the forced on/off target that is not used.
4168H	Online registration error	The number of registered settings of the device test with execution conditions has exceeded 32.	Disable the settings of the device test with execution conditions registered in the CPU module. Alternatively, reduce the number of executional conditioned device tests to be registered at a time.
4169H	Online registration error	No settings of the device test with execution conditions are registered.	Check the number of registered settings of the device test with execution conditions in the CPU module, and disable the settings.
416AH	Online registration error	The specified execution conditions do not exist. (Device test with execution conditions)	Check whether the execution conditions (program block, step No., and execution timing) specified for disabling settings are registered in the CPU module.
416BH	Online registration error	Other than the ladder program was specified for the registration of the device test with execution conditions.	Check and correct the program block specified when the settings of the device test with execution conditions are registered or disabled.
41C1H	File related error	The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41C2H	File related error	File open specification data for file access is wrong.	Execute again after checking the specification data.
41C3H	File related error	Simultaneously accessible files exceeded the maximum.	Execute again after decreasing file operations.
41C4H	File related error	Simultaneously accessible files exceeded the maximum.	Execute again after decreasing file operations.
41C5H	File related error	• The specified file does not exist. ■When CPU Module Logging Configuration Tool is used • When an attempt was made to re-register the data logging with the previous settings, the corresponding file did not exist.	Execute again after checking the file.
41C7H	File related error	The specified file/folder or drive (memory) does not exist.	Execute again after checking the file/folder or drive (memory).
41C8H	File related error	The size of the specified file has exceeded that of the existing file.	• Execute again after checking the size of the specified file. • If the error recurs after re-execution, the file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41C9H	File related error	• Access to the file sector has failed. • The format information data of the target drive (memory) is abnormal.	Back up data in the CPU module, and then initialize the memory.
41CAH	File related error	• Access to the file sector has failed. • The format information data of the target drive (memory) is abnormal.	Back up data in the CPU module, and then initialize the memory.
41CBH	File related error	The file name is specified in a wrong method.	Execute again after checking the file name.
41CCH	File related error	• The specified file does not exist. Or, the specified subdirectory does not exist. ■When CPU Module Logging Configuration Tool is used • The data logging was started in a state that sub-folders for storing data logging files (or folders) cannot be created or accessed. Or, sub-folders cannot be created or accessed while the data logging is being performed or the logged data is being saved.	Execute again after checking the name of the file and subdirectory.
41CDH	File related error	• An access to the file is prohibited in the system. ■When CPU Module Logging Configuration Tool is used • The data logging was started in a state that files (or folders) cannot be created or accessed because a file (or folder) with the same name exists. Or, files (folders) cannot be created or accessed while the data logging is being performed or the logged data is being saved.	• Do not access the specified file or subdirectory. • Execute again after checking the file and subdirectory. • Execute again after checking the open mode of the file.
41CEH	File related error	The file cannot be written because the specified file is read-only.	Execute again after checking the attribute of the specified file.

Error code	Error name	Error details and cause	Action
41CFH	File related error	The specified drive (memory) has been used exceeding the capacity.	Execute again after checking the drive (memory) capacity.
41D0H	File related error	The specified drive (memory) has no free space. Or, the number of files in the directory of the specified drive (memory) has exceeded the maximum.	• Execute again after increasing the free space of the drive (memory). • Delete files in the drive (memory), and execute the function again.
41D1H	File related error	• The file name is specified in a wrong method. • The SD memory card is disabled by SM606 (SD memory card forced disable instruction).	• Execute again after checking the file name. • If the error recurs after re-execution, the file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory. • Cancel the SD memory card forced disable instruction.
41D5H	File related error	The file of the same name exists.	Forcibly execute the request, or execute after changing the file name.
41D6H	File related error	The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41D7H	File related error	The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41D8H	File related error	The specified file is being accessed.	Execute again after a while.
41DFH	File related error	The specified drive (memory) is write-protected.	Execute again after canceling the write protect of the specified drive (memory).
41E0H	File related error	The specified drive (memory) is abnormal or does not exist.	• Execute the operation again after checking that the SD memory card has been inserted. • Back up data, and then initialize the memory.
41E1H	File related error	Access to the SD memory card has failed.	Back up data, and then write the data to the data memory.
41E4H	File related error	Access to the SD memory card has failed.	• Execute the operation again after checking that the SD memory card has been inserted. • Execute the operation again after replacing the SD memory card. • Back up data, and then initialize the memory.
41E7H	File related error	The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41E8H	File related error	The format information data of the specified drive (memory) is abnormal.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
41E9H	File related error	The specified file is being accessed.	Execute again after a while.
41EBH	File related error	The file name is specified in a wrong method.	Execute again after checking the file name.
41ECH	File related error	The file system of the specified drive (memory) is logically corrupted.	• The file information data may be corrupted. • Back up data in the CPU module, and then initialize the memory.
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.

Error code	Error name	Error details and cause	Action
41F8H	File related error	The data is being accessed from another engineering tool.	Data are being written to the program memory or being transferred to the backup memory. Access the file after the currently-performed processing ends.
41FAH	File related error	Program was written beyond the area where the program can be executed.	Execute again after reducing either the already written program or newly written program.
41FBH	File related error	The specified file is already being manipulated from the engineering tool.	Execute again after the currently performed operation is completed.
41FCH	File related error	• An attempt was made to initialize the drive (memory) being used.	Stop all accesses to the specified drive (memory), and execute the request again.
41FDH	File related error	There are no data written to the data memory.	Write all the required files to the programmable controller.
41FEH	File related error	• An SD memory card is not inserted. • The SD memory card is being disabled. • The SD memory card is disabled by SM606 (SD memory card forced disable instruction). ■When CPU Module Logging Configuration Tool is used • The data logging was started when the CPU module is in the following state: no SD memory card is inserted; the CARD READY.LED is not on (green); or the SD memory card is forcibly disabled.	• Insert the SD memory card. • Remove the SD memory card, and insert it again. • Cancel the SD memory card forced disable instruction.
41FFH	File related error	The type of the SD memory card differs.	Check the type of the SD memory card.
4200H	Online module change function error	In a redundant system with redundant extension base unit, the system switching occurred when the online module change is in progress.	Check SD1617 (Online module change progress status) with the CPU module of the control system and resume the online module change operation.
4201H	Online module change function error	In a redundant system with redundant extension base unit, the request cannot be executed because a module on the main base unit of the control system is being changed online.	Issue the request again after the online module change processing ends.
4202H	Online module change function error	• The request cannot be executed because the specified module is being changed online. • The request cannot be executed because the redundant function module is restarting.	• Issue the request again after the online module change processing ends. • If the online module change operation cannot be completed, power off the system, and change the module. • Issue the request again after the restart of the redundant function module has completed.
4203H	Online module change function error	In a redundant system with redundant extension base unit, modules on the main base unit of the standby system cannot be changed online.	Power off the standby system before changing modules on the main base unit of the standby system online.
4204H	Online module change function error	In a redundant system with redundant extension base unit, the online module change cannot be executed to the modules on an extension base unit from the CPU module of the standby system.	Execute the online module change from the CPU module of the control system.
421DH	Online module change function error	Redundant operations cannot be performed because the redundant function module is being changed online or is restarting.	Perform redundant operations after the online module change processing ends or the restart has completed.
4240H	Redundant system related error	Any of the following operations is requested to the CPU module in the standby system, but cannot be performed because they are not supported. • Operation mode change • System switching • Memory copy from control system to standby system • Control system forced start-up while waiting for a start-up of the other system • External input/output forced on/off function	Perform these operations to the CPU module in the control system.

Error code	Error name	Error details and cause	Action
4241H	Redundant system related error	Data communications cannot be performed with the other system because of any of the following reasons: • The CPU module of the other system is powered off or in a reset state. • The tracking communication stops due to an error in the CPU module of the own or other system. • Tracking cables are disconnected or incorrectly connected, or failed. • The redundant function module of the own or other system is being changed online, a module communication test is being performed for the module, a hardware failure has occurred in the module, or the module is restarting. • System A/B has not been determined in the CPU modules.	• If a WDT error has occurred in the CPU module of the own or other system, eliminate the error cause and then execute the operation again. • Set the own system as system A or B and the other system as the other. • Check that there is no error or failure in the CPU module, tracking cables, redundant function modules, and perform a retry. Or, perform a retry after the online module change processing or the module communication test ends. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.
4243H	Redundant system related error	The request cannot be executed because a stop error has occurred in the CPU module of the standby system.	Check the stop error, eliminate the error cause, and then execute the request again.
4244H	Redundant system related error	The request cannot be executed because the operating status of the CPU module in the control system differs from that of the CPU module in the standby system.	Match the operating status of the CPU module between the systems, and execute the request again.
4246H	Redundant system related error	The request cannot be executed because the operation mode is being changed or the systems are being switched.	Execute the request again after the operation mode change processing or the system switching processing ends.
4247H	Redundant system related error	The request cannot be executed because the memory copy function is being executed.	Execute the request again after the memory copy processing ends. The completion status can be checked in the following special relay areas. • SM1654 (Memory copy being executed): Off • SM1655 (Memory copy completion): On
4248H	Redundant system related error	Data communications cannot be performed with the system specified as a connection destination (request destination module I/O number) because of either of the following reasons: • The request was issued during the system switching processing. • The system specified as a connection destination (request destination module I/O number) does not exist.	Check the following, and execute the request again. • The system switching processing has completed. • The redundant system has started up successfully. • When the memory copy with the special relay and special register is used, 03D1H (Standby system CPU module) is stored in SD1653.
4249H	Redundant system related error	The request cannot be executed because the system setting (system A or B, control system or standby system) has not been determined.	Set one system as system A or control system and the other system as system B or standby system, and execute the request again. Or, do not specify any connection destination system (request destination module I/O number), and execute the request again.
424AH	Redundant system related error	The request cannot be executed because the system A, system B, control system, or standby system is set as a connection destination (request destination module I/O number).	Do not specify any connection destination system (request destination module I/O number), and execute the request again.
424BH	Redundant system related error	The request cannot be executed because system switching is disabled because of either of the following reasons: • SM1646 (System switching by a user) is off. • The DCONTSW instruction is being executed.	Perform either of the following operations to enable system switching, and execute the request again. • Turn on SM1646 (System switching by a user). • Execute the ECONTSW instruction.
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.

Error code	Error name	Error details and cause	Action
4256H	Redundant system related error	The request cannot be executed because a timeout error has occurred in tracking communications.	Check if the tracking cables have been properly connected. If the same error code is displayed again even after the tracking cables are connected properly, the possible cause is a hardware failure of the CPU module, redundant function module, or tracking cable. Please consult your local Mitsubishi representative.
4258H	Redundant system related error	The operation mode cannot be changed because the control system in separate mode is in a state waiting for transition to RUN.	Change the operating status of the CPU module whose PROGRAM RUN LED is flashing to RUN by using the RUN/STOP/RESET switch or remote operation, and change the operation mode again.
4259H	Redundant system related error	The operation mode cannot be changed from separate mode to backup mode because the communication route differs from that of when the mode was changed from backup mode to separate mode.	Change the operation mode using the same communication route of when the mode was changed from backup mode to separate mode.
425AH	Redundant system related error	The system cannot be started as the control system because of either of the following reasons: • The system A/B setting has not been determined. • The system is being changed to the control system. • A start-up operation has been performed in the standby system.	Take the following actions: • Set one system as system A and the other system as system B, restart the CPU module. • Check that the system has been changed to the control system. • Check the system status (control system/standby system).
425BH	Redundant system related error	In a redundant system with redundant extension base unit, a function that is not supported by an intelligent function module on an extension base unit has been executed via the intelligent function module.	• Execute the function via a module on the main base unit. • Check the command data of the SLMP/MC protocol.
425EH	Redundant system related error	Data communications cannot be performed with the other system because a module communication test is being executed on a redundant function module.	Retry data communications after the module communication test ends.
425FH	Redundant system related error	The memory copy function cannot be executed because the CPU module models differ between the control system and the standby system.	Match the CPU module model between the systems, and execute the function again.
4269H	Any other error	The remote RUN (function) cannot be executed.	Execute the function again after a while.
426AH	Any other error	The date and time data specified by an engineering tool (personal computer) are less than one hour from the start time of daylight saving time.	• Specify the time to other than one less than one hour from the start time of daylight saving time using an engineering tool (personal computer). Then, execute the function again. • Check and Correct the start time of the daylight saving time setting.
426CH	Any other error	The specified operation cannot be performed.	Check the settings or the operating status of the recording function.
426DH	Any other error	The operation cannot be performed because the recording function is running.	Perform the operation after the recording function stops.
4270H	Debug related function error	Data logging is being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data) to another memory.	• Register data logging to the memory where the data logging is being performed. Or, stop the data logging being performed and register again. ■When CPU Module Logging Configuration Tool is used • Start the data logging to the memory where the data logging is being performed. Or, stop the data logging being performed, and start the data logging.
4271H	Debug related function error	The specified data logging is already being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).	Stop the data logging. Or, write, delete, or register data logging to the setting number where no data logging is being performed.

Error code	Error name	Error details and cause	Action
4272H	Debug related function error	The trigger logging specifying the device as a trigger condition is being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).	• Change the trigger condition. Or, stop the trigger logging being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data), and register another trigger logging. ■When CPU Module Logging Configuration Tool is used • Change the trigger condition. Or, stop the trigger logging being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data), and start another trigger logging.
4275H	Debug related function error	Auto logging is being performed.	After the auto logging, replace the SD memory card and execute again.
4276H	Debug related function error	The function that cannot be executed during the data logging (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data) was executed.	Stop the data logging, and then execute the function.
4277H	Debug related function error	• The number of saved files exceeded the specified number. ■When CPU Module Logging Configuration Tool is used • The data logging was started in a state where the number of saved files has exceeded the specified number. (The operation when the number of saved files exceeded is set to "Stop".) Or, the data logging was started in a state where the number of saved files has exceeded the specified number. (The operation when the number of saved files exceeded is set to "Overwrite".)	• The number of files saved in the storage destination memory has exceeded the setting value. Delete files, or change the storage destination and then register. ■When CPU Module Logging Configuration Tool is used • The number of files saved in the storage destination memory has exceeded the setting value. Delete files or change the storage destination, and then start the data logging.
4278H	Debug related function error	• An attempt was made to register data logging in a state where the saved file number has reached its maximum, FFFFFFFF. Or, the number reached to the maximum during the execution. ■When CPU Module Logging Configuration Tool is used • The data logging was started in a state where the saved file number has reached its maximum, FFFFFFFF. Or, the number reached to the maximum during the execution.	• The saved file number in the storage target memory has reached its maximum, FFFFFFFF. Delete files, or change the storage destination and then register. ■When CPU Module Logging Configuration Tool is used • The saved file number in the storage target memory has reached its maximum, FFFFFFFF. Delete files or change the storage target memory, and then perform the data logging.
427AH	Debug related function error	• The common setting file is corrupted. ■When CPU Module Logging Configuration Tool is used • The data logging was started to the memory where the corrupted common setting file exists.	• Write the common settings to the target memory again. ■When CPU Module Logging Configuration Tool is used • Write the common settings to the target memory again.
427BH	Debug related function error	• The data logging with the same file storage destination is being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data). ■When CPU Module Logging Configuration Tool is used • The data logging with the same file storage destination is being performed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).	• Stop the data logging destined for the same storage, and then register. Or, change the storage destination of the file, and then register. ■When CPU Module Logging Configuration Tool is used • Stop the data logging destined for the same storage, and then perform another data logging. Or, change the storage destination of the file, and then register.
4280H	File transfer function error	A file transfer test was executed from another CPU Module Logging Configuration Tool during execution of a file transfer test.	Execute the file transfer test again after the ongoing test is completed.
4282H	Debug related function error	The registration was performed with the internal buffer capacity set to 0.	Check and correct the internal buffer capacity setting.
4283H	Debug related function error	An attempt was made to register trigger logging in a state that the specified number of records before trigger has exceeded the number of records that can be collected within the internal buffer capacity.	• Check and correct the internal buffer capacity setting. • Reduce the number of records before trigger.

Error code	Error name	Error details and cause	Action
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.
4289H	Debug related function error	Items that cannot be set as data collection conditions are set.	Check and correct the data collection condition setting.
428AH	Data logging function error	The data logging has started while • Data is being written to the programmable controller online. • The global label assignment information is being written to the programmable controller (data memory).	• Start the data logging after the online data write to the programmable controller has completed. • Start the data logging after the writing of the global label assignment information to the programmable controller has completed.
428CH	Data logging function error	Execution fails because any of the following files does not exist. • CPU parameter file • Global label setting file • Program file with the target program name	Write the following files to the CPU module. • CPU parameter file • Global label setting file • Program file with the target program name
4290H	Data logging function error	The program file or global label setting file was changed during the realtime monitor where a label (global label or local label) is specified as a monitor target, timing condition, or trigger condition.	• Do not change the program file or global label setting file during the realtime monitor where a label (global label or local label) is specified. • Stop the realtime monitor where a label (global label or local label) is specified. • Check and correct the setting of the special register (SD940), and execute the function again.
4291H	Debug related function error	Online program change has been executed during the execution of the real-time monitor where a step number is specified as "Timing" or "Trigger condition".	After the completion of the online program change, check the step number specified and execute real-time monitor again.
4292H	Debug related function error	Real-time monitor does not begin during the execution of real-time monitor.	Start real-time monitor after the stop of the real-time monitor being executed.
4293H	Debug related function error	Execution fails because the internal buffer exceeds its maximum capacity.	Check and correct the settings of internal buffer capacity, and then try again.
4294H	Debug related function error	Execution fails because the settings of internal buffer capacity has been changed during the execution of functions that consume the internal buffer.	Deactivate the functions that consume the internal buffer, and then try again. Or, restore the internal buffer capacity to the previous state, and then try again.
42A3H	Safety related error	The safety operation mode specification details lie outside the range.	Change the safety operation mode again.
42A4H	Safety related error	An attempt was made to change the safety operation mode to TEST MODE while in TEST MODE, or to change the safety operation mode to SAFETY MODE while in SAFETY MODE.	Check whether an attempt was made to change the current safety operation mode to the same mode.
42A5H	Safety related error	An operation that cannot be executed in SAFETY MODE was performed.	Change the safety operation mode to TEST MODE and execute again.
42A7H	Safety related error	The safety operation mode cannot be changed to SAFETY MODE.	• Change the CPU module operating status to other than RUN and execute again. • Execute again after completing writing of the safety program, safety FB file, safety CPU parameters, safety module parameters, safety global label setting file, and standard/safety shared label setting file.
42A8H	Safety related error	The safety operation mode cannot be changed to TEST MODE.	Change the safety operation mode again.
42A9H	Safety related error	A CRC error occurred during communication with the Safety CPU.	• Take measures to reduce noise, and retry data communications. • If the same error code is displayed again, the possible cause is a hardware failure of the applicable module. Please consult your local Mitsubishi representative.
42B0H		Communications between the CPU module and the SIL2 function module failed.	
42B4H		An error was detected in the safety mutual monitoring between the CPU module and the SIL2 function module.	

Error code	Error name	Error details and cause	Action
42B8H	Safety related error	The size of the program stored in the program memory and the size of the program being executed differ.	• Restart the Safety CPU. • If the same error code is displayed again, the possible cause is a hardware failure of the applicable module. Please consult your local Mitsubishi representative.
42B9H	Safety related error	The Safety CPU received a request from an unsupported device.	Send the request to the Safety CPU again from a supported device.
42BAH	Safety related error	An SD memory card was specified for the write destination for the safety sequence program, safety FB file, safety CPU parameters, safety module parameters, safety global label setting file, and standard/safety shared label setting file.	Try the write operation again.
42BBH	Safety related error	The maximum capacity for safety programs and safety FB files that can be stored in the specified drive (memory) has been exceeding.	• Delete the safety programs or safety FB files stored in the Safety CPU drive (memory), and execute again. • Execute again after formatting Safety CPU drive (memory). • Reduce the safety program and safety FB file to be written to less than 40K steps and execute again.
42BCH	Safety related error	A file mismatch has been detected between the CPU module and the SIL2 function module.	Check that the SIL2 function module is mounted and format the memory. Write all files to the CPU module. Then, reset and run the CPU module and perform the processing again.
42BEH	Safety related error	The number of safety sequence programs, safety FB files, safety CPU parameters, safety module parameters, safety global label setting files, and standard/safety shared label setting files exceeded the maximum number of files that can be stored on the specified drive (memory).	Execute again after deleting the safety sequence programs, safety FB files, safety CPU parameters, safety module parameters, safety global label setting files, and standard/safety shared label setting files on the drive (memory).
42BFH	Safety related error	Initialization of the CPU module has been completed, but the SIL2 function module cannot be initialized.	To initialize the SIL2 function module, mount the module to the right of the CPU module. Power on and reset the CPU module. Then, initialize the module again.
42C0H	Redundant system related error	The SFC program was written by the online change to the CPU module that does not support the SFC program in the redundant mode.	Check that the SFC program is not written to the CPU module that does not support the SFC program in the redundant mode.
42C1H	Redundant system related error	Since the operation mode is changed or the systems were switched during the initial processing or the initial processing (when switched to RUN), it takes time to complete the processing.	• Check whether the processing ends or not after a while. • Take measures to reduce noise. • If the same error code is displayed again even after the redundant system is restarted, the possible cause is a hardware failure of the CPU module, redundant function module, or tracking cable. Please consult your local Mitsubishi representative.
42C2H	Redundant system related error	The memory copy function is executed to the CPU module that does not support the selection writing of program restoration information while no program restoration information is in the CPU module of the control system.	• Execute the memory copy after the program restoration information is written to the CPU module of the control system. • Replace the CPU module of the control system with the one that supports the selection writing of program restoration information.
42C3H	Redundant system related error	The request cannot be executed because the other system is not a standby system.	Change the other system to a standby system and try again.
42C4H	Redundant system related error	The request cannot be executed because the operation is not supported by the other system.	Replace the CPU module with the one that supports the requested operation and try again.
42C5H	Redundant system related error	• In a redundant system with redundant extension base unit, access to a module on an extension base unit is attempted when the standby system is specified in "Specify Redundant CPU" in the "Specify Connection Destination" window. • In a redundant system with redundant extension base unit, access to a module on an extension base unit, such as a file operation, is attempted either connecting the device directly to the CPU module of the standby system or via a module on the main base unit of the standby system. • In a redundant system with redundant extension base unit, access to other stations via a module on the main base unit of the standby system and a module on an extension base unit is attempted.	• Specify the control system in "Specify Redundant CPU" in the "Specify Connection Destination" window and access a module on an extension base unit. • Access a module on an extension base unit either connecting the device to the CPU module of the control system directly or via a module on the main base unit of the control system. • Access other stations via the control system in a redundant system with redundant extension base unit.

Error code	Error name	Error details and cause	Action
42DEH	Redundant system related error	The request cannot be executed because the safety operation modes of two CPU modules differ between the control system and standby system.	Match the safety operation modes of two CPU modules between the control system and standby system, and execute the request again.
42DFH	Redundant system related error	The request cannot be executed because an error has been detected in the communications between the CPU modules of the control system and the standby system.	Check if the tracking cables have been properly connected. If the same error code is displayed again even after the tracking cables are connected properly, the possible cause is a hardware failure of the CPU module, SIL2 function module, redundant function module, or tracking cable. Please consult your local Mitsubishi representative.
42F0H	Safety related error	The request cannot be executed because the safety operation mode change is in progress.	Execute the request again after completion of the safety operation mode change that is currently in progress.
42F3H	Safety related error	The request cannot be executed because the SIL2 function module cannot be accessed.	• Check that the SIL2 function module is mounted and execute the request again. • Check that the safety function module or SIL2 function module operates normally and execute the request again.
42F5H	Safety related error	A file mismatch has been detected between the CPU module of the request side and the specified CPU module or between the SIL2 function modules.	Check that the SIL2 function module is mounted and format the memory. Write all files to the CPU module. Then, reset and run the CPU module and perform the processing again.
4300H	Security function error (user authentication)	• An operation requiring authentication was executed when no authentication was received from the CPU module. • An operation requiring authentication was executed from a startup source or function that does not support user authentication.	• Log in to the CPU module. • Perform operation from a device which supports the user authentication function.
4309H	Security function error (user authentication)	The specified user name has not been registered in the CPU module.	• Register user management information in the CPU module. • Log in with a registered user name.
430CH	Security function error (user authentication)	• An unauthorized file access has been attempted by a user with an insufficient access level. • A file was accessed by an unauthorized user (access level) with the user authentication function enabled (file access control enabled). • The CPU module does not support the access level of the user.	• Access files within the authorized scope. • Files should be accessed by authorized users. • Files should be accessed by users with an access level supported by the CPU module.
430DH	Security function error (user authentication)	An operation requiring authentication was executed from a startup source that does not support the user authentication function with the user authentication function enabled.	Perform the operation from a device that supports the user authentication function.
4310H	Security function error (user authentication)	• Logon failed.	• Log on with a registered user name/password. • If the password authentication is disabled (lockout), wait for the specified time and try again.
4311H	Security function error (user authentication)	• Logon failed.	• Wait for a while and try again.
4314H	Security function error (user authentication)	A user authentication operation was performed with the user authentication function disabled.	Perform the operation after enabling the user authentication function.
4317H	Security function error (user authentication)	An attempt was made to change the user management information of the logged on user.	The operation should be performed after the logged on user logs off.
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 unauthorized operation has been attempted by a user with an insufficient access level. • The CPU module does not support the access level of the user.	• Check the access level of the user. • Files should be accessed by users with an access level supported by the CPU module.
431CH	Security function error (user authentication)	The number of simultaneously logged on users exceeds the maximum value, and therefore the system is unable to authenticate the applicable user.	• Unnecessary logged on users should log off. • Adjust the number of log-ons to within the maximum value.
431DH	Security function error (user authentication)	An operation that requires the authentication was performed with no user management information registered in the CPU module.	Register user management information in the CPU module.

Error code	Error name	Error details and cause	Action
431EH	Security function error (user authentication)	• The access level of the operator does not match with the level registered in the CPU module.	• Log on with a user whose access level is registered in the CPU module.
431FH	Security function error (user authentication)	• The operation has been performed from the engineering tool that does not support the user authentication mode.	• Use the engineering tool that supports the user authentication mode.
4320H	Security function error (user authentication)	• The log-off operation has been performed when the operation is specified for the both systems but the own system is in the log-off state.	• In the redundant mode, perform the log-off operation while it is specified for the other system.
4321H	Security function error (user authentication)	• The user authentication function has not been executed properly. • The user authentication function has been executed on the CPU module whose operation mode is the redundant mode while a different operation mode is set with the engineering tool. • The request cannot be executed because a password change or a user information change took a long time and a timeout error has occurred.	• Execute the request again after a while. • Set the operation mode to the redundant mode with the engineering tool and execute the function again. • Change the password or the user information again. • If the same error code is displayed again, the possible cause is a hardware failure of the target module. Please consult your local Mitsubishi representative.
4322H	Security function error (user authentication)	• The request cannot be executed because another user authentication function is being executed. • The request cannot be executed because the memory copy function to the other system is being executed. • The request cannot be executed because an error has occurred while another user authentication function was being executed. • The request cannot be executed because the user authentication synchronization function for both systems is being executed.	• Execute the request again after another user authentication function has been completed. • Execute the request again after the memory copy function to the other system has been completed. • Execute the request again after a while.
4323H	Security function error (user authentication)	• The log-off operation has been performed when the operation is specified for the both systems but the other system is in the log-off state.	• Perform the log-off operation while it is specified for the own system.
4324H	Security function error (user authentication)	• Logon failed. • The user information of the programmable controller does not match between the connected system and the other system.	• Log on with a registered user name/password. • Set the user information of the both systems again. • To match the user information with those of the control system, execute memory copy.
4325H	Security function error (user authentication)	• Log-on to the both systems cannot be performed because the number of users who have simultaneously logged on to the both systems exceeds the maximum number. • The user information of the programmable controller does not match between the connected system and the other system.	• Log on with a registered user name/password. • Set the user information of the both systems again. • To match the user information with those of the control system, execute memory copy.
4326H	Security function error (user authentication)	• An operation was performed from the engineering tool with the version where enhanced vulnerability-measures were not taken.	• Use the engineering tool with the version where enhanced vulnerability-measures are taken.
432CH	Security function error (user authentication)	• An operation was performed with the combination of the following systems: a system with a CPU module where the enhanced vulnerability-measures are not taken; a system with a CPU module where the "setting to communicate only with GX Works3 with the vulnerability-measures enhanced version" is enabled.	• In the connected system or in another system, disable the "setting to communicate only with GX Works3 with the vulnerability-measures enhanced version".
433CH	Maintenance and inspection error	The error was not cleared. (Error clear was performed during execution of error clear.)	• Execute again after a while. If the same error code is displayed again, the possible cause is a hardware failure of the target module. Please consult your local Mitsubishi representative.
433DH	Maintenance and inspection error	The target module cannot handle the error clear.	Check the target module. (Check the module in which the error occurred.)
4400H	Security function error	A file protected by a password has been opened without unlocking the password.	Enter a correct password and perform password authentication.
4401H	Security function error	• Read password authentication has failed when required. • The file password format is incorrect.	• Set a correct read password and perform password authentication. • Access the file with the correct method.
4402H	Security function error	• Write password authentication has failed when required. • The file password format is incorrect.	• Set a correct write password and perform password authentication. • Access the file with the correct method.

Error code	Error name	Error details and cause	Action
4403H	Security function error	Both passwords for reading and for writing that are set upon Create, Change, Delete, or Disable do not match the previous ones.	Set correct passwords for both reading and writing, and perform password authentication.
4404H	Security function error	A file error was detected before or after performing Create, Change, or Delete.	• Initialize the drive including the target file by initializing the memory. • Write the target file to the programmable controller again, and then register or cancel the file password.
4408H	Security function error	Password authentication failed when access was required.	Set a correct password and perform password authentication again.
4409H	Security function error	Password authentication failed when access was required.	Set a correct password and perform password authentication again 1 minute later.
440AH	Security function error	Password authentication failed when access was required.	Set a correct password and perform password authentication again 5 minutes later.
440BH	Security function error	Password authentication failed when access was required.	Set a correct password and perform password authentication again 15 minutes later.
440CH	Security function error	Password authentication failed when access was required.	Set a correct password and perform password authentication again 60 minutes later.
440DH			
440EH	Security function error	The security function was activated and password authentication cannot be performed.	Set a correct password and perform password authentication again after a certain period of time.
440FH	Security function error	An operation was performed to the firmware update prohibited file with a file password set.	Disable the file password setting.
4410H	Security function error	The file of the locked CPU module is accessed without the security key authentication.	• Register the security key which locks the CPU module to the engineering tool. • When the project is opened, lock the project with the security key which locks the CPU module. • When the CPU module is locked, the access control target file cannot be accessed using the following functions or external devices. · FTP server function · SLMP/MC protocol · GOT · EZSocket
4412H	Security function error	The security key cannot be registered to the CPU module due to the failure of the internal memory where the security key is registered. Or, the security key of the CPU module cannot be deleted.	Hardware failure of the CPU module. Replace the CPU module.
4413H	Security function error	Since the CPU module is locked and 32 engineering tools are reading and writing programs simultaneously, another engineering tool cannot read or write programs.	Wait until the number of engineering tools which are reading and writing programs decreases to 31 or less, and read or write programs.
4414H	Security function error	The request cannot be executed because the CPU module is locked.	Do not execute the request because it cannot be executed by the CPU module which is locked.
4415H	Security function error	The request cannot be executed because the CPU module is not locked.	Do not execute the request because it cannot be executed by the CPU module which is not locked.
4416H	Security function error	The request cannot be executed because the CPU module lock or unlock processing is being performed.	Execute the request again after the lock or unlock processing ends.
4417H	Security function error	An attempt was made to write or delete the security key with no extended SRAM cassette inserted in the CPU module.	Check that an extended SRAM cassette is correctly inserted, and then perform the operation again.
4418H	Security function error	The security key cannot be changed or deleted because any locked program exists in the CPU module.	Unlock all the programs in the CPU module.
4422H	Security function error	The access target CPU module does not support the security key information stored in the engineering tool.	Change the security key information version of the engineering tool in accordance with the version supported by the target CPU module.
4424H	Security function error	The security key set to the file written by executing the file batch online change function does not match the one registered to the CPU module (or extended SRAM cassette).	Set the same security key registered to the CPU module to the file, and then execute the file batch online change function.
4425H	Security function error	The file with a security key was written by executing the batch file online change function, but no security key is registered to the CPU module (or extended SRAM cassette).	Clear the security key set to the file, and then execute the online change function.
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.

Error code	Error name	Error details and cause	Action
4801H	iQ Sensor Solution related error	• When the iQ Sensor Solution data backup/restoration is requested, the target module does not exist. • When the iQ Sensor Solution data restoration is requested, the specified backup folder does not exist. • The value in the special register (SD) related to the iQ Sensor Solution data backup/restoration is out of range.	• Check and correct the value in the special register, and execute the function again. • Check that the backup data created by the MELSEC iQ-R series module is specified.
4802H	iQ Sensor Solution related error	The number of backup folders used by the iQ Sensor Solution data backup/restoration function has reached to its limit.	• Delete an existing backup folder(s), and execute the function again. • Set "Automatic specification (Folder deletion supported)" in the special register for setting a backup target folder number.
4803H	iQ Sensor Solution related error	The system file does not exist in the specified backup data, or the system file is corrupted.	Specify another backup data.
4804H	iQ Sensor Solution related error	The backup file does not exist in the specified backup data, or the backup file is corrupted.	Specify another backup data.
4805H	iQ Sensor Solution related error	• When the iQ Sensor Solution data backup function is executed, no backup target device exists. • The data backup function was executed to a device which does not support the function.	• Check and correct the value in the special register, and execute the function again. • Execute the function to a device which supports the function.
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. • The SD memory card is disabled by SM606 (SD memory card forced disable instruction).	• Insert or re-insert an SD memory card, and execute the function again. • Enable the SD memory card operation, and execute the function again.
4807H	iQ Sensor Solution related error	Data communications cannot be performed with a target device of the iQ Sensor Solution data backup/restoration function.	• Check the operation of the target device. • Check the network status between the target device, such as cables, hubs, and routers. • The line may be busy. Perform a retry after a while. • Check and correct the communication timeout time of data backup/restoration specified in the special register.
4808H	iQ Sensor Solution related error	• When the backup data is restored to the device supporting iQ Sensor Solution, data communications cannot be performed with the restoration target device. • When backup data is restored to the device supporting iQ Sensor Solution, the device of the specified backup data and the restoration target device do not match.	• Check the operation of the target device. • Check the connections and network status between the target device, such as cables, hubs, and routers. • The line may be busy. Perform a retry after a while. • Check and correct the specified backup data and the device supporting iQ Sensor Solution (manufacturer, model name, version) specified as the restoration target device.
4809H	iQ Sensor Solution related error	The data backup/restoration function of iQ Sensor Solution was executed to the module which does not support the function.	• Replace the module with the one which supports the function. • Execute the function to the module which supports the function.
480AH	iQ Sensor Solution related error	The function was executed to the CC-Link module whose operation had been switched from the standby master station to master station.	After powering off and on the master station and standby master station, execute the function again.
480BH	iQ Sensor Solution related error	The function was executed for the CC-Link module to which "Read Model Name of Slave Station" is not set in the setting of the automatic detection function of connected devices.	Select "Read Model Name of Slave Station" in the setting of the automatic detection function of connected devices, and execute the function again.
480CH	iQ Sensor Solution related error	The specified command cannot be executed because the automatic detection of connected device function of iQ Sensor Solution is being executed.	Execute the command again after the automatic detection processing ends.
480DH	iQ Sensor Solution related error	• The specified command cannot be executed because the communication setting reflection function of iQ Sensor Solution is being executed. • A communication timeout occurred in a command of a device supporting iQ Sensor Solution.	• Execute the command again after the communication setting reflection processing ends. • Check and correct the communication time check setting value using the engineering tool.
480EH	iQ Sensor Solution related error	• The specified command cannot be executed because the monitor function of iQ Sensor Solution is being executed. • The specified command cannot be executed because the sensor parameter read/write function of iQ Sensor Solution is being executed. • A communication timeout occurred in a command of a device supporting iQ Sensor Solution.	• Execute the function again after a while. • Execute the command again after the sensor parameter read/write processing ends. • Check and correct the communication time check setting value using the engineering tool.

Error code	Error name	Error details and cause	Action
4810H	iQ Sensor Solution related error	The series of the CPU module where the iQ Sensor Solution data restoration function is to be executed differs from the series of the CPU module where the iQ Sensor Solution data backup function was executed. Or the restoration target module differs from the backup target module.	Check the series of the CPU module where the data backup function was executed and the backup target module.
4811H	iQ Sensor Solution related error	The series of the CPU module where the iQ Sensor Solution data backup function is to be executed differs from the series of the CPU module where last time the function was executed. Or the last backup target module differs from the current backup target module.	• Execute the data backup function to the module where the data backup function was executed last time. • Set a different folder number.
4812H	iQ Sensor Solution related error	The functions that cannot be executed simultaneously with the data backup/restoration function of iQ Sensor Solution, such as the file transfer function (FTP server/client), are being executed.	Execute the function again after a while.
4902H	Any other error	The communications have stopped because in the simple CPU communication function, an error occurred in another simple CPU communication setting No. for which the same communication destination is specified.	Eliminate the error cause of the simple CPU communication setting No. with an error.
4905H	Any other error	The capacity of label area used exceeded its limit.	• Check the program, delete the unused local and global label definitions, compile the program, and then write the program to the programmable controller. • Change the label area size in parameter, compile the program, and then write the program to the programmable controller.
4906H	Any other error	The file register data cannot be cleared by either of the following reasons: • The QDRSET instruction was executed more than once after the CPU module was powered on or reset. • The file register setting parameter is set to the one other than "Use Common File Register in All Programs".	• If the QDRSET instruction is being executed, power off and on or reset the CPU module. • The operation cannot be performed if the file register setting parameter is not set to "Use Common File Register in All Programs".
4907H	Any other error	The target CPU module does not support "Access from External Device" of "Global Label Setting".	Disable "Access from External Device", and then write the data to the programmable controller again.
4908H	Any other error	The specified program execution type is not supported.	Check the program execution type specified.
4909H	Any other error	A second SFC program has been started while an SFC program was running.	Check the execution status of the SFC program.
490AH	Any other error	An SFC program is stopped while "Stop Mode" is set to "Hold output after stop".	Set "Stop Mode" to "Stop output after stop".
490BH	Any other error	A program has started or stopped while the CPU module is in the STOP or PAUSE state.	Set the CPU module to the RUN state and execute the program again.
490CH	Any other error	A control system execution program has started or stopped in CPU module of the standby system when the redundant system is in backup mode.	• Execute the program in the CPU module of the control system. • Check the name of the program specified.
490DH	Any other error	An SFC program has started or stopped when the operation mode of the CPU module is redundant mode.	Check the name of the program specified.
4A00H	Network error	• Access to the specified station cannot be made since the routing parameters, network number/station, or network station <-> IP-related information setting are not set to the start source CPU module and/or relay CPU module. • For routing via a multiple CPU system, the control CPU module of the network module for data routing or the CPU module for data routing has not started. • The third byte of the IP address (network number) specified for the IP communication test is the same as that of the CPU module where the test is executed. • The CPU module that performs IP packet transfer is not the control CPU of the CC-Link IE module, which is on the path that IP packets travel.	• Set the routing parameters, network number/station, or network station <-> IP-related information setting for accessing the specified station to the related stations. • Retry after a while. Or, start communication after checking that the system for data routing has started. • Do not use the same third byte of the IP address (network number) specified for the IP communication test with that of the CPU module where the test is executed. • Set the CPU module that performs IP packet transfer as the control CPU of the CC-Link IE module, which is on the path that IP packets travel.
4A01H	Network error	• The network of the number set to the routing parameters does not exist. • The specified CPU module cannot be communicated through the network that is not supported by the CPU module.	• Check and correct the routing parameters set to the related stations. • Set communication through the network that is supported by the specified CPU module.

Error code	Error name	Error details and cause	Action
4A02H	Network error	Access to the specified station cannot be made.	• Check the network module for error, or check that the modules are not in offline. • Check if the network numbers/PC numbers are correctly set.
4A03H	Network error	A request for network test was issued.	Check the request of the SLMP/MC protocol.
4A05H	Link related error (file related error)	121 stations or more are specified to the station number.	Check the station number.
4A10H	Link related error (file related error)	The number of files in the specified folder exceeded the limit.	Reduce the number of files in the specified folder.
4A20H	IP communication test error	• The upper 2 bytes of the IP addresses do not match between the CPU module and the request destination device on the same path that IP packets travel. • The upper 2 bytes of the IP addresses do not match between the CPU module and the CC-Link IE module on the same path that IP packets travel. • The upper 2 bytes of the IP addresses do not match between the CC-Link IE modules on the same path that IP packets travel. • The upper 2 bytes of the IP addresses do not match between the request source device and the CPU module connected to the request destination device by Ethernet.	• Check and correct the IP address settings of the CPU module. • Check and correct the IP address of the request destination device. • Check and correct the IP address of the CC-Link IE module. • Check and correct the IP address of the request source device.
4A21H	IP communication test error	• The 3rd byte (Network No.) or 4th byte (Station No.) of the IP address of the CPU module is the number that is not available for CC-Link IE. • The 3rd byte (Network No.) or 4th byte (Station No.) of the IP address of the request destination device is the number that is not available for CC-Link IE.	• Check and correct the IP address settings of the CPU module. • Check and correct the IP address of the request destination device.
4A22H	IP communication test error	The IP address is not set to the CC-Link IE module on the path that IP packets travel.	• Set the IP address to the CC-Link IE module used as a master station. • Check the communication status with the master station when the CC-Link IE module is used as a local station. • Replace the CC-Link IE module (master station) with the one that supports the IP packet transfer function. • Conduct the IP communication test again after the CC-Link IE module is started up. • Check and correct the IP address of the request destination device.
4A23H	IP communication test error	• The CPU module on the path that IP packets travel does not support the IP packet transfer function. • Routing parameters are set so that IP packets are routed to CPU modules that are incompatible with IP packet transfer.	• Replace the CPU module with the one supporting the IP packet transfer function. • Correct routing parameter so that IP packets are routed to the CPU module that supports the IP packet transfer function. • Check and correct the IP address of the request destination device.
4A24H	IP communication test error	• The CC-Link IE module on the path that IP packets travel does not support the IP packet transfer function. • Routing parameters are set so that IP packets are routed to the network module that does not support the IP packet transfer function. • The 3rd byte (Network No.) of the IP address of the device on the path that IP packets travel is overlapping with the network No. of the module connected to the CPU module and does not support the IP packet transfer function. • The 3rd byte (Network No.) of the IP address of the request destination device is overlapping with the network No. of the module connected to the CPU module and does not support the IP packet transfer function.	• Replace the CC-Link IE module with the one supporting the IP packet transfer function. • Correct routing parameter so that IP packets are routed to the CC-Link IE module that supports the IP packet transfer function. • Check and correct the setting so that the 3rd byte (Network No.) of the IP address of the device on the path that IP packets travel does not overlap with the network No. of the module connected to the CPU module and does not support the IP packet transfer function. • Check and correct the setting so that the 3rd byte (Network No.) of the IP address of the request destination device does not overlap with the network No. of the module connected to the CPU module and does not support the IP packet transfer function. • Check and correct the IP address of the request destination device.

Error code	Error name	Error details and cause	Action
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. • The target CPU No. specified does not exist.	• Check the error occurred in the specified access destination or the relay station, and take an action. • Check the connection destination (request destination module I/O number or PC number) in the request data of SLMP/MC protocol. • Check the stop error, and take an action. • Check and correct the target CPU No.
4B02H	Target module error	The request is not addressed to the CPU module.	Perform operation for the module that can execute the specified function.
4B03H	Target module error	• The specified route is not supported by the specified CPU module version. • The communication target CPU module is not mounted. • A device mounted on the specified route does not support communications.	• Check whether the specified route is supported or not. • Check the mounting status of the CPU module. • Check the stop error, and take an action.
4B04H	Target module error	The specified connection destination (request destination module I/O number) is not supported.	An invalid value is set as the start I/O number of the connection target module. Set the start I/O number of the target module correctly, and retry data communications.
4C00H	Data logging function error	There is not enough free space for storing the result file in the target memory.	Increase the free space, and create the result file again.
4C01H	Data logging function error	• Writing of the result file to the target memory has not completed successfully because the SD memory card is write-protected. • Writing of the result file to the target memory has not completed successfully because the folder/file structure is incorrect.	• Unlock the write protect switch of the SD memory card, and write the result file again. • Check that the target memory is broken. • Check that the folder or file to be used in the target memory is deleted.
4C02H	Data logging function error	The SD memory card was removed while the data logging function was being executed (data logging status: Waiting RUN Not collected, Waiting to establish collection conditions Not collected, Waiting start Not collected, Pause, Collecting, Waiting trigger Collecting before trigger, Collecting after trigger, or Saving in progress). Or, writing to the SD memory card has not completed successfully.	• Insert the SD memory card and execute the function again. • Replace the SD memory card and execute the function again.
4C03H	Data logging function error	The number of files in the root directory and subdirectory in the target memory exceeded the limit.	• Increase the free space of the drive (memory), and execute the function again. • Delete files in the drive (memory), and execute the function again.

Error code	Error name	Error details and cause	Action
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.
4C07H	Data logging function error	A data logging is attempted to start with a data storage memory that is different from the one for the data logging already registered.	Check and correct the setting for the data logging to be started.
4C08H	Data logging function error	Three or more of the data logging that the CPU built-in memory (function memory) is specified as the data storage memory are attempted to start.	• Check and correct the setting for the data logging to be started. • Check that the version of the CPU module is designed to make it possible to start three or more settings of logging operation with "CPU built-in memory (function memory)" specified for the data storage destination memory.
4C09H	Data logging function error	Free space in the target memory (data memory) is not enough.	Increase the free space in the target memory (data memory) and execute the function again.
4C0AH	Data logging function error	The target drive (data memory) is abnormal	Initialize the memory to bring the drive (data memory) back to normal state.
4C0BH	Data logging function error	• A data logging file that is being transferred is deleted. • Reading from (an access to) a data logging file is failed. • The specified file does not exist. Or, the specified subdirectory does not exist.	• Check and correct the number of files to be saved in the file switching setting. • Check that the data logging file is not deleted. • Check that an SD memory card is inserted. • Check the file name and subdirectory name. Then, execute the function again.
4C0CH	Data logging function error	• The data logging file transfer function is executed while the access to the SD memory card is forcibly disabled. • The SD memory card is removed during the data logging file transfer function is executed.	• Clear the disabled state of the SD memory card and execute the function again. • Insert the SD memory card and execute the function again.
4C0DH	Data logging function error	A data logging is attempted to start during a transfer of a result file.	Restart the data logging after the completion of the data logging file transfer function.
4C0EH	Data logging function error	A file transfer is stopped by the data logging file transfer stop request.	Do not send the data logging file transfer stop request.
4C0FH	Data logging function error	The program file or global label setting file was changed while the data logging function specifying the label (global label or local label) as the collection start condition, data collection target, or trigger condition was being executed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).	• Do not change the program file or global label setting file while the data logging function specifying the label (global label or local label) is being executed (data logging status: RUN waiting (no collection), Condition waiting (no collection), Start waiting (no collection), Pause, Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data). • Stop the data logging specifying the label (global label or local label). • Check and correct the setting of SD940 (Stop direction at file change on label specification), and execute the function again.
4C10H	CPU module data backup/restoration function error	• The maximum allowable capacity is exceeded. • The maximum allowable number of files is exceeded. • The upper limit of the backup folder number is exceeded. • The maximum length (255 characters) of the file path is exceeded.	• Increase the free space of the SD memory card and CPU module, and execute the function again. • Delete files in the SD memory card and CPU module, and execute the function again. • Delete backup data in the SD memory card and CPU module, and execute the function again. • Check and correct the folder structure or folder/file names of backup target data, and execute the data backup function again.

Error code	Error name	Error details and cause	Action
4C11H	CPU module data backup/restoration function error	• An SD memory card is not inserted. • The SD memory card is disabled by SM606 (SD memory card forced disable instruction).	• Insert or re-insert an SD memory card, and execute the function again. • Enable the SD memory card operation, and execute the function again.
4C12H	CPU module data backup/restoration function error	Reading/writing of data from/to an SD memory card completed with an error.	• Check that an SD memory card is inserted, and execute the function again. • Replace an SD memory card, and execute the function again. • The backup data may have been corrupted. Execute the data restoration function using another backup data.
4C13H	CPU module data backup/restoration function error	Reading/writing of data from/to the CPU built-in memory completed with an error.	• Back up data in the CPU built-in memory, initialize the memory, and write the data back to the original memory. Then, execute the data backup/restoration function. • The possible cause is a hardware failure of the restoration target CPU module. Execute the data restoration function to another CPU module.
4C14H	CPU module data backup/restoration function error	• The CPU module data backup/restoration function cannot be executed because a file password is set to the data. • Data was restored to the CPU module where the same data with a file password has already been stored.	Delete file passwords, and execute the CPU module data backup/restoration function.
4C15H	CPU module data backup/restoration function error	• Any of the following functions that cannot be executed simultaneously with the CPU module data backup/restoration function is being executed: the file transfer function (FTP), data backup/restoration (iQ Sensor Solution function), IP address change function, firmware update function (via the engineering tool), memory copy function, or online module change function. • The CPU module data backup/restoration function is executed when the following functions are being executed. · Online change · File transfer function (FTP server)	Execute the function again after a while.
4C16H	CPU module data backup/restoration function error	• The automatic backup setting by specification of day and time is turned on while the setting value (automatic backup date and time) are out of range. • The automatic backup setting by specification of time and day of the week is turned on while the setting value (automatic backup time and day of the week) are out of range.	• Review the setting value (automatic backup date and time) and turn on the automatic backup setting by specification of day and time. • Review the setting value (automatic backup time and day of the week) and turn on the automatic backup setting by specification of time and day of the week.
4C17H	CPU module data backup/restoration function error	The model of the restoration target CPU module differs from the model of the backup source CPU module.	Execute the data restoration function to the CPU module whose model is the same as that of the backup source CPU module.
4C18H	CPU module data backup/restoration function error	Data was restored while the operating status of the CPU module is in RUN or PAUSE.	Change the operating status of the CPU module to STOP, and execute the function again.
4C19H	CPU module data backup/restoration function error	• The data restoration function was executed with backup files ($BKUP_CPU_INF.BSC and BKUP_CPU.BKD) not structured properly. • Data (file(s)) is missing in the backup file ($BKUP_CPU_INF.BSC) in the backup data folder. • The data restoration function was executed with a folder where no backup files ($BKUP_CPU_INF.BSC, BKUP_CPU.BKD, and BKUP_CPU_DEVLAB.BKD) are stored.	The backup data may have been corrupted. Execute the data restoration function using another backup data.
4C1AH	CPU module data backup/restoration function error	• A folder with a value that matches the restoration target date folder setting value or number folder setting value does not exist in the SD memory card. • The restoration target data setting value is out of range. • The restoration target date folder setting value or number folder setting value is out of range.	• Check and correct the restoration target date folder setting value or number folder setting value, and execute the function again. • Check and correct the restoration target data setting value, and execute the function again. • Disable the automatic restoration function with the SD CARD OFF button when not using it. Then, execute the function again.

Error code	Error name	Error details and cause	Action
4C1BH	CPU module data backup/restoration function error	The data restoration function was executed to the CPU module whose status (such as programs, parameters, and file structure) differs from that of when the data backup function was executed.	• Match the CPU module status to the one at the time of backup, and execute the function again. • Set all data as the restoration target data, and execute the automatic data restoration function.
4C1CH	CPU module data backup/restoration function error	• An SD memory card is not inserted. • The SD memory card is disabled by SM606 (SD memory card forced disable instruction). • An SD memory card is write-protected.	• Insert or re-insert an SD memory card, and execute the function again. • Enable the SD memory card operation, and execute the function again. • Cancel the write protection, and execute the function again.
4C1EH	CPU module data backup/restoration function error	The status of the SFC program (such as step status and transition conditions) was changed during execution of the data backup function.	Take measures so that the status of the SFC program does not change during execution of the data backup function, and execute the function again.
4C1FH	CPU module data backup/restoration function error	The specified command cannot be executed because the CPU module data backup/restoration function is being executed.	Execute the command again after the data backup/restoration processing ends.
4C20H	CPU module data backup/restoration function error	The data backup/restoration function was executed while the CPU module was in a state where this function could not be executed.	• Deselect the "Access from External Device" checkbox in the label setting window or delete labels with this item selected, and initialize the memory which is written the global label assignment information. Then, execute the data backup function again. • Initialize the memory where the global label assignment information is written and execute the restoration function again. • Check the firmware version of the CPU module and execute the automatic data restoration function again.
4C21H	CPU module data backup/restoration function error	The number of backup data stored in an SD memory card exceeds the upper limit value.	• Delete backup data in the SD memory card, and execute the function again. • Check and correct the settings of the upper limit value of the number of backup data, and execute the function again.
4C22H	CPU module data backup/restoration function error	Bit 5 of SD944 has been turned on while the set value of SD1353 is out of the allowable range.	Check and correct the set value of SD1353, and turn on bit 5 of SD944.
4C23H	CPU module data backup/restoration function error	Upper limit value of the number of the backup data cannot be changed because a CPU data folder has already been in an SD memory card.	After deleting the CPU data folder in the SD memory card and turning off bit 5 of SD944, turn on bit 5 of SD944 again.
4C24H	CPU module data backup/restoration function error	The data cannot be backed up because more backup data than the upper limit value of the number of backup data exists in the SD memory card.	• Delete the backup data exceeding the upper limit value, and execute the function again. • Check and correct the settings of the upper limit value of the number of backup data, and execute the function again.
4C26H	CPU module data backup/restoration function error	The automatic restoration function with the SD CARD OFF button cannot be executed because the button has been pressed for more than 10 seconds after the READY LED had begun to flash.	Release the SD CARD OFF button within 10 seconds after the READY LED begins to flash. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.
4C40H	File transfer function (FTP client) error	• When files are specified by using wild card characters for the file transfer function instruction, the number of files matched exceeds the upper limit of the transferable number of files. • When files are specified by using wild card characters for the file transfer function instruction, no files are matched.	• Check and correct the wild card specification. • Check if the specified folder path exists.
4C43H	File transfer function (FTP client) error	The number of processing completed files for sending or acquiring FTP client file is mismatched with the total number of processing files.	Execute the function again.
4C44H	File transfer function (FTP client) error	The file transfer function (FTP client) is executed while the following functions are being executed. • CPU module data backup/restoration function • iQ Sensor Solution data backup/restoration function	Execute the File transfer function (FTP client) again after the mentioned function is completed.
4C50H	Special relay and special register function error	The free space in the data memory is not enough.	Increase the free space in the target memory (data memory) and execute the function again.

Error code	Error name	Error details and cause	Action
4C51H	Special relay and special register function error	Writing to the data memory is not completed successfully.	Initialize the memory to bring the drive (data memory) back to normal state.
4C52H	Special relay and special register function error	The function memory clear request or the data memory file transfer request is sent during a data logging.	Do not send the function memory clear request or the data memory file transfer request during a data logging.
4C53H	Special relay and special register function error	The function memory clear request or the data memory file transfer request is sent during execution of the following functions. • File batch online change • Data transfer to the data memory • The global label assignment information is being written to the programmable controller (data memory).	Send the function memory clear request or the data memory file transfer request after the completion of the functions.
4C54H	Special relay and special register function error	The request cannot be executed because the CPU module is not in the STOP state.	Set the operating status of the CPU module to STOP, and execute the request again.
4C55H	Special relay and special register function error	The number of files in the root directory and subdirectory in the data memory exceeded the limit.	• Increase the free space in the data memory and execute the request again. • Delete files in the data memory, and execute the request again.
4C56H	Special relay and special register function error	• Reading from (an access to) a file is failed. • The specified file does not exist. Or, the specified subdirectory does not exist.	• Check that the file is not deleted. • Check the file name and subdirectory name. Then, execute the function again.
4D40H	Firmware update function error (Via engineering tool)	Access to the flash ROM of the module to be updated has failed.	Perform the firmware update to the module again.
4D41H	Firmware update function error (Via engineering tool)	• Access to the module to be updated has failed. • The module is unable to perform the firmware update. • An incorrect firmware update file (a firmware update file not for the module to be updated) has been used. • An invalid firmware update file has been used.	Check the following and perform the firmware update again. • No hardware failure occurs in the base unit or the module. • The module started up normally. • The module is able to perform the firmware update. • The correct firmware update file for the module is set in the engineering tool. • The name or contents of the firmware update file are not changed from their original state.
4D44H	Firmware update function error (Via engineering tool)	• A firmware update file of the version that cannot be installed on the module used has been used. • The module does not support the firmware update. • The module information cannot be read from the module to be updated.	• Use the module with a firmware version that supports the firmware update using the engineering tool. • Check if the module supports the firmware update. • Check that the module is mounted properly, reset the CPU module, and perform the firmware update again.
4D45H	Firmware update function error (Via engineering tool)	The firmware update is disabled.	Enable the firmware update and perform the operation again.
4D46H	Firmware update function error (Via engineering tool)	The engineering tool and the CPU module are connected incorrectly. (The cable connection and/or connection settings in the engineering tool are not correct.)	• Check the cable connection and/or the settings in the "Specify Connection Destination" window that the CPU No.1 is connected via USB or an Ethernet port ("Ethernet Port Direct Connection"/"Connection via HUB"). • Check that the "Target PLC" is set to "Not Specified" in the "Specify Connection Destination" window of the engineering tool.
4D47H	Firmware update function error (Via engineering tool)	• The operation cannot be performed because the firmware update is being performed from another engineering tool. • The operation cannot be performed because the CPU No.1 was not reset after the last firmware update. • A communication error occurred in the last firmware update.	• Perform the firmware update to the module again after the completion of the update from the other engineering tool. • Manually reset the CPU module and perform the firmware update again.
4D48H	Firmware update function error (Via engineering tool)	• The firmware update cannot be performed due to a CPU module stop error. • The module may be faulty.	• Check the parameters. • Check whether the module is mounted correctly. • If the same error code is displayed again, please consult your local Mitsubishi representative.

Error code	Error name	Error details and cause	Action
4D49H	Firmware update function error (Via engineering tool)	• The CPU No.1 has been powered off or reset during the firmware update processing. • The engineering tool has been exited or a communication error has occurred during the firmware update processing.	Perform the firmware update again.
4D4AH	Firmware update function error (Via engineering tool)	• The firmware update has been performed to the module that the CPU No.1 does not support. • The firmware update has been performed using the firmware update file that the CPU No.1 does not support. • An invalid firmware update file has been used.	• Update the CPU No.1 to the latest firmware version and perform the firmware update again. • Ensure that the name or contents of the firmware update file is not changed from its original state.
4D4BH	Firmware update function error (Via engineering tool)	The CPU modules of CPU No.2 and later do not support the firmware update using the engineering tool.	Update the CPU modules of CPU No.2 and later to the latest version by performing the firmware update using an SD memory card, and perform the operation again.
4D4CH	Firmware update function error (Via engineering tool)	• The module to be updated cannot be updated on the extension base unit. • The firmware update file set in the engineering tool cannot be used for the module on the engineering tool.	• Mount the module to be updated to the main base unit and perform the firmware update again. • Set the correct firmware update file for the module to be updated in the engineering tool, and perform the firmware update again.
4D4DH	Firmware update function error (Via engineering tool)	A firmware data error has been detected during the firmware update processing.	Perform the firmware update again.
4D4EH	Firmware update function error (Via engineering tool)	• The specified operation cannot be performed because the firmware update is being performed. • The specified operation cannot be performed because the CPU No.1 was not reset after the firmware update.	Reset the CPU module after the completion of the firmware update and perform the specified operation again.
4D4FH	Firmware update function error (Via engineering tool)	The firmware update has been performed to the module controlled by the CPU module of CPU No.2 or later.	• Set its control CPU to the CPU No.1 and perform the firmware update again. • Update the firmware of CPU No.1 to the latest version and perform the firmware update again.
4D50H	Firmware update function error (Via engineering tool)	• A remote operation has been performed to the CPU module of CPU No.2 or later during the firmware update processing. • The firmware update has been performed while the CPU module of CPU No.2 or later is the RUN state.	• Reset the CPU module and perform the firmware update again. • Change the operating status of the CPU module of CPU No.2 or later to STOP. Reset the CPU module and perform the firmware update again.
4D52H	Firmware update function error (Via engineering tool)	The control CPU does not support firmware updates using the engineering tool.	Update the control CPU to a version supporting the firmware update function, and perform the firmware update again.
4D53H	Firmware update function error (Via engineering tool)	Performed an update on a module controlled by a CPU module executing another function.	Check that the control CPU is not executing another function, and perform the firmware update again.
4D5AH	Firmware update function error (Via engineering tool)	The firmware update cannot be performed because a tracking cable is connected.	Disconnect the tracking cable and try again.
4D5BH	Firmware update function error (Via engineering tool)	In a redundant system with redundant extension base unit, the firmware update has been executed while both systems are powered on.	Power off the standby system. Reset the CPU module of the control system and execute the firmware update again.

Error codes related to the online module change function

The following table lists the codes of errors related to the online module change function.
The code of an error detected during online module change is stored in SD1618 (Online module change error code).
The code of an error when a disable request is executed during online module change is stored in SD1619 (Disable request error code during online module change).

Error code	Error name	Error details and cause	Action
4110H	Online module change error	The online module change function was executed while the CPU module was in an error state (stop error).	The module cannot be changed online. Power off the programmable controller, and replace the module.
4111H		In a multiple CPU system, the online module change function was executed even though other CPU modules have not started up.	Execute the function again after all the CPU modules in the system start up.
4202H		• An attempt was made to change two modules directly at the same time. • A module that is not ready to be removed was removed.	• Continue processing for the module being changed online. Two modules cannot be changed online at the same time. For the other module, power off the programmable controller, and replace the module. • Turn on SM1602 (Module removal request flag) and check that the value, 5, is stored in SD1617 (Online module change progress status). Then, continue processing from the step for mounting a module.
4205H		The online module change function was executed in a multiple CPU system where a CPU module that does not support this function is included.	The module cannot be changed online. Power off the programmable controller, and replace the module.
4206H		An attempt was made to change the inter-module synchronization target module online.	The module cannot be changed online. Power off the programmable controller, and replace the module.
4210H		The selected module is out of the valid range.	Check and correct the target base unit number and/or slot number, and turn on SM1600 (Module selection request flag) again.
4214H		The model of the newly-mounted module differs from that of the module before online change. Or, the newly-mounted module is not upward compatible (module mismatch).	Mount the same model or the upward compatible module, and execute the request again. • Mount a module of the same model, and perform the module recognition processing again.
4215H		• The module recognition processing is performed without the module being mounted. • The module selection processing is performed to the slot that is set as "Empty". • The module newly-mounted to the system is not mounted correctly.	• Mount a module, and perform the module recognition processing again. • Check and correct the target base unit number and/or slot number, and turn on SM1600 (Module selection request flag) again. • Mount the module correctly.
4216H		The module newly-mounted to the system online has failed. (The CPU module cannot access to the module.) • The module has failed. • The module newly-mounted to the system is not mounted correctly.	Use another module, and execute the request again. Or, power off the system, and replace the module with the one that supports the online module change function. • Replace the module with the normal one. • Mount the module correctly.
4218H		• The online change target module (before change) does not support the online module change function. • The module newly-mounted to the system online (after change) does not support the online module change function. • The online module change function was executed to change the MELSEC-Q series module.	• If an error occurs at the stage of selecting a module, the module cannot be changed online. Power off the programmable controller, and replace the module. • If an error occurs at the stage of recognizing a module, replace the module with the one that supports the online module change function, and continue the processing. • The module cannot be changed online. Power off the programmable controller, and replace the module.
421AH		The online module change function was executed to change a module controlled by another CPU module.	Execute the function from the control CPU module.
4222H		The request cannot be executed because the redundant function module is restarting.	Issue the request again after the restart of the redundant function module has completed.

Codes of errors detected by other than the self-diagnostic function (6F00H to 6FFFH)

The following table lists the codes of errors, other than those detected by the self-diagnostic function of the CPU module.

Error code	Error name	Error details and cause	Action
6F00H	Transient execution error	The internal buffer area for transient processing is used to the maximum.	Temporarily stop transient transmissions, or reduce the frequency. Then, perform transmissions again. Or, add the COM instruction to the 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			
6F24H			
6F25H	Module communication test error	The operating status of the CPU module is in a state other than STOP. (A test cannot be executed.)	Change the operating status of the CPU module to STOP, and execute a module communication test.
6F27H	Module communication test being executed	• A module communication test execution request was received during the test. • A module communication test termination request was received during the test.	Since the module communication test is being executed, execute the request again after the completion of the test.
6F29H	Module communication test error	A module communication test execution request was received in the state where tracking cables are connected incorrectly.	Connect one end of a tracking cable to the IN connector and the other end to the OUT connector of the test target redundant function module, and execute the request again.
6F30H	Transient execution error (redundant function module communication being stopped)	Data communications cannot be performed with the other system because of any of the following reasons: • The CPU module of the other system is powered off or in a reset state. • The tracking communication stops due to an error in the CPU module of the own or other system. • Tracking cables are disconnected or incorrectly connected, or failed. • The redundant function module of the own or other system is being changed online, a module communication test is being performed for the module, or a hardware failure has occurred in the module.	• If a WDT error has occurred in the CPU module of the own or other system, eliminate the error cause and then execute the operation again. • Check that there is no error or failure in the CPU module, tracking cables, redundant function modules, and perform a retry. Or, perform a retry after the online module change processing or the module communication test ends. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, redundant function module, base unit, or tracking cable. Please consult your local Mitsubishi representative.
6F40H	Transient execution error	The number of transient transmission requests exceeded the upper limit of simultaneously processable requests.	Temporarily stop transient transmissions, or reduce the frequency. Then, perform transmissions again.
6F44H	Transient frame error	An invalid transient frame was received.	Check and correct the request data (send data) on the request source side, and perform transmissions again.
6F4AH			
6F4BH			
6F4CH			
6F4DH			
6F4EH			
6F60H	Module error	An error has been detected in the redundant function module.	Please consult your local Mitsubishi representative.

Codes of errors detected by other than the self-diagnostic function (C000H to CFFFH)

The following table lists the codes of detected errors related to the Ethernet-equipped module.

These error codes are not stored in SD0 because they are not detected by the self-diagnostic function of the CPU module.

Error code	Error name	Error details and cause	Action
C000H to CFBFH	MELSEC iQ-R Ethernet User's Manual (Application)		

38 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 194 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.

38.1 How to Read the Event List

The event list contains the following information.

Item	Description
Event code	ID number assigned to an event
Event type	Type of an event
Event category	Category of an event
Detected event	Description of a detected event
Detailed information 1 to 3	Details of a detected event

Detailed information

The following table lists the details of information displayed in the detailed information 1 to 3.

Detailed information	Item	Description
Detailed information 1	Operation source information	Information on the operation source • Connection port (such as Ethernet and USB) • I/O number • Network number • Station number • IP address • User name (when the user authentication function is enabled)
	Event history file information	Information on the event history file
	Module information	Information on the target module (I/O number)
	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
	Automatic standby system recovery information	Information on the automatic standby system recovery

Detailed information	Item	Description
Detailed information 2	Communication speed and communication mode	Information on the communication speed and the communication mode
	Communication status	Information on the communication status
	Security key operation information	Information on the corresponding security key
	Remote password information	Information on the corresponding remote password
	File password information	Information on the corresponding file password
	Blocked IP address information	Information on the blocked IP address
	Drive/file information	Information on the corresponding drive name and file name
	Drive number and file name	Information on the corresponding drive number and file name
	Copy source drive/file information	Information on the corresponding drive name and file name
	Operation target information	Information on the operation target: I/O number
	Clock information (before change)	Clock information before change
	Remote operation type information	Information on the remote operation type
	User authentication information	Information on the user authentication function status (enabled/disabled)
	User management information copy	Information on user management information copy (succeeded/failed)
	Login/logout information	Login/logout information of the user authentication function
	File access control information	Information on file access control: access level
	Program start information	Information on the start of a specified program
	Program stop information	Information on the stop of a specified program
	System configuration information	Information on the system configuration
	Target 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) supporting iQ Sensor Solution
	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

38.2 Event List

The following table lists events related to the CPU module.

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
00100	System	Info	Link-up	■CPU module The CPU module has entered into the link-up state as a result of an operation such as connecting a network cable between the CPU module and an external device. ■Redundant function module The redundant function module has entered into the communication enable state. * This event code does not indicate the start of tracking communication.	Operation source information	Communication speed and communication mode	—	—	—
00110			Communication start/end (TCP)	Data communications with an external device through TCP connection started. Or, data communications with an external device through TCP connection ended.		Communication status			
00120			FTP connection start/end	FTP connection with an external device started. Or, FTP connection with an external device ended.					
00130			Receive frame error	A receive frame error was detected.		—			
00140			SNTP server time synchronization failure	Time setting by the time synchronization function failed because there was no response from the SNTP server.					
00400			Power-on/reset	The CPU module has been power-on or reset.	Start-up information*2				
00401			Initial processing cancelled	The CPU module has been powered off or reset during initial processing.	Information on the initial processing cancellation				
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				
00440			Safety operation mode undefined	The safety operation mode was undefined, and the system started in TEST MODE.	—				
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				
					1	2	3	4	5
00800	System	Warning	Link-down	■CPU module The CPU module has entered into the link-down state as a result of an operation such as disconnecting a network cable between the CPU module and an external device. ■Redundant function module The redundant function module has entered into the communication disable state as a result of any of the following: • Any tracking cable was disconnected. • The other system was powered off. • An error occurred in the cable, connector, or module.	Operation source information	Communication speed and communication mode	—	—	—
00904			Socket communication send error	Sending a message over socket communication failed.		—			
00906			Alive check error	The alive status of an external device could not be checked within the period specified by the response monitoring timer.					
00907			Divided message receive timeout error	• All the data could not be received within the period specified by the response monitoring timer. • Data of the total data length could not be received. • The remaining part of the message divided into the TCP/IP level could not be received within the period specified by the response monitoring timer.					
00908			IP composition timeout error	An IP composition timeout error occurred. (The specified period of time expired without receiving all of the divided data.)					
00909			TCP-specified port number error	A port number being used for opened connection was set (for TCP/IP).					
0090A			UDP-specified port number error	A port number being used for opened connection was set (for UDP/IP).					

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
00A00	System	Warning Error	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				
00F01			Auto memory copy (control system)	The control system automatically copied its memory to the standby system.	—				
00F02			Auto memory copy (standby system)	The memory of the control system was automatically copied to the standby system.					
00F03			Automatic standby system recovery	The standby system has recovered automatically.	Automatic standby system recovery information				
00F04			Automatic standby system recovery failed	The automatic standby system recovery has failed.					
00F05			Latch data clear	Due to no battery and a memory error, the latch data was cleared to zero.	—				
1000 and after			When a self-diagnostic error occurs, the error is stored as an event.						

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
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.		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.					
10600			User authentication function enabled/ disabled	The user authentication function was enabled or disabled.		User authentication information			
10601			Login succeeded	The login processing of the user authentication function completed successfully.		Login/logout information			
10602			Login failed	The login processing of the user authentication function failed.					
10603			Log off successful	The log off processing of the user authentication function completed successfully.					
10604			Log off failed	The log off processing of the user authentication function failed.					
10605			Auto logout	The system automatically logged out the user by executing the user authentication function.					
10606			User management information update	The user management information was updated. (The information was added, changed, or deleted.)		—			
10607			User management information registration (Administrators)	An operator (Administrators) was registered to the user management information.					

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
10608	Security	Info	All-users log off successful	The all-users log off processing of the user authentication function completed successfully.	Operation source information	Login/logout information	—	—	—
10609			All-users log off failed	The all-users log off processing of the user authentication function failed.					
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			
14600			User management information copy successful (copy source)	Copy of the user management information to the other system completed successfully.		User management information copy			
14601			User management information copy failed (copy source)	Copy of the user management information to the other system failed.					
14602			User management information copy successful (copy destination)	Copy of the user management information completed successfully.					
14603			User management information copy failed (copy destination)	Copy of the user management information failed.					
18200		Warning	Initialization of all programmable controller information	All information in the programmable controller was initialized.		—			
20100	Operation	Info	Error clear	The error was cleared.		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.					
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.					

Event code	Event type	Event category	Detected event	Description	Detailed information				
					1	2	3	4	5
20610	Operation	Info	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				
24000			Clock setting	The clock data was set.	Operation source information	Clock information (before change)	Clock information (after change)		
24001			Remote operation request accepted	A remote request (RUN, STOP, or PAUSE) was accepted.		Remote operation type information	—		
24100			Operating status change (RUN)	The operating status of the CPU module was changed to RUN.	—	—			
24101			Operating status change (STOP)	The operating status of the CPU module was changed to STOP.					
24102			Operating status change (PAUSE)	The operating status of the CPU module was changed to PAUSE.					
24120			Starting the program	The program was started.	Operation source information	Program start information			
24121			Stopping the program	The program was stopped.		Program stop information			
24200			Creation of new folders, writes to files/folders[*1]	• A new folder was created. • A new file was created or data was written to a file.		Drive/file information			
24201			File copy[*1]	A file was copied.		Copy source drive/file information	Copy destination drive/file information		
24202			Folder/file rename[*1]	A folder name or file name was changed.					
24300			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				
					1	2	3	4	5
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.		Device/label information/ Device/label clearing information			
2A202			Folder/file deletion*1	A folder or file was deleted.		Drive/file information			
2B000			System switching (by a user)	The systems were switched due to the cause on the user side.		System switching information			
2B001			Memory copy execution by the engineering tool (control system)	The memory copy function was executed by the engineering tool.		—			
2B002			Memory copy execution by the special relay and special register (control system)	The memory copy function was executed by the special relay and special register.	—				
2B003			Memory copy execution by a user (standby system)	The memory copy function was executed by a user.					
2B004			Control system forced start-up	One system was forcibly started as a control system while waiting for a start-up of the other system.	Operation source information	Control system start-up cause information			

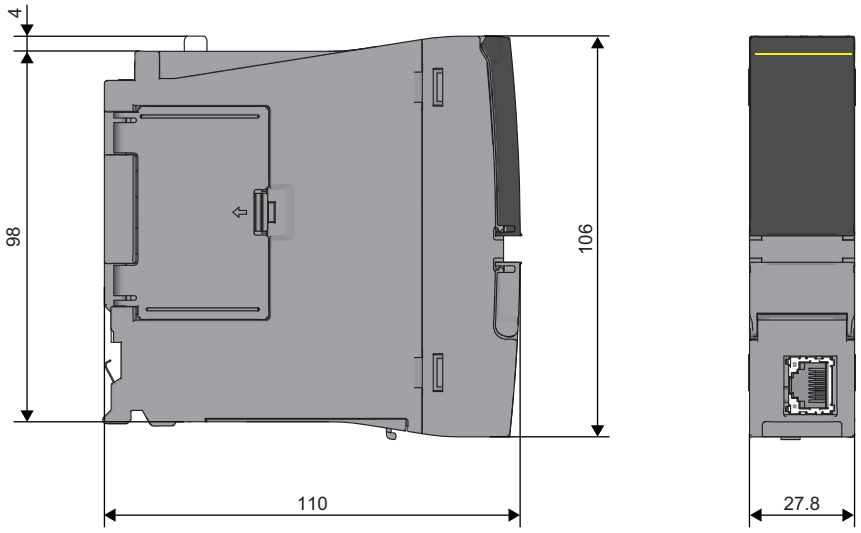
*1 As for file-related events such as writing and deleting a file, the following files are targeted:
· Program file
· FB program file
· Parameter file
· Safety program file
· Safety FB program file
· Safety parameter file

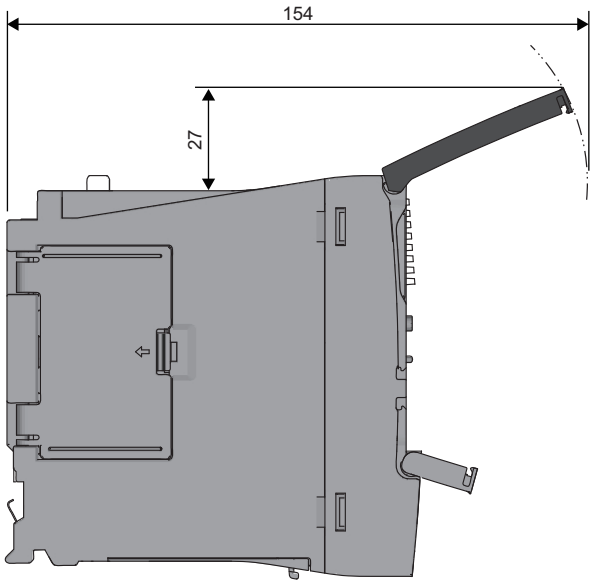
*2 The engineering tool displays "Start-up information" in detailed information 1 only when the CPU module with firmware version "06" or later is used.

APPENDICES

Appendix 1 External Dimensions

CPU module

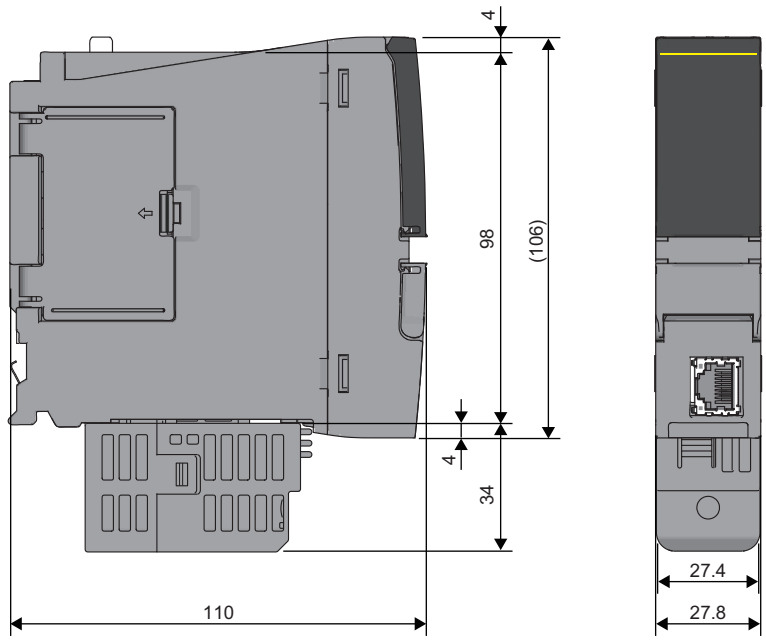




(Unit: mm)

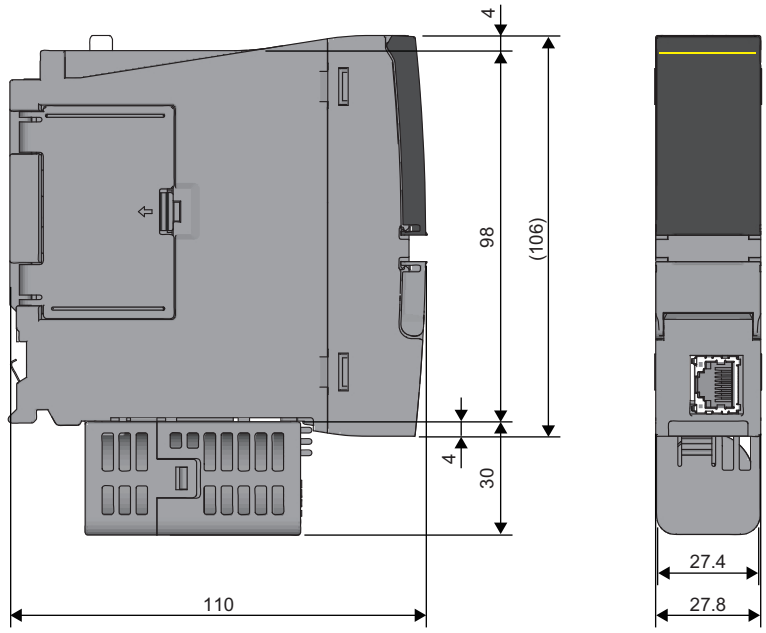
A

■With the Q7BATN-SET



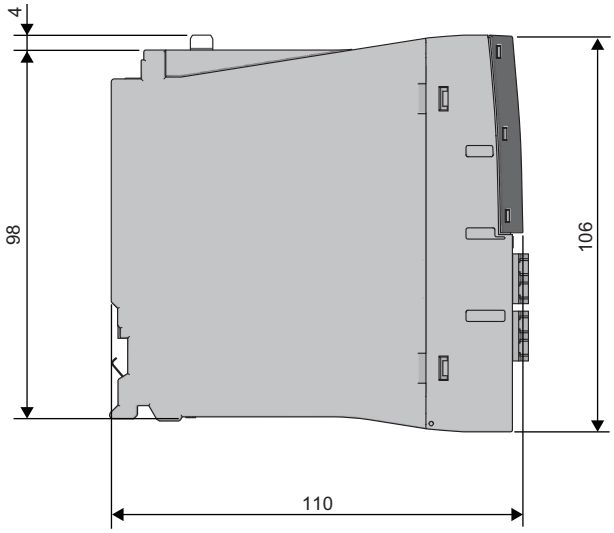
(Unit: mm)

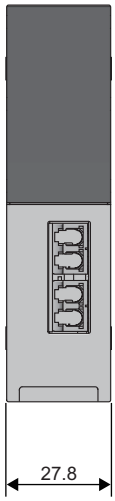
■With the Q7BAT-SET



(Unit: mm)

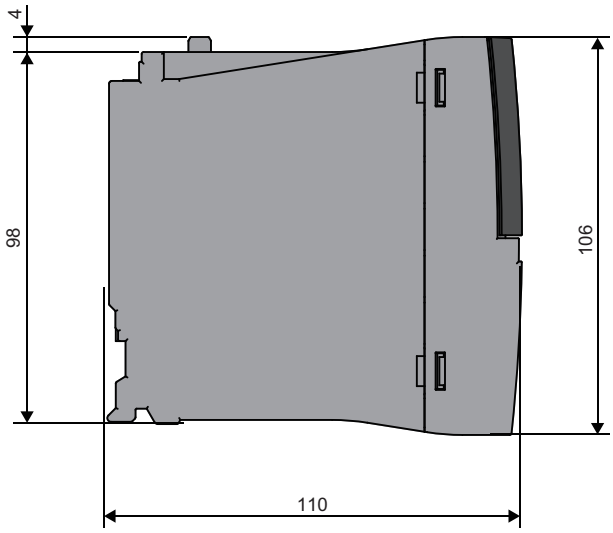
Redundant function module





(Unit: mm)

SIL2 function module





(Unit: mm)

A

Appendix 2 Compliance with EMC and Low Voltage Directives

Method of ensuring compliance

To ensure that Mitsubishi programmable controllers maintain EMC and Low Voltage Directives when incorporated into other machinery or equipment, certain measures may be necessary. Please refer to one of the following manuals.

- MELSEC iQ-R Module Configuration Manual (SH-081262ENG)
- Safety Guidelines (IB-0800525)

The CE mark on the side of the programmable controller indicates compliance with EMC and Low Voltage Directives.

Additional measures

To ensure that this product maintains EMC and Low Voltage Directives, please refer to one of the following manuals.

- MELSEC iQ-R Module Configuration Manual (SH-081262ENG)
- Safety Guidelines (IB-0800525)

Appendix 3 Functional Availability by CPU Module Model

The following table lists the availability of functions by CPU module model.

Rn: RnCPU, RnEN: RnENCPU, RnP(P): Process CPU (process mode), RnP(R): Process CPU (redundant mode), RnPSF: SIL2 Process CPU, RnSF: Safety CPU

○: Supported, ×: Not supported

Function		Availability*1					
		Rn	RnEN	RnP(P)	RnP(R)	RnPSF	RnSF
Constant scan		○	○	○	○	○	○
Device/label access service processing setting		○	○	○	○	○	○
Device/label access service processing constant wait function		×	×	○	○	×	×
Interrupt function	Multiple interrupt function	○	○	○	○	○	○
Output mode setting when the status changed from STOP to RUN		○	○	○	○	○	○
Device/label memory area setting		○	○	○	○	○	○
Internal buffer capacity setting		○*3	○	○	○	×	○
SD memory card forced disable		○*3	○	○	○	○	○
Clock function		○	○	○	○	○	○
Writing data to the CPU module	Writing data to the programmable controller	○	○	○	○	○	○
	Online change (ladder block)	○	○	○	○	○	○
	Online change (files)	○	○	○	○	○	○
RAS function	Scan monitoring function	○	○	○	○	○	○
	Self-diagnostics function	○	○	○	○	○	○
	FB hierarchy information	○	○	○	○	×	×
	Error clear	○	○	○	○	○	○
	Event history function	○	○	○	○	○	○
	Online module change	×	×	○	○	○	×
	Program cache memory auto recovery function	○	○	○	○	○	○
Remote operation		○	○	○	○	○	○
Boot operation		○*3	○	○	○	×	×
Monitor function	Circuit monitor	○	○	○	○	○	○
	Device/buffer memory batch monitor	○	○	○	○	○	○
	Watch	○	○	○	○	○	○
	Program monitor list	○	○	○	○	○	○
	Interrupt program monitor list	○	○	○	○	○	○
	Real-time monitor function	○	○	×	×	×	○
	Scan time measurement	○	○	×	×	×	×
	Scan time clear	○	○	×	×	×	○
	Specified program monitor	○	○	×	×	×	×
Test function	External input/output forced on/off function	○	○	○	○	×	○
	Device test with execution condition	○	○	○	○	×	○
Data logging function		○*3	○	○	○	×	○
Recording function		○*2	○	×	×	×	○
Debug function	Memory dump function	○*3	○	×	×	×	○
SFC function		○	○	○	○	×	○
Database function	Database access instruction	○*2	○	×	×	×	×
	CPU module database access (from external device) function	○*2	○	×	×	×	×
PID control function		○	○	○	○	○	○
Process control function		×	×	○	○	○	×
CPU module data backup/restoration function		○*2	○	○	○	×	○

A

Function		Availability[*1]					
		Rn	RnEN	RnP(P)	RnP(R)	RnPSF	RnSF
Multiple CPU system function	Out-of-group I/O fetch	○	○	○	×	×	○
	Multiple CPU synchronized startup	○	○	○	×	×	○
	Data communications between CPU modules	○	○	○	×	×	○
	Multiple CPU synchronous interrupt	○	○	○	×	×	○
Security function	User authentication function	×	×	×	×	○	○
	Block password function	○	○	○	○	○	○
	Security key authentication function	○	○	○	○	○	○
	File password function	○	○	○	○	○	○
	IP filter function	○	○	○	○	○	○
	Remote password function	○	○	○	○	○	○
	Write-protect function for device data (from outside the CPU module)	○	○	×	×	×	×
Sequence scan synchronization sampling function		○	○	×	×	×	○
Label initialization function	Label initialization after converting all programs (reassignment)	×	×	○	○	○	×
	Label initialization with label initial values	×	×	○	○	○	×
Routing setting		○	○	○	○	○	○
Label access setting from external device		○[*2]	○	○	○	○	○
Latch function	Latch with a battery	○	○	○	○	○	○
	Latch with a battery-less option cassette	○[*2]	○	×	×	×	×
Initial device/label value setting		○	○	○	○	○	○
Redundant function	Operation mode change	×	×	×	○	×	×
	System switching	×	×	×	○	○	×
	Tracking transfer	×	×	×	○	○	×
	Memory copy from control system to standby system	×	×	×	○	○	×
	System consistency check	×	×	×	○	○	×
	Program execution in both systems	×	×	×	○	○	×
	Redundant system operation setting	×	×	×	○	○	×
	Redundant function module communication test	×	×	×	○	○	×
	Setting for redundant system with redundant extension base unit	×	×	×	○	×	×
	Automatic standby system recovery	×	×	×	○	×	×
	Replacement/addition of an extension cable (online)	×	×	×	○	×	×
Safety operation mode		×	×	×	×	○	○
Continuous RUN prevention in TEST MODE		×	×	×	×	×	○
Safety diagnostic function		×	×	×	×	○	○
Safety data identify check		×	×	×	×	○	○
Safety communication function		×	×	×	×	○	○
Ethernet function		📖MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)					
SLMP communication function		○	○	○	○	○	○
Inter-module synchronization function		○	○	○	×	×	○
CC-Link IE Field Network Basic function		📖CC-Link IE Field Network Basic Reference Manual					
iQ Sensor Solution function		📖iQ Sensor Solution Reference Manual					
Firmware update function	Update using the engineering tool	○	○	○	○	×	×
	Update using an SD memory card	○[*3]	○	○	○	×	×

*1 Some functions have restrictions on the firmware version and the production information of the CPU module used or the version of the engineering tool used. Refer to the following.
☞ Page 566 Added and Enhanced Functions

*2 The R00CPU, R01CPU, and R02CPU does not support the use of these functions.

*3 The R00CPU does not support the use of these functions.

Appendix 4 List of Special Relay Areas

The following table lists items in the list.

Item	Description
No.	Special relay number
Name	Special relay name
Data stored	Data stored in the special relay and its meaning
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (program, engineering tool, GOT, or other testing operations from external device) • U/S: User and system <Set timing> • Every END: Data is set every time END processing is performed. • Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN). • Status change: Data is set when the status is changed. • Error: Data is set when an error occurs. • Instruction execution: Data is set when an instruction is executed. • Request: Data is set when requested by a user (using the special relay). • Writing: Data is set when a user performs a writing operation. • During END: Data is set when END processing is performed. • Power-on to RUN or STOP to RUN: Data is set when the operating status changes from power-on to RUN or from STOP to RUN. • System switching: Data is set when two systems are switched (between the control system and the standby system)

Point

Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

A

Diagnostic information

The following is the list of special rely areas relating to the diagnostic information.

No.	Name	Data stored	Details	Set by (setting timing)
SM0	Latest self-diagnostic error (including annunciator ON)	Off: No error On: Error	• This relay turns on when the self-diagnostics returns an error (including the case when an error is detected by turning the annunciator ON). • The ON state is maintained even after the error has been later cleared.	S (Error)
SM1	Latest self-diagnostic error (not including annunciator ON)	Off: No error On: Error	• This relay turns on when the self-diagnostics returns an error (not including the case when an error is detected by turning the annunciator ON). • The ON state is maintained even after the error has been later cleared. • This relay does not turn on when errors that are notified by the PALERT instruction or the PABORT instruction.	S (Error)
SM50	Error reset	Off→On: Error reset request On→Off: Error reset complete	• This relay clears the error state when the mode transfers from off to on. • This relay switches from on to off when the error reset has been completed.	U/S (Status change)
SM51	Battery low latch	Off: Normal On: Battery low	• This relay switches to on when the battery voltage of the CPU module drops below the specified value. • The ON state is maintained even after the battery voltage has been later recovered to the normal value. • This relay synchronizes with BAT LED.	S (Error)
SM52	Battery low	Off: Normal On: Battery low	This relay has the same function as SM51 except for switching to off after the battery voltage has been recovered to a normal value.	S (Error)
SM53	AC/DC DOWN	Off: No AC/DC DOWN detection On: AC/DC DOWN is detected	• This relay switches to on when a momentary power failure within 20ms is detected while the AC power supply module is in use. This relay can be reset when power is turned off and on. (In a redundant system with redundant extension base unit, if a momentary power failure occurs in a module on an extension base unit, the power failure will be detected in both systems.) • This relay switches to on when a momentary power failure within 10ms is detected while the DC power supply module is in use. This relay can be reset when power is turned off and on. (In a redundant system with redundant extension base unit, if a momentary power failure occurs in a module on an extension base unit, the power failure will be detected in both systems.)	S (Error)
SM56	Instruction execution fault	Off: Normal On: Instruction execution fault state	• This relay switches to on when an error which can be classified as instruction execution fault is returned. • The ON state is maintained even after the error has been later cleared.	S (Error)
SM60	Fuse Blown	Off: Normal On: Fuse blown is detected	• This relay is on when at least one output module is in fuse blown state and the ON state is maintained even after later recovering to the normal state. • The fuse blown state check is also done for output modules on the remote I/O station.	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.	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.	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 register monitors the status of the power supply module mounted on the main base unit and counts the number of momentary power failures. • This relay turns off the flags (SM152 and SM153) of the power supply 1 and 2 when the CPU module starts up. • When one of the two power supply modules is powered off, this relay turns off the corresponding flag to one powered off.	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.	S (Status change)

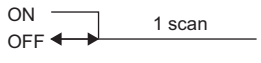
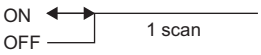
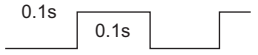
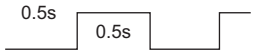
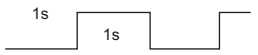
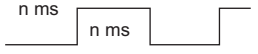
System information

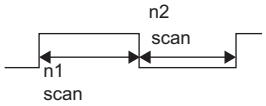
The following is the list of special relay areas relating to the system information.

No.	Name	Data stored	Details	Set by (setting timing)
SM203	STOP contact	Off: Other than STOP state On: STOP state	This relay is on in STOP state.	S (Status change)
SM204	PAUSE contact	Off: Other than PAUSE state On: PAUSE state	This relay is on in PAUSE state. Note that this relay is on during the END processing of the scan which the specified PAUSE contact turns on if PAUSE state is generated at the PAUSE contact.	S (Status change)
SM210	Clock data set request	Off→On: Setting request is detected On→Off: Setting is completed	• Clock data stored in SD210 to SD216 is written into the CPU module when this relay is switched from off to on. • This relay switches from on to off when writing of clock data stored in SD210 to SD216 into the clock element is completed.	U/S (Status change)
SM211	Clock data set error	Off: No error On: Error	This relay switches to on when an error is generated in values from SD210 to SD216, and to off when no error is generated.	S (Request)
SM213	Clock data read request	Off: Non-processing On: Reading request	Clock data is loaded into SD210 to SD216 when this relay is in the ON state.	U

System clock

The following is the list of special relay areas relating to the system clock.

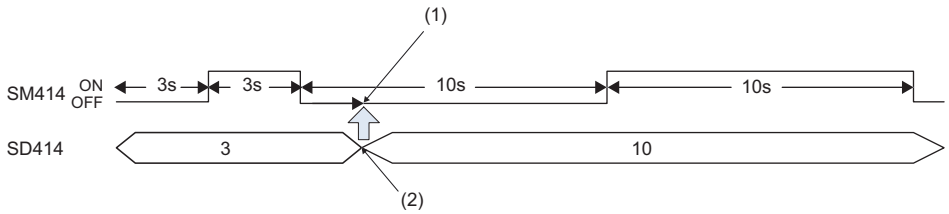
No.	Name	Data stored	Details	Set by (setting timing)
SM400	Always On		Always On	S (Power-on to RUN/ STOP to RUN/every END)
SM401	Always Off		Always Off	S (Power-on to RUN/ STOP to RUN/every END)
SM402	After RUN, ON for 1 scan only		• This relay is on during only one scan after RUN mode starts. • This relay is enabled only for the scan execution type program.	S (Status change/every END)
SM403	After RUN, OFF for 1 scan only		• This relay is off during only one scan after RUN mode starts. • This relay is enabled only for the scan execution type program.	S (Status change/every END)
SM409	0.01 second clock		• This relay repeats on/off at 5ms intervals. • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)
SM410	0.1 second clock		• This relay repeats on/off at certain intervals. • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)
SM411	0.2 second clock		• This relay repeats on/off at certain intervals. • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)
SM412	1 second clock		• This relay repeats on/off at certain intervals. • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)
SM413	2 second clock		• This relay repeats on/off at certain intervals. • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)
SM414	2n second clock		• This relay repeats on/off at regular intervals specified in SD414 (in units of seconds). (When the value in SD414 is changed, the elapsed time count that has started when the ON/OFF state of SM414 last changed continues, and the ON/OFF state changes when the new specified time is elapsed.[*1]) • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)
SM415	2n millisecond clock		• This relay repeats on/off at regular intervals specified in units of milliseconds on the SD415. (When the value in SD415 is changed, the elapsed time count that has started when the ON/OFF state of SM415 last changed continues, and the ON/OFF state changes when the new specified time is elapsed.[*1]) • This relay does not turn on/off per scan, but turns on/off whenever the specified time interval is reached. • The initial state when the CPU module is powered on or reset is off.	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SM420	User timing clock No.0		• This relay repeats on/off at specified scan intervals. • The initial state when the CPU module is powered on or reset is off. • The on/off scan interval is set using the DUTY instruction. (n1: ON scan interval, n2: OFF scan interval)	S (Every END)
SM421	User timing clock No.1			
SM422	User timing clock No.2			
SM423	User timing clock No.3			
SM424	User timing clock No.4			

*1 The following figures show operation examples of SM414/SM415 when a value in SD414/SD415 is changed.

Ex.

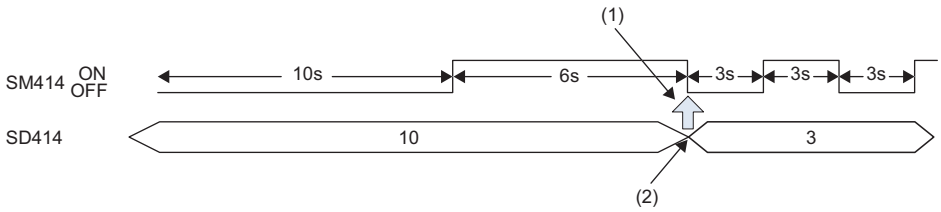
When a value in SD414 is changed from 3 to 10:



(1) The elapsed time after the ON/OFF state of SM414 changes remains.
(2) Value change

Ex.

When a value in SD414 is changed from 10 to 3:



(1) If the new interval in SD414 has already elapsed after the last change of the ON/OFF state of SM414, the ON/OFF state changes as soon as a value in SD414 is changed.
(2) Value change

Drive information

The following is the list of special relay areas relating to the drive information.

No.	Name	Data stored	Details	Set by (setting timing)
SM600	Memory card usable flags	Off: Disabled On: Enabled	This relay is on when an SD memory card is enabled (This relay switches to on when a valid SD memory card is inserted and prepared for use).	S (Status change)
SM601	Memory card protect flag	Off: Not protected On: Protected	This relay is on when the write protect switch of the SD memory card is set to on.	S (Status change)
SM603	Memory card (drive 2) flag	Off: No SD memory card inserted On: SD memory card inserted	This relay is on when an SD memory card is inserted. (This relay switches to on when an SD memory card is inserted regardless of its state (enabled/disabled) and type).	S (Status change)
SM604	Memory card in-use flag	Off: Not used On: In use	This relay is on when the SD memory card is being used.	S (Status change)
SM605	Memory card remove/ insert prohibit flag	Off: Allowed to remove/insert On: Not allowed to remove/insert	Turn on this relay to prohibit the SD memory card from being removed/inserted. When this relay is on, if SM607 is set to on, the system switches it to off.	U/S

No.	Name	Data stored	Details	Set by (setting timing)
SM606	SD memory card forced disable instruction	Off: Cancel instruction On: Forced disable instruction	• This relay can be switched to on to issue the instruction that forces the SD memory card to be disabled. However, whenever any function accesses the SD memory card, the disabling process does not start until the access is completed. • This relay can be switched to off to cancel the instruction that forces the SD memory card to be disabled.	U/S (Status change)
SM607	SD memory card forced disable status flag	Off: Not disabled On: Disabled	• This relay switches to on when the SD memory card is disabled by switching SM606 to on. • This relay switches to off when the forcibly-disabled SD memory card is enabled by switching SM606 to off.	S (Status change)
SM626	Extended SRAM cassette insertion flag	Off: Not inserted On: Inserted	This relay is on when the extended SRAM cassette is inserted.	S (Status change)
SM628	Program memory write error	Off: No write operation/normal On: Write error	This relay switches to on when a write error is detected during write operation to the program memory. This relay switches to off when the write instruction is issued.	S (Writing)
SM629	Program memory write flag	Off: No write operation On: Executing write operation	This relay is on when the write process to the program memory is in progress. The relay is switched to off when the write process is completed.	S (Writing)
SM630	Program memory overwrite count error flag	Off: The number of rewrite operations is less than 100000 On: The number of rewrite operations reaches 100000	This relay switches to on when the number of program memory rewriting operations reaches 100000 (CPU module must be replaced).	S (Writing)
SM632	Data memory write error	Off: No write operation/normal On: Write error	This relay switches to on when a write error is detected during write operation to the data memory. This relay switches to off when the write instruction is issued.	S (Writing)
SM633	Data memory write flag	Off: No write operation On: Executing write operation	This relay is on when the write process to the data memory is in progress. The relay is switched to off when the write process is completed.	S (Writing)
SM634	Number of rewriting operations error to data memory flag	Off: The number of rewrite operations is less than 100000 On: The number of rewrite operations reaches 100000	This relay switches to on when the number of data memory rewriting operations reaches 100000 (CPU module must be replaced).	S (Writing)
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 less than 100000 On: The number of rewrite operations reaches 100000	This relay switches to on when the number of system memory (Flash ROM)*1 rewrite operations reaches 100000 (CPU module must be replaced).	S (Writing)

*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 while the interrupt program is being executed.	S (Instruction execution)
SM701	Number of output characters selection	Off: Outputs until reaching NULL code On: Outputs 16 characters	• ASCII codes are output until reaching NULL (00H) code, when SM701 is off. • 16 characters of ASCII codes are output, when SM701 is on. • This relay saves/returns while the interrupt program is being executed.	U
SM702	Search method	Off: Sequential search On: Dichotomizing search	• The search method in the search instruction can be specified by using this relay. • To use the dichotomizing search, data must be sorted. • This relay saves/returns while the interrupt program is being executed.	U
SM703	Data sort instruction sort order	Off: Ascending On: Descending	• This relay can be used to specify how to arrange data in the data sort instruction: ascending or descending. • This relay saves/returns while the interrupt program is being executed.	U
SM704	Block comparison	Off: Mismatch is detected On: Completely match	• This relay switches to on when all the data conditions are met in the block data comparison instruction. • This relay saves/returns while the interrupt program is being executed.	S (Instruction execution)
SM709	DT/TM instruction improper data detection flag	Off: No improper data On: Improper data is detected	• This relay switches to on when the comparison target data is date data or cannot be recognized as clock data, or the comparison target device (three words) exceeds the specified device range in the DT/TM instruction. • This relay saves/returns while the interrupt program is being executed.	S (Instruction execution)/U
SM752	Dedicated instruction End bit control flag	Off: Automatically controlled On: Not automatically controlled	This relay can be used to set whether the system automatically controls the on/off operation of End bit being used for the dedicated instruction and other (control of End bit which is registered in the End processing of the instruction completion and turned on for only the next one scan (Off: turned on for only one scan after the instruction is completed as well as normal/error End bit, On: turned on when the instruction is completed and the On status is maintained after that)).	U
SM753	File being accessed	Off: Not in progress On: In progress	• This relay is on during file access by the SP.FWRITE/SP.FREAD/SP.DEVST instruction. • This relay is on while the SP.FTPPUT/SP.FTPGET instruction is being executed. • This relay is on during access to the SD memory card and data memory.	S (Status change)
SM754	BIN/DBIN instruction error control flag	Off: Executes error detection On: No execute error detection	This relay can be switched to on when the error detection is not desirable in the BIN/DBIN instruction.	U
SM755	Scaling data check settings	Off: Performs data check On: Not perform data check	This relay can be used to enable/disable the check whether the scaling data is sorted in ascending order when the SCL/DSCL/SCL2/DSCL2 instruction is being executed.	U

No.	Name	Data stored	Details	Set by (setting timing)
SM756	Module access completion wait control flag	Off: Not wait the completion On: Waits the completion	This relay can be used to determine whether the system must wait until the access is completed before starting the next instruction when the write access instruction to the buffer memory of another module is being executed.	U
SM775	Selection of refresh processing during the COM instruction execution	Off: Executes all the refresh processing On: Executes the refresh processing specified in SD775	This relay can be used to select the refresh processing target when the COM instruction is executed from two options: all refresh processing or only refresh processing specified in SD775.	U
SM776	Local device setting at CALL	Off: Disables local devices On: Enables local devices	This relay can be used to determine whether the local devices on the subroutine program called when the CALL instruction is being executed are enabled or not.	U
SM777	Local device setting in interrupt programs	Off: Disables local devices On: Enables local devices	This relay can be used to determine whether the local devices are enabled or not when the interrupt program is being executed. However, local devices/safety local devices of a program file in the storage location are always used in the standard program/safety program, regardless of the setting of SM777.	U
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
SM816	Hold mode (S.IN instruction)	Off: Value not held On: Value held	Whether to hold the output value or not is specified when the input value is found to be exceeding the valid range during the range check processing of the S.IN instruction.	U
SM817	Hold mode (S.OUT instruction)	Off: Value not held On: Value held	This relay is used to specify whether or not to hold the output values of the S.OUT1, S.OUT2, and S.DUTY instructions if a sensor error occurs.	U

Ethernet function

The following is the list of special relay areas relating to the Ethernet function.

No.	Name	Data stored	Details	Set by (setting timing)
SM1520	IP address storage area write request	Off→On: Write request exists On→Off: Writing completed	• When this relay is changed from off to on, IP address settings stored in SD1518 to SD1525 are written to the IP address storage area (system memory*1) of the CPU module. • Once write to the IP address storage area (system memory*1) is completed (regardless of whether it was successful or not), this relay is turned off.	S (Status change)/U
SM1521	IP address storage area write error	Off: No error On: Error	This relay is turned to on when write to the IP address storage area (system memory*1) failed, and off when it was successful.	S (Status change)
SM1522	IP address storage area clear request	Off→On: Clear request exists On→Off: Storage area cleared	When this relay is changed from off to on, the IP address storage area (system memory*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 error	Off: No error On: Error	This relay is turned to on when clear of the IP address storage area (system memory*1) failed, and off when it was successful.	S (Status change)
SM1524	Initial processing successful completion state	Off: Completed with an error On: Completed normally	This relay is turned on when the initial processing for the Ethernet function was completed successfully.*2 When the initial processing was completed with an error, SM1525 is turned on and SM1524 stays off.	S (Status change)
SM1525	Initial processing error completion state	Off: Completed normally On: Completed with an error	This relay is turned on when the initial processing for the Ethernet function was completed with an error.*2 When the initial processing was completed successfully, SM1524 is turned on and SM1525 stays off.	S (Status change)

*1 This is the memory used by the system when the CPU module is executing functions.

*2 Ethernet function initial processing involves updating parameters set for data communication to Ethernet-equipped modules to enable communication with external devices. Initial processing of Ethernet-equipped modules is performed by setting Ethernet parameters, writing them to the CPU module, and then powering off and on or resetting the CPU module. Note that if Ethernet parameters are not set, initial processing of Ethernet-equipped modules will be performed based on default parameters. (SM1524 (initial processing successful completion status) turns on the moment initial processing is complete, and communication with external devices becomes possible.)

Online module change function

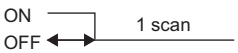
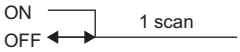
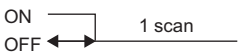
The following is the list of special relay areas relating to the online module change function.

No.	Name	Data stored	Details	Set by (setting timing)
SM1600	Module selection request flag	Off→On:Requested	This relay is turned on to select an online change target module. When changing a module directly, the system turns on this relay upon removal of the target module. The relay can be turned on only when the value set in SD1617 is 0 (Normal operation). The relay turns off upon completion of the online module change processing. If the selection cancel is requested, the relay turns off after the selection is cancelled.	S (Status change)/U (Request)
SM1601	Module selection completion flag	Off: No module selected On: Selected	This relay turns on when an online change target module has been selected. The relay turns off upon completion of the online module change processing.	S (Status change)
SM1602	Module removal request flag	Off→On: Requested	This relay is turned on to request a removal of the selected module. When changing a module directly, the system turns on this relay upon removal of the target module. The relay can be turned on only when the value set in SD1617 is 2 (Module selected). The relay turns off upon completion of the online module change processing.	S (Status change)/U (Request)
SM1603	Module removal ready flag	Off: Not prepared On: Ready	This relay turns on when the selected module is ready to be removed. The relay turns off upon completion of the online module change processing.	S (Status change)
SM1604	Module removal completion flag	Off: Not completed On: Completed	This relay turns on when the selected module has been removed. The relay turns off upon completion of the online module change processing.	S (Status change)
SM1605	Module mounting completion flag	Off: Not completed On: Completed	This relay turns on when a new module has been mounted. The relay turns off upon completion of the online module change processing.	S (Status change)
SM1606	Module recognition request flag	Off→On: Requested	This relay is turned on to request recognition of the newly-mounted module. When changing a module directly, the system turns on this relay upon mounting of the module. The relay can be turned on only when the value set in SD1617 is 6 (Module mounted). The relay turns off upon completion of the online module change processing.	S (Status change)/U (Request)
SM1607	Module recognition completion flag	Off: Not recognized On: Recognized	This relay turns on when the newly-mounted module is recognized by the system. The relay turns off upon completion of the online module change processing.	S (Status change)
SM1608	Module control resumption request flag	Off→On: Requested	This relay turns on to start control of the replaced module. When changing a module directly, the system turns on this relay upon recognition of the module. The relay can be turned on only when the value set in SD1617 is 8 (Module recognized). The relay turns off upon completion of the online module change processing.	S (Status change)/U (Request)
SM1609	Online module change completion flag	Off: Not completed On: Completed	The relay turns on upon completion of the online module change processing. This relay turns off in the next scan.	S (Status change)
SM1615	Module selection cancellation request flag	Off→On: Module selection cancellation requested	This relay is turned on to cancel a module selection request. The relay can be turned on only when the value set in SD1617 is 2 (Module selected). The relay turns off after the selection is cancelled.	S (Status change)/U (Request)
SM1616	Online module change availability flag	Off: Disabled On: Enabled	This relay turns on when only the Process CPU is used in a single CPU system.	S (Initial)
SM1617	Online module change status flag	Off: Function not executed On: Function being executed	This relay turns on when SM1600 is turned on to start the online module change processing. The relay turns off upon completion of the online module change processing.	S (Status change)
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)

No.	Name	Data stored	Details	Set by (setting timing)
SM1619	Disable request flag during online module change	Off: No disable request On: Disable request detected	This relay turns on when a disable request is issued during the online module change processing. The relay turns off upon completion of the online module change processing.	S (Status change)

Redundant function

The following is the list of special relay areas relating to the Redundant function.

No.	Name	Data stored	Details	Set by (setting timing)
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
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

No.	Name	Data stored	Details	Set by (setting timing)
SM1673	Tracking transfer completion flag	Off: Transfer not completed On: Transfer completed	• This flag is stored a result of the tracking transfer operated in the preceding END processing. • The flag turns on if any of the tracking transfers in block 1 to 64 has been normally completed, and turns off if the tracking transfers failed due to causes such as tracking communication error.	S (Status change)
SM1679	Error reset (the other system)	Off→On: Standby system error reset requested On→Off: Standby system error reset completed	• This relay is turned from off to on to clear a continuation error occurred in the standby system. • This relay turns from on to off when the error reset has been completed. • Initial value is off.	U/S (Status change)
SM1680	Error of the other system monitoring	Off: No error On: Error	• This relay turns on if an error occurs on communications with the other system when an initial processing (including when the system is determined while waiting for the other system starts up) or an END processing is performed. (This relay turns on when the bit of SD1648 turns on.) • This relay turns off when an error is cleared.	S (Initial/every END/ system switching)
SM1681	Latest self-diagnostic error (including annunciator ON) (the other system)	Off: No error On: Error	• This relay turns on if a diagnostic error occurs in the CPU module in the other system. (The relay also turns on if an error is detected by an annunciator.) • The SM0 status for the CPU module in the other system is reflected.	S (Every END)
SM1682	Latest self-diagnostic error (not including annunciator ON) (the other system)	Off: No error On: Error	• This relay turns on if a self-diagnostic error occurred in the CPU module in the other system. (The relay remains off if an error is detected by an annunciator.) • The SM1 status for the CPU module in the other system is reflected.	S (Every END)
SM1683	Detailed information 1: Flag in use (the other system)	Off: Not used On: In use	• This relay turns on when there is detailed information 1 for an error occurred in the CPU module in the other system. • The SM80 status for the CPU module in the other system is reflected.	S (Every END)
SM1684	Detailed information 2: Flag in use (the other system)	Off: Not used On: In use	• This relay turns on when there is detailed information 2 for an error occurred in the CPU module in the other system. • The SM112 status for the CPU module in the other system is reflected.	S (Every END)
SM1754	Waiting for the start-up of the other system	Off: Started up (own system) On: Waiting for the start-up (the other system)	• This relay turns on while waiting for the start-up of the other system after powered on. • This relay turns off under the following conditions. · Own system starts up as the control system or the standby system after succeeding in tracking communications with the other system. · Own system is started up as the control system by a certain operation while waiting for the start-up of the other system.	S (Status change)
SM1756	Wait timeout for receiving cyclic data after system switching	Off: No timeout On: Timeout	This relay turns on when the receipt of the cyclic data after system switching is not completed within the cyclic data receipt waiting time[*1] while the setting to wait cyclic data receive after system switching is enabled. This relay turns off when the timeout does not occur.	S (At system switching)

*1 ☞ Page 556 Waiting time for cyclic data receive after system switching (Twcyc)

Safety information

The following is a list of special relay areas relating to safety information.

No.	Name	Data stored	Details	Set by (setting timing)
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. (Clear by turning power off and on or resetting.)	S (Status change)
SM1904	Safety communication setting (1st module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1904.	S (Initial)
SM1905	Safety communication setting (2nd module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1905.	S (Initial)
SM1906	Safety communication setting (3rd module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1906.	S (Initial)
SM1907	Safety communication setting (4th module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1907.	S (Initial)
SM1908	Safety communication setting (5th module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1908.	S (Initial)
SM1909	Safety communication setting (6th module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1909.	S (Initial)
SM1910	Safety communication setting (7th module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1910.	S (Initial)
SM1911	Safety communication setting (8th module)	Off: Not set On: Set	• This relay stores the safety communication setting status. • When safety communications are available (the relay is on), the start I/O number of the target module is stored in SD1911.	S (Initial)

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

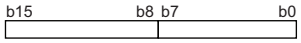
Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

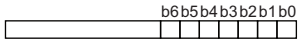
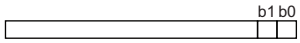
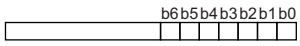
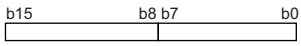
A

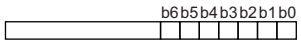
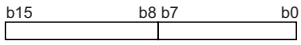
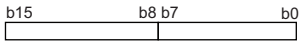
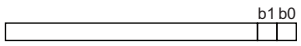
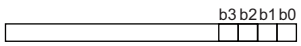
Diagnostic information

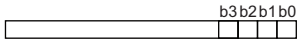
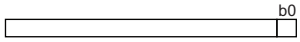
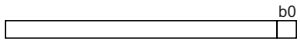
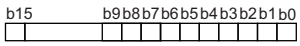
The following is the list of special register areas relating to the diagnostic information.

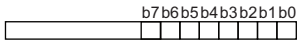
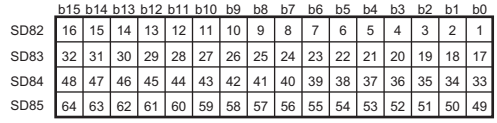
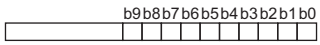
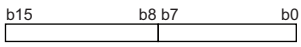
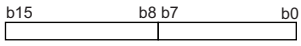
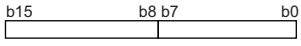
No.	Name	Data stored	Details	Set by (setting timing)
SD0	Latest self-diagnostics error code	Latest self-diagnostics error code	Error codes are stored in a hexadecimal value when the diagnostics detects an error.	S (Error)
SD1	Latest self-diagnostics error time	Latest self-diagnostics error time	The year 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		

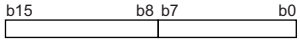
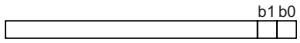
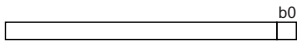
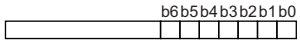
No.	Name	Data stored	Details	Set by (setting timing)
SD53	AC/DC DOWN	The number of AC/DC DOWN detections	The value of this register increments by one and stored as a BIN code, each time input voltage drops to 85% (AC power)/65% (DC power) or less of the nominal value while the CPU module is carrying out an operation. A counting cycle from 0 to 65535 to 0 is repeated. (In a redundant system with redundant extension base unit, if a momentary power failure occurs in a module on an extension base unit, counting will be performed in both systems.)	S (Error)
SD60	Number of module with blown fuse	Number of module with blown fuse	• The lowest I/O number of module in which a fuse blew is stored. • The fuse blown state check is also done for output modules on the remote I/O station.	S (Error)
SD61	I/O module verify error module number	I/O module verify error module number	• The lowest I/O number of the module which has an error detected by the I/O module verification is stored. • I/O module verification is also done for modules on the remote I/O station.	S (Error)
SD62	Annunciator number	Annunciator number	The annunciator number which is detected first is stored.	S (Instruction execution)
SD63	Number of annunciators	Number of annunciators	The number of detected annunciator is stored.	S (Instruction execution)
SD64 to SD79	Table of detected annunciator numbers	Detected annunciator number	• Numbers of activated annunciators are sequentially registered into SD64 to SD79, each time an annunciator (F) is turned on by the SET F instruction. • Number of the annunciator deactivated by the RST F instruction is removed from register areas SD64 to SD79, and the numbers of annunciators which were lined up behind the removed one move forward one by one. When the number of detected annunciators has already reached 16, the number of a newly detected annunciator is not stored into SD64 to SD79. For details on the annunciator, refer to the annunciator (F) (☞ Page 291 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 50: Safety station system configuration information	S (Error)

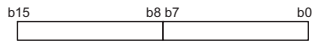
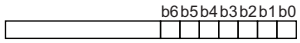
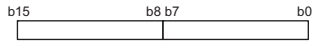
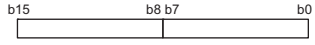
No.	Name	Data stored	Details	Set by (setting timing)
SD81 to SD111	Detailed information 1	Detailed information 1	• Detailed information 1 corresponding to the error code (SD0) is stored. • The type of the detailed information 1 can be obtained using SD80 (the value of the "Detailed information 1 information category code" stored in SD80 corresponds to the following figures (1), (2), (4) to (7), (24), (27), (34), (37) and (38)). (1) Program position information ■SD81: With or without specification  b0: Argument number b4: Sequence step number b5: FB number b6: File name ■SD82: Argument number (stored in the range from 1) ■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)

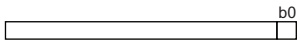
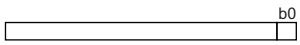
No.	Name	Data stored	Details	Set by (setting timing)
SD81 to SD111	Detailed information 1	Detailed information 1	(5) System configuration information ■SD81: With or without specification  b0: I/O number b1: Slot number b2: Base number b3: Power supply number b4: CPU number b5: Network number b6: Station number ■SD82: I/O number ■SD83  • b0 to b7: Slot number (0 to 11) • b8 to b15: Base number (0: Main base unit) ■SD84  • b0 to b7: Power supply number (1 to 2: Power supply 1 to 2) • b8 to b15: CPU number: (1 to 4: CPU No.1 to 4) ■SD85: Network number ■SD86: Station number 0 to 120 (0 is stored for the master station.) (6) Number of times information When there is no specification, 0 is set to each SD. ■SD81: With or without specification  b0: Number of times (set value) b1: Number of times (actual measurement value) ■SD82 to SD83: Number of times (set value) When the number of times (set value) is not specified, 0 is set to each SD. • SD82: Lower word of the number of times (set value) • SD83: Upper word of the number of times (set value) ■SD84 to SD85: Number of times (actual measurement value) When the number of times (actual measurement value) is not specified, 0 is set to each SD. • SD84: Lower word of the number of times (actual measurement value) • SD85: Upper word of the number of times (actual measurement value) (7) Time information When there is no specification, 0 is set to each SD. ■SD81: With or without specification  b0: Time (value set) (ms) b1: Time (value set) (μs) b2: Time (actual measurement value) (ms) b3: Time (actual measurement value) (μs)	S (Error)

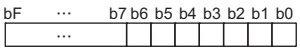
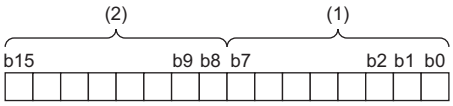
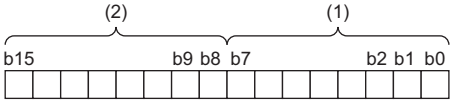
No.	Name	Data stored	Details	Set by (setting timing)
SD81 to SD111	Detailed information 1	Detailed information 1	■SD82: Time (value set) (ms) ■SD83: Time (value set) (μs) ■SD84: Time (actual measurement value) (ms) ■SD85: Time (actual measurement value) (μs) (24) Failure information Failure information is system information. (27) System switching information ■SD81: With or without specification  b0: System switching cause b1: System switching instruction ID number b2: Cause of system switching failure b3: Control system/standby system transition ■SD82: System switching cause 1: Power-off, reset, hardware failure 2: Stop error 3: System switching request from a network module 16: System switching request by using the SP.CONTSW instruction 17: System switching request using an engineering tool ■SD83: System switching instruction ID number ■SD84: Cause of system switching failure 1: Tracking communications disabled 2: Tracking communication timeout 3: Stop error of the standby system 4: Operating status mismatch between both systems 5: Memory copy being executed 6: Online change being executed 7: A failure of a network module detected on the standby system 8: System switching being executed 9: A redundant function module being changed online 10: System switching disabled on the standby system by using the DCONTSW instruction 14: Safety operation mode mismatch between both systems 15: A failure of safety tracking data detected ■SD85: Control system/standby system transition 1: Control system → Standby system 2: Standby system → Control system (34) TEST MODE continuous RUN prevention setting ■SD81: With or without specification  b0: 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 b5: System switching 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  	S (Error)
SD81 to SD111	Detailed information 1	Detailed information 1	(50) Safety station system configuration information ■SD81: With or without specification  b0: Own station I/O number b1: Slot number on the own station base unit b2: Own station base number b3: Own station power supply number b4: Own station CPU number b5: Network number b6: Station number b7: Connection number b8: Slot number on another station base unit b9: Another station base number ■SD82: Own station I/O number ■SD83: Own station base number/Slot number on the own station base unit  • b0 to b7: Slot number on the base unit (0 to 11) • b8 to b15: Base number (0: Main base unit) ■SD84: Own station CPU number/Own station power supply number  • b0 to b7: Power supply number (1 to 2: Power supply 1 to 2) • b8 to b15: CPU number: (1 to 4: CPU No.1 to 4) ■SD85: Network number ■SD86: Station number 0 to 120 (0 is stored for the master station.) ■SD87: Connection number 0 to 120 ■SD88: Another station base number/Slot number on another station base unit  • b0 to b7: Slot number on the base unit (0 to 11) • b8 to b15: Base number (0: Main base unit)	S (Error)

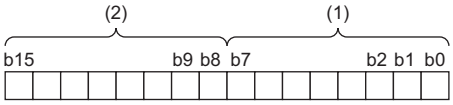
No.	Name	Data stored	Details	Set by (setting timing)
SD112	Detailed information 2 information category	Detailed information 2 information category code	Detailed information 2 information category code is stored.  • b0 to b7: Information category code • b8 to b15: Unused (fixed to 0) The following codes 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)
SD113 to SD143	Detailed information 2	Detailed information 2	• Detailed information 2 corresponding to the error code (SD0) is stored. • The type of information can be checked in SD112. (The value of the "Detailed information 2 information category code" stored in SD112 corresponds to the following (2) to (5), (25), (28), (29) and (57).) (2) Drive number and file name ■SD113: With or without specification  b0: Drive number b1: File name ■SD114: Drive number ■SD115 to SD122: File name (first eight characters of Unicode string) (3) Annunciator number ■SD113: With or without specification  b0: Annunciator number ■SD114: Annunciator number (4) Parameter information ■SD113: With or without specification  b0: Parameter type b1: Parameter storage location b2: I/O number b3: Parameter number b4: Network number b5: Station number b6: System information	S (Error)

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) ■SD116  • b0 to b7: Power supply number (1 to 2: Power supply 1 to 2) ■SD117: Network number ■SD118: Station number 0 to 120 (0 is stored for the master station.)	S (Error)

No.	Name	Data stored	Details	Set by (setting timing)
SD113 to SD143	Detailed information 2	Detailed information 2	(25) Process control instruction processing information ■SD113: With or without specification  b0: Process control instruction processing information ■SD114: Processing description The following are processing blocks stored in SD114 (Processing details). 1: — (Instruction with no processing block) 2: Range check 3: Input limiter 4: Engineering value inverse transformation 5: Digital filter 6: Input addition 7: Variation rate & upper/lower limiter 8: Reset windup 9: Output conversion 10: Output ON time conversion 11: Variation rate check 12: Integrated value calculation 13: Control cycle determination 14: SV setting 15: Tracking 16: Gain (Kp) operation 17: PID operation 18: Deviation check 19: PID operation 1 (operation processing of Bn or Cn) 20: PID operation 2 (operation processing of Dn) 21: PID operation 3 (operation processing of ΔMV) 22: PIDP operation 23: Operating time monitor 24: SPI operation 25: IPD operation 26: BPI operation 27: Engineering value transformation 28: Variation rate limiter 29: Ratio calculation 30: Upper/lower limit check 31: Loop stop 32: MV correction 33: Two-position (on/off) control 34: Three-position (on/off) control 35: Operation constant check 36: SV count-up 37: MVPGS operation 38: Output 39: Input check 40: Timeout determination 41: Step manipulated value set 42: Sampling cycle determination 43: Response waveform observation 44: Identification processing 45: PID constants calculation (28) Program error information ■SD113: With or without specification  b0: Program error code ■SD114: Program error code	S (Error)

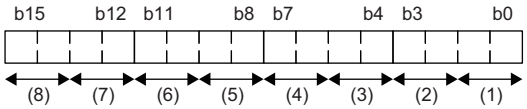
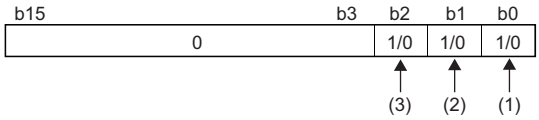
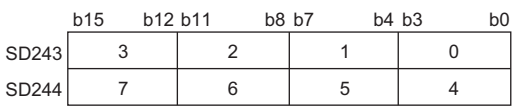
No.	Name	Data stored	Details	Set by (setting timing)
SD113 to SD143	Detailed information 2	Detailed information 2	(29) Other station error information (CC-Link IE Field) ■SD113: With or without specification  b0: Error classification b1: Error code b2: Date (yyyymmdd) b3: Time (hhmmss) b4: Day of the week b5: Detailed error information 1 b6: Detailed error information 2 b7 to bF: Detailed error information 3 to detailed error information 10 ■SD114: Error classification ■SD115: Error code ■SD116 and SD117: Date (yyyymmdd) • SD116: Lower word of date (yyyymmdd) • SD117: Upper word of date (yyyymmdd) ■SD118 and SD119: Time (hhmmss) • SD118: Lower word of time (hhmmss) • SD119: Upper word of time (hhmmss) ■SD120: Day of the week ■SD121: Detailed error information 1 ■SD122: Detailed error information 2 ■SD123 to SD130: Detailed error information 3 to 10	S (Error)
SD150	Power-off/power supply voltage drop detection status	Power-off/power supply voltage drop detection status (bit pattern) 0: Power-on/normal voltage 1: Power-off/voltage drop detected/no power supply module	• For the redundant power supply base unit or redundant extension base unit, the power supply module status (power is shut off, power supply voltage drop (not including a momentary power failure) is detected, or an empty slot for the power-supply module exists) is stored using the following bit pattern.  (1) Powered-off/power supply voltage dropped (power supply module 1) b0:Main base unit (2) Powered-off/power supply voltage dropped (power supply module 2) b8: Main 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.  (1) Failure detection status of the power supply module 1 b0:Main base unit (2) Failure detection status of the power supply module 2 b8: Main 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.	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.  (1) Details of the invalid power supply module 1 b0:Main base unit (2) Details of the invalid power supply module 2 b8: Main 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)
SD200	Status of switch	Status of CPU switch	The switch status of the CPU module is stored as follows: 0: RUN, 1: STOP	S (Switch change)
SD201	LED status	CPU-LED state	This register stores the information that indicates LED status (0: off, 1: on, 2: flashing (fast/slow)) of the CPU module in the following bit patterns.  (1) READY (2) ERROR (3) PROGRAM RUN (4) USER (5) BATTERY (6) CARD READY (7) CARD ACCESS (8) FUNCTION	S (Status change)
SD203	Operating status of CPU	Operating status of CPU	The operating status of the CPU module is stored as follows: 0: RUN, 2: STOP, 3: PAUSE	S (Every END)
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)
SD206	Pair version	Pair version	The 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)
SD243 SD244	Number of base slots	Number of base slots	The number of slots of the base unit, which is specified in the Base / power supply / extension cable setting of system parameters, is stored. When the number of slots of the base unit is not specified by the system parameter, that of the mounted base unit is stored. (In a redundant system with redundant extension base unit, the number is stored in the CPU module of both systems.)  0: Main 1 to 7: Extension 1 to 7	S (Initial)

A

No.	Name	Data stored	Details	Set by (setting timing)
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)
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)		
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)

No.	Name	Data stored	Details	Set by (setting timing)
SD306	Number of points assigned to the file register	ZR (L)	• The number of points assigned to the ZR device is stored in 32 bits. • The number of points is stored even when the number of points assigned to ZR is 32K or less.	S (Initial)
SD307		ZR (H)		
SD308	Number of points assigned to refresh devices	RD (L)	• The number of points assigned to the RD device is stored in 32 bits. • The number of points is stored even when the number of points assigned to RD is 32K or less.	S (Initial)
SD309		RD (H)		
SD312	File register block number	File register block number	The block number of the file register currently selected is stored.	S (Status change)

System clock

The following is the list of special register areas relating to the system clock.

No.	Name	Data stored	Details	Set by (setting timing)
SD412	One second counter	The number of counts that is counted once per second.	• The value in this register increments by one for each second after the CPU module enters in RUN mode. • A counting cycle from 0 to 65535 to 0 is repeated.	S (Status change)
SD414	2n second clock setting	Unit setting for 2n second clock	• The n value of the 2n second clock is stored (Default: 30). • Configurable range is -32768 to 32767 (0 to FFFFH).	U
SD415	2n ms clock setting	Unit setting for 2n ms clock	• The n value for the 2n ms clock is stored. (Default: 30). • Configurable range is -32768 to 32767 (0 to FFFFH).	U
SD420	Scan counter	The number of counts that is counted once for each scan.	• The value in this register increments by one for each scan after the CPU module enters in RUN mode (however, the count is skipped for scans by the initial execution type program). • A counting cycle from 0 to 65535 to 0 is repeated.	S (Every END)

Fixed scan function information

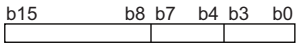
The following is the list of special register areas relating to the fixed scan function information.

No.	Name	Data stored	Details	Set by (setting timing)
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 (it is measured in increments of μs). SD518: stores a value in the ms place (storage range: 0 to 65535) SD519: stores a value in the μs place (storage range: 0 to 999) • These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN.	S (Every END)
SD519		Initial scan time (unit: μs)		
SD520	Current scan time	Current scan time (unit: ms)	• The current scan time is stored into SD520 and SD521 (it is measured in increments of μs). SD520: stores a value in the ms place (storage range: 0 to 65535) SD521: stores a value in the μs place (storage range: 0 to 999) Example: If the current scan time is 23.6ms, the following values are stored: SD520 = 23 SD521 = 600 • These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN. • In a redundant system, when the operating status of the CPU module is RUN state, these areas are cleared to 0 at the system switching.	S (Every END)
SD521		Current scan time (unit: μs)		
SD522	Minimum scan time	Minimum scan time (unit: ms)	• The minimum value of the scan time other than one for the initial execution program is stored into SD522 and SD523 (it is measured in increments of μs). SD522: stores a value in the ms place (storage range: 0 to 65535) SD523: stores a value in the μs place (storage range: 0 to 999) • These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN. • In a redundant system, when the operating status of the CPU module is RUN state, these areas are cleared to 0 at the system switching.	S (Every END)
SD523		Minimum scan time (unit: μs)		
SD524	Maximum scan time	Maximum scan time (unit: ms)	• The maximum value of the scan time other than one for the initial execution program is stored into SD524 and SD525 (it is measured in increments of μs). SD524: stores a value in the ms place (storage range: 0 to 65535) SD525: stores a value in the μs place (storage range: 0 to 999) • These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN. • In a redundant system, when the operating status of the CPU module is RUN state, these areas are cleared to 0 at the system switching.	S (Every END)
SD525		Maximum scan time (unit: μs)		
SD526	END processing time	END processing time (unit: ms)	• The time period from completion of a scan program until start of the next scan is stored into SD526 to SD527 (it is measured in increments of μs). SD526: stores a value in the ms place (storage range: 0 to 65535) SD527: stores a value in the μs place (storage range: 0 to 999) • These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN. • In a redundant system, when the operating status of the CPU module is RUN state, these areas are cleared to 0 at the system switching.	S (Every END)
SD527		END processing time (unit: μs)		
SD528	Constant scan wait time	Constant scan wait time (unit: ms)	• The waiting time specified in the constant scan setting process is stored into SD528 and SD529 (it is measured in increments of μs). SD528: stores a value in the ms place (storage range: 0 to 65535) SD529: stores a value in the μs place (storage range: 0 to 999) • These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN. • In a redundant system, when the operating status of the CPU module is RUN state, these areas are cleared to 0 at the system switching.	S (Every END)
SD529		Constant scan wait time (unit: μs)		

No.	Name	Data stored	Details	Set by (setting timing)
SD530	Scan program execution time	Scan program execution time (unit: ms)	• The execution time of the scan program for one scan is stored into SD530 and SD531 (it is measured in increments of μs). SD530: stores a value in the ms place (storage range: 0 to 65535) SD531: stores a value in the μs place (storage range: 0 to 999) • These areas are cleared to 0 when the operating status of the CPU module is switched from STOP to RUN. • In a redundant system, when the operating status of the CPU module is RUN state, these areas are cleared to 0 at the system switching.	S (Every END)
SD531		Scan program execution time (unit: μs)		

Drive information

The following is the list of special register areas relating to the drive information.

No.	Name	Data stored	Details	Set by (setting timing)
SD600	Memory card mounting status	SD memory card type	This register indicates the type of mounted SD memory cards.  • b0 to b3: Fixed to 0 • b4 to b7 0: Does not exist 4: SD memory card b8 to b15: Fixed to 0	S (Initial, card insertion/ removal)
SD604	SD memory card (drive 2) usage status	SD memory card (drive 2) usage status	Usage status of the SD memory card is stored using the following bit pattern. (On indicates being used.) b0: Event history b1: Module extension parameter*1 b2: Label communication data b3 to b15: Not used	S (Status change)
SD606	SD memory card (drive 2) capacity	SD memory card (drive 2) capacity: the lower digits (unit: K bytes)	The capacity of the SD memory card is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial, card insertion/ removal)
SD607		SD memory card (drive 2) capacity: the higher digits (unit: K bytes)	The capacity of the SD memory card is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial, card insertion/ removal)
SD610	SD memory card (drive 2) free space	SD memory card (drive 2) free space: the lower digits (unit: K bytes)	The amount of free space of the SD memory card is stored in increments of 1K byte.	S (At change)
SD611		SD memory card (drive 2) free space: the higher digits (unit: K bytes)	The amount of free space of the SD memory card is stored in increments of 1K byte.	S (At change)
SD614	Device/label memory (drive 3) usage status	Device/label memory (drive 3) usage status	Usage status of the device/label memory is stored using the following bit pattern. (On indicates being used.) b0: File register b1 to b15: Not used	S (Status change)
SD616	Device/label memory (drive 3) capacity	Device/label memory (drive 3) capacity: the lower digits (unit: K bytes)	The capacity of the device/label memory is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)
SD617		Device/label memory (drive 3) capacity: the higher digits (unit: K bytes)	The capacity of the device/label memory is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)
SD618	Device/label memory (file storage area) capacity	Device/label memory (file storage area) (drive 3) capacity: the lower digits (unit: K bytes)	The capacity of the device/label memory (file storage area) is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)
SD619		Device/label memory (file storage area) (drive 3) capacity: the higher digits (unit: K bytes)	The capacity of the device/label memory (file storage area) is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)
SD620	Data memory (drive 4) usage status	Data memory (drive 4) usage status	Usage status of the data memory is stored using the following bit pattern. (On indicates being used.) b0: Event history b1: Module extension parameter*1 b2: Label communication data b3 to b15: Not used	S (Status change)
SD622	Data memory (drive 4) capacity	Data memory (drive 4) capacity: the lower digits (unit: K bytes)	The capacity of the data memory is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)
SD623		Data memory (drive 4) capacity: the higher digits (unit: K bytes)	The capacity of the data memory is stored in increments of 1K byte (the amount of free space for a formatted memory card is stored).	S (Initial)
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)

No.	Name	Data stored	Details	Set by (setting timing)
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) 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.) b1 to b15: Not used	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".)

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)
SD771	Specification of the number of write instruction executions to data memory	Specification of the number of write instruction executions to data memory	In this register, the maximum number of the data memory writing instruction (SP.DEVST) executions per day is specified. When the number of data memory writing instruction executions exceeds the value specified in this register, an error is generated. Setting range is from 1 to 32767. If a value is set outside the range, an error is generated during the data memory writing instruction execution.	U
SD775	Selection of refresh processing during the COM instruction execution	Selection of refresh processing during the COM instruction execution	In this register, whether or not each processing is executed during the COM instruction execution is selected (Default: 0). The specification on the SD775 is effective when SM775 is turned on. b15 ··· b13 ··· b6 ··· b4 b3 b2 b1 b0 ■ Refresh processing (0: Not executed, 1: Executed) b0: I/O refresh b1: Link refresh of the CC-Link module b2: Link refresh of the CC-Link IE Controller Network module b3: Intelligent function module refresh b6: Link refresh of the CC-Link IE Field Network module ■ Device/label access service processing execution (0: Executed, 1: Not executed) b15: Device/label access service processing (communications with the engineering tool, GOT, or other external devices)	U
SD792 and SD793	PID limit setting (for complete derivative)	0: Limit restriction applied 1: No limit restriction	The limit restriction for each PID loop is specified as follows: (for the PIDCONT instruction) SD792: b15 16 to b1 2, b0 1 SD793: 32 to 18, 17 1 to 32: Loop 1 to 32	U
SD794 and SD795	PID limit setting (for incomplete derivative)	0: Limit restriction applied 1: No limit restriction	The limit restriction for each PID loop is specified as follows: (for the S.PIDCONT instruction) SD794: b15 16 to b1 2, b0 1 SD795: 32 to 18, 17 1 to 32: Loop 1 to 32	U
SD816 SD817	Basic period	Execution cycle	An execution cycle (unit: second) of process control instructions is set in real number.	U

No.	Name	Data stored	Details	Set by (setting timing)
SD818	Bumpless function availability setting for the S.PIDP instruction	0: Enabled 1: Disabled	The availability of the bumpless function for the S.PIDP instruction is set.	U
SD820	Dummy device	Dummy device	A dummy device used in process control instructions is set.	U
SD821				

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: b15 ... b1 b0 SD1400: I15 to I1 I0 SD1401: I31 to I17 I16 SD1463: I1023 to I1009 I1008	S (at execution)

User authentication function

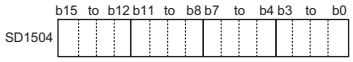
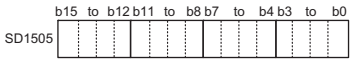
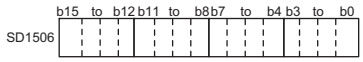
The following is the list of special register area relating to the user authentication function.

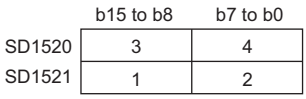
No.	Name	Data stored	Details	Set by (setting timing)
SD1468	Number of login users	Number of login users	The number of users logged in to the CPU module is stored. The value is incremented by one every time the login operation to the programmable controller or files is performed. The value is decremented by one every time the logout operation is performed.	S (Status change)

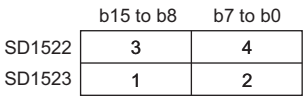
A

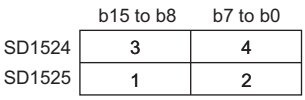
Ethernet function

The following is the list of special register areas relating to the Ethernet function.

No.	Name	Data stored	Details	Set by (setting timing)
SD1504	Open completion signal	In this register, open completion status is stored.	Open statuses of connection No.1 to 16 are stored. (0: Close/Open not completed, 1: Open completed).  b0: Connection 1 b1: Connection 2 b2 to b15: Connection 3 to 16 For details on the on/off timing, refer to the following. MELSEC iQ-R Ethernet User's Manual (Application)	S (Status change)
SD1505	Open request signal	In this register, open request status is stored.	Open processing statuses of connection No.1 to 16 are stored. (0: No open request, 1: Open request exists).  b0: Connection 1 b1: Connection 2 b2 to b15: Connection 3 to 16 For details on the on/off timing, refer to the following. MELSEC iQ-R Ethernet User's Manual (Application)	S (Status change)
SD1506	Socket communications receive status signal	In this register, receive status is stored.	Receiving statuses of connections No.1 to 16 are stored. (0: No data received, 1: Data receiving completed)  b0: Connection 1 b1: Connection 2 b2 to b15: Connection 3 to 16 For details on the on/off timing, refer to the following. MELSEC iQ-R Ethernet User's Manual (Application)	S (Status change)

<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="6">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). 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), the value of IP address stored in the IP address storage area (system memory) is stored.


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). 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), the value of subnet mask pattern stored in the IP address storage area (system memory) is stored.


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 gateway IP address (lower)</td><td rowspan="2">• Specify the default gateway IP address to be stored into the IP address storage area (system memory). 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), the value of default gateway IP address stored in the IP address storage area (system memory) is stored.


1 to 4: First to fourth byte</td><td rowspan="2">S (Status change)/U</td></tr>
<tr><td>SD1525</td><td>Default gateway 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) is 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) 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>

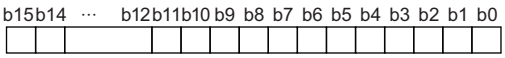
Online module change function

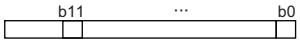
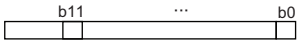
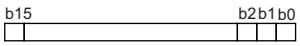
The following is the list of special register areas relating to the online module change function.

No.	Name	Data stored	Details	Set by (setting timing)
SD1600	Module selection (base unit No.)	Base unit No. where the online change target module is mounted	The base unit number where the online change target module is mounted is specified. 0: Main base unit 1: Extension base unit 1 ⋮ 7: Extension base unit 7 FFFFH: Not specified (Default) The stored value returns to its default (FFFFH) upon completion of the online module change processing. When changing a module directly, the system stores the corresponding base unit number upon removal of the target module.	S (Status change)/U (Request)
SD1601	Module selection (slot No.)	Slot No. where the online change target module is mounted	The slot number where the online change target module is mounted is specified. 0: Slot No.0 1: Slot No.1 ⋮ 11: Slot No.11 FFFFH: Not specified (Default) The 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 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 461 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 461 Error codes related to the online module change function)	S (Status change)

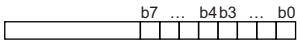
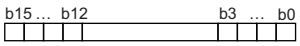
Redundant function

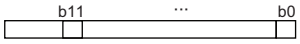
The following is the list of special register areas relating to the redundant function.

No.	Name	Data stored	Details	Set by (setting timing)
SD1642	BACKUP/SEPARATE LED flashing cause	BACKUP/SEPARATE LED flashing cause	Flashing cause of BACKUP/SEPARATE LED is stored using the following bit pattern.  Corresponding bits to each cause turns on. b1: Tracking communications disabled b3: Stop error of the standby system b4: Operating status mismatch between both systems b5: Memory copy being executed b6: Online change being executed b7: A failure of a network module detected on the standby system b8: System switching being executed b9: A redundant function module being changed online b10: System switching disabled on the standby system by using the DCONTSW instruction b14: Safety operation mode mismatch between both systems b15: A failure of safety tracking data detected • 0 is stored as initial value. • When the multiple causes occurred at the same time, bits corresponding to each cause turn on simultaneously. If the tracking communications are disabled or the redundant function module is being changed online, the other causes turn off. • When the redundant function module in the other system is being changed online, "Tracking communications disabled" is stored in own system. • Even if a cause of the system switching failure has occurred, the system is switched when the system switching causes which have higher priority, such as power-off or reset of the CPU module, have occurred. (☞ Page 219 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)
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 14: Safety operation mode mismatch between both systems 15: A failure of safety tracking data detected • This register is initialized with 0 when own system is powered on. • Zero is stored in this register upon completion of system switching.	S (At system switching)

No.	Name	Data stored	Details	Set by (setting timing)
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
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)

No.	Name	Data stored	Details	Set by (setting timing)
SD1653	Memory copy destination I/O number	Memory copy destination I/O number	• Before SM1653 is turned off and on, the I/O number of the memory copy destination (CPU module in the standby system: 03D1H) is stored. • The initial value is 0.	U
SD1654	Memory copy completion status	Memory copy completion status	This register stores the execution status of the memory copy. When the value other than 0 is stored, the memory copy is completed with an error or cannot be executed. 0H: Completed normally Other than 0H: Completed with an error or cannot be executed. For details on the values stored when an error occurs, refer to the error codes. (☞ Page 389 List of Error Codes)	S (Status change)
SD1662	Tracking transfer data receive completion wait time	Value to be added to the tracking transfer data receive completion wait time	• Specify the value to be added to waiting time for completion of tracking data reception for the CPU module in the control system. • Specify the value in units of ms. (Range is 0 to 2100ms.) If the value out of the range is specified, the system recognized the value as 2100ms. • The initial value is 0ms.	U
SD1664	Tracking transfer error count	Tracking transfer error count	• For the tracking transfer of the device/label data at each scan, the number is added (+1) when the tracking transfer cannot be executed because of disconnection of the tracking cables, power-off, reset, stop error of the CPU module in the standby system, or error in the redundant function module. • A counting cycle from 0 to 65535 to 0 is repeated.	S (Error)
SD1667 to SD1670	Tracking transfer trigger	Off: No trigger On: Trigger	• When data is transferred based on the tracking transfer setting of redundant settings, specify the target block as trigger. ■SD1667 b0 to b15: Block 1 to block 16 ■SD1668 b0 to b15: Block 17 to block 32 ■SD1669 b0 to b15: Block 33 to block 48 ■SD1670 b0 to b15: Block 49 to block 64 • When "Transfer Automatically" is selected for the tracking block No.1 auto transfer setting, b0 of SD1667 is turned on by the system at power-on or when the CPU module is switched from STOP to RUN. In other cases, turn on bit0 of SD1667 to b15 of SD1670 by the user. • Initial values for b0 of SD1667 to b15 of SD1670 are off (no trigger)	S (Initial)/U
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)

No.	Name	Data stored	Details	Set by (setting timing)
SD1680	CPU module operation information (the other system)	CPU module operation information (the other system)	This register stores the operating status of the CPU module in the other system in the following bit pattern. When communications with the other system are disabled, 00FFH is stored.  b0 to b3: 0H: RUN, 2H: STOP, 3H: PAUSE, FH: Communications with the other system disabled b4 to b7: 0H: No error, 1H: Continuation error, 2H: Stop error, FH: Communications with other systems disabled Communications with the other system are disabled in the following states. • The CPU module in the other system is powered off or is being reset. • A hardware failure has occurred in own or the other system. • Tracking cables are not connected or disconnected.	S (Initial/every END)/system switching
SD1681	Latest self-diagnostic error code (the other system)	Latest error code (the other system)	This register stores an error code for the error occurred in the other system with hexadecimal. SD0 of the CPU module in the other system is reflected.	S (Every END)
SD1682 to SD1688	Latest self-diagnostic error time (the other system)	Latest self-diagnostic error time (the other system)	• This register stores the error time in the other system. • Data configuration is the same as SD1 to SD7. • The values in SD1 to SD7 of the CPU module in the other system are reflected.	S (Every END)
SD1689	Detailed information 1 information category (the other system)	Information category code of the detailed information 1 (the other system)	• This register stores the category code of the detailed information 1 for the error that has occurred in the other system. • Data configuration is the same as SD80. • The value in SD80 of the CPU module in the other system is reflected.	S (Every END)
SD1690 to SD1720	Detailed information 1 (the other system)	Detailed information 1 (the other system)	• This register stores the detailed information 1 corresponding to the error that has occurred in the other system. • Data configuration is the same as SD81 to SD111. • The values in SD81 to SD111 of the CPU module in the other system are reflected.	S (Every END)
SD1721	Detailed information 2 information category (the other system)	Information category code of the detailed information 2 (the other system)	• This register stores the category code of the detailed information 2 for the error that has occurred in the other system. • Data configuration is the same as SD112. • The value in SD112 of the CPU module in the other system is reflected.	S (Every END)
SD1722 to SD1752	Detailed information 2 (the other system)	Detailed information 2 (the other system)	• This register stores the detailed information 2 corresponding to the error that has occurred in the other system. • Data configuration is the same as SD113 to SD143. • The values in SD113 to SD143 of the CPU module in the other system are reflected.	S (Every END)
SD1754	Cause of control system start-up	Cause of start-up as the control system	This register stores the cause that one of the redundant system has been started up as the control system. • This register is initialized with 0 when the CPU module is powered off and on or is reset.  b0 to b3: Cause of start-up by specific operations to start up the system as the control system 0: No specific operation, 1: Switch operation (RUN→STOP→RUN), 2: Online operation, 4: Contact input (X) b12 to b15: Information when the system is started up as the control system 0: Before start-up, 1: Start-up as the control system by starting up the other system, 2: Start-up as the control system by the specific operation	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SD1755	Tracking cable connection status	Tracking cable connection status	• This register stores the tracking cable connection status when an initial processing (including when the system is determined while waiting for the other system starts up) or an END processing is performed. 0H: Normal 12H: Cable disconnection of the OUT side 13H: Cable insertion error of the OUT side 14H: Communications on the OUT side being established 21H: Cable disconnection of the IN side 22H: Cable disconnection of the IN or OUT side 23H: Cable disconnection of the IN side or cable insertion error of the OUT side 24H: Cable disconnection of the IN side or communications on the OUT side being established 31H: Cable insertion error of the IN side 32H: Cable insertion error of the IN side or cable disconnection of the OUT side 33H: Cable insertion error of the IN or OUT side 34H: Cable insertion error of the IN side or communications on the OUT side being established 41H: Communications on the IN side being established 42H: Communications on the IN side being established or cable disconnection of the OUT side 43H: Communications on the IN side being established or cable insertion error of the OUT side 44H: Communications on the IN or OUT side being established • The tracking cable status can be checked with this register when bit of SD1648 turns on.	S (Initial/every END)
SD1756	Module information on wait timeout for receiving cyclic data after system switching	Timeout occurrence status (bit pattern) Off: No timeout On: Timeout	• When the receipt of the cyclic data after system switching is not completed within the cyclic data receipt waiting time*1 while the setting to wait cyclic data receive after system switching is enabled, this register turns on the bit corresponding to the slot on the main base unit on which a timeout has occurred. (The bits of the positions where no modules are mounted turn off.) • When a timeout does not occur, this register turns off the corresponding bit.  b0 to b11: Slot No.0 to slot No.11	S (At system switching)

*1 ☞ Page 556 Waiting time for cyclic data receive after system switching (Twcyc)

A

Safety information

The following is the list of special register areas relating to the safety information.

No.	Name	Data stored	Details	Set by (setting timing)
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 (it is measured in increments of μs). SD1890: The ms part is stored. (Range: 0 to 65535) SD1891: The μs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.	S (Status change)
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 (it is measured in increments of μs). SD1892: The ms part is stored. (Range: 0 to 65535) SD1893: The μs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.	S (Status change)
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 (it is measured in increments of μs). SD1894: The ms part is stored. (Range: 0 to 65535) SD1895: The μs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.	S (Status change)
SD1895		Maximum safety cycle processing time (unit: μs)		
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 Safety communication target station (1st module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)
SD1905	Start I/O number of Safety communication target station (2nd module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)

No.	Name	Data stored	Details	Set by (setting timing)
SD1906	Start I/O number of Safety communication target station (3rd module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)
SD1907	Start I/O number of Safety communication target station (4th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)
SD1908	Start I/O number of Safety communication target station (5th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)
SD1909	Start I/O number of Safety communication target station (6th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)
SD1910	Start I/O number of Safety communication target station (7th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)
SD1911	Start I/O number of Safety communication target station (8th module)	0 to FFH: Start I/O number FFFFH: Not set	• The value obtained by dividing the start I/O number of the station targeted for safety communications by 16 is stored. • When the safety communication function is not used, FFFFH is stored.	S (Initial)

Appendix 6 List of Safety Special Relay Areas

The following table lists items in the list.

Item	Description
No.	Safety special relay number
Name	Safety special relay name
Description	Data stored in the safety special relay and its meaning
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (safety program, engineering tool)*1 • U/S: User and system <Set timing> • Every END: Data is set every time END processing is performed. • Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN). • Status change: Data is set when the status is changed. • Error: Data is set when an error occurs. • Instruction execution: Data is set when an instruction is executed. • Request: Data is set when requested by a user (using the safety special relay). • Writing: Data is set when a user performs a writing operation. • During END: Data is set when END processing is performed. • Power-on to RUN or STOP to RUN: Data is set when the operating status changes from power-on to RUN or from STOP to RUN.

*1 As an operation from the GOT or other external devices, only data read is available.

Point

Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

System clock

The following is the list of safety special relay areas relating to the system clock.

No.	Name	Data stored	Details	Set by (setting timing)
SA\SM400	Always on		This relay is always on.	S (Power-on to RUN or STOP to RUN)
SA\SM401	Always off		This relay is always off.	S (Power-on to RUN or STOP to RUN)
SA\SM444	On at the first safety program execution		This relay turns on at the first safety program execution.	S (Power-on to RUN or STOP to RUN)
SA\SM1800	ON for only the first safety program execution after system switching (standby system to control system)		This relay turns on only the first safety cycle processing after the standby system was switched to the control system.	S (Status change)

A

Safety information

The following is the list of safety special relay areas relating to safety information.

No.	Name	Data stored	Details	Set by (setting timing)
SA\SM1008	Safety refresh communication status of each module (1st module)	Off: Normal On: Communication error	The safety refresh communication status of the first CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1904) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1008 to SA\SD1015.)	S (Status change)
SA\SM1016	Safety refresh communication status of each module (2nd module)	Off: Normal On: Communication error	The safety refresh communication status of the second CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1905) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1016 to SA\SD1023.)	S (Status change)
SA\SM1024	Safety refresh communication status of each module (3rd module)	Off: Normal On: Communication error	The safety refresh communication status of the third CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1906) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1024 to SA\SD1031.)	S (Status change)
SA\SM1032	Safety refresh communication status of each module (4th module)	Off: Normal On: Communication error	The safety refresh communication status of the fourth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1907) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1032 to SA\SD1039.)	S (Status change)
SA\SM1040	Safety refresh communication status of each module (5th module)	Off: Normal On: Communication error	The safety refresh communication status of the fifth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1908) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1040 to SA\SD1047.)	S (Status change)
SA\SM1048	Safety refresh communication status of each module (6th module)	Off: Normal On: Communication error	The safety refresh communication status of the sixth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1909) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1048 to SA\SD1055.)	S (Status change)
SA\SM1056	Safety refresh communication status of each module (7th module)	Off: Normal On: Communication error	The safety refresh communication status of the seventh CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1910) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1056 to SA\SD1063.)	S (Status change)
SA\SM1064	Safety refresh communication status of each module (8th module)	Off: Normal On: Communication error	The safety refresh communication status of the eighth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1911) is stored. (The safety refresh communication status of each safety connection is stored in SA\SD1064 to SA\SD1071.)	S (Status change)
SA\SM1088	Module switch request for safety communication information	Off: Module specification completed On: Module specification requested	The SA\SD1088 content is updated when this relay changes from off to on. This relay turns off when SA\SD1090 to SA\SD1097 and SA\SD1104 to SA\SD1223 data update is complete.	U (Request)/S (Status change)

Appendix 7 List of Safety Special Register Areas

The following table lists items in the list.

Item	Description
No.	Safety special register number
Name	Safety special register name
Description	Data stored in the safety special register
Details	Detailed description of the data stored
Set by (setting timing)	Set side of data (system or user) and timing when data is set by the system <Set by> • S: System • U: User (safety program, engineering tool)*1 • U/S: User and system <Set timing> • Every END: Data is set every time END processing is performed. • Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN). • Status change: Data is set when the status is changed. • Error: Data is set when an error occurs. • Instruction execution: Data is set when an instruction is executed. • Request: Data is set when requested by a user (using the safety special relay). • Switch change: Data is set when the switch of the CPU module is changed. • Card insertion/removal: Data is set when an SD memory card is inserted or removed. • Writing: Data is set when a user performs a writing operation. • During END: Data is set when END processing is performed.

*1 As an operation from the GOT or other external devices, only data read is available.

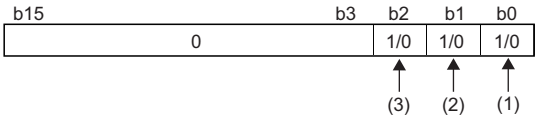
Point

Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

A

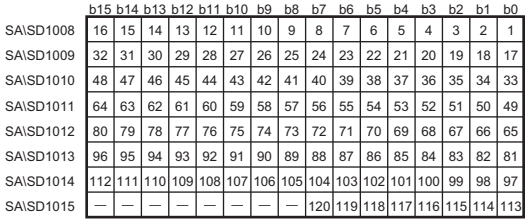
System information

The following is the list of safety special register areas relating to system information.

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

The following is the list of safety special register areas relating to safety information.

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 first CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1904) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223.  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 second CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1905) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1024 to SA\SD1031	Safety refresh status of each safety connection (3rd module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the third CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1906) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1032 to SA\SD1039	Safety refresh status of each safety connection (4th module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the fourth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1907) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)

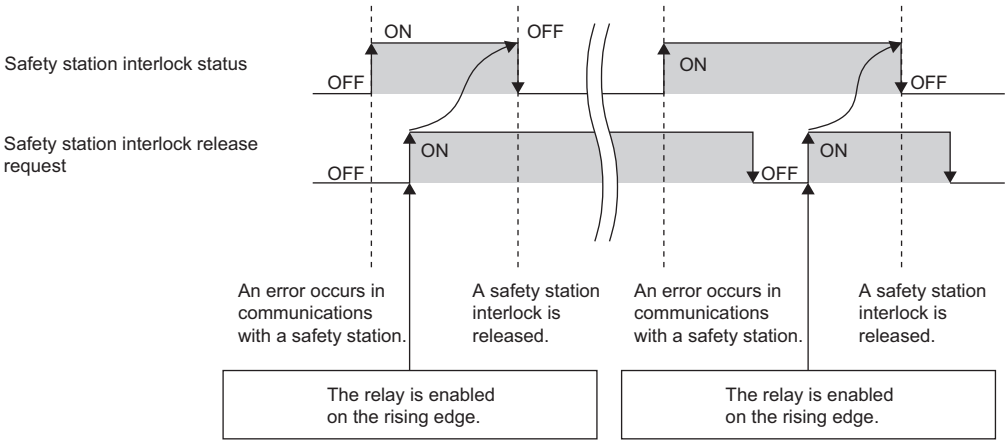
No.	Name	Data stored	Details	Set by (setting timing)
SA\SD1040 to SA\SD1047	Safety refresh status of each safety connection (5th module)	0: Safety communications normal, safety connection not used, own station 1: Communication error	• The communication status of safety stations connected to the fifth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1908) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
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 sixth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1909) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	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 seventh CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1910) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	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 eighth CC-Link IE Field Network master/local module (the one whose start I/O number is stored in SD1911) is stored. • A communication error means that the safety station is not in any of the communication states stored in SA\SD1104 to SA\SD1223. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1088	Module setting for safety communication information	1 to 8:Target module	• The n-th 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	Target module for safety communication information	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 · 10H to 15H: Performing safety initial communications*2 · 30H: Safety communication stopped (This value is stored when the safety communication error occurs and the systems are switched in interlock status.) · 1000H or later: For details on the values stored when an error occurs, refer to the list of error codes. (☞ Page 389 List of Error Codes)	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
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 safety connection turns on. 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)
SA\SD1240 to SA\SD1247	Interlock release request for each safety connection (1st module)	0: Do not release the interlock. 1: Release the interlock.	Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1 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 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 safety connection turns on. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	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.	Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1 The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	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 safety connection turns on. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	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.	Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1 The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	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 safety connection turns on. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	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.	Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1 The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	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 safety connection turns on. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SA\SD1304 to SA\SD1311	Interlock release request for each safety connection (5th module)	0: Do not release the interlock. 1: Release the interlock.	Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1 The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	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 safety connection turns on. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	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.	Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1 The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	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 safety connection turns on. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1336 to SA\SD1343	Interlock release request for each safety connection (7th module)	0: Do not release the interlock. 1: Release the interlock.	Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1 The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	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 safety connection turns on. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	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.	Turn off and on the bit corresponding to the safety connection to release the interlock. (The bit does not automatically turn off after execution is complete.)*1 The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	U (Request)
SA\SD1600 to SA\SD1607	Safety I/O hold status of each safety connection (1st module)	0: Not holding safety input/output 1: Holding safety input/output	The safety input/output hold status of safety stations connected to the 1st CC-Link IE Field Network module (the one whose start I/O number is stored in SD1904) is stored. b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SA\SD1600: 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 SA\SD1601: 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 SA\SD1602: 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 SA\SD1603: 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 SA\SD1604: 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 SA\SD1605: 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 SA\SD1606: 112 111 110 109 108 107 106 105 104 103 102 101 100 99 98 97 SA\SD1607: — — — — — — — — 120 119 118 117 116 115 114 113 1 to 120: Safety connection number —: Fixed to 0	S (Status change)
SA\SD1608 to SA\SD1615	Safety I/O hold status of each safety connection (2nd module)	0: Not holding safety input/output 1: Holding safety input/output	The safety input/output hold status of safety stations connected to the 2nd CC-Link IE Field Network module (the one whose start I/O number is stored in SD1905) is stored. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1616 to SA\SD1623	Safety I/O hold status of each safety connection (3rd module)	0: Not holding safety input/output 1: Holding safety input/output	The safety input/output hold status of safety stations connected to the 3rd CC-Link IE Field Network module (the one whose start I/O number is stored in SD1906) is stored. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1624 to SA\SD1631	Safety I/O hold status of each safety connection (4th module)	0: Not holding safety input/output 1: Holding safety input/output	The safety input/output hold status of safety stations connected to the 4th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1907) is stored. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SA\SD1632 to SA\SD1639	Safety I/O hold status of each safety connection (5th module)	0: Not holding safety input/output 1: Holding safety input/output	The safety input/output hold status of safety stations connected to the 5th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1908) is stored. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1640 to SA\SD1647	Safety I/O hold status of each safety connection (6th module)	0: Not holding safety input/output 1: Holding safety input/output	The safety input/output hold status of safety stations connected to the 6th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1909) is stored. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1648 to SA\SD1655	Safety I/O hold status of each safety connection (7th module)	0: Not holding safety input/output 1: Holding safety input/output	The safety input/output hold status of safety stations connected to the 7th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1910) is stored. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)
SA\SD1656 to SA\SD1663	Safety I/O hold status of each safety connection (8th module)	0: Not holding safety input/output 1: Holding safety input/output	The safety input/output hold status of safety stations connected to the 8th CC-Link IE Field Network module (the one whose start I/O number is stored in SD1911) is stored. The bit arrangement of the safety connection number is the same as that of the 1st module. (Note that the SA\SD numbers differ.)	S (Status change)

*1 A safety station interlock release request is executed when the bit rises, and the interlock status is then released. To release the interlock again, it is necessary to turn the request off and on again. After eliminating the cause of any communication errors, turn on the request with the CPU module. When performing safety communications between CPU modules, perform this operation with both CPU modules.



*2 If the safety communications do not start when the safety communication status keep performing safety initial communications (10H to 15H), check the safety communications partner station status or network communication status.

Appendix 8 Buffer Memory

The buffer memory is memory used with the following applications.

Module	Application
CPU module	Stores values such as Ethernet function setting values.
SIL2 function module	Stores the status of the SIL2 function module (such as diagnostics information and system information). (☞ Page 493 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))

A

SIL2 function module

The following table lists the SIL2 function module buffer memory addresses.

Address	Name	Data stored
Un\G0	Latest self-diagnostics error code	Error codes are stored in a hexadecimal value when the self-diagnostics detects an error.
Un\G1	Latest self-diagnostics error time	The year value (four digits) for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G2		The month value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G3		The day value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G4		The hour value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G5		The minute value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G6		The second value for the date/time when Un\G0 data was updated is stored as a BIN code.
Un\G7		The day value for the date/time when Un\G0 data was updated is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)
Un\G10 to Un\G25	Self-diagnostics error code 1 to 16	The maximum of 16 types of error codes are stored into Un\G10 onwards when the self-diagnostics detects an error. (The same error code as one already stored in Un\G10 onwards is not stored.) The 17th error code onwards are not stored. Also, error codes are not stored when 16 types of error codes have already been stored in Un\G10 to Un\G25.
Un\G50	Error clear	Set 1 in Un\G50 to clear the error. After the error is cleared, 0 is automatically set in Un\G50 by the system.
Un\G52	LED status	This buffer memory area stores the information that indicates the LED status (0: off, 1: on, 2: flashing) in the following bit patterns. • b0 to b1: READY • b2 to b3: ERROR • b4 to b5: PROGRAM RUN • b6 to b7: SAFETY COM RUN • b8 to b9: SAFETY COM ERR • b10 to b11: TEST
Un\G54	Safety cycle processing time execution cycle error count	The detected number of times that cycles for which safety cycle processing was not executed is stored.
Un\G60	Pair version	The pair version is stored in decimal notation.
Un\G62	Current safety cycle time (ms unit)	• The current safety cycle processing time is stored. (The time is measured in 1 μs units.) Un\G62: The ms part is stored. (Range: 0 to 65535) Un\G63: The μs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.
Un\G63	Current safety cycle time (μs unit)	
Un\G64	Minimum safety cycle time (ms unit)	• The minimum safety cycle processing time value is stored. (The time is measured in 1 μs units.) Un\G64: The ms part is stored. (Range: 0 to 65535) Un\G65: The μs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.
Un\G65	Minimum safety cycle time (μs unit)	
Un\G66	Maximum safety cycle time (ms unit)	• The maximum safety cycle processing time value is stored. (The time is measured in 1 μs units.) Un\G66: The ms part is stored. (Range: 0 to 65535) Un\G67: The μs part is stored. (Range: 0 to 999) • The measurement is stored even when the status is STOP.
Un\G67	Maximum safety cycle time (μs unit)	
Un\G70	Firmware version	The firmware version is stored.
Un\G80	Detailed information 1 information category	The detailed information 1 information category code is stored.
Un\G81 to Un\G111	Detailed information 1	• Detailed information 1 corresponding to the error code (Un\G0) is stored. • The stored information contains the following eight types. 0: N/A 1: Program position information 2: Drive number and file name 4: Parameter information 5: System configuration information 6: Number of times information 7: Time information 24: Failure information 34: Continuous RUN prevention setting in TEST MODE • The type of detailed information 1 can be obtained using Un\G80. (The bit patterns for values in the "Detailed information 1 information category code" stored in Un\G80 correspond to those in SD81 to SD111.) (☞ Page 493 List of Special Register Areas)
Un\G112	Detailed information 2 information category	The detailed information 2 information category code is stored.

Address	Name	Data stored
Un\G113 to Un\G143	Detailed information 2	• Detailed information 2 corresponding to the error code (Un\G0) is stored. • The stored information contains the following six types. 0: N/A 2: Drive number and file name 3: Annunciator number 4: Parameter information 5: System configuration information 25: Process control instruction processing information 29: Other station error information (CC-Link IE Field) • The type of detailed information 1 can be obtained using Un\G112. (The bit patterns for values in the "Detailed information 2 information category code" stored in Un\G112 correspond to those in SD113 to SD143.) (☞ Page 493 List of Special Register Areas)
Un\G150	Program memory writing	• b0: This relay switches to on when a write error is detected during write operation to the program memory. This relay switches to off when the write instruction is issued. • b1: This relay is on when the write process to the program memory is in progress. The relay is switched to off when the write process is completed. • b2: This relay switches to on when the number of program memory rewriting operations reaches 100000. (The module must be replaced.)
Un\G152	Program memory write (transfer) status	This register displays the write (transfer) status to the program memory with a percentage (0 to 100%). The initial value is "0". Upon completion of writing, this register is set to "100". It is set to "0" at the time when the write command is issued.
Un\G153 to Un\G154	Program memory write count index	• This register indicates the index value for the number of write operations to the program memory up to now (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value continues to be counted even when it exceeds 100000). If the index value exceeds 100000, the module must be replaced.
Un\G155	Data memory writing	• b0: This relay switches to on when a write error is detected during write operation to the data memory. This relay switches to off when the write instruction is issued. • b1: This relay is on when the write process to the data memory is in progress. The relay is switched to off when the write process is completed. • b2: This relay switches to on when the number of data memory rewriting operations reaches 100000. (The module must be replaced.)
Un\G157	Data memory write (transfer) status	This register displays write (transfer) status to the data memory with a percentage. (0 to 100%). The initial value is "0". Upon completion of writing, this register is set to "100". It is set to "0" at the time when the write command is issued.
Un\G158 to Un\G159	Index for the number of data memory write operations	• This register indicates the index value for the number of write operations to the data memory up to now (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value continues to be counted even when it exceeds 100000). If the index value exceeds 100000, the module must be replaced.
Un\G160	System memory writing	• b0: This relay switches to on when a write error is detected during write operation to the system memory (Flash ROM)*1. This relay switches to off when the write instruction is issued. • b1: This relay switches to on when the write process to the system memory (Flash ROM)*1 is in progress, and switches to off when the write process is complete. • b2: This relay switches to on when the number of system memory (Flash ROM)*1 rewrite operations reaches 100000. (The SIL2 function module or safety function module must be replaced.)
Un\G163 to Un\G164	Index for the number of system memory write operations	• This register indicates the index value for the number of write operations to the system memory (Flash ROM)*1 up to now. (stored as a 32-bit BIN value). However, the number of write operations is not equal to the index value. • When the index value exceeds 100000, an error is generated (the index value continues to be counted even when it exceeds 100000). If the index value exceeds 100000, the module must be replaced.

*1 This is the memory used by the system when the SIL2 function module is executing functions.

Appendix 9 Processing Time

The scan time of the CPU module is the sum of the instruction execution time, the program execution time, and the END processing time. Each of the processing time that constitutes the scan time is as follows.

Point

For the availability of functions depending on the CPU module, refer to the description of each function.

Instruction execution time

The instruction execution time is the total of each instruction processing time used for the program executed by the CPU module. For the processing time of each instruction, refer to the following.

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

Program execution time

The program execution time is the processing time when multiple programs are executed. When the interrupt program is executed, the overhead time during the interrupt program being executed is also included.

Processing time when multiple programs are executed (program switching time)

When multiple programs are executed, the scan time becomes longer.

■When multiple programs run

The processing time [μs] with multiple programs run is given by: the number of program files × 1.2.

Overhead time at execution of interrupt/fixed scan execution type program

This section describes each overhead time when each of the interrupt programs and fixed scan execution type programs is executed.

Point

The processing time for the interrupt program and fixed scan execution type program is as shown below.
Instruction processing time for each instruction + Overhead time + Each refresh processing time (when setting)

■Overhead time when executing the interrupt program

The overhead time when executing the interrupt program includes the one before executing the interrupt program and the other when the interrupt program finished.

• Overhead time before executing the interrupt program

Interrupt factor	Condition	Overhead time
Internal timer interrupt (I28 to I31)	Saves/stores the file register (R) block number.	33.2μs
	Not save/store the file register (R) block number.	20.6μs
Interrupt from module (I0 to I15, I50 to I1023)	Saves/stores the file register (R) block number.	42.2μs
	Not save/store the file register (R) block number.	30.1μs

• 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.	30.1μs
	Not save/store the file register (R) block number.	18.6μs
Interrupt from module (I0 to I15, I50 to I1023)	Saves/stores the file register (R) block number.	35.2μs
	Not save/store the file register (R) block number.	23.4μs

■Overhead time when executing the fixed scan execution type program

The following table lists the overhead times when executing the fixed scan execution type program.

Condition	Overhead time
Saves/stores the file register (R) block number.	65.3μs
Not save/store the file register (R) block number.	41.9μs

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 493 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	Without an extended SRAM cassette	0.36
	With an extended SRAM cassette	0.62
KM2		476

■Overhead time when ending safety cycle processing

The overhead time when ending safety cycle processing is 130μs.

Processing time of safety communications

This section describes the processing time of safety communications.

Safety communications processing time = Input processing time of safety communications[*1] + Output processing time of safety communications[*2]

*1 This value is added even when the module to be connected in the safety communication setting is a safety output module only.

*2 This value is added even when the module to be connected in the safety communication setting is a safety input module only.

■Input processing time of safety communications

(1) When using the following module: a CC-Link IE Field Network master/local module with the safety protocol version set only to "1" in the safety communication setting[*1]

*1 The safety protocol version can be set in the safety communication setting using the engineering tool whose version is "1.110Q" or later. The safety protocol version cannot be set using the engineering tool whose version is earlier than "1.110Q".

$T1 = (N \times KM1) + (M \times KM2) + KM3$ [μs]

- T1: Input processing time of safety communications for (1)
- N: Number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting
- M: Number of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting

Constant	Constant value	
	Firmware version of the CPU module is earlier than 14	Firmware version of the CPU module is 14 or later
KM1	11	11
KM2	15	15
KM3	110	110

(2) When using the following module: a CC-Link IE Field Network master/local module with the safety protocol version set only to "2" in the safety communication setting[*2]

*2 The safety protocol version can be set in the safety communication setting using the engineering tool whose version is "1.110Q" or later. The safety protocol version cannot be set using the engineering tool whose version is earlier than "1.110Q".

$T2 = (N \times KM1) + (M \times KM2) + KM3$ [μs]

- T2: Input processing time of safety communications for (2)
- N: Number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting
- M: Number of the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting

Constant	Constant value	
	Firmware version of the CPU module is earlier than 14	Firmware version of the CPU module is 14 or later
KM1	—	16
KM2	—	20
KM3	—	128

(3) When using the following modules in combination: a CC-Link IE Field Network master/local module with the safety protocol version set to "1" in the safety communication setting, the module with the version set to "2"[*3]

*3 The safety protocol version can be set in the safety communication setting using the engineering tool whose version is "1.110Q" or later. The safety protocol version cannot be set using the engineering tool whose version is earlier than "1.110Q".

T3 = (N1 × KM1) + (N2 × KM2) + ((M1 + M2) × KM3) + (M3 × KM4) + KM5 + KM6 [μs]

- T3: Input processing time of safety communications for (3)
- N1: Number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting
- N2: Number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting
- M1: Number of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting
- M2: Number of the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting
- M3: Total number of the CC-Link IE Field Network master/local modules mounted

Constant	Constant value	
	Firmware version of the CPU module is earlier than 14	Firmware version of the CPU module is 14 or later
KM1	—	12
KM2	—	17
KM3	—	8
KM4	—	13
KM5	—	135
KM6	—	73

■Output processing time of safety communications

(1) When using the following module: a CC-Link IE Field Network master/local module with the safety protocol version set only to "1" in the safety communication setting[*1]

*1 The safety protocol version can be set in the safety communication setting using the engineering tool whose version is "1.110Q" or later. The safety protocol version cannot be set using the engineering tool whose version is earlier than "1.110Q".

T1 = (N × KM1) + (M × KM2) + KM3 [μs]

- T1: Output processing time of safety communications for (1)
- N: Number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting
- M: Number of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting

Constant	Constant value	
	Firmware version of the CPU module is earlier than 14	Firmware version of the CPU module is 14 or later
KM1	6	6
KM2	15	15
KM3	89	89

(2) When using the following module: a CC-Link IE Field Network master/local module with the safety protocol version set only to "2" in the safety communication setting[*2]

*2 The safety protocol version can be set in the safety communication setting using the engineering tool whose version is "1.110Q" or later. The safety protocol version cannot be set using the engineering tool whose version is earlier than "1.110Q".

T2 = (N × KM1) + (M × KM2) + KM3 [μs]

- T2: Output processing time of safety communications for (2)
- N: Number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting
- M: Number of the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting

Constant	Constant value	
	Firmware version of the CPU module is earlier than 14	Firmware version of the CPU module is 14 or later
KM1	—	9.5
KM2	—	24
KM3	—	105

(3) When using the following modules in combination: a CC-Link IE Field Network master/local module with the safety protocol version set to "1" in the safety communication setting, the module with the version set to "2"[*3]

*3 The safety protocol version can be set in the safety communication setting using the engineering tool whose version is "1.110Q" or later. The safety protocol version cannot be set using the engineering tool whose version is earlier than "1.110Q".

T3 = (N1 × KM1) + (N2 × KM2) + ((M1 + M2) × KM3) + (M3 × KM4) + KM5 + KM6 [μs]

- T3: Output processing time of safety communications for (3)
- N1: Number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting
- N2: Number of connections of the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting
- M1: Number of the CC-Link IE Field Network master/local modules with the safety protocol version set to "1" in the safety communication setting
- M2: Number of the CC-Link IE Field Network master/local modules with the safety protocol version set to "2" in the safety communication setting
- M3: Total number of the CC-Link IE Field Network master/local modules mounted

Constant	Constant value	
	Firmware version of the CPU module is earlier than 14	Firmware version of the CPU module is 14 or later
KM1	—	8
KM2	—	10
KM3	—	14
KM4	—	12
KM5	—	114
KM6	—	114

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 × 8.1.

END processing time

The END processing time includes the following:

- Common processing time
- I/O refresh processing time
- Link refresh processing time for the network module
- Intelligent function module refresh processing time
- Prolonged time of END processing when executing each function
- Device/label access service processing time
- Processing time of the identification check for safety data
- Tracking transfer time (☞ Page 549 Increase in the scan time due to tracking transfer)

Common processing time

The common processing time by the system is 2.78ms to 137.07ms[*1] for one unit of CPU module (module is not mounted).

*1 When the scan time is around 20ms, the processing time will be approximately 10.60ms; when the scan time is around 100ms, the processing time will be approximately 17.50ms.

I/O refresh processing time

The I/O refresh processing time for module mounted on main base unit can be calculated by the following formula.

I/O refresh processing time [μs] = ((Number of input refresh points[*1] × KM1) + (Number of unit having number of input points × KM2) + KM3[*3]) + ((Number of output refresh points[*2] × KM4) + (Number of unit having number of output points × KM5) + KM6[*3])

*1 The value indicates the numeric value that is obtained through dividing the number of input points by 16.

*2 The value indicates the numeric value that is obtained through dividing the number of output points by 16.

*3 When the number of I/O points is 0, this value is handled as 0.

Condition			Constant value
MELSEC iQ-R series module	Input	KM1	0.04
		KM2	0.56
		KM3	11.2
	Output	KM4	0.02
		KM5	1.13
		KM6	7.58

Link refresh processing time for the network module

This section describes the link refresh processing time for the network module.

■Link refresh processing time for the CC-Link IE Controller Network module

The link refresh processing time between the CPU module and the CC-Link IE Controller Network module on the main base unit is calculated by the following formulas.

$\alpha T, \alpha R$ [ms] = KM1 + KM2 × ((LB + LX + LY + SB) ÷ 16 + LW + SW) + αU

αU [ms] = KM3 × (SBU ÷ 16 + SWU)

- αT: Link refresh time (sending side)
- αR: Link refresh time (receiving side)
- αU: Unit label (SB/SW) refresh time
- LB: Total number of points of link relay (LB) that is refreshed by the station*1
- LW: Total number of points of link register (LW) that is refreshed by the station*1
- LX: Total number of points of link input (LX) that is refreshed by the station*1
- LY: Total number of points of link output (LY) that is refreshed by the station*1
- SB: Number of points of link special relay (SB)*2
- SW: Number of points of link special register (SW)*2
- SBU: Number of points of link special relay (SB)*3
- SWU: Number of points of link special register (SW)*3

Condition	Constant value
KM1 ($\times 10^{-3}$)	58.17
KM2 ($\times 10^{-3}$)	0.02
KM3 ($\times 10^{-3}$)	0.04

*1 Total number of points here indicates the one of link devices that have been set in "Refresh Setting" and "Network Configuration Settings". Note that the number of points that has been assigned in the reserved station is excluded.

*2 Number of points here indicates the one of when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.

*3 Number of points here indicates the one of when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.

■Link refresh processing time for the CC-Link IE Field Network module

The link refresh processing time between the CPU module and the CC-Link IE Field Network module on the main base unit is calculated by the following formulas.

$\alpha T, \alpha R$ [ms] = KM1 + KM2 × ((RX + RY + SB) ÷ 16 + RWr + RWw + SW) + αU

αU [ms] = KM3 × (SBU ÷ 16 + SWU)

- αT: Link refresh time (sending side)
- αR: Link refresh time (receiving side)
- αU: Unit label (SB/SW) refresh time
- RX: Total number of points of remote input (RX) refreshed by the master station/local station*1
- RY: Total number of points of remote output (RY) refreshed by the master station/local station*1
- RWw: Total number of points of remote register (RWw) refreshed by the master station/local station*1
- RWr: Total number of points of remote register (RWr) refreshed by the master station/local station*1
- SB: Number of points of link special relay (SB)*2
- SW: Number of points of link special register (SW)*2
- SBU: Number of points of link special relay (SB)*3
- SWU: Number of points of link special register (SW)*3

Condition	Constant value
KM1 ($\times 10^{-3}$)	58.17
KM2 ($\times 10^{-3}$)	0.02
KM3 ($\times 10^{-3}$)	0.04

*1 Total number of points here indicates the one of link devices that have been set in "Refresh Setting" and "Network Configuration Settings".

*2 Number of points here indicates the one of when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.

*3 Number of points here indicates the one of when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.

*4 The base units here indicates the type of base unit where the network module targeted for refresh is mounted.

A

■Link refresh processing time for the CC-Link module

The link refresh processing time between the CPU module and the CC-Link module on the main base unit is calculated by the following formulas. (Remote net Ver.1 mode, Remote net Ver.2 mode)

$\alpha T, \alpha R$ [ms] = KM1 + KM2 × ((RX + RY + SB) ÷ 16 + RWr + RWw + SW) + αU

αU [ms] = KM3 × (SBU ÷ 16 + SWU)

- αT: Link refresh time (sending side)
- αR: Link refresh time (receiving side)
- αU: Unit label (SB/SW) refresh time
- RX: Total number of points of remote input (RX) refreshed by the master station/local station[*1*4]
- RY: Total number of points of remote output (RY) refreshed by the master station/local station[*1*4]
- RWw: Total number of points of remote register (RWw) refreshed by the master station/local station[*1*4]
- RWr: Total number of points of remote register (RWr) refreshed by the master station/local station[*1*4]
- 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 ($\times 10^{-3}$)	58.17
KM2 ($\times 10^{-3}$)	0.02
KM3 ($\times 10^{-3}$)	0.04

*1 Total number of points here indicates the one of link devices that have been set in "Refresh Setting" and "Network Configuration Settings".

*2 Number of points here indicates the one of when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.

*3 Number of points here indicates the one of when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.

*4 When a CC-Link module is mounted on a main base unit of a redundant system, the number of points is counted as 0 because configuring the settings for RX/RY/RWw/RWr in the link refresh setting does not enable the settings even if attempted.

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.

Point

The refresh processing time described in this manual is for the case when the "Target" is set to "Device" in the refresh setting. For the refresh processing time of when the refreshing target is set to the module label or refresh data register (RD), refer to the manual for the module used.

Refresh processing time of one module [μs] = Read refresh time*1 + Write refresh time*1

*1 When the number of settings of read refresh (Module → CPU module) or write refresh (CPU module → Module) is 0, each processing time is 0.

■Read refresh time [μs]

Number of read refresh settings × KM1 + Refresh time for the first item (A) + Refresh time for the second item (A) + ··· + Refresh time for the nth item (A) + KM2

■Write refresh time [μs]

Number of write refresh settings × KM4 + Refresh time for the first item (B) + Refresh time for the second item (B) + ··· + Refresh time for the nth item (B) + KM5

- A: KM3 × Number of refresh transfer (word) [μs]
- B: KM6 × Number of refresh transfer (word) [μs]
- n: Number of blocks for refresh settings*1
- KM1 to KM6: Constant value as shown below

Condition			Constant value
MELSEC iQ-R series module	KM1	Module on the main base unit → CPU module	1.79
	KM2	Module on the main base unit → CPU module	20.8
	KM3	Module on the main base unit → CPU module	0.09
	KM4	CPU module → Main base unit	0.96
	KM5	CPU module → Main base unit	15.04
	KM6	CPU module → Main base unit	0.02

*1 The number of blocks for refresh settings can be checked in "Auto Refresh Setting Total Counts" of "Module Parameter List" window. For details, refer to the manual for the module used.

A

Prolonged time of END processing when executing each function

This section describes the prolonged time of END processing when executing each function.

■Latch processing time

If the latch function is used and the range for the latch time setting is effective, the scan time becomes longer. Also when the time setting is configured for the latch time setting, the scan time may be prolonged in the next END processing after the specified time has passed. The increase in scan time when the latch range is set can be calculated by the following formula.

Increase in scan time [μs] = (KM1 × Number of settings of latch range setting[*1]) + (KM2 × (Number of points of bit device with latch specified ÷ 16 + Number of points of word device with latch specified + Number of points of double word device with latch specified × 2)) + KM3

*1 The latch range (1) and the latch range (2) are counted as different device types.

Condition		Constant value
When set to "each scan"	KM1	1.00
	KM2	0.09
	KM3	1.20
When set to "hour"	KM1	1.0
	KM2	0.004
	KM3	17.5

■Processing time when file register is used

When "Use File Register of Each Program" has been set through setting the file register, the scan time becomes longer. The increase in the scan time [ms] is given by: the number of program files × 0.016.

Point

When "Use Common File Register in All Programs" has been set, the scan time may not become longer.

■File batch online change

When the file batch online change is executed, the scan time becomes longer. The increase in scan time [ms] can be calculated by the following formula.

Calculation formula	Constant value
(KM1 × Number of program files[*1]) + (KM2 × Number of steps in the largest program[*2]) + KM3	KM1: 0.018 KM2: 0.06×10^{-3} KM3: 0.9
(KM4 × Number of program files to be written[*3]) + (KM5 × Total number of steps in programs[*4])	KM4: 2.34 KM5: 0.06×10^{-3}

*1 This indicates the number of program files that are written to the CPU module.
*2 This indicates the number of steps in the largest program file among the program files to which the file batch online change is performed.
*3 This indicates the number of program files to which the file batch online change is performed.
*4 This indicates the total number of steps in the program files to which the file batch online change is performed.

Device/label access service processing time

This section describes the device/label access service processing time when "Processing time = One time" in the device/label access service processing setting.

Condition			Device/label access service processing time		
			For operations for both systems	For operations for one system	
				Via redundant function module	Without redundant function module
Connected via USB	Ladder block change during RUN (online program change)	100 steps are inserted into the head of program of 10K steps.	1.0ms maximum*1	—	—
	Monitor data registration	Data register (D) (Number of device points = 32 points) is registered to monitor.	—	0.03ms maximum	0.03ms maximum
		Safety data register (SA\D) (Number of device points = 32 points) is registered to monitor.	—	0.03ms maximum	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 10K steps.	1.0ms maximum*1	—	—
	Monitor data registration	Data register (D) (Number of device points = 32 points) is registered to monitor.	—	0.07ms maximum	0.07ms maximum
		Safety data register (SA\D) (Number of device points = 32 points) is registered to monitor.	—	0.07ms maximum	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 10K steps.	1.0ms maximum*1	—	—
	Monitor data registration	Data register (D) (Number of device points = 32 points) is registered to monitor.	—	0.07ms maximum	0.07ms maximum
		Safety data register (SA\D) (Number of device points = 32 points) is registered to monitor.	—	0.07ms maximum	0.07ms maximum

*1 When data are written to the program using a pointer (P) during RUN, the processing time is extended depending on the pointer number used. For example, when the data are written to the program using P8191 during RUN, the processing time is extended up to 3.0ms.

Processing time of the identification check for safety data

The extended scan time when executing the identification check for safety data is 800 μs under the following conditions.

- Safety programs: 20K steps
- Create safety global label setting file (32 word definition)
- Create standard/safety shared label setting file (32 word definition)
- Safety cycle time: 10ms
- Perform safety communications for 3 connections
- Scan time: 1.5ms (when the identification check for safety data not executed)

Process control function processing time

The process control function processing time is the time required to process the control loop when the process control function is executed. The following table lists the processing time of some control loop examples.

Loop type	Configuration	Processing time
Two-degree-of-freedom PID control (S2PID)	S.IN, S.PHPL, S.2PID, and S.OUT1 instructions	268μs
PID control (SPID)	S.IN, S.PHPL, S.PID, and S.OUT1 instructions	252μs
Monitor (SMON)	S.IN and S.PHPL instructions	123μs

Point

For the processing time of each process control instruction, refer to the following.

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

Redundant function processing time

This section describes the processing time in the redundant function.

Increase in the scan time due to tracking transfer

The following describes the increase in the scan time of the CPU module due to tracking transfer.

The increase in the scan time determined by the following calculation formula is a rough standard for a system start-up. Check the increase in the scan time with the actual systems.

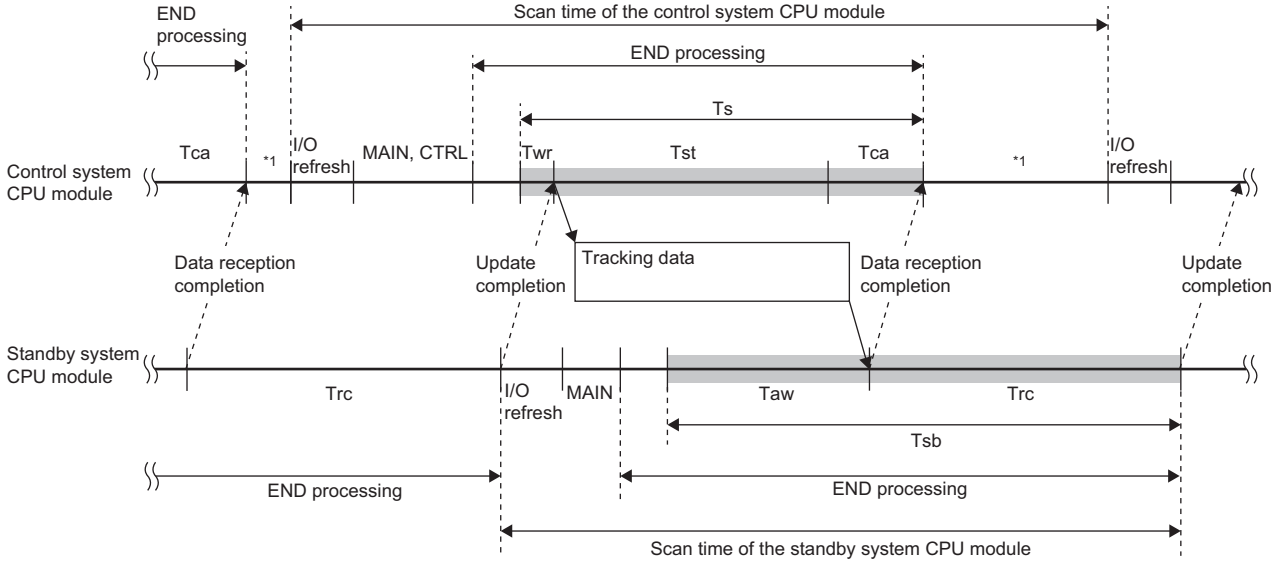
Point

The scan time can be checked in SD520 and SD521 (Current scan time). (☞ Page 508 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 550 Waiting time for completion of the previous tracking data reflection (Twr) Tst: Page 550 Tracking data send time (Tst) Tca: Page 552 Waiting time for completion of tracking data reception (Tca) α: Page 552 Other extended times (α)
Increase in the scan time of the CPU module in the standby system (Tsb)	Tsb = Taw + Trc + α[ms]	Taw: Page 552 Waiting time for tracking data reception (Taw) Trc: Page 553 Tracking data reflection time (Trc) α: Page 552 Other extended times (α)

Ex.

When CTRL is a control system execution program and MAIN is a program executed in both systems (in configuration with the main base unit only)



*1 If the constant scan is used, waiting time for the constant scan is generated.

■Waiting time for completion of the previous tracking data reflection (Twr)

The following describes the waiting time (Twr) for completion of the previous tracking data reflection in the CPU module of the control system.

Twr = 1 + Tdrm - Toth [ms]

- Tdrm: Maximum time taken for reflection of tracking data in the CPU module of the standby system
- Toth: Scan time excluding Ts (Increase in the scan time) in the CPU module of the control system

Tdrm is determined as follows.

- When no extended SRAM cassette is inserted: 1 + (No. of tracking blocks × 300.0×10^{-6}) + (Total size of tracking data [word] × 45.0×10^{-6})
- When an extended SRAM cassette is inserted: 1 + (No. of tracking blocks × 300.0×10^{-6}) + (Total size of tracking data [word] × 106.0×10^{-6})

If Tdrm - Toth is smaller than 0, Twr is handled as 1.

Twr is not generated in the first scan in which the tracking transfer is started.

■Tracking data send time (Tst)

The following describes the tracking data send time (Tst) in the CPU module of the control system. The calculation method differs depending on whether an extended SRAM cassette is inserted or not.

Extended SRAM cassette inserting status	Tracking data send time
Not inserted	$0.58 + (26.7 \times 10^{-6}) \times D1 + (43.5 \times 10^{-6}) \times D2 + (1.5 \times 10^{-3}) \times D5 + (66.0 \times 10^{-6}) \times D6$ [ms]
Inserted	$0.58 + (26.7 \times 10^{-6}) \times D1 + (113.5 \times 10^{-6}) \times D2 + (1.5 \times 10^{-3}) \times D5 + (66.0 \times 10^{-6}) \times D6$ [ms]

- D1: Size [word] of tracking data in the system data, signal flow memory (standard/safety), and refresh data register (RD)
- D2: Size [word] of tracking data of global devices (standard/safety), local devices (standard/safety), global labels (standard/safety), local labels (standard/safety), and standard/safety shared labels
- D5: Number of tracking transfer settings of global devices
- D6: Total size [word] of tracking data in the system data (safety)*1, signal flow memory (safety), local devices (safety), global labels (safety), local labels (safety), and global devices (safety)
- E1: Additional time when the SFC program is used = 3.5 [ms] (Only when the SFC program is used)
- F1: Additional time when the PID control instruction is used = 0.02 [ms] (Only when the PID control instruction is used)

*1 For the size of the system data (safety), refer to the value for D4 "■Variable depending on safety communication status" in "System data".

D1 is determined as follows.

Item	Size
System data	■Fixed data 8225 ■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 231 Tracking transfer setting)

*2 The CC-Link module that was set with the parameter is the target.

*3 This applies when the tracking device/label setting for the redundant settings of the CPU parameter is set to "Transfer collectively" (default) or when the tracking device/label setting is set to "Detailed setting" and "Module Label (Extension) Setting" is set to "Transfer".

*4 This applies when the tracking device/label setting of the redundant settings of the CPU parameter is set to "Detailed setting" and "Module Label (Extension) Setting" is set to "Do Not Transfer".

D2 is determined as follows.

Item	Size*5
Global device (other than the refresh data register (RD))	Follows the tracking transfer settings. (☞ Page 231 Tracking transfer setting)
Local device	Total size of local devices of the device/label memory area detailed setting × the number of programs on the program setting*7
Global label*6	Follows the global label settings.
Local label*6	Follows the local label settings.

*5 The size of the devices/labels to be transferred (tracked) can be checked by clicking the [Size Calculation] button in "Detailed Setting" of "Device/Label Detailed Setting". (☞ Page 234 Detailed setting)

*6 Even though global labels/local labels are deleted by the online change, the size of the tracking data of the global labels/local labels are the same as the one before the online change. The size is updated when the program and global label setting file to which all data is rebuilt (reassigned) are written to the programmable controller and the operating status of the CPU module is switched from STOP to RUN.

*7 This indicates the number of programs where local devices are set to use in the "Setting of Device/File Use Or Not" window of the program setting of the CPU parameter.

D4 is determined as follows.

Item	Size
System data	■Fixed data 7602 ■Data that varies depending on the tracking device/label setting 16 × (No. of tracking settings of T, ST, C, and LC) + 8 × (No. of global device tracking settings of other devices) ■Variable depending on safety communication status*8 136 + (29 × (Number of connections set in the safety communication setting of all the modules*9))
Signal flow memory (standard/safety)	(Total number of steps of control system execution programs, or total number of steps of rising/falling instruction in an FB)/16 (The digit after the decimal point is round up.)
Refresh data register (RD)	Follows the tracking transfer settings. (☞ Page 231 Tracking transfer setting)

*8 When calculating D7, use this value for "System data (safety)".

*9 Since the maximum number of mountable CC-Link IE Field Network module is 8, the maximum number of connections that can be set in the safety communication setting is 960 (120 connections × 8 modules).

D5 is determined as follows.

Item	Size*10
Global device (standard) (other than the refresh data register (RD))	Follows the tracking transfer settings. (☞ Page 231 Tracking transfer setting)
Global device (safety)	Follows the safety device area in the device/label memory are settings.
Local device (standard)	Total size of local devices of the device/label memory area detailed setting × the number of programs on the program setting*12
Local device (safety)	Safety device/label area capacity minus safety device area capacity and safety label area capacity specified in the device/label memory area settings
Global label (standard)*11	Follows the global label settings.
Global label (safety)	Follows the safety global label settings.
Local label (standard)*11	Follows the local label settings of each standard program.
Local label (safety)	Safety label area capacity specified in the device/label memory area setting minus the total size of labels defined in the safety global label setting file
Standard/safety shared label	Follows the standard/safety shared label setting file.

*10 The size of the devices/labels to be transferred (tracked) can be checked by clicking the [Size Calculation] button in "Detailed Setting" of "Device/Label Detailed Setting". (☞ Page 234 Detailed setting)

*11 Even though global labels/local labels are deleted by the online change, the size of the tracking data of the global labels/local labels are the same as the one before the online change. The size is updated when the program and global label setting file to which all data is rebuilt (reassigned) are written to the programmable controller and the operating status of the CPU module is switched from STOP to RUN.

*12 This indicates the number of programs where local devices are set to use in the "Setting of Device/File Use Or Not" window of the program setting of the CPU parameter.

■Waiting time for completion of tracking data reception (Tca)

The following describes the waiting time for completion of tracking data reception (Tca) in the CPU module of the control system.

Tca = 2 [ms]

■Other extended times (α)

The following describes other extended times (α) in the CPU module of the control system.

α = 0.6 + α1 [ms]

When an error occurs during a tracking transfer, the scan time may be extended (α1).α 1 is determined as follows.

Item	Time for α1
Disconnection, pulling out, inserting of the other tracking cable	6 [ms]
Momentary power failure or power-off in the standby system	16 to 56 [ms]
Hardware failure in the standby system	50 to Total size of tracking data [word] × 26.7 × 10^{-6} + 10^{*1} [ms]

*1 Maximum extended time for the redundant function module

Point

When using the constant scan, take one of the following measures.

- The setting time of the constant scan is set by adding the above extended time (α1) when an error occurs. (☞ Page 95 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 190 Error Clear)

■Waiting time for tracking data reception (Taw)

The following describes the waiting time for tracking data reception (Taw) in the CPU module of the standby system.

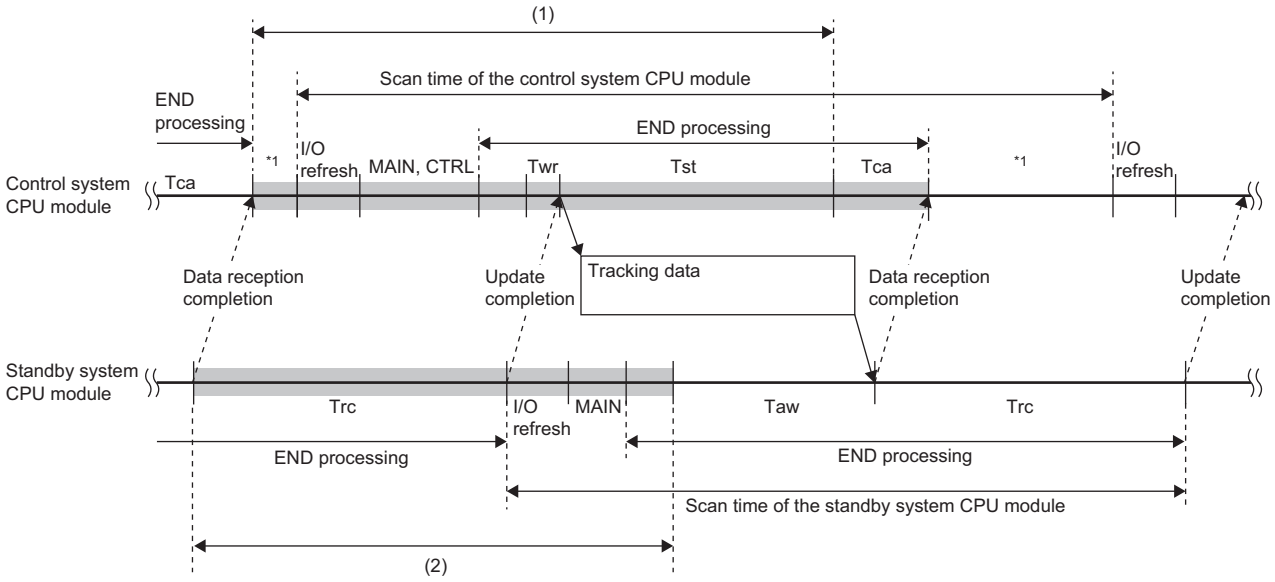
The waiting time for tracking data reception is determined by two types of period; (1) the period from when the control system CPU module receives a receive completion notification until next time it receives the notification and (2) the period when the standby system CPU module sends a receive completion notification until it starts receiving tracking data.

When (1) ≤ (2): 0 [ms]

When (1) > (2): (1) - (2) [ms]

Ex.

When CTRL is a control system execution program and MAIN is a program executed in both systems (in configuration with the main base unit only)



*1 If the constant scan is used, waiting time for the constant scan is generated.

- For the I/O refresh processing time for (1) and (2) and END processing time for others, refer to the following.

☞ Page 541 END processing time

- If the values of each item for (1) and (2) vary, the waiting time for tracking data reception also varies.

■Tracking data reflection time (Trc)

The following describes the tracking data reflection time (Trc) in the CPU module of the standby system. The calculation method differs depending on whether an extended SRAM cassette is inserted or not.

Extended SRAM cassette inserting status	Tracking data reflection time
Not inserted	$1 + (20.0 \times 10^{-6}) \times D1 + (37.2 \times 10^{-6}) \times D2 + (682.3 \times 10^{-6}) \times D6 + (300.0 \times 10^{-6}) \times F1$ [ms]
Inserted	$1 + (20.0 \times 10^{-6}) \times D1 + (106.0 \times 10^{-6}) \times D2 + (682.3 \times 10^{-6}) \times D6 + (300.0 \times 10^{-6}) \times F1$ [ms]

- D1: Size [word] of tracking data in the system data, signal flow memory (standard/safety), and refresh data register (RD)
- D2: Size [word] of tracking data of global devices (standard/safety), local devices (standard/safety), global labels (standard/safety), local labels (standard/safety), and standard/safety shared labels
- D6: Total size [word] of tracking data in the system data (safety), signal flow memory (safety), local devices (safety), global labels (safety), local labels (safety), and global devices (safety)
- F1: Number of transferred blocks

For the calculation formula for the size of tracking data, refer to the following.

☞ Page 550 Tracking data send time (Tst)

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		One scan
System switching request by the SP.CONTSW instruction		0ms
System switching request from the engineering tool		0ms

Time required for system switching

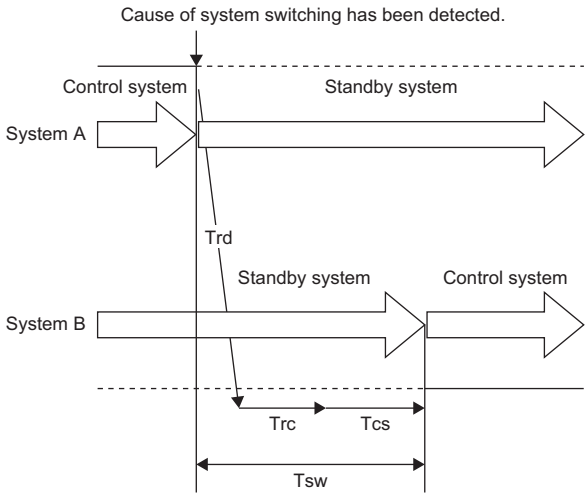
The following describes the time required for the CPU module in the new control system to start operating as the new control system after the detection of a system switching cause in the control system.

The time required for system switching determined by the following calculation formula is a rough standard for a system start-up. Check the time with the actual systems.

Tsw = Trd + Trc + Tcs [ms]

- Tsw: Time required for system switching*1
- Trd: Delay until the CPU module in the standby system receives the system switching request sent from the control system
- Trc: Time required for reflection of tracking data in the CPU module of the standby system
- Tcs: Setup time required for the CPU module in the standby system to start operating as the new control system

*1 This is the maximum system switching time.



System switching cause	Processing time per each system switching cause			
	Trd	Trc	Tcs	
• Power supply module failure*3, power-off • Reset	■When the standby system scan time is longer than 19.26ms*2 0.517 + Standby system scan time - 19.26 [ms] ■When the standby system scan time is 19.26ms or shorter*2 0.517ms	0ms	27ms*6	The following time is added according to the parameter setting and system configuration. • When the signal flow memory is tracked: 0ms • When the signal flow memory is not tracked: 2ms
• CPU module failure • Base unit failure*4 • CPU module stop error (major) • Redundant function module failure*4	■When the standby system scan time is longer than 10ms*2 0.517 + Standby system scan time - 10 [ms] ■When the standby system scan time is 10ms or shorter*2 0.517ms			
CPU module stop error (moderate)	Increase in the scan time of the CPU module in the control system (Ts) by tracking transfer*5 - Waiting time for completion of the previous tracking data reflection (Twr)	☞ Page 555 Tracking data reflection time (Trc)*5	40ms*7	
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 555 Tracking data reflection time (Trc)		
System switching request by the SP.CONTSW instruction				
System switching request from the engineering tool				

*2 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".

*3 When the CC-Link IE Field Network module has not been mounted or depending on the state of the power supply module failure, the systems may not be switched. 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:
· Via the CC-Link IE Field Network module: Trd = 400ms (maximum) + Standby system scan time

*4 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
· Via the CC-Link IE Field Network module: Trd = Standby system scan time

*5 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.

*6 The value in this item varies depending on the scan time. The table shows the value for the scan time of 200ms. The value is Tcs=7 when the scan time is 25ms.

*7 The value in this item varies depending on the scan time. The table shows the value for the scan time of 200ms. The value is Tcs=36 when the scan time is 25ms.

■Tracking data reflection time (Trc)

Refer to the following.

☞ Page 553 Tracking data reflection time (Trc)

Extended time until initial output after system switching (Tjo)

This section describes the extended time until initial output after system switching (Tjo).

■Time required until values are output to the network module

The following describes the time required for the CPU module that has started operating as the new control system to output values to the network module for the first time.

- Standard control

Tjo = (Sc + Twcyc) - Twc - Ts - Toref [ms]

- Tjo: Extended time from completion of a system switching to the first output
- Sc: Scan time of the CPU module in the control system
- Twcyc: Waiting time for cyclic data receive after system switching (☞ Page 556 Waiting time for cyclic data receive after system switching (Twcyc))
- Twc: Constant scan waiting time
- Ts: Increase in the scan time of the CPU module in the control system by tracking transfer
- Toref: Output refresh of the intelligent function module (CPU module → Intelligent function module)

- Safety control

Item	Calculation formula
Normal value	Tso = (Msw or Lsw) + Tsc + 1.5× TM + 4.5 × RM + Tsio [ms]
Maximum value	Tso = (Msw or Lsw) + Tsc + 2 × TM + 9 × RM + Tsio [ms]

- Tso: Extended time from completion of a system switching to the first output
- Msw: Master station switching time
- Lsw: Line switching time
- Tsc: Safety cycle time
- Tsio: Safety cycle processing time
- TM: Transmission interval monitoring time
- RM: Safety refresh monitoring time

Waiting time for cyclic data receive after system switching (Twcyc)

The following shows the calculation formula for waiting time for cyclic data receive after system switching.

When 'Setting to wait cyclic data receive after system switching' of the CPU parameter is enabled in the redundant line structure of the CC-Link IE Field Network module, the waiting time for cyclic data after system switching (Twcyc) is added to the delay time until initial output after system switching (Tjo). When the setting is disabled (default), 0 [ms] is added.

Twcyc = Lsw + 2LS [ms]

- Lsw: Line switching time of the CC-Link IE Field Network module
- LS: Link scan time of the CC-Link IE Field Network module

For the calculation formula for Lsw and LS, refer to the following.

📖MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

Memory copy time

The following table shows the time taken for the memory copy under the following conditions:

[Condition]

- CPU module: R120PSFCPU
- Sequence scan time: 5ms
- The SD memory card of the standby system CPU module has been formatted.
- Data in the SD memory card is mismatched between the control system and standby system.

Memory copy target memory and size of data to be transferred	Memory copy time
Size of data transferred from the data memory and program memory: 154K bytes Size of data transferred from the SD memory card: 128K bytes	50s
Size of data transferred from the data memory and program memory: 292K bytes Size of data transferred from the SD memory card: 512K bytes	70s

Memory copy time depends on the data to be copied and the use of the SD memory card. Use the time above as a rough standard for memory copy.

Appendix 10 Parameter List

This section lists parameters.

Point

For the parameters that are not described in this chapter, refer to the manuals for each module used.

System parameters

The following is the list of system parameters.

Item			Parameter No.
I/O Assignment	Base/Power/Extension Cable Setting	Slot	0201H
		Base, Power Supply Module, Extension Cable	0203H
	I/O Assignment Setting	Points, Start XY, Module Status Setting	0200H
		Control PLC Settings	0202H
		Module Name	0203H
	Points Occupied by Empty Slot Batch Setting		0100H
	Redundant Module Group Setting		0400H

CPU parameters

The following is the list of CPU parameters.

Item			Parameter No.
Name Setting	Title Setting		3100H
	Comment Setting		3101H
Operation Related Setting	Timer Limit Setting		3200H
	RUN-PAUSE Contact Setting		3201H
	Remote Reset Setting		3202H
	Output Mode Setting of STOP to RUN		3203H
	Module Synchronous Setting		3207H
	Clock Related Setting		3209H
Interrupt Settings	Fixed Scan Interval Setting		3A00H
	Fixed Scan Execution Mode Setting		3A00H
	Interrupt Enable Setting in Executing Instruction		3A00H
	Block No. Save/Recovery Setting		3A00H
	Interrupt Priority Setting from Module		3A01H
Service Processing Setting	Device/Label Access Service Processing Setting		3B00H
File Setting	File Register Setting		3300H
	Initial Value Setting		3301H
	Label Initial Value Reflection Setting		3302H
	File Setting for Device Data Storage		3303H
Memory/Device Setting	Cassette Setting	Extended SRAM Cassette	3404H
	Device/Label Memory Area Setting	Device/Label Memory Area Capacity Setting	3400H
		Device Points	3401H
		Local Device	3405H
		Latch Range Setting	3407H
		Latch Type Setting of Latch Type Label	3408H
	Index Register Setting		3402H
	Refresh Memory Setting		3403H
	Device Latch Interval Setting		3406H
	Pointer Setting		340BH

A

Item			Parameter No.
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 Online Module Change Function Setting	Save Destination	3507H*1
		Set Save Volume of Per File	3507H*1
		Direct Change Setting	3505H
Program Setting	Program Setting	Program Name	3700H
		Execution Type	3700H
		Type (Fixed Scan)	3700H
		Type (Event)	3701H
		Detail Setting Information	—
		Refresh Group Setting	3700H
		Device/File Use or not	3700H
		Both Systems Program Executions Setting	3700H
	FB/FUN File Setting		3702H
Routing Setting	Routing Setting		3800H
Redundant System Settings	Tracking Setting	Signal Flow Memory Tracking Setting	5001H
		Tracking Device/Label Setting	5001H
		Safety Tracking Setting	5A30H

*1 When the device/label operation save setting is set to "Do not save", the value is 3504H.

Module parameters

The following are the lists of module parameters.

Ethernet function

For the module parameters relating to the Ethernet function, refer to the following.

MELSEC iQ-R Ethernet User's Manual (Application)

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
	Safety I/O Hold Time		5A21H

Memory card parameters

The following is the list of memory card parameters.

Item		Parameter No.
Setting of File/Data Use or Not in Memory Card	Data for Label Communication	2010H
	Module Extended Parameter	

A

Appendix 11 Target List and Operation Details of the Device/Label Access Service Processing Setting

Target list

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

Applicable function	Description
Functions to perform read/write to files accessed by programs	When read/write is performed to the relevant files during execution of a program, file inconsistency may occur. Therefore, such operation is performed during the END processing.
Functions to perform write to devices/labels	When write is performed to devices/labels during execution of a program, the result of operation processing may become inconsistent. Therefore, such operation is performed during the END processing.

Communication functions via the SLMP/MC protocol

Among communication functions using the SLMP/MC protocol, those to which the device/label access service processing setting can be applied are as follows.

Function			Command
Device memory	Multiple blocks batch read		0406 (00□0)
	Multiple blocks batch write		1406 (00□0)
	Batch read	In units of bits	0401 (00□1)
		In units of words	0401 (00□0)
	Batch write	In units of bits	1401 (00□1)
		In units of words	1401 (00□0)
	Random read		0403 (00□0)
	Test (Random write)	In units of bits	1402 (00□1)
		In units of words	1402 (00□0)
	Monitor		0802 (00□0)
Files	New file creation*1		1820 (0000)
	File copy*1	File password function incompatible	1824 (0000)
		File password function compatible	1824 (0004)
	File open*1	File password function incompatible	1827 (0000)
		File password function compatible	1827 (0004)
	File read*1		1828 (0000)
	File write*1		1829 (0000)
	File close*1		182A (0000)
Programmable controller CPU monitoring	Registration*2		0630 (0000)
	Clear*2		0631 (0000)

*1 Only the file register file can support it. (Other file access is always executed asynchronously from the program.)

*2 When registration/clear command is executed, it does not affect the scan time. However, since the serial communication module will periodically access into the device memory of the CPU module when Function is enabled, it becomes the target for Device/Label access service processing setting.

Communication function using an engineering tool

Of the communication functions using an engineering tool, the functions targeted for the setting of the Device/Label access service processing are indicated.

Function	
Writing data to the programmable controller	File register file
	Device data (Local device also included)
	Global label and local label data
Reading data from the programmable controller	File register file
	Device data (Local device also included)
	Global label and local label data
	Device data storage file
Monitor function	Circuit monitor
	Device/buffer memory batch monitor
	Label batch monitor (local label included)
	Buffer memory monitor
Ethernet function	File transfer (FTP server)*1
	File transfer (FTP client)*1

*1 Only the file register file can support it. (Other file access is always executed asynchronously from the program.)

Operation details

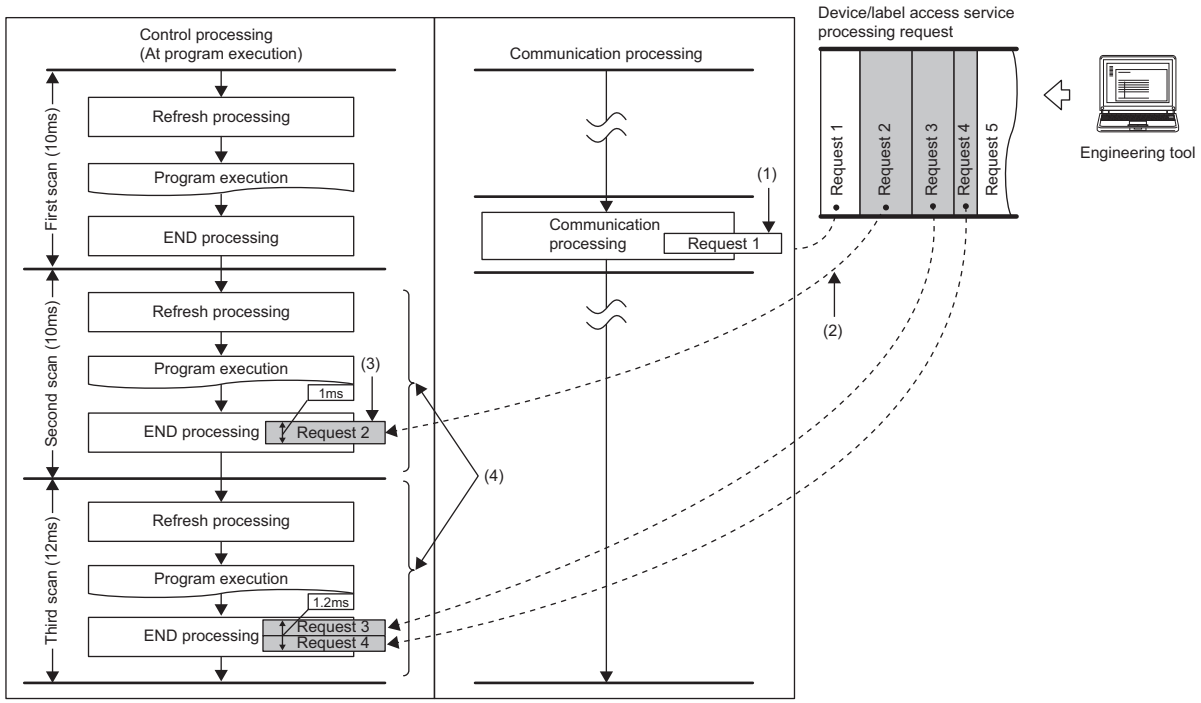
Operations enabled by setting details of the device/label access service processing setting are as follows.

Execute the process as scan time proceeds

This setting is useful to execute the device/label access service processing in a way commensurate with the system size. It allows the system to be designed without considering the device/label access service processing time because it is determined as a function of the scan time.

Ex.

When "Scan time ratio = 10%" is set



(1) Synchronization with program is unnecessary.

(2) Multiple requests are processed until the specified device/label access service processing ratio (10%) is exceeded. When the specified device/label access service processing ratio is exceeded, the remaining requests are processed in the END processing of the next scan. Also, for scan which has the device/label access service processing time shorter than 0.1ms, it is assumed as "The device/label access service processing time per scan = 0.1ms".

(3) Operations such as access to device, which are synchronized with the program, are processed in the END processing.

(4) The maximum device/label access service processing time available varies because the scan time (program execution time) varies.

Point

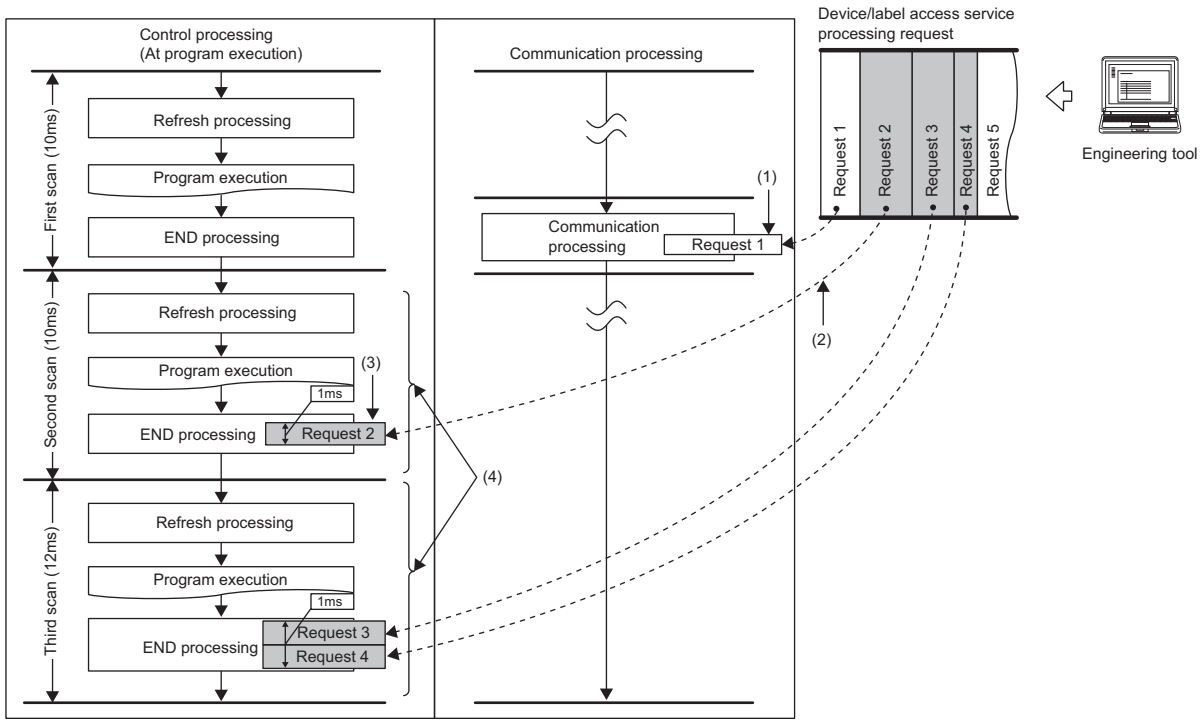
For operations such as access to devices, which are synchronized with the program, adjust the time by this setting because they are processed during the END processing. If no request data for the device/label access service processing exists, the scan time is shortened by the specified ratio as the CPU module proceeds to the next scan without waiting for requests.

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.

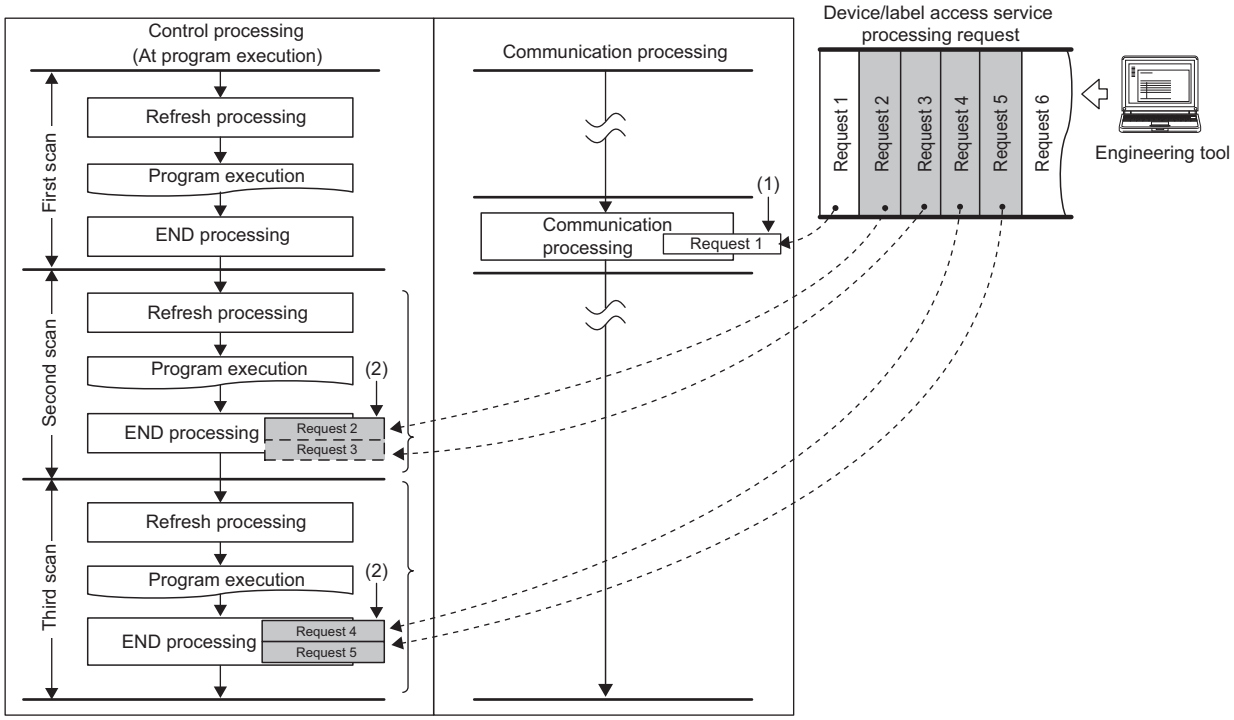
A

Set Processing Counts

This setting is useful to stably execute the device/label access service processing in a system where requests come from multiple peripherals. It provides stable communication in a system where multiple peripherals exist because the CPU module can execute the device/label access service processing based on the number of request sources.

Ex.

When "Processing counts = 2" is set



(1) Synchronization with program is unnecessary.

(2) Two requests are processed in one END processing independently of the requested processing time.

Point

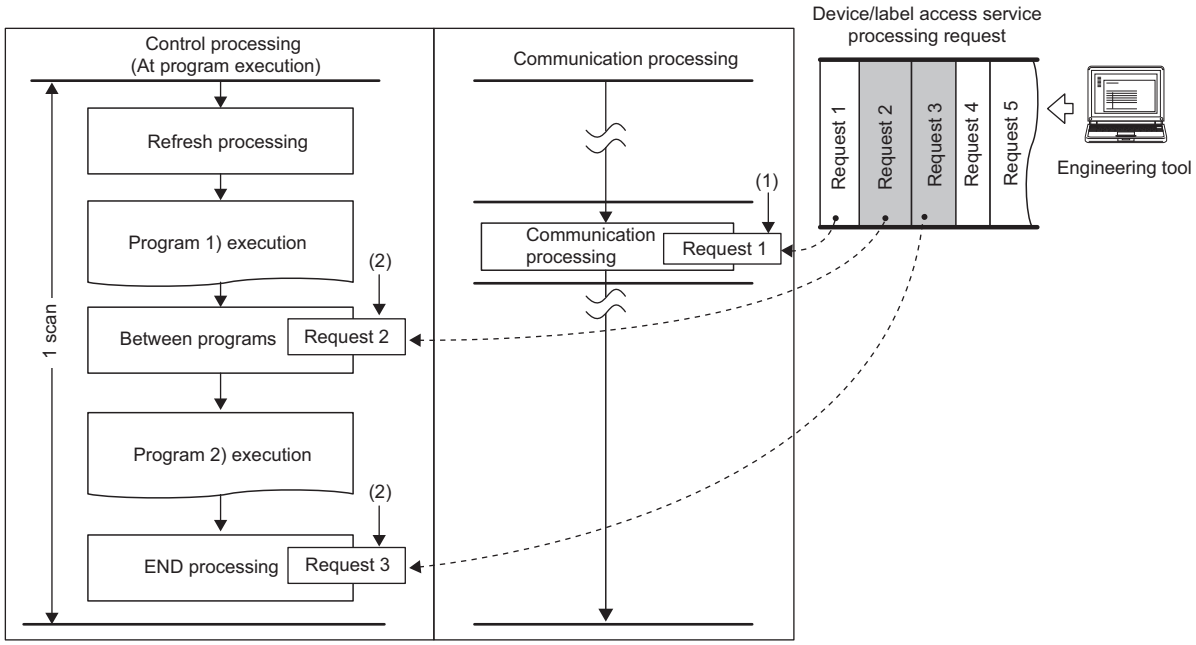
For operations such as access to devices, which are synchronized with the program, adjust the number of executions by this setting because they are processed during the END processing. If no request data for the device/label access service processing exists, the CPU module proceeds to the next scan without waiting for requests.

Execute END Processing between Programs

This setting is useful to give priority to the device/label access service processing in a system with a large number of programs. Because operations such as access to devices are performed between program executions and during the END processing, requests as many as the number of programs can be processed during a single scan. As a result, it can reduce the response for the device/label access service processing.

Ex.

When "Execute END Processing between Programs" is enabled



(1) Synchronization with program is unnecessary.

(2) Requests for operations such as access to devices are processed between program executions and during the END processing.

A

Appendix 12 Added and Enhanced Functions

This section describes added and enhanced functions of the CPU module and the engineering tool, as well as the firmware versions of the CPU module and software versions of the engineering tool corresponding to the functions.

Added or enhanced function	F: Firmware version of the CPU module S: Software version of the engineering tool	Reference
Access level "Assistant Developers" of the user authentication function	F: 08 S: 1.080J	GX Works3 Operating Manual
Setting to Wait Receiving Cyclic Data after Switching System*1	F: 04 S: 1.060N	Page 260 Setting to wait cyclic data receive after system switching
User authentication function (enhanced vulnerability-measures taken)	F: 12 S: 1.105K	Page 209 User Authentication Function of CPU Modules
Supported Safety protocol version: 2	F: 14 S: 1.110Q	MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

*1 The firmware version of the usable RJ71GF11-T2 is "35" or later.

INDEX

Symbols

"$MELPRJ$" folder 152

A

Accuracy of constant scan 96
Accuracy of timers 297
Annunciator (F) 66,291
ASCII 338

B

BACKUP LED 33
Backup mode 24
Battery 29,46
BATTERY LED 29
Bit 338
Block number 316
Block switching method 316
BOOL 338
Buffer memory 24

C

CARD ACCESS LED 29
CARD READY LED 29
CC-Link IE Controller Network-equipped module 25
CC-Link IE Field Network-equipped master/local module 25
Character string constant 370
Class 340
Clearing file registers 317
Constant 368
Constant scan 66,95
Control system 24
Control system execution program 24
COUNTER 338,339
Counter 338,339
Counter (C) 66,302
CPU buffer memory 64,152
CPU buffer memory access device 311
CPU module 64,473
CPU parameters 557
CTRL LED 33

D

Data memory 64,152
Data register (D) 66,305
Data type 338,339,340
Decimal 368
Decimal constant (K) 368
Device assignment 336
Device initial value 360
Device supporting iQSS 25
Device/label access service processing 97
Device/label memory 64,70,147
DINT 338
Double word [signed] 338
Double word [unsigned]/bit string [32 bits] 338
Double-precision real number 338,369
DWORD 338

E

Edge relay (V) 66,293
END processing 93
ERR LED 33
Error clear 190
Error code 386
ERROR LED 29,35,56,375
Ethernet interface module with built-in CC-Link IE 25
Ethernet port 29,64
Ethernet-equipped module 25
Event execution type program 113
Event history 194
Event history file 195
Extended SRAM cassette 47,68
External dimensions 64,68,70,473

F

File 158
File password 206
File register (R/ZR) 67,316
File register setting 317
File registers common to all programs 317
Fixed scan execution mode 111
Fixed scan execution type program 108
Fixed scan interval 109
Function (FUN) 337
Function block (FB) 337
Function device 306
Function input (FX) 67,306
FUNCTION LED 29
Function output (FY) 67,306
Function register (FD) 67,306

G

Generic data type (ANY type) 340
Global device 325
Global label 336,337,340
Global pointer 320

H

Hexadecimal 368
Hexadecimal constant (H) 368
High-speed timer 295
High-speed timer (T/ST) 295

I

I/O refresh 89
Identification check for safety data 265,267
Index modification 312
Index register (Z) 67
Index register setting 313
Indirect specification 329

Initial execution type program 107
Initial processing (when powered on) 89
Initial processing (when switched to RUN) 89
Initial scan time 94
Initial scan time monitoring time 94
Initialization of labels after rebuilt all (reassignment) 364
Input (X) 66,290
INT 338
Intelligent function module device 67
Internal current consumption (5VDC) 64,70
Internal relay (M) 66,290
Interrupt factor 323
Interrupt pointer (I) 67,323
Interrupt pointer number 323
Interrupt priority 133
Interrupt using the internal timer (I28 to I31) 66
IP filter 206

J

Jump instructions 320

L

L ERR LED 34
Label initial value 360
Label Initial value reflection setting 366
Latch (1) 353
Latch (2) 353
Latch clear 359
Latch relay (L) 66,290
LCOUNTER 338,339
Librarization of programs 118
Link direct device 308
LINK LED 34
Link refresh 309
Link register (W) 66,305
Link relay (B) 66,291
Link special register (SW) 66,305
Link special relay (SB) 66,293
Local label 337,340
Local pointer 321
Long counter 338,339
Long counter (LC) 66,302
Long index register (LZ) 67
Long retentive timer 338,339
Long retentive timer (LST) 66,295
Long timer 338,339
Long timer (LT) 66,294
Low-speed timer 295
Low-speed timer (T/ST) 295
LREAL 338
LRETENTIVETIMER 338,339
LTIMER 338,339

M

Master control instructions 319
Maximum counting speed for counters 304
Memory capacity 64,70
Memory card parameters 559
Memory configuration 145
Memory copy 242
MEMORY COPY LED 33
Module access device 310
Module communication test 261
Module label 336
Module label assignment area 318
Momentary power failure 144
Multiple interrupt 133
Multiple interrupt function 133

N

Nesting (N) 319
Nesting structure 319
New control system 24
New standby system 24
Number of array elements 344
Number of FB files 66
NZ2MC-2MBSE 68
NZ2MC-8MBSE 68

O

Online change 173
Online change (ladder block) 173
Operation mode of the system 75
Optical connector (IN) 34
Optical connector (OUT) 34
Output (Y) 66,290
Output mode at operating status change (STOP to RUN) 142
Overflow 369
Own system 24

P

Parameter No. 557
PAUSE state 139
PID control 203
POINTER 338
Pointer 338
Pointer (P) 67,320
Pointer setting 322
Priority for interrupt factors 324
Process control function 204
Process control instructions 205
Processing time 536
Production information marking 30,34,35
Program arrangement change 118
Program block 337
Program cache memory 145
Program capacity 64,70
Program executed in both systems 24
Program execution type 106
Program list monitor 94
Program memory 64,70,145
PROGRAM RUN LED 29,35,57
Programming language 66

Q

Q6BAT 46

R

READY LED 29,35,375
REAL 338
Real constant (E) 369
Real number 369

Redundant function module 24
Redundant system 24
Refresh data register (RD) 67,318
Refresh memory 64,152
Refresh memory setting 318
Remote head module 25
Remote operation 200
Remote password 206
Reserved area for online change 175
Reset 56
Resetting counters 303
Retentive timer 338,339
Retentive timer (ST) 66,295
RETENTIVETIMER 338,339
RUN LED 33
RUN state 139
RUN/STOP/RESET switch 29,56,57
RUN-PAUSE contacts 202

S

SAFETY COM ERR LED 35
SAFETY COM RUN LED 35
Safety counter 66,330,333
Safety cycle time 91
Safety data register 66,330,333
Safety device 330
Safety global device 335
Safety global label 350
Safety input 66,330,331
Safety internal relay 66,330,331
Safety link register 66,330,333
Safety link relay 66,330,331
Safety local device 335
Safety local label 350
SAFETY MODE 76
SAFETY MODE (wait-for-restart) 76
Safety operation mode 76
Safety output 66,330,331
Safety retentive timer 66,330,332
Safety special register 67,330
Safety special relay 67,330
Safety system device 67,330,334
Safety timer 66,330,332
Safety user device 66,330,331
Saving/restoring of the file register (R) block number 130
Saving/restoring of the index register (Z, LZ) . . . 131
SBY LED 33
Scan execution type program 108
Scan monitoring function 182
Scan time 94
Scan time monitoring time 182
Scan time monitoring time setting 182
SD CARD OFF button 29
SD memory card 48,154
SD/RD LED 29
Security key authentication 206
SEPARATE LED 33
Separate mode 24
Serial number method 316
Setting RUN-PAUSE contacts 202
Setting time zone 170
Shift JIS 338
Signal flow memory 64,153,232,275
SIL2 function module 24
SIL2 mode 24
SIL2 Process CPU 24
Single-precision real number 338,369
Special register (SD) 67,307
Special relay (SM) 67,307
SPEED LED 29
Standard CPU 25
Standard programmable controller 25
Standard/safety shared label 24,350
Standby system 24
Standby type program 118
Station sub ID number 25
STOP state 139
STRING 338
String 338
String [Unicode] 338
Structure 339,345
Structure array 346
Subroutine call instruction 122
SYS A LED 33
SYS B LED 33
System A 24
System B 24
System clock 171
System consistency check 247
System device 306
System label 336
System parameters 557
System switching 165,212,478

T

TEST LED 35
TEST MODE 76
TIME 338
Time 338
Time setting 167
Time zone 170
TIMER 338,339
Timer 338,339
Timer (T) 66,294
Timer current value 296
Timer limit setting 296
Timer time limit value 296
Tracking cable 24
Tracking transfer 165,225,478
Troubleshooting by symptom 377

U

Underflow 369
Unicode 338
USB port 29,64
User authentication function 206
User device 290
USER LED 29

W

Watchdog timer 182
Weight 65,68,70
WORD 338
Word [signed] 338
Word [unsigned]/bit string [16 bits] 338
WSTRING 338

I

REVISIONS

*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
October 2025	SH(NA)-082503ENG-A	First edition

Japanese manual number: SH-082502-A

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2025 MITSUBISHI ELECTRIC CORPORATION

CONDITIONS OF USE FOR THE PRODUCT

(1) The Product has been certified by a third-party certification body as being in compliance with the IEC 61508 safety standards and meets the requirements of the IEC 61511 safety standards. However, this does not guarantee that the Product is free from malfunctions or failures. The user of this Product shall comply with any and all applicable safety standard, regulation or law and take appropriate safety measures for the system in which the Product is installed or used and shall take the second or third safety measures other than the Product. Mitsubishi Electric is not liable for damages that could have been prevented by compliance with any applicable safety standard, regulation or law.

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

(a) power plants,

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

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

(d) amusement equipments,

(e) incineration and fuel devices,

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

(g) mining and drilling,

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

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

WARRANTY

Please confirm the following product warranty details before using this product.

1. Limited Warranty and Product Support.

a. Mitsubishi Electric Corporation ("MELCO") warrants that for a period of eighteen (18) months after date of delivery from the point of manufacture or one year from date of Customer's purchase, whichever is less, Mitsubishi programmable logic controllers (the "Products") will be free from defects in material and workmanship.

b. At MELCO's option, for those Products MELCO determines are not as warranted, MELCO shall either repair or replace them or issue a credit or return the purchase price paid for them.

c. For this warranty to apply:

(1) Customer shall give MELCO (i) notice of a warranty claim to MELCO and the authorized dealer or distributor from whom the Products were purchased, (ii) the notice shall describe in reasonable details the warranty problem, (iii) the notice shall be provided promptly and in no event later than thirty (30) days after the Customer knows or has reason to believe that Products are not as warranted, and (iv) in any event, the notice must given within the warranty period;

(2) Customer shall cooperate with MELCO and MELCO's representatives in MELCO's investigation of the warranty claim, including preserving evidence of the claim and its causes, meaningfully responding to MELCO's questions and investigation of the problem, grant MELCO access to witnesses, personnel, documents, physical evidence and records concerning the warranty problem, and allow MELCO to examine and test the Products in question offsite or at the premises where they are installed or used; and

(3) If MELCO requests, Customer shall remove Products it claims are defective and ship them to MELCO or MELCO's authorized representative for examination and, if found defective, for repair or replacement. The costs of removal, shipment to and from MELCO's designated examination point, and reinstallation of repaired or replaced Products shall be at Customer's expense.

(4) If Customer requests and MELCO agrees to effect repairs onsite at any domestic or overseas location, the Customer will pay for the costs of sending repair personnel and shipping parts. MELCO is not responsible for any re-commissioning, maintenance, or testing on-site that involves repairs or replacing of the Products.

d. Repairs of Products located outside of Japan are accepted by MELCO's local authorized service facility centers ("FA Centers"). Terms and conditions on which each FA Center offers repair services for Products that are out of warranty or not covered by MELCO's limited warranty may vary.

e. Subject to availability of spare parts, MELCO will offer Product repair services for (7) years after each Product model or line is discontinued, at MELCO's or its FA Centers' rates and charges and standard terms in effect at the time of repair. MELCO usually produces and retains sufficient spare parts for repairs of its Products for a period of seven (7) years after production is discontinued.

f. MELCO generally announces discontinuation of Products through MELCO's Technical Bulletins. Products discontinued and repair parts for them may not be available after their production is discontinued.

2. Limits of Warranties.

a. MELCO does not warrant or guarantee the design, specify, manufacture, construction or installation of the materials, construction criteria, functionality, use, properties or other characteristics of the equipment, systems, or production lines into which the Products may be incorporated, including any safety, fail-safe and shut down systems using the Products.

b. MELCO is not responsible for determining the suitability of the Products for their intended purpose and use, including determining if the Products provide appropriate safety margins and redundancies for the applications, equipment or systems into which they are incorporated.

c. Customer acknowledges that qualified and experienced personnel are required to determine the suitability, application, design, construction and proper installation and integration of the Products. MELCO does not supply such personnel.

d. MELCO is not responsible for designing and conducting tests to determine that the Product functions appropriately and meets application standards and requirements as installed or incorporated into the end-user's equipment, production lines or systems.

e. MELCO does not warrant any Product:

(1) repaired or altered by persons other than MELCO or its authorized engineers or FA Centers;

(2) subjected to negligence, carelessness, accident, misuse, or damage;

(3) improperly stored, handled, installed or maintained;

(4) integrated or used in connection with improperly designed, incompatible or defective hardware or software;

(5) that fails because consumable parts such as batteries, backlights, or fuses were not tested, serviced or replaced;

(6) operated or used with equipment, production lines or systems that do not meet applicable and commensurate legal, safety and industry-accepted standards;

(7) operated or used in abnormal applications;

(8) installed, operated or used in contravention of instructions, precautions or warnings contained in MELCO's user, instruction and/or safety manuals, technical bulletins and guidelines for the Products;

(9) used with obsolete technologies or technologies not fully tested and widely accepted and in use at the time of the Product's manufacture;

(10) subjected to excessive heat or moisture, abnormal voltages, shock, excessive vibration, physical damage or other improper environment; or

(11) damaged or malfunctioning due to Acts of God, fires, acts of vandals, criminals or terrorists, communication or power failures, or any other cause or failure that results from circumstances beyond MELCO's control.

f. All Product information and specifications contained on MELCO's website and in catalogs, manuals, or technical information materials provided by MELCO are subject to change without prior notice.

g. The Product information and statements contained on MELCO's website and in catalogs, manuals, technical bulletins or other materials provided by MELCO are provided as a guide for Customer's use. They do not constitute warranties and are not incorporated in the contract of sale for the Products.
h. These terms and conditions constitute the entire agreement between Customer and MELCO with respect to warranties, remedies and damages and supersede any other understandings, whether written or oral, between the parties. Customer expressly acknowledges that any representations or statements made by MELCO or others concerning the Products outside these terms are not part of the basis of the bargain between the parties and are not factored into the pricing of the Products.
i. THE WARRANTIES AND REMEDIES SET FORTH IN THESE TERMS ARE THE EXCLUSIVE AND ONLY WARRANTIES AND REMEDIES THAT APPLY TO THE PRODUCTS.
j. MELCO DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

3. Limits on Damages.

a. MELCO'S MAXIMUM CUMULATIVE LIABILITY BASED ON ANY CLAIMS FOR BREACH OF WARRANTY OR CONTRACT, NEGLIGENCE, STRICT TORT LIABILITY OR OTHER THEORIES OF RECOVERY REGARDING THE SALE, REPAIR, REPLACEMENT, DELIVERY, PERFORMANCE, CONDITION, SUITABILITY, COMPLIANCE, OR OTHER ASPECTS OF THE PRODUCTS OR THEIR SALE, INSTALLATION OR USE SHALL BE LIMITED TO THE PRICE PAID FOR PRODUCTS NOT AS WARRANTED.
b. The Product has been certified by a third-party certification body as being in compliance with the IEC 61508 safety standards and meets the requirements of the IEC 61511 safety standards. However, this does not guarantee that the Product is free from malfunctions or failures. 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.

INFORMATION AND SERVICES

For further information and services, please contact your local Mitsubishi Electric sales office or representative.
Visit our website to find our locations worldwide.

MITSUBISHI ELECTRIC Factory Automation Global Website
Locations Worldwide
www.MitsubishiElectric.com/fa/about-us/overseas/

TRADEMARKS

Microsoft, 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.

TRANSLATION OF THE ORIGINAL INSTRUCTIONS

SH(NA)-082503ENG-A(2510)MEE
MODEL: RNPSFCPU-U-E

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA 461-8670, JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.