



Mitsubishi Programmable Controller

**MELSEC iQ-R**  
series

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

---

-R04CPU  
-R08CPU  
-R08PCPU  
-R16CPU  
-R16PCPU  
-R32CPU  
-R32PCPU  
-R120CPU  
-R120PCPU



# SAFETY PRECAUTIONS

---

(Read these precautions before using this product.)

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

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



## WARNING

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

---



## CAUTION

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

---

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

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

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

## [Design Precautions]

---



## WARNING

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

## [Design Precautions]

---

### **WARNING**

- 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 write any data to the "system area" and "write-protect area" of the buffer memory in the module. Also, do not use any "use prohibited" signals as an output signal from the CPU module to each module. Doing so may cause malfunction of the programmable controller system. For the "system area", "write-protect area", and the "use prohibited" signals, refer to the user's manual for the module used.
  - If a communication cable is disconnected, the network may be unstable, resulting in a communication failure of multiple stations. Configure an interlock circuit in the program to ensure that the entire system will always operate safely even if communications fail. Failure to do so may result in an accident due to an incorrect output or malfunction.
  - To maintain the safety of the programmable controller system against unauthorized access from external devices via the network, take appropriate measures. To maintain the safety against unauthorized access via the Internet, take measures such as installing a firewall.
-

## [Design Precautions]

---

### CAUTION

---

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

## [Installation Precautions]

---

### **WARNING**

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

## [Installation Precautions]

---

### **CAUTION**

- Use the programmable controller in an environment that meets the general specifications in the Safety Guidelines included with the base unit. Failure to do so may result in electric shock, fire, malfunction, or damage to or deterioration of the product.
  - To mount a module, place the concave part(s) located at the bottom onto the guide(s) of the base unit, and push in the module until the hook(s) located at the top snaps into place. Incorrect interconnection may cause malfunction, failure, or drop of the module.
  - When using the programmable controller in an environment of frequent vibrations, fix the module with a screw.
  - Tighten the screws within the specified torque range. Undertightening can cause drop of the screw, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
  - When using an extension cable, connect it to the extension cable connector of the base unit securely. Check the connection for looseness. Poor contact may cause malfunction.
  - When using an SD memory card, fully insert it into the SD memory card slot. Check that it is inserted completely. Poor contact may cause malfunction.
  - Securely insert an extended SRAM cassette into the cassette connector of the CPU module. After insertion, close the cassette cover and check that the cassette is inserted completely. Poor contact may cause malfunction.
  - Do not directly touch any conductive parts and electronic components of the module, SD memory card, extended SRAM cassette, or connector. Doing so may cause malfunction or failure of the module.
-

## [Wiring Precautions]

---

### **WARNING**

---

- Shut off the external power supply (all phases) used in the system before installation and wiring. Failure to do so may result in electric shock or cause the module to fail or malfunction.
  - After installation and wiring, attach the included terminal cover to the module before turning it on for operation. Failure to do so may result in electric shock.
-

## [Wiring Precautions]

---

### CAUTION

- Individually ground the FG and LG terminals of the programmable controller with a ground resistance of 100 ohms or less. Failure to do so may result in electric shock or malfunction.
  - Use applicable solderless terminals and tighten them within the specified torque range. If any spade solderless terminal is used, it may be disconnected when the terminal screw comes loose, resulting in failure.
  - Check the rated voltage and terminal layout before wiring to the module, and connect the cables correctly. Connecting a power supply with a different voltage rating or incorrect wiring may cause fire or failure.
  - Connectors for external devices must be crimped or pressed with the tool specified by the manufacturer, or must be correctly soldered.
  - Securely connect the connector to the module. Poor contact may cause malfunction.
  - Do not install the control lines or communication cables together with the main circuit lines or power cables. Keep a distance of 100mm or more between them. Failure to do so may result in malfunction due to noise.
  - Place the cables in a duct or clamp them. If not, dangling cable may swing or inadvertently be pulled, resulting in damage to the module or cables or malfunction due to poor contact. Do not clamp the extension cables with the jacket stripped.
  - Check the interface type and correctly connect the cable. Incorrect wiring (connecting the cable to an incorrect interface) may cause failure of the module and external device.
  - Tighten the terminal screws or connector screws within the specified torque range. Undertightening can cause drop of the screw, short circuit, fire, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
  - When disconnecting the cable from the module, do not pull the cable by the cable part. For the cable with connector, hold the connector part of the cable. For the cable connected to the terminal block, loosen the terminal screw. Pulling the cable connected to the module may result in malfunction or damage to the module or cable.
  - Prevent foreign matter such as dust or wire chips from entering the module. Such foreign matter can cause a fire, failure, or malfunction.
  - A protective film is attached to the top of the module to prevent foreign matter, such as wire chips, from entering the module during wiring. Do not remove the film during wiring. Remove it for heat dissipation before system operation.
  - Programmable controllers must be installed in control panels. Connect the main power supply to the power supply module in the control panel through a relay terminal block. Wiring and replacement of a power supply module must be performed by qualified maintenance personnel with knowledge of protection against electric shock. For wiring, refer to the MELSEC iQ-R Module Configuration Manual.
  - For Ethernet cables to be used in the system, select the ones that meet the specifications in the user's manual for the module used. If not, normal data transmission is not guaranteed.
-



## [Startup and Maintenance Precautions]

---

### **WARNING**

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

## [Startup and Maintenance Precautions]

---

### **CAUTION**

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

## [Startup and Maintenance Precautions]

---

### CAUTION

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

## [Operating Precautions]

---

### CAUTION

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

## [Disposal Precautions]

---

### CAUTION

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

## [Transportation Precautions]

---

### CAUTION

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

# CONDITIONS OF USE FOR THE PRODUCT

---

(1) Mitsubishi programmable controller ("the PRODUCT") shall be used in conditions;

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

(2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries.

MITSUBISHI SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI'S USER, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.

("Prohibited Application")

Prohibited Applications include, but not limited to, the use of the PRODUCT in;

- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
- Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
- Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.

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

## INTRODUCTION

---

Thank you for purchasing the Mitsubishi MELSEC iQ-R series programmable controllers.

This manual describes the memory, functions, devices, and parameters of the relevant products listed below.

Before using this product, please read this manual and the relevant manuals carefully and develop familiarity with the functions and performance of the MELSEC iQ-R series programmable controller to handle the product correctly.

When applying the program examples provided in this manual to an actual system, ensure the applicability and confirm that it will not cause system control problems.

Please make sure that the end users read this manual.

### Relevant products

---

R04CPU, R08CPU, R08PCPU, R16CPU, R16PCPU, R32CPU, R32PCPU, R120CPU, R120PCPU

# CONTENTS

---

SAFETY PRECAUTIONS .....	1
CONDITIONS OF USE FOR THE PRODUCT .....	9
INTRODUCTION .....	9
RELEVANT MANUALS .....	19
TERMS .....	20

## PART 1 PROGRAMMING

---

<b>CHAPTER 1 RUNNING A PROGRAM</b> .....	<b>22</b>
<b>1.1 Scan Configuration</b> .....	<b>22</b>
Initial processing (when powered on or switched to RUN) .....	22
I/O refresh .....	22
Program operation .....	22
END processing .....	23
<b>1.2 Scan Time</b> .....	<b>23</b>
Initial scan time .....	23
<b>1.3 Data Communication and I/O Processing</b> .....	<b>24</b>
Data communication .....	24
I/O processing and response delay .....	24
Refresh mode .....	25
Direct mode .....	28
<b>1.4 Program Flow</b> .....	<b>30</b>
<b>1.5 Program Execution Type</b> .....	<b>31</b>
Initial execution type program .....	31
Scan execution type program .....	32
Fixed scan execution type program .....	32
Event execution type program .....	37
Standby type program .....	41
Execution type change .....	42
Group setting for refresh .....	43
<b>1.6 Program Type</b> .....	<b>44</b>
Subroutine program .....	44
Interrupt program .....	45
<b>CHAPTER 2 OPERATION PROCESSING ACCORDING TO OPERATING STATUS OF THE CPU MODULE</b> .....	<b>57</b>
<b>CHAPTER 3 OPERATION PROCESSING AT MOMENTARY POWER FAILURE</b> .....	<b>59</b>
<b>CHAPTER 4 MEMORY CONFIGURATION OF THE CPU MODULE</b> .....	<b>61</b>
<b>4.1 Memory Configuration</b> .....	<b>61</b>
Program memory/program cache memory .....	62
Device/label memory .....	63
Data memory .....	64
Refresh memory .....	64
CPU buffer memory .....	64
SD memory card .....	64

4.2	File Size Unit in Memory	65
4.3	Memory Operation	65
4.4	Files	66
	File types and storage memory	66
	File operation available	67
	File size	68

## PART 2 FUNCTIONS

<b>CHAPTER 5 SCAN MONITORING FUNCTION</b>		<b>73</b>
5.1	Scan Time Monitoring Time Setting	73
5.2	Watchdog Timer Reset	73
5.3	Precautions	74
	Measurement error of watchdog timer	74
	Resetting the watchdog timer when repeatedly executing a program	74
	Scan time when the WDT reset instruction is used	74
<b>CHAPTER 6 CLOCK FUNCTION</b>		<b>75</b>
6.1	Time Setting	75
	Changing the clock data	75
	Reading the clock data	75
	Precautions on the clock data	76
6.2	Setting Time Zone	76
6.3	System Clock	77
	Special relay used for system clock	77
	Special register used for system clock	77
<b>CHAPTER 7 ONLINE CHANGE</b>		<b>78</b>
7.1	Online Ladder Block Change	78
	Editable contents	78
	Range changeable in a single session	79
	Steps reserved for online ladder block change	79
	Online ladder block change during the boot operation	79
	Setting the initial value for registering/changing label definition	80
	Precautions	80
<b>CHAPTER 8 INTERRUPT FUNCTION</b>		<b>84</b>
8.1	Multiple Interrupt Function	84
	Interrupt priority	84
	Multiple interrupt execution sequence	87
	Precautions	88
<b>CHAPTER 9 PID CONTROL FUNCTION</b>		<b>89</b>
<b>CHAPTER 10 PROCESS CONTROL FUNCTION</b>		<b>91</b>
<b>CHAPTER 11 CONSTANT SCAN</b>		<b>93</b>
11.1	Setting Constant Scan	93
11.2	Accuracy of Constant Scan	93
11.3	Precautions	94

<b>CHAPTER 12 DATABASE FUNCTION</b>	<b>95</b>
<b>12.1 Usage Procedure</b>	<b>95</b>
Creating databases	95
Operating databases	96
<b>12.2 List of Database Access Instructions</b>	<b>96</b>
<b>12.3 Creating Unicode Text Files</b>	<b>97</b>
Database specifications	97
Setting details of Unicode text file	98
<b>12.4 Transactions for Databases</b>	<b>101</b>
<b>12.5 Folder Configuration of Databases</b>	<b>101</b>
<b>12.6 Timing of Database Update</b>	<b>102</b>
<b>12.7 Precautions</b>	<b>102</b>
<b>CHAPTER 13 REMOTE OPERATION</b>	<b>103</b>
<b>13.1 Remote RUN/STOP</b>	<b>103</b>
Executing method	103
Precautions	104
<b>13.2 Remote PAUSE</b>	<b>104</b>
Executing method	104
<b>13.3 Setting RUN-PAUSE contacts</b>	<b>105</b>
<b>13.4 Remote RESET</b>	<b>105</b>
Enabling remote RESET	105
Executing method	105
<b>CHAPTER 14 DEVICE/LABEL MEMORY AREA SETTING</b>	<b>106</b>
<b>14.1 Default Capacity of Each Area</b>	<b>106</b>
<b>14.2 The Setting Range of the Capacity of Each Area</b>	<b>107</b>
<b>14.3 Device/Label Memory Area Setting</b>	<b>108</b>
<b>14.4 Device Setting</b>	<b>109</b>
Range of use of device points	109
<b>CHAPTER 15 INTERNAL BUFFER CAPACITY SETTING</b>	<b>110</b>
<b>CHAPTER 16 DEVICE/LABEL INITIAL VALUE SETTINGS</b>	<b>111</b>
<b>16.1 Setting Initial Device/Label Values</b>	<b>111</b>
Setting initial device values	111
Setting initial label values	112
<b>16.2 Applicable Devices/Labels</b>	<b>113</b>
<b>16.3 Precautions</b>	<b>113</b>
<b>CHAPTER 17 OUTPUT MODE AT OPERATING STATUS CHANGE (STOP TO RUN)</b>	<b>114</b>
<b>17.1 Operation When the Operating Status Is Changed from STOP to RUN</b>	<b>114</b>
Outputting the output (Y) status before STOP	114
Clearing the output (Y) (output after one scan)	114
<b>17.2 Setting the Output Mode</b>	<b>115</b>
<b>CHAPTER 18 LABEL ACCESS SETTING FROM EXTERNAL DEVICE</b>	<b>116</b>
<b>18.1 Configuration Procedure</b>	<b>116</b>
<b>18.2 Label Communication Data Storage Location and Operation at the Time of Each Setting Operation</b>	<b>117</b>
<b>18.3 File Operation</b>	<b>117</b>

<b>CHAPTER 19 ROUTING SETTING</b>	<b>118</b>
19.1 Setting Method	118
19.2 Setting Example	119
19.3 Precautions	119
<b>CHAPTER 20 LATCH FUNCTION</b>	<b>120</b>
20.1 Types of Latch	120
20.2 Applicable Devices and Labels	120
Applicable devices	120
Applicable labels	121
20.3 Latch Setting	121
Setting latch on devices	121
Setting latch on labels	125
20.4 Clearing Latch Range Data	125
20.5 Precautions	125
<b>CHAPTER 21 MEMORY CARD FUNCTION</b>	<b>126</b>
21.1 SD Memory Card Forced Disable	126
How to forcibly disable the SD memory card	126
Operation of functions accessing the SD memory card	127
Canceling SD memory card forced disable	127
Precautions	127
21.2 Boot Operation	128
Boot operation procedure	128
Specifiable file types	129
Allowed maximum number of boot files	129
Configuring the boot setting	130
Operation when security functions are enabled	131
Precautions	131
<b>CHAPTER 22 DEVICE/LABEL ACCESS SERVICE PROCESSING SETTING</b>	<b>132</b>
22.1 Operations Enabled by Setting Details	133
Execute the Process as Scan Time Proceeds	134
Set Processing Time	135
Set Processing Counts	136
Execute END processing between programs	137
22.2 Setting Method	138
22.3 Precautions	138
<b>CHAPTER 23 DATA LOGGING FUNCTION</b>	<b>139</b>
23.1 Data To be Collected	140
Number of points of data	140
Data type	140
Data to be collected	140
23.2 Data Collection Conditions	141
Each scan	141
Time specification	141
Interruption occurrence	142
Condition specification	143
23.3 Logging Type	146

Operating procedure for continuous logging . . . . .	146
Operating procedure for trigger logging. . . . .	147
<b>23.4 Trigger Condition . . . . .</b>	<b>148</b>
Condition specification. . . . .	148
When trigger instruction executed. . . . .	149
<b>23.5 States of the Data Logging Function . . . . .</b>	<b>150</b>
Data logging states . . . . .	150
Storage states . . . . .	150
LED status. . . . .	151
<b>23.6 Steps Until the Collected Data is Saved . . . . .</b>	<b>152</b>
Switching to a storage file . . . . .	152
<b>23.7 Internal Buffer . . . . .</b>	<b>154</b>
Internal buffer capacity setting. . . . .	154
Amount of internal buffer consumed . . . . .	154
<b>23.8 Data Logging File . . . . .</b>	<b>155</b>
Storage format of data logging files. . . . .	155
<b>23.9 Missing Data . . . . .</b>	<b>155</b>
Conditions under which missing data occurs . . . . .	155
<b>23.10 Special Relay and Special Register Used by the Data Logging Function . . . . .</b>	<b>155</b>
<b>23.11 Other Functions. . . . .</b>	<b>156</b>
Setting the behavior at the time of a transition to RUN. . . . .	156
Auto logging. . . . .	157
SD memory card replacement. . . . .	158
<b>23.12 SD Memory Card Life When the Data Logging Function is Used . . . . .</b>	<b>160</b>
<b>23.13 Precautions to Take When Using the Data Logging Function . . . . .</b>	<b>161</b>
<b>CHAPTER 24 MEMORY DUMP FUNCTION . . . . .</b>	<b>166</b>
<hr/>	
<b>24.1 Object Data . . . . .</b>	<b>166</b>
Data to be collected. . . . .	166
<b>24.2 Trigger Condition. . . . .</b>	<b>167</b>
Device specification. . . . .	167
Error code specification. . . . .	168
Combining trigger conditions. . . . .	169
<b>24.3 Procedure for Memory Dump . . . . .</b>	<b>169</b>
<b>24.4 Flow of Data Collection . . . . .</b>	<b>170</b>
Effect on the scan time . . . . .	170
<b>24.5 Memory Dump File . . . . .</b>	<b>171</b>
Save file name. . . . .	171
<b>24.6 States of the Memory Dump Function. . . . .</b>	<b>171</b>
Memory dump status. . . . .	171
LED status. . . . .	171
<b>24.7 Sizes of Files Used for the Memory Dump Function . . . . .</b>	<b>172</b>
Capacity of the memory dump setting file . . . . .	172
Capacity of the memory dump file. . . . .	172
<b>24.8 Special Relay and Special Register Used in the Memory Dump Function . . . . .</b>	<b>173</b>
<b>24.9 Precautions for the Memory Dump Function . . . . .</b>	<b>173</b>
<b>CHAPTER 25 REAL-TIME MONITOR FUNCTION . . . . .</b>	<b>176</b>
<hr/>	
<b>CHAPTER 26 RAS FUNCTIONS . . . . .</b>	<b>178</b>
<hr/>	



<b>26.1</b>	<b>Self-Diagnostics Function</b> .....	<b>178</b>
	Self-diagnostics timing .....	178
	Check method of error .....	178
	CPU Module Operation Upon Error Detection Setting .....	179
<b>26.2</b>	<b>Error Clear</b> .....	<b>182</b>
<b>26.3</b>	<b>Event History Function</b> .....	<b>184</b>
	Event history settings .....	184
	Logging of the event history .....	185
	Viewing the event history .....	187
	Clearing the event history .....	187
<b>26.4</b>	<b>Program Cache Memory Auto Recovery Function</b> .....	<b>188</b>

---

**CHAPTER 27 MULTIPLE CPU SYSTEM FUNCTION** **189**

<b>27.1</b>	<b>Out-of-group I/O Fetch</b> .....	<b>190</b>
	Accessing controlled module .....	190
	Accessing non-controlled module .....	190
<b>27.2</b>	<b>Operation Settings</b> .....	<b>193</b>
	Stop setting .....	193
	Synchronous startup setting .....	194
	Clock data .....	197
<b>27.3</b>	<b>Multiple CPU Parameter Checking</b> .....	<b>198</b>
<b>27.4</b>	<b>Data Communication Between CPU Modules</b> .....	<b>199</b>
	Memory to be used .....	199
	Fixed scan communication setting .....	204
	Error detection setting .....	204
	Module-by-module data guarantee .....	205
	Communication through refresh .....	207
	Communication through direct access .....	212
	Data guarantee by program .....	215
	Communication between CPU modules in error state .....	217
<b>27.5</b>	<b>Multiple CPU Synchronous Interrupt</b> .....	<b>218</b>
	Execution timing .....	219
	Multiple interrupt .....	219
	Precautions .....	219

---

**CHAPTER 28 SECURITY FUNCTION** **221**

**PART 3 DEVICES AND CONSTANTS**

---

**CHAPTER 29 DEVICES** **224**

---

<b>29.1</b>	<b>Device List</b> .....	<b>224</b>
<b>29.2</b>	<b>User Device</b> .....	<b>226</b>
	Input (X) .....	226
	Output (Y) .....	226
	Internal relay (M) .....	227
	Latch relay (L) .....	227
	Link relay (B) .....	227
	Annunciator (F) .....	227
	Link special relay (SB) .....	229
	Edge relay (V) .....	229

Timer . . . . .	230
Counter . . . . .	238
Data register (D) . . . . .	241
Link register (W) . . . . .	241
Link special register (SW) . . . . .	241
<b>29.3 System Device . . . . .</b>	<b>242</b>
Function device (FX/FY/FD) . . . . .	242
Special relay (SM) . . . . .	242
Special register (SD) . . . . .	242
<b>29.4 Link Direct Device . . . . .</b>	<b>243</b>
Specification method . . . . .	243
Specification range . . . . .	243
Difference from link refresh . . . . .	244
<b>29.5 Module Access Device . . . . .</b>	<b>245</b>
Precautions . . . . .	246
<b>29.6 CPU Buffer Memory Access Device . . . . .</b>	<b>246</b>
Specification method . . . . .	246
<b>29.7 Index Register (Z/LZ) . . . . .</b>	<b>247</b>
Types of Index registers . . . . .	247
Index register setting . . . . .	248
<b>29.8 File Register (R/ZR) . . . . .</b>	<b>249</b>
Specification method . . . . .	249
Setting file registers . . . . .	250
Clearing file registers . . . . .	250
<b>29.9 Refresh Data Register (RD) . . . . .</b>	<b>251</b>
Refresh memory setting . . . . .	251
<b>29.10 Nesting (N) . . . . .</b>	<b>252</b>
<b>29.11 Pointer (P) . . . . .</b>	<b>253</b>
Global pointer . . . . .	253
Local pointer . . . . .	254
Pointer setting . . . . .	255
<b>29.12 Interrupt Pointer (I) . . . . .</b>	<b>256</b>
Interrupt factors of the interrupt pointer numbers . . . . .	256
The priority for the interrupt pointer numbers and interrupt factors . . . . .	257
<b>29.13 Other Devices . . . . .</b>	<b>258</b>
Network No. specification device (J) . . . . .	258
I/O No. specification device (U) . . . . .	258

---

**CHAPTER 30 CONSTANTS 259**

<b>30.1 Decimal Constant (K) . . . . .</b>	<b>259</b>
<b>30.2 Hexadecimal Constant (H) . . . . .</b>	<b>259</b>
<b>30.3 Real Constant (E) . . . . .</b>	<b>260</b>
Setting range for real numbers . . . . .	260
Activity when operation is in progress . . . . .	260
Expression on program . . . . .	260
<b>30.4 Character String Constant . . . . .</b>	<b>261</b>
Rules for special character . . . . .	261

---

**CHAPTER 31 HOW TO USE DEVICES CONVENIENTLY 262**

<b>31.1 Global Device . . . . .</b>	<b>262</b>
-------------------------------------	------------

<b>31.2</b>	<b>Local Device</b> .....	<b>262</b>
	Devices available as local device .....	262
	Local device area .....	263
	When local device is used in subroutine program: .....	263
	When local device is used in interrupt and other programs: .....	263
	Clearing local device .....	263
	Setting method for the local devices .....	264
	Specification method for the local devices .....	265
	Precautions .....	265
<b>31.3</b>	<b>Index Modification</b> .....	<b>266</b>
	16-bit index modification .....	266
	32-bit index modification .....	266
	Device for which Index modification can be performed .....	266
	Combination of index modification .....	266
	Precautions .....	267
<b>31.4</b>	<b>Indirect Specification</b> .....	<b>269</b>
	Indirect address of device .....	269
	Devices that can allow indirect specification .....	269

## **APPENDICES** **270**

<b>Appendix 1</b>	<b>Error Codes</b> .....	<b>270</b>
	Error code system .....	270
	Operation when an error occurs .....	272
	How to clear errors .....	272
	List of error codes .....	272
<b>Appendix 2</b>	<b>Event List</b> .....	<b>312</b>
	Guide for reference of event list .....	312
	Event list .....	313
<b>Appendix 3</b>	<b>Troubleshooting by Symptom</b> .....	<b>316</b>
	When the POWER LED of the power supply module turns off .....	316
	When the READY LED of the CPU module turns off .....	316
	When the specific extension base unit cannot be recognized .....	317
	When the specific Q series extension base unit cannot be recognized .....	317
	When data cannot be written to the programmable controller .....	317
	When data cannot be read from the programmable controller .....	318
	When Ethernet function cannot be used .....	318
<b>Appendix 4</b>	<b>List of Special Relay Areas</b> .....	<b>319</b>
<b>Appendix 5</b>	<b>List of Special Register Areas</b> .....	<b>335</b>
<b>Appendix 6</b>	<b>Processing Time</b> .....	<b>358</b>
	Instruction execution time .....	358
	Program execution time .....	358
	END processing time .....	360
	Data logging function processing time .....	368
	Memory dump function processing time .....	369
	Real-time monitor function processing time .....	369
	Database function processing time .....	370
	Process control function processing time .....	370
<b>Appendix 7</b>	<b>Parameter List</b> .....	<b>371</b>
	System parameters .....	371
	CPU parameters .....	372

Memory card parameters .....	373
<b>Appendix 8 List of the Targets of Device/Label Access Service Processing Setting .....</b>	<b>374</b>
Communication functions via the MC protocol .....	374
Communication function using an engineering tool .....	375
<b>Appendix 9 How to Use CPU Module Logging Configuration Tool .....</b>	<b>376</b>
System configuration .....	376
Specifications .....	379
Data logging procedure .....	390
Menu items and setting methods .....	391
<b>Appendix 10 Added and Enhanced Functions .....</b>	<b>417</b>
<b>INDEX</b>	<b>419</b>
<hr/>	
REVISIONS .....	422
WARRANTY .....	423
TRADEMARKS .....	424

# RELEVANT MANUALS

Manual name [manual number]	Description	Available form
MELSEC iQ-R CPU Module User's Manual (Application) [SH-081264ENG] (this manual)	Memory, functions, devices, and parameters of the CPU module	Print book e-Manual EPUB PDF
MELSEC iQ-R CPU Module User's Manual (Startup) [SH-081263ENG]	Performance specifications, procedures before operation, and troubleshooting of the CPU module	Print book e-Manual EPUB PDF
MELSEC iQ-R Ethernet User's Manual (Application) [SH-081257ENG]	Functions, parameter settings, programming, troubleshooting, I/O signals, and buffer memory of Ethernet	Print book e-Manual EPUB PDF
MELSEC iQ-R Programming Manual (Program Design) [SH-081265ENG]	Program specifications, such as ladder programs and ST programs, and labels	e-Manual EPUB PDF
MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks) [SH-081266ENG]	Instructions for the CPU module, dedicated instructions for the intelligent function modules, and standard functions/function blocks	e-Manual EPUB PDF

## Point

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

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

# TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
Buffer memory	Memory in an intelligent function module for storing data such as setting values and monitored values. When integrated into the CPU module, this memory refers to a memory for storing data such as setting values and monitored values of the Ethernet function, and data used for data communication of the multiple CPU system function.
Control CPU	A CPU module that controls connected I/O modules and intelligent function modules. The multiple CPU system allows the user to assign this control to any CPU module on a module-by-module basis.
CPU module	A generic term for the MELSEC iQ-R series CPU module
CPU Module Logging Configuration Tool	Software to configure data logging settings and to manage collected data
Dedicated instruction	An instruction for using functions of the module
Device	A device (X, Y, M, D, or others) in a CPU module
Engineering tool	The product name of the software package for the MELSEC programmable controllers
Ethernet-equipped module	A generic term for the RJ71EN71 (when the Ethernet function is used) and MELSEC iQ-R series CPU module (when the Ethernet function is used)
Global label	A label that is valid for all the program data when multiple program data are created in the project. There are two types of global label: a module specific label (module label), which is generated automatically by GX Works3, and an optional label, which can be created for any specified device.
GX LogViewer	Software to display data collected by data logging
I/O module	A generic term for the input module, output module, I/O combined module, and interrupt module
Intelligent function module	A module that has functions other than input and output, such as an A/D converter module and D/A converter module
Label	A label that represents a device in a given character string
Module label	A label that represents one of memory areas (I/O signals and buffer memory areas) specific to each module in a given character string. For the module used, GX Works3 automatically generates this label, which can be used as a global label.
Network module	A generic term for the following modules: <ul style="list-style-type: none"> <li>• Ethernet interface module</li> <li>• CC-Link IE Controller Network module</li> <li>• Module on CC-Link IE Field Network</li> <li>• MELSECNET/H module</li> <li>• MELSECNET/10 module</li> </ul>
POU	A unit that configures a program. Units are categorized and provided in accordance with functions. A program that is composed of POUs (program organization units) allows the lower-layer processing, when the program is multi-layered, to be divided into several units by processing and function, enabling the creation of programs based on each unit.
Process CPU	A generic term for the R08PCPU, R16PCPU, R32PCPU, and R120PCPU
Program block	A group of POUs that configure a program
Programmable controller CPU	A generic term for the R04CPU, R08CPU, R16CPU, R32CPU, and R120CPU
RAS	The abbreviation for Reliability, Availability, and Serviceability. This term refers to the overall usability of automated equipment.
Windows Vista <sup>®</sup>	A generic term for Microsoft <sup>®</sup> Windows Vista <sup>®</sup> Home Basic Operating System, Microsoft <sup>®</sup> Windows Vista <sup>®</sup> Home Premium Operating System, Microsoft <sup>®</sup> Windows Vista <sup>®</sup> Business Operating System, Microsoft <sup>®</sup> Windows Vista <sup>®</sup> Ultimate Operating System, and Microsoft <sup>®</sup> Windows Vista <sup>®</sup> Enterprise Operating System
Windows <sup>®</sup> 7 or later	A generic term for Microsoft <sup>®</sup> Windows <sup>®</sup> 7 Starter Operating System, Microsoft <sup>®</sup> Windows <sup>®</sup> 7 Home Premium Operating System, Microsoft <sup>®</sup> Windows <sup>®</sup> 7 Professional Operating System, Microsoft <sup>®</sup> Windows <sup>®</sup> 7 Ultimate Operating System, Microsoft <sup>®</sup> Windows <sup>®</sup> 7 Enterprise Operating System, Microsoft <sup>®</sup> Windows <sup>®</sup> 8 Operating System, Microsoft <sup>®</sup> Windows <sup>®</sup> 8 Pro Operating System, Microsoft <sup>®</sup> Windows <sup>®</sup> 8 Enterprise Operating System, Microsoft <sup>®</sup> Windows <sup>®</sup> 8.1 Operating System, Microsoft <sup>®</sup> Windows <sup>®</sup> 8.1 Pro Operating System, and Microsoft <sup>®</sup> Windows <sup>®</sup> 8.1 Enterprise Operating System
Windows <sup>®</sup> XP	A generic term for Microsoft <sup>®</sup> Windows <sup>®</sup> XP Home Edition Operating System, and Microsoft <sup>®</sup> Windows <sup>®</sup> XP Professional Operating System

Part 1 consists of the following chapters.

1 RUNNING A PROGRAM

---

2 OPERATION PROCESSING ACCORDING TO OPERATING STATUS OF THE CPU MODULE

---

3 OPERATION PROCESSING AT MOMENTARY POWER FAILURE

---

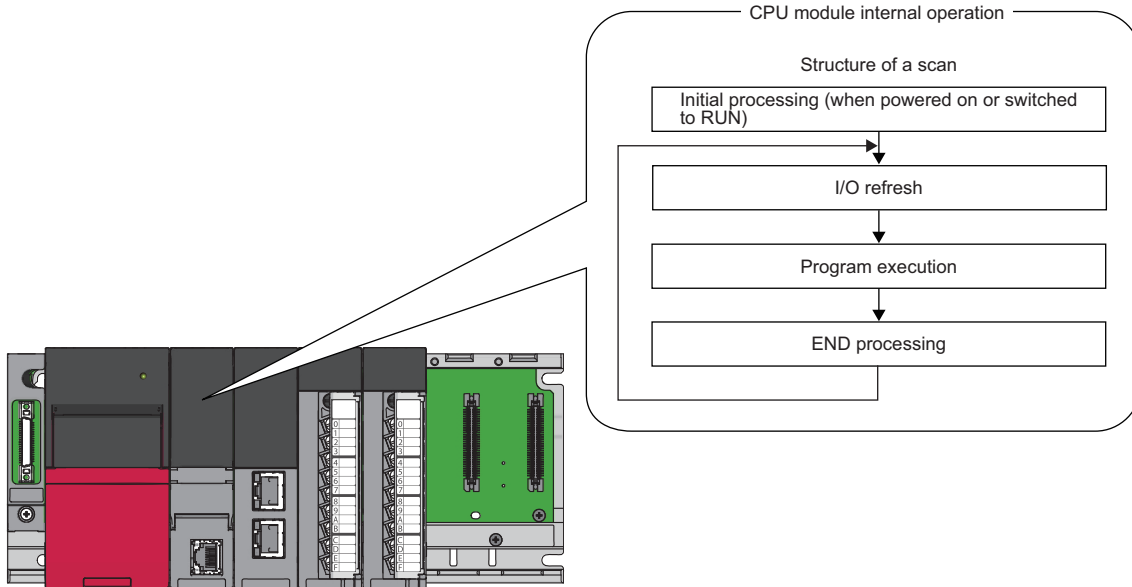
4 MEMORY CONFIGURATION OF THE CPU MODULE

---

# 1 RUNNING A PROGRAM

## 1.1 Scan Configuration

The following shows the scan configuration of the CPU module.



### Initial processing (when powered on or switched to RUN)

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

- Booting from an SD memory card
- Checking each parameter and program
- 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

### I/O refresh

The module performs the following before starting program operation.


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

#### Point

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

### Program operation



According to the program settings, the module executes from step 0 through the END/FEND instruction for each program.

This program is referred to as a main routine program. A main routine program can be divided into subroutine programs. (  Page 44 Subroutine program)



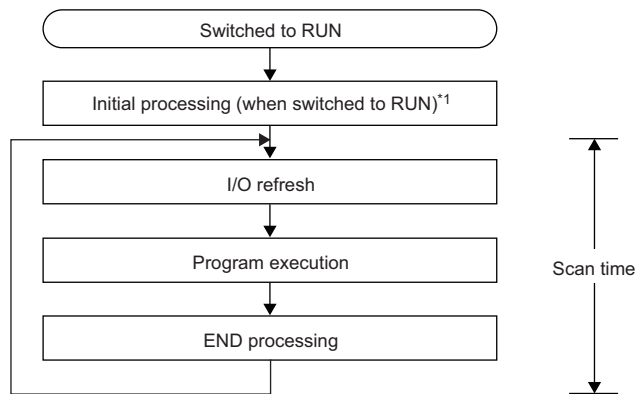
## END processing

Performs the following processing.

- Network module 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 374 List of the Targets of Device/Label Access Service Processing Setting).
- Watchdog timer reset (  Page 73 Watchdog Timer Reset)
- Refresh between CPU modules (for the multiple CPU system configuration)
- Device collection by the data logging function (when the collection interval is set to "Each scanning cycle" or "Time specification")
- Self-diagnostics processing
- Sets a value to the special relay/special register (for those with the set timing specified as the END processing timing)

## 1.2 Scan Time

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




\*1 The initial scan time includes this processing.

### Initial scan time


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

#### How to check the initial scan time

See below.

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

#### Initial scan time monitoring

Monitors at the initial scan time monitoring time. (  Page 73 Scan Time Monitoring Time Setting)

#### ■Precautions on the initial scan time monitoring time

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

# 1.3 Data Communication and I/O Processing

## Data communication

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

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

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

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

## I/O processing and response delay

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

### Refresh mode

The CPU module performs I/O processing collectively at a specified timing. The following table lists the refresh timing which can be specified. (☞ Page 25 Refresh mode)

Refresh timing	Description
When END processing is performed	The refresh processing is performed in every END processing*1.
At the execution time of specified program	The refresh processing is performed before and after specified program execution. (☞ Page 43 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 28 Direct mode)

### Differences between refresh mode and direct mode

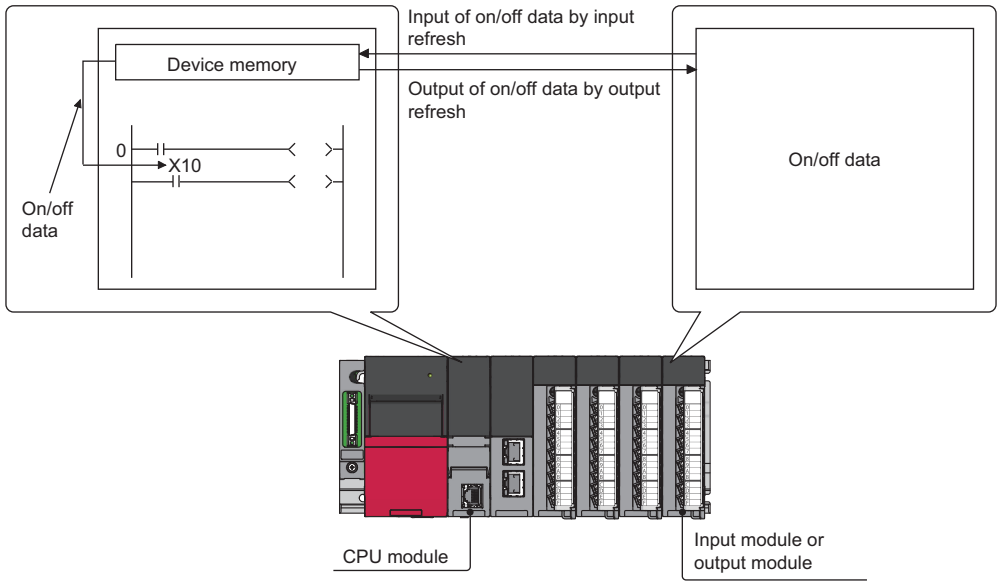
The direct mode directly accesses I/O modules at execution of an instruction. Therefore, data is input faster than when it is input in refresh mode. Processing time required for each instruction, however, takes longer. The following table lists the availability of the refresh mode and the direct mode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

Item	Refresh mode	Direct mode
Input/output modules	Enabled	Enabled
Input/output of intelligent function modules		Enabled
Remote input/output in CC-Link IE or CC-Link		Disabled

# Refresh mode

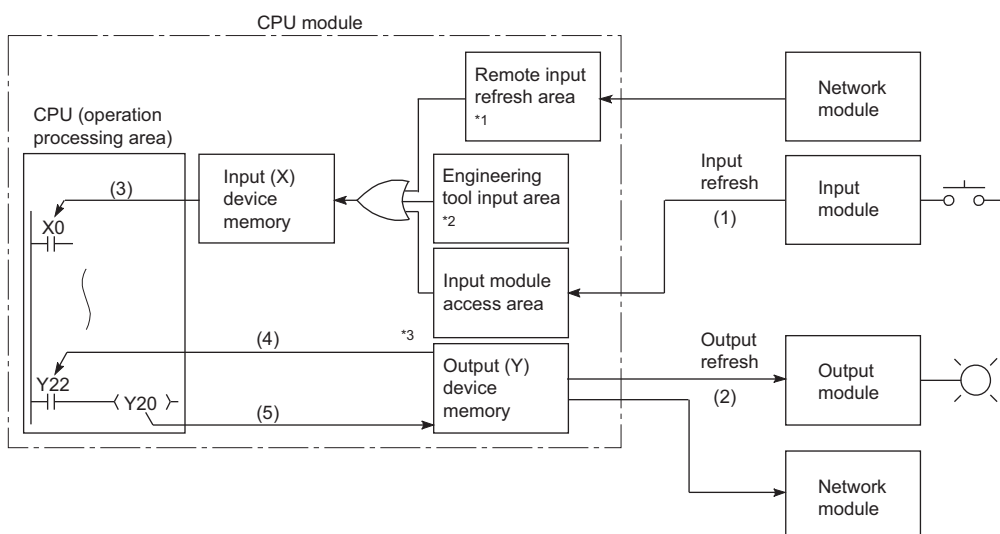
The CPU module performs I/O processing collectively at a specified timing.

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 auto refresh is set to the input (X) in the CC-Link IE 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.

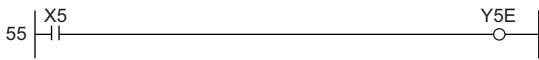
### Output

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).

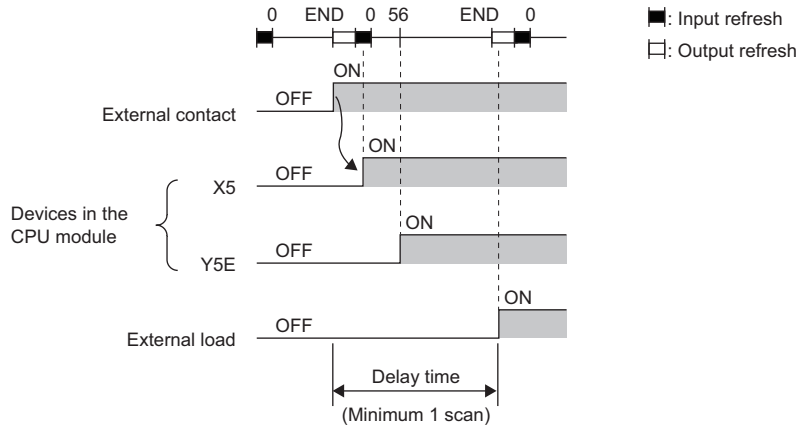
## Response delay

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.

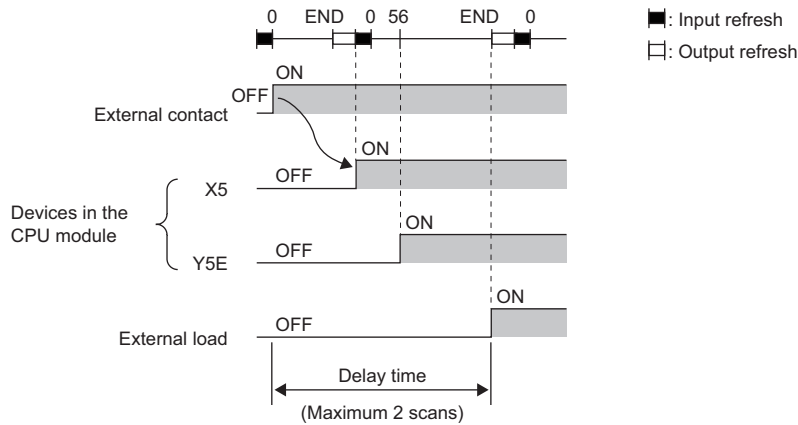
[Example] A program that turns on the output Y5E when the input X5 turns on



- Y5E turns on the earliest

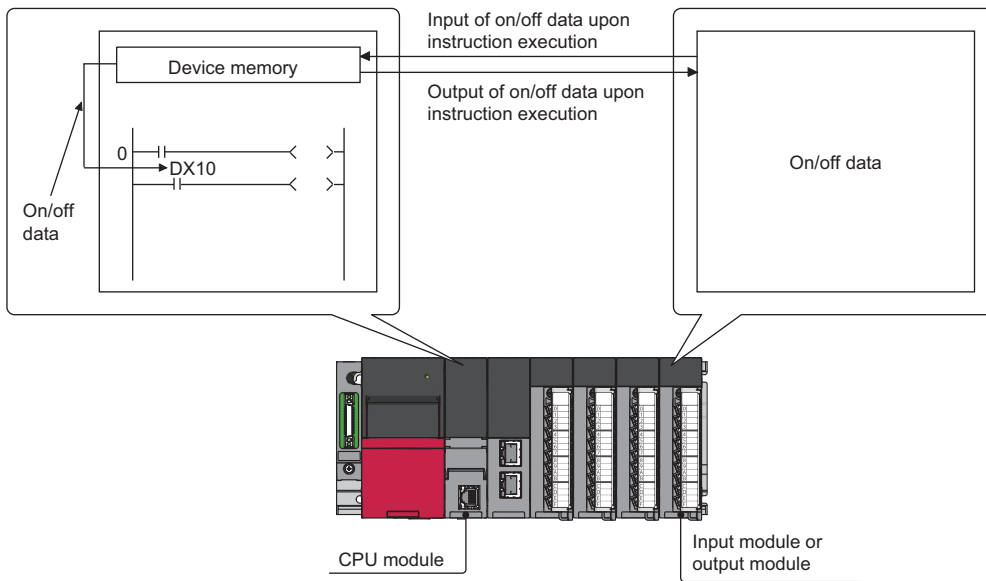


- Y5E turns on the latest



# Direct mode

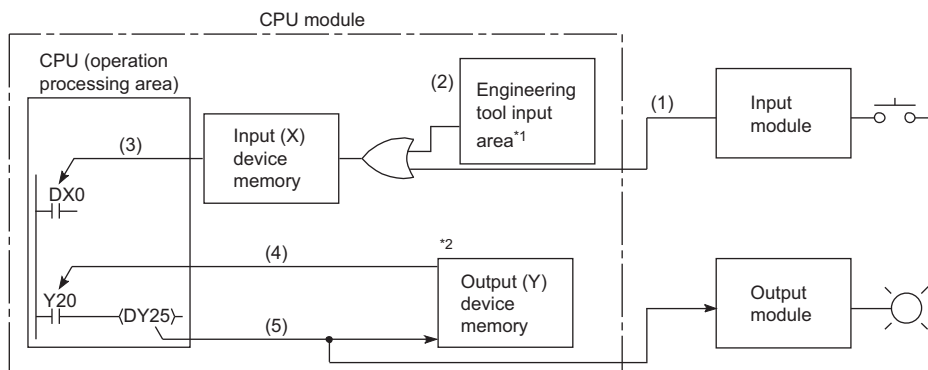
The CPU module performs I/O processing when each instruction is executed in a program.



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:

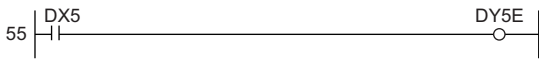
- Test operation of an engineering tool
- Writing data from the network module
- Writing data from an external device using SLMP

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.

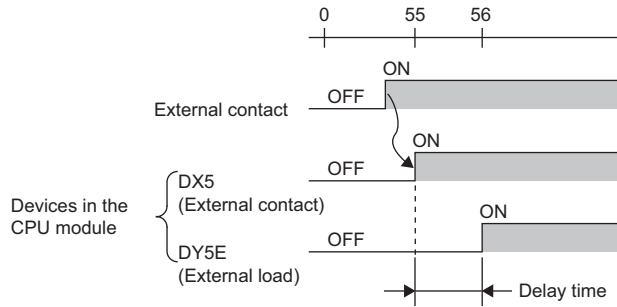
## Response delay

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.

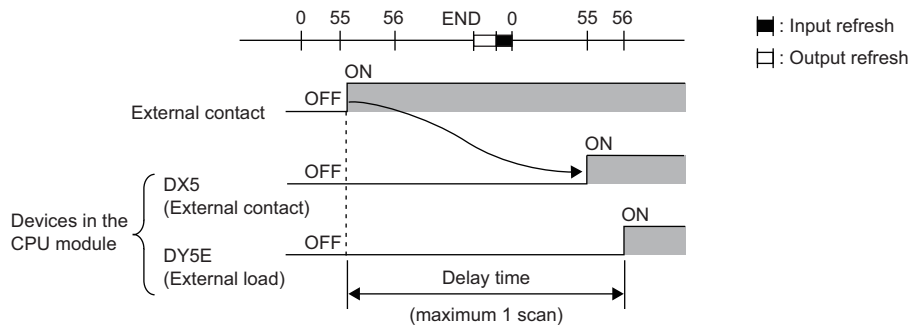
[Example] A program that turns on the output DY5E when the input DX5 turns on



- Y5E turns on the earliest

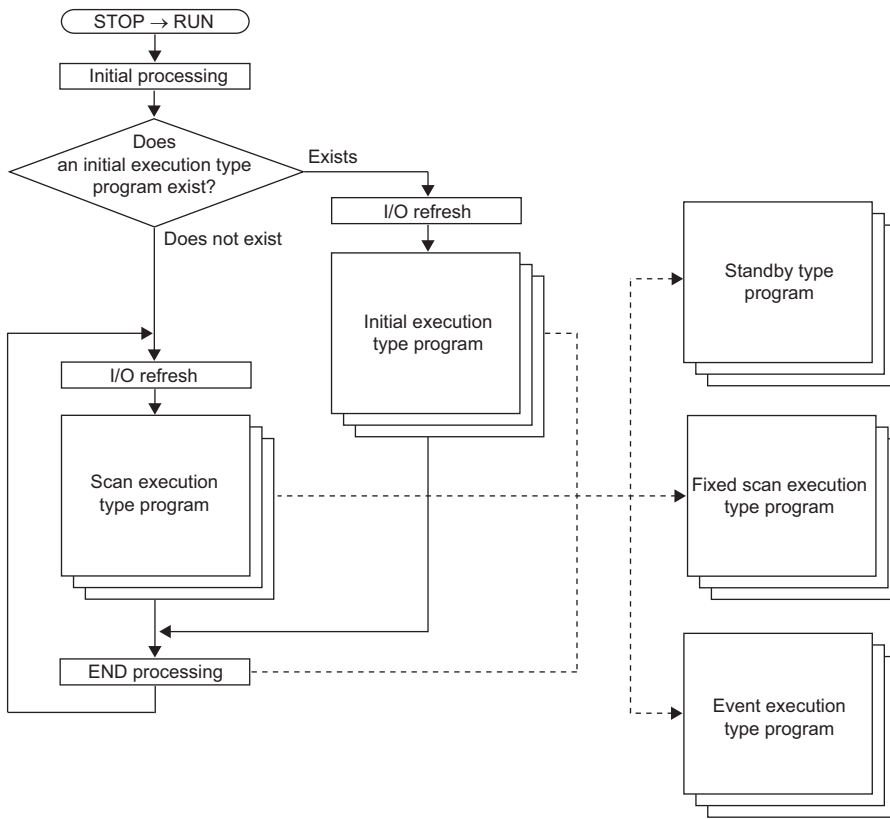


- Y5E turns on the latest



# 1.4 Program Flow

Programs are executed in order when the CPU module is switched to the RUN state according to the program execution type and execution sequence settings (☞ Page 31 Program Execution Type, ☞ Page 42 Execution type change).



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

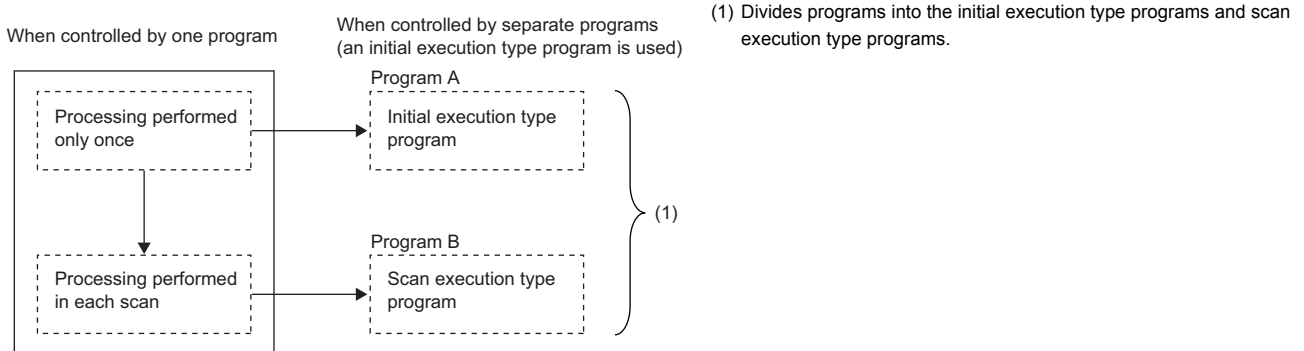


# 1.5 Program Execution Type

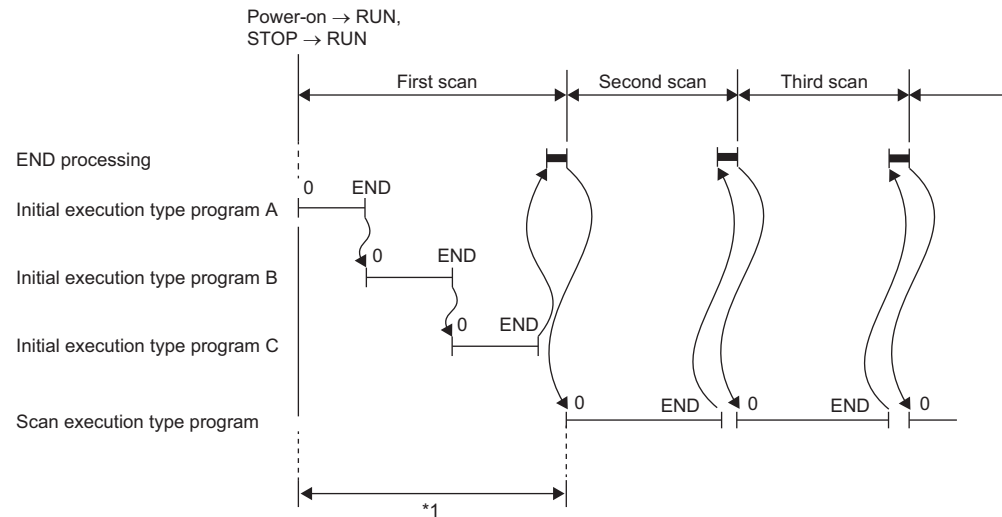
Sets the execution condition of the program. (☞ Page 42 Execution type change)

## Initial execution type program

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



Note that the execution time of the initial execution type program is the same as the initial scan time. (☞ Page 23 Initial scan time) When multiple initial execution type programs are executed, the execution time of the initial execution type program will be the total time until all the initial execution type programs finish execution.



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

### Precautions

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

#### ■Restrictions on program creation

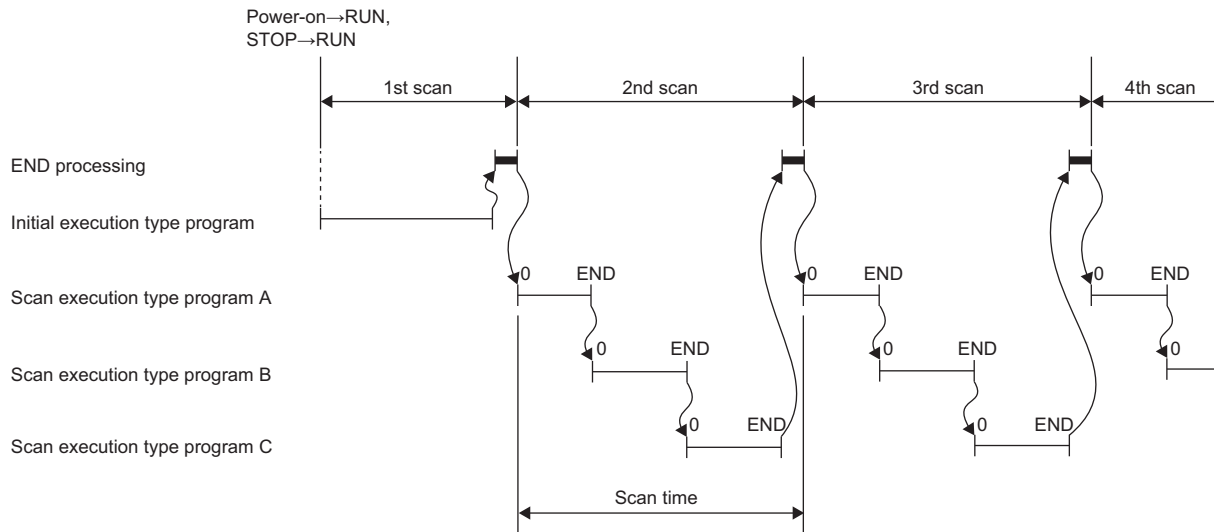
In the initial execution type program, do not use an instruction which requires several scans to complete (i.e., an instruction with a completion device).

**Ex.**

e.g. SEND and RECV instructions

# Scan execution type program

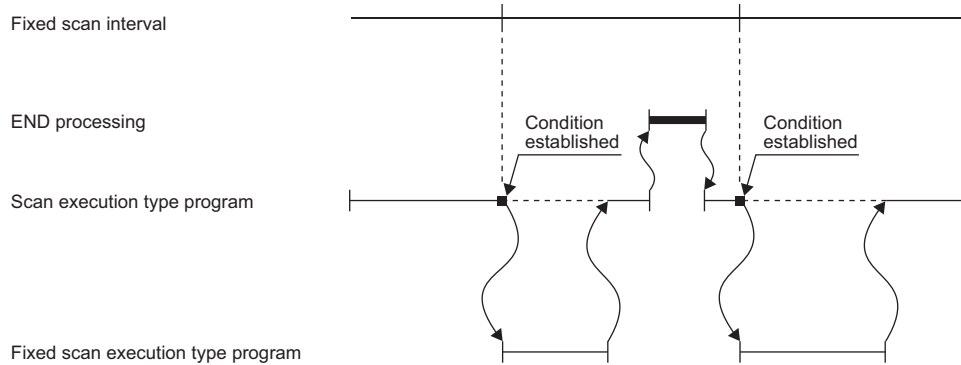
Executed only once per every scan starting from the scan following the scan in which the initial execution type program was executed.



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

# Fixed scan execution type program


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



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

## Fixed scan interval setting

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

 [CPU Parameter] ⇒ "Program Setting"

### Operating procedure

"Program Setting" window

Item	Setting
Program Setting	
Program Setting	<Detailed Setting>

"Detailed Setting" window

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

"Fixed Scan Execution Type Detailed Setting" window

Item	Setting
Fixed Scan Interval	
Unit	ms

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.	<ul style="list-style-type: none"> <li>• When "ms" is selected: 0.5 to 60000ms (in units of 0.5ms)</li> <li>• When "s" is selected: 1 to 60s (in units of 1s)</li> </ul>	—
Unit	Selects the unit for the fixed scan interval.	<ul style="list-style-type: none"> <li>• ms</li> <li>• s</li> </ul>	ms

## Action when the execution condition is satisfied

Performs the following action.

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

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

### ■If there are multiple fixed scan execution type programs

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

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

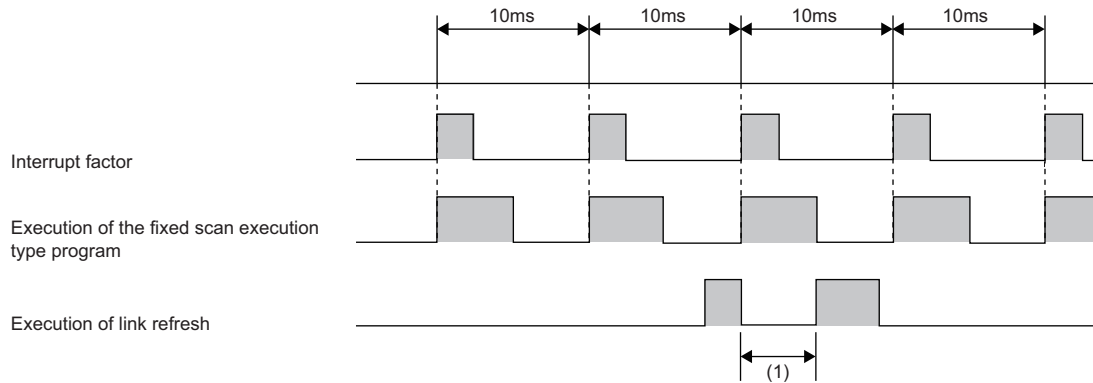
Operates according to the fixed scan execution mode setting.

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

Operates according to the fixed scan execution mode setting.

### ■If an interrupt factor occurs during link refresh

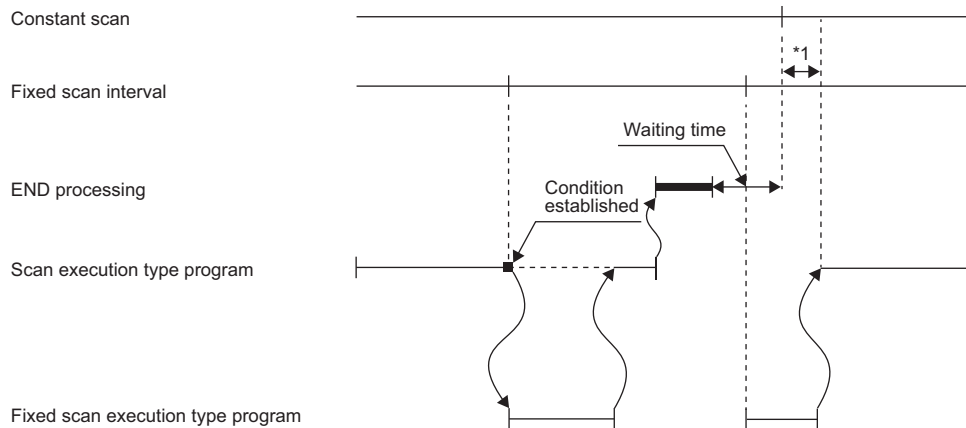
Suspends link refresh and executes the fixed scan execution type program. Even while station unit block guarantee is enabled for cyclic data during refresh of such links as CC-Link IE, if the fixed scan execution type program uses a device specified as the refresh target, station unit block guarantee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



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

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

Executes the fixed scan execution type program.



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

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

If an interrupt program (including an interrupt which triggers the event execution type program) is triggered while the fixed scan execution type program is being executed, the program operates in accordance with the interrupt priority. (Page 84 Multiple Interrupt Function)

## Processing when the fixed scan execution type program starts

The same processing as when the interrupt program starts. (☞ Page 54 Processing at interrupt program startup)

### Fixed scan execution mode

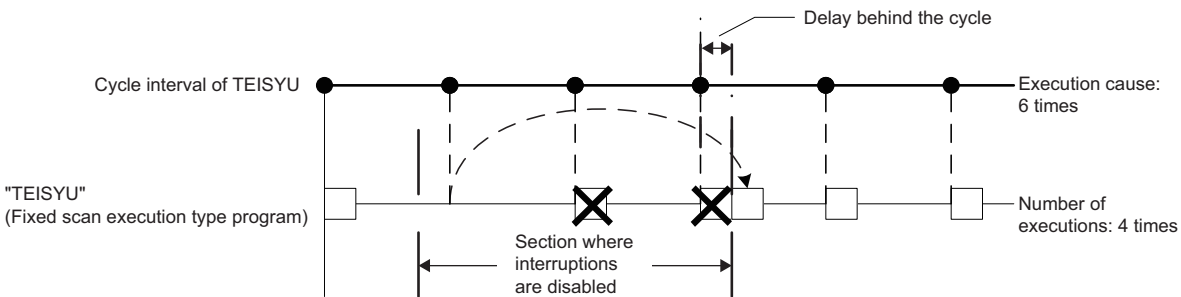
For fixed scan interrupts (I28 to I31, I48, I49) triggered by the fixed scan execution type program or the internal timer of the CPU module, this mode specifies the program execution operation which should be performed when more than one interrupt occurs (☞ Page 46 Operation upon occurrence of an interrupt 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

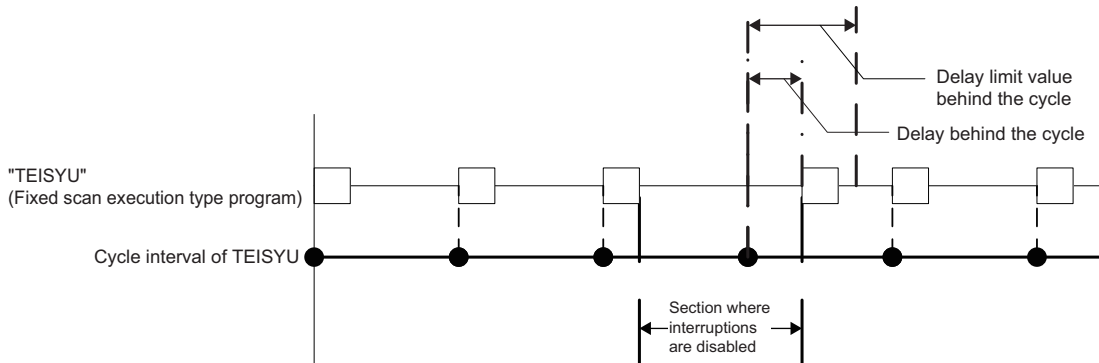
The program is executed for all the pending number of executions so that it can be executed the same number of times as execution causes.



(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 exist, it is performed within the range of delay limit values (☞ Page 36 Delay limit values from the period) However, even if more than one pending execution exists, it is performed only once.



#### ■Fixed scan execution mode setting

Use the fixed scan execution mode setting.

☞ [CPU Parameter] ⇒ "Interrupt Settings" ⇒ "Fixed Scan Execution Mode Setting"

#### Window

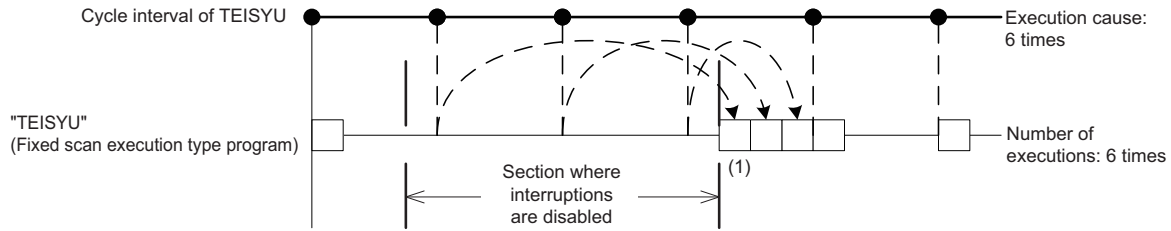


#### Displayed items

Item	Description	Setting range	Default
Fixed Scan Execution Mode	When fixed scan characteristics are prioritized, the program is executed within the range of delay limit values from the period. When the number of executions is prioritized, the program is executed for all the pending number of executions.	<ul style="list-style-type: none"> <li>• Precede Fixed Scan</li> <li>• Execution Count Takes Priority</li> </ul>	Precede Fixed Scan

## ■ Delay limit values from the period

Sets the allowable range of delay (misalignment) from the period and the waiting program is executed if the interrupt is enabled within this range. If the interrupt is enabled outside the range, the program is not executed.



Delay limit values from the period differ as follows.

- For the fixed scan execution type program

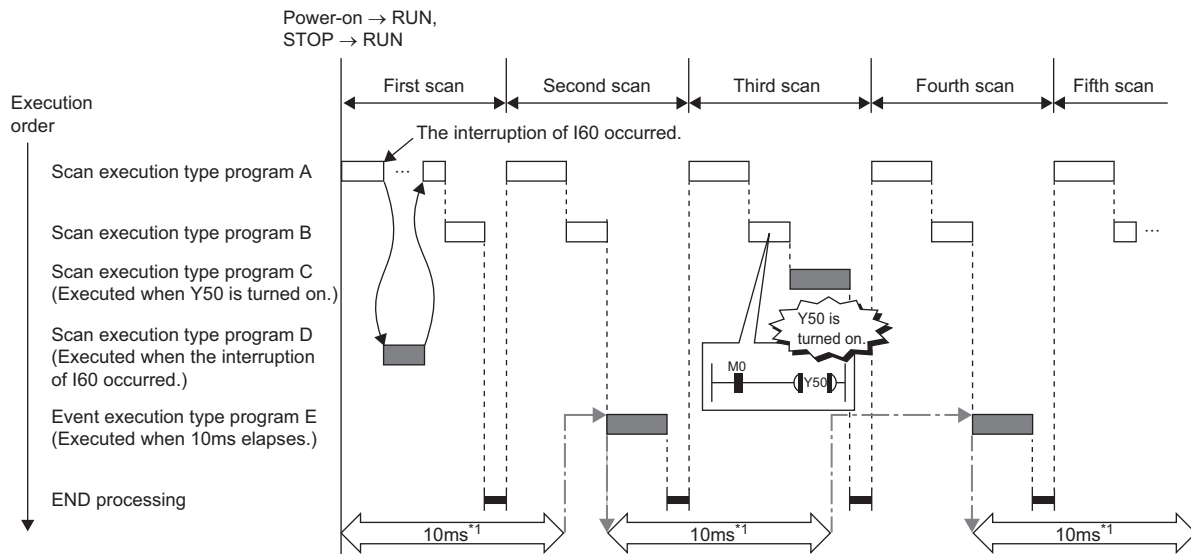
The greatest common divisor of the fixed scan intervals of all the fixed scan execution type programs. For example, if there are fixed scan execution type programs with 2ms, 4ms, and 12ms scan periods, the delay limit values from the period is 2ms.

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

Fixed to the same value as the fixed scan interval. For example, if the periods are I28: 100ms and I29: 20ms, the delay limit values from the period are I28: 100ms and I29: 20ms.

## Event execution type program

This type of program starts execution when triggered by a specified event. (☞ Page 37 Trigger type) The program is executed at the execution turn specified in program setting of the CPU parameters, and if execution conditions of specified trigger are met when the execution turn of the event execution type program comes, the program is executed.



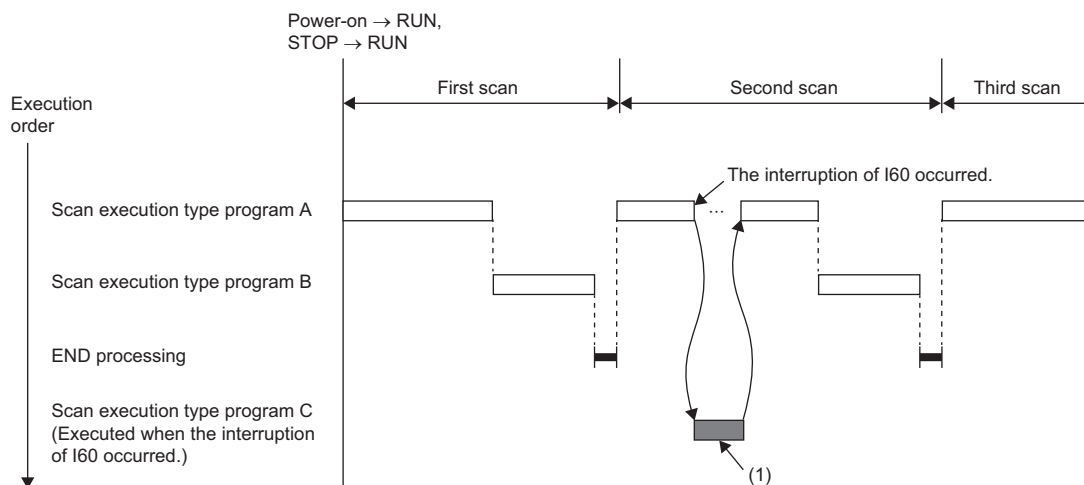
\*1 Measurement of elapsed time is 10ms or more because it is determined depending on the scan time.

### Trigger type

The following lists the triggers for the event execution type programs. (☞ Page 40 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.

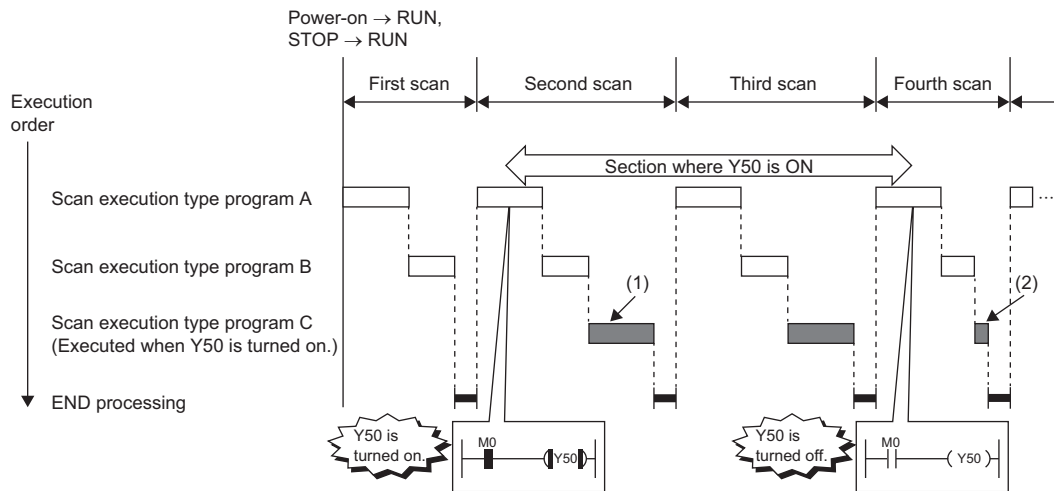
Interrupt pointers (I) which can be specified are I0 to I15, I28 to I31, I44, I45, I48, I49, I50 to I1023.

#### 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 46 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 <sup>*2</sup> , ZR <sup>*2</sup> , RD, Un\G, Jn\W, Jn\SW

\*1 A local device or index-modified device cannot be specified.

\*2 This is not available when "Use File Register of Each Program" is enabled. However, if no file register is assigned or if specification is out of the range, interrupt is not be turned on (TRUE).

### Point

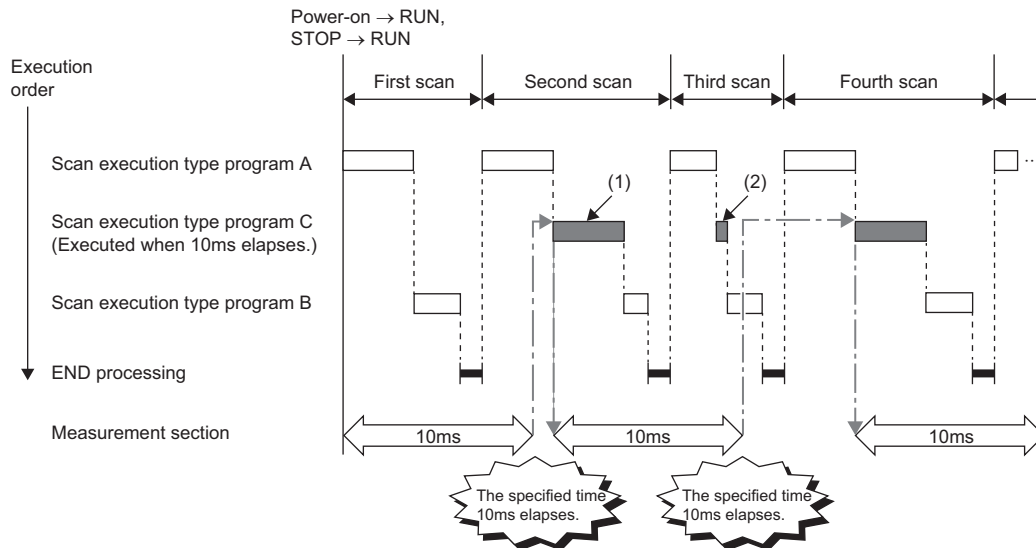
To measure the timer and the retentive timer continuously even when the trigger execution condition of the event execution type program is not met, use the long timer (LT) and the long retentive timer (LST). The timer (T) and the retentive timer (ST) can be used in the event execution type program. However, they do not measure time when the trigger execution condition of the event execution type program is not met. Set "Clear Output and Current Value of Timer" to "Clear" when the long timer (LT) and the long retentive timer (LST) are used. (☞ Page 40 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 passes, 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 passes.

### 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

Item	Setting
Program Setting	<Detailed Setting>

"Detailed Setting" window


Execute Order	Program Name	Execution Type		Refresh Group Setting	Device/File Use or not
		Type	Detailed Setting Information		
1	MAIN	Event	Bit ON:Do Not Clear:	(Do not Set)	<Detailed Setting>
2					

"Event Execution Type Detailed Setting" window

Item	Setting
Trigger Type	ON of Bit Data (TRUE)
Interruption Occurrence	
ON of Bit Data (TRUE)	
Clear Output and Current Value of Timer	Do Not Clear
Passing Time	
Unit	ms
Clear Output and Current Value of Timer	Do Not Clear

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

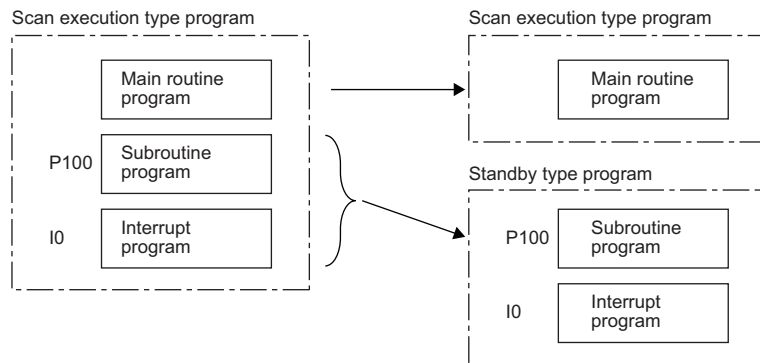
Item	Description	Setting range	Default
Interruption occurrence	Sets the interrupt pointer used as the trigger.	I0 to I15, I28 to I31, I44, I45, I48, I49, I50 to I1023	—
ON of Bit Data (TRUE)	Sets the device used as the trigger.	 Page 38 Bit data ON (TRUE)	—
Clear Output and Current Value of Timer	Sets 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.	<ul style="list-style-type: none"> <li>• Do Not Clear</li> <li>• Clear</li> </ul>	Do Not Clear
Passing Time	Unit	<ul style="list-style-type: none"> <li>• When "ms" is selected: 1 to 65535ms (in units of 1ms)</li> <li>• When "s" is selected: 1 to 65535s (in units of 1s)</li> </ul>	ms
	Clear Output and Current Value of Timer	<ul style="list-style-type: none"> <li>• Do Not Clear</li> <li>• Clear</li> </ul>	Do Not Clear

## Standby type program

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

### Librarization of programs

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



### Program arrangement change

Used when programs available for all systems are prepared and only those required are executed. For example, a program set in advance as the standby type with a parameter can be changed to scan execution type and executed. (☞ Page 42 Using an instruction)

### Execution method

The standby type program is executed as follows.

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

# Execution type change

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

## Using parameter settings

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

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

### Operating procedure

"Program Setting" window

Item	Setting
Program Setting	
Program Setting	<Detailed Setting>

"Detailed Setting" window

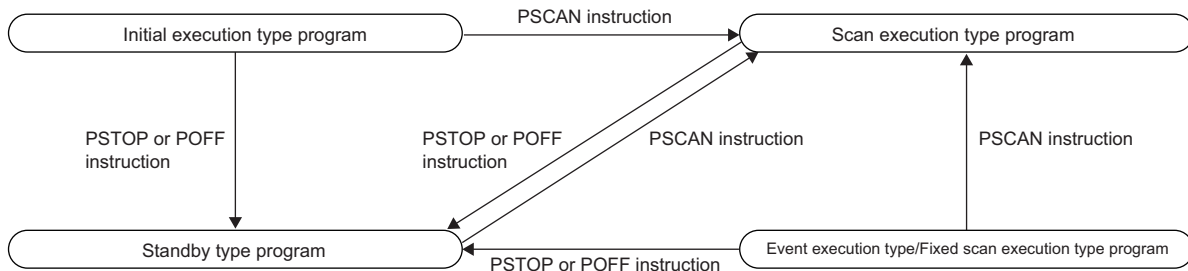
Execute Order	Program Name	Execution Type		Refresh Group Setting	Device/File Use or not
		Type	Detailed Setting Information		
1	MAIN	Scan		(Do not Set)	<Detailed Setting>
2	MAIN1	Fixed Scan		(Do not Set)	<Detailed Setting>
3	MAIN2	Standby		(Do not Set)	<Detailed Setting>

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

## Using an instruction

Use the following instructions.

- PSCAN
- PSTOP
- POFF



### Change timing

The execution type change timing is shown below.

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

\*1 Stops execution at END processing after instruction execution.

### Precautions

The following lists the precautions when changing the execution type.

- If a fixed scan execution type program or event execution type program is changed to another execution type, it cannot be restored to the original execution type.
- If an instruction is executed multiple times in a single scan for a single program, the program operates for the instruction executed last.


## Group setting for refresh

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

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

### Refresh Group Setting

A group number is set to each program.

 [CPU Parameter] ⇒ "Program Setting" ⇒ "Program Setting" ⇒ "Detailed Setting"

#### Window

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

#### Displayed items

Item	Description	Setting range	Default
Refresh Group Setting	Sets the group number of each program which is specified for each module.	<ul style="list-style-type: none"> <li>(Do not Set)</li> <li>Groups [1] to [64]</li> </ul>	(Do not Set)

#### Point

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

### Refresh setting of each module

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

 [Module Parameter] ⇒ "Refresh Setting"

#### Window

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

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

#### Point

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

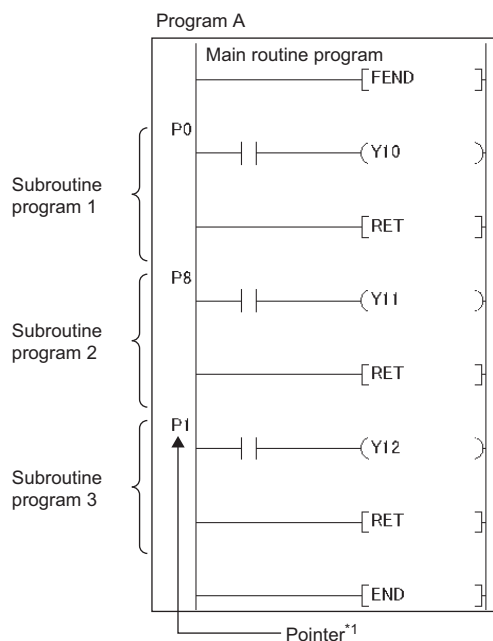
# 1.6 Program Type

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

## Subroutine program

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

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



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

### Point

- Making a program a standby type allows it to be managed as a separate program. (☞ Page 41 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 252 Nesting (N))

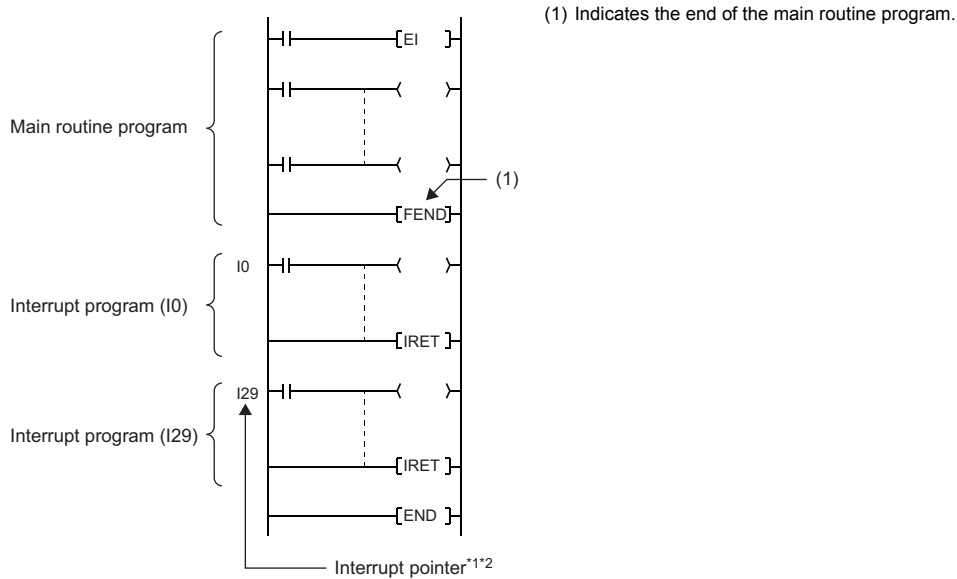
## Precautions

The following lists the precautions when using the subroutine program.

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

# Interrupt program

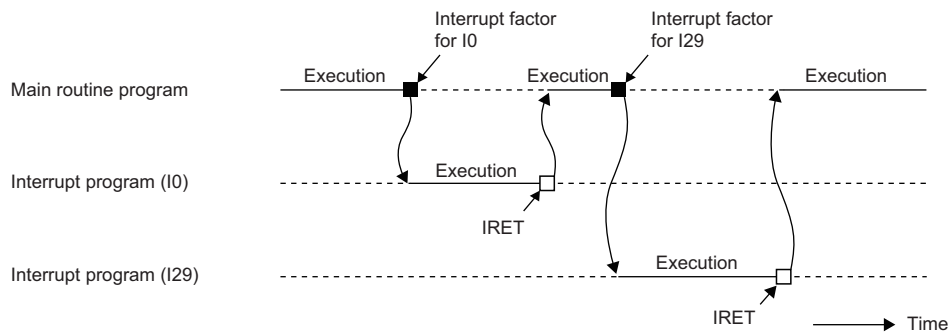
A program from an interrupt pointer (I) through the IRET instruction.



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

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

When an interrupt factor occurs, the interrupt program corresponding to the interrupt pointer number is executed (Page 257 The priority for the interrupt pointer numbers and interrupt factors) However, before that, interrupt must be enabled by executing the EI instruction.



**Point**

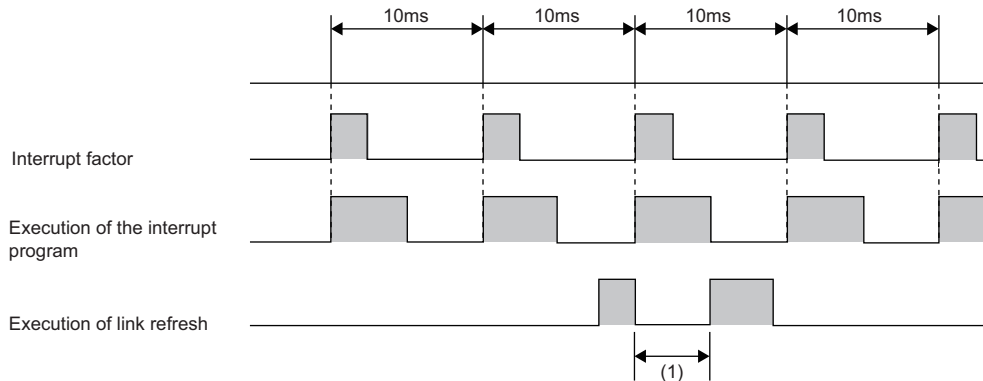
Making a program a standby type allows it to be managed as a separate program. (Page 41 Standby type program)

## Operation upon occurrence of an interrupt factor

The following shows the operation when an interrupt factor occurs.

### ■If an interrupt factor occurs during link refresh

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



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

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

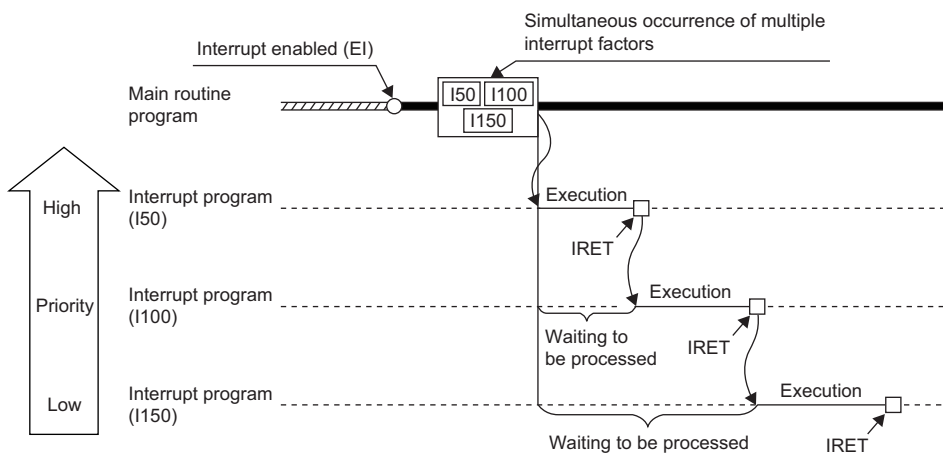
Executes the interrupt program for the interrupt factor.

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

If an interrupt such as a fixed scan execution type program (including an interrupt which triggers the event execution type program) is triggered while an interrupt program is being executed, the program operates in accordance with the interrupt priority. (☞ Page 84 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 87 Multiple interrupt execution sequence)

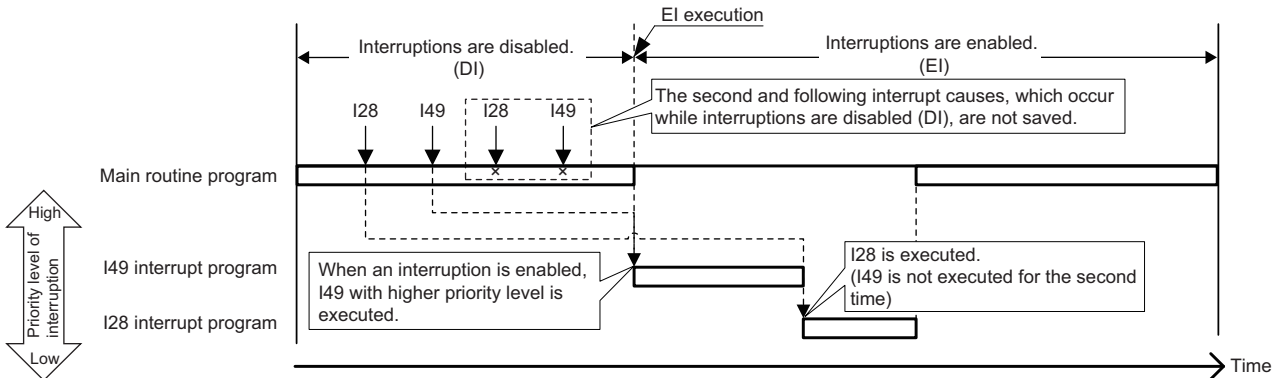




**■ If an interrupt factor occurs when interrupt is disabled (DI)**

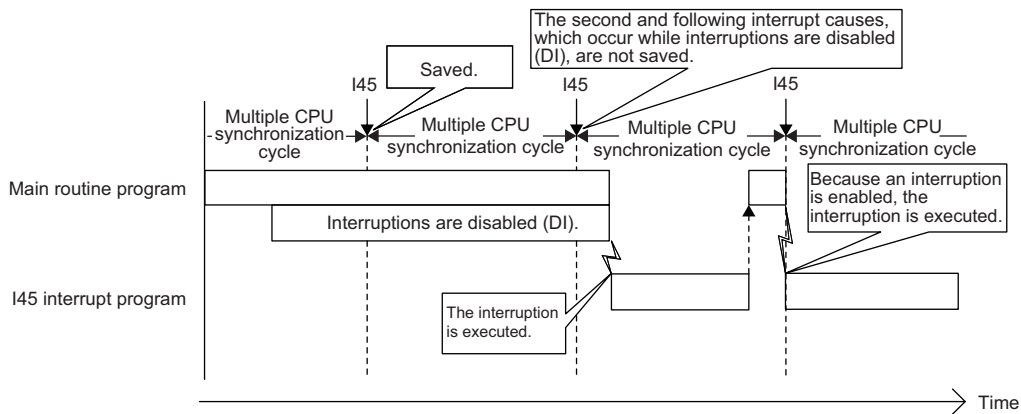
- For I0 to I15, I28 to I31, I48, I49, and I50 to I1023

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



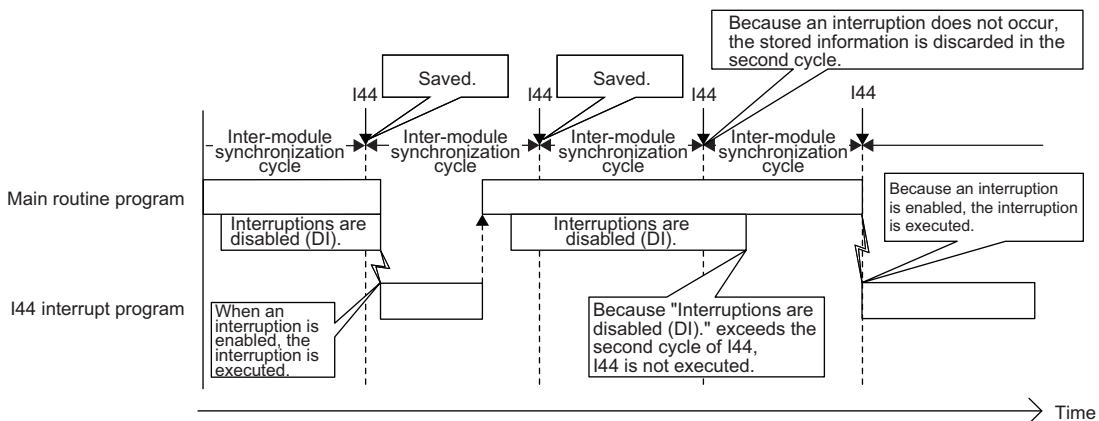
- For I45

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



- For I44

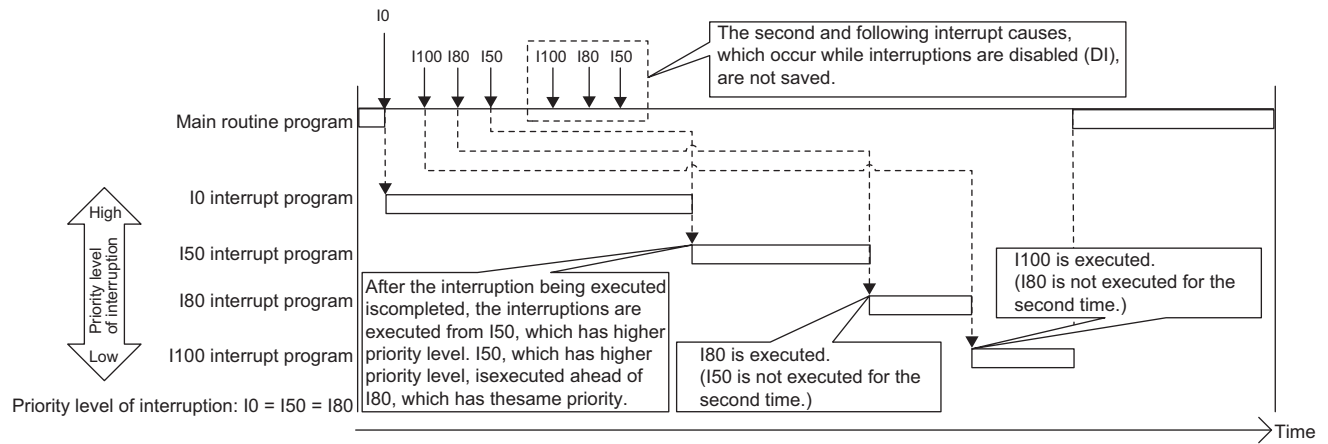
If interrupt is enabled before the next cycle, the I44 interrupt program will be executed when the interrupt is enabled. If interrupt continues to be disabled beyond the start of the next cycle (the second cycle), the memorized information will be discarded (even when the interrupt is enabled, the I44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



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

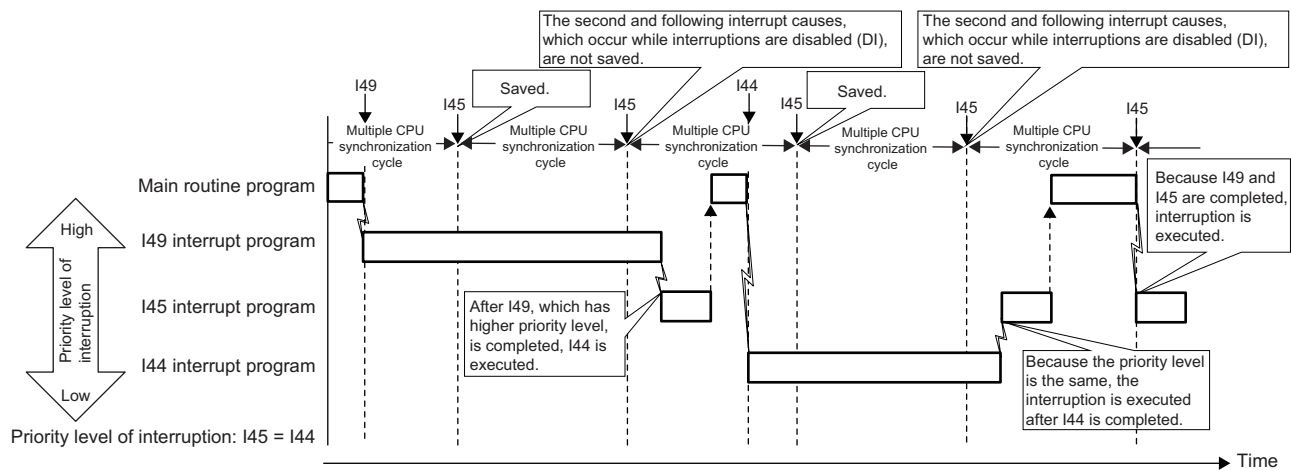
- For I0 to I15 and I50 to I1023

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



- For I45

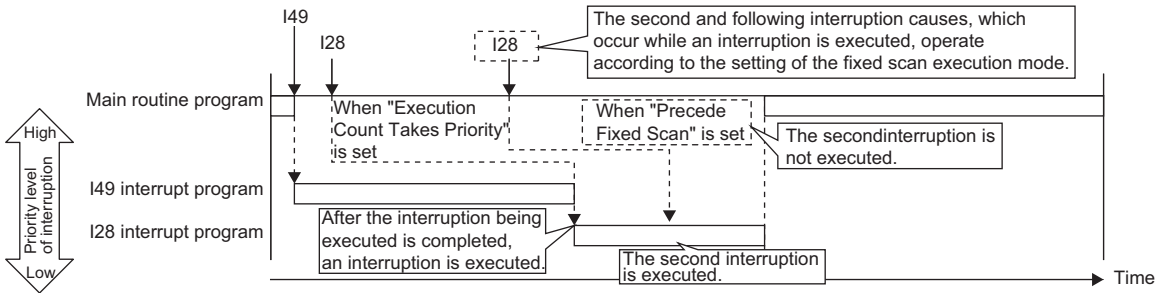
The occurred interrupt factor is memorized, and the I45 interrupt program will be executed after the running interrupt program finishes. Even if the I45 interrupt occurs multiple times, its interrupt factor will be memorized only once.



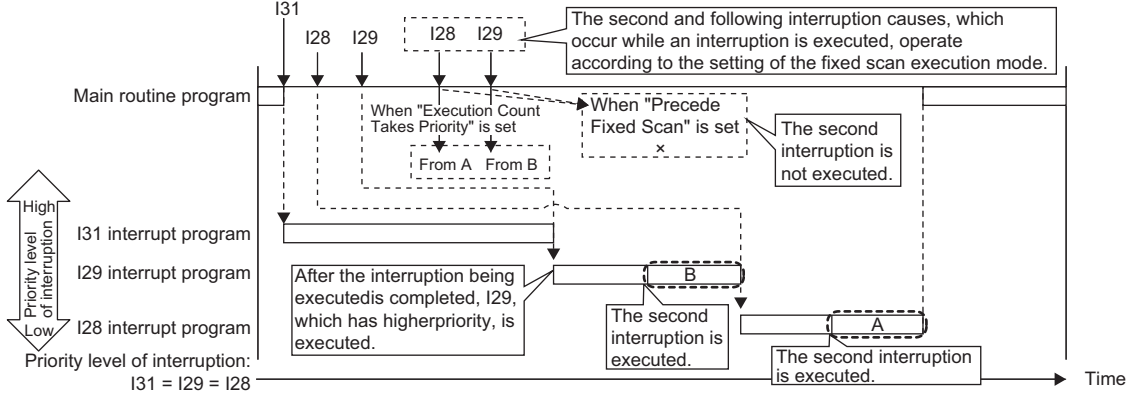
- For I28 to I31, I48, and I49

The occurred interrupt factor is memorized, and the interrupt program corresponding to the factor will be executed after the running interrupt program finishes. If the same interrupt factor occurs multiple times, it will be memorized once but operation at the second and later occurrences depends on setting of the fixed scan execution mode (Page 35 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 the interruption, which has lower priority level than the one being executed, occurs

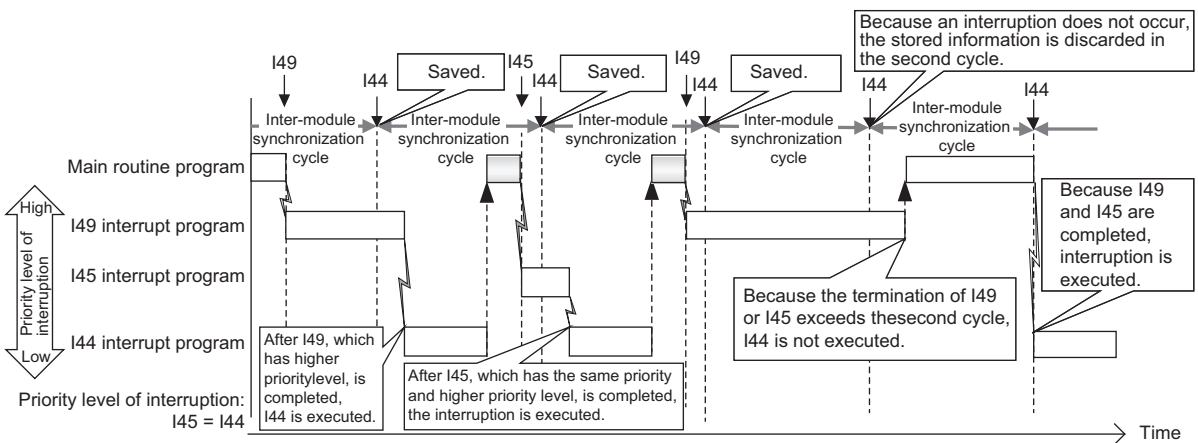


When the interruption, which has the same priority level as the one being executed, occurs



- For I44

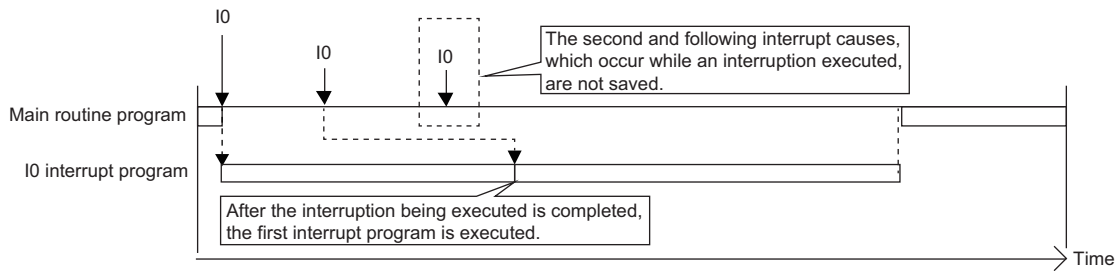
If the running interrupt program finishes before the next cycle, the I44 interrupt program will be executed when the running interrupt program finishes. If the running interrupt program continues beyond the start of the next cycle (the second cycle), the memorized information will be discarded (even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



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

- For I0 to I15 and I50 to I1023

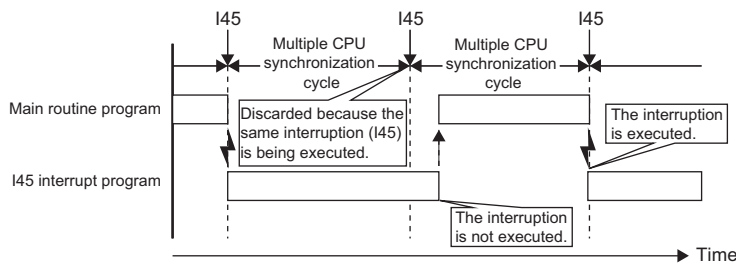
The interrupt factor that has occurred is memorized, and the interrupt program corresponding to the factor will be executed when the interrupt is enabled. Even if the same interrupt factor occurs multiple times, it will be memorized only once.



- For I45

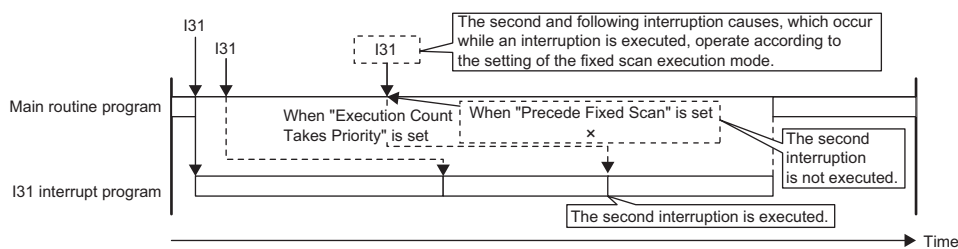
If an interrupt factor which is the same as that for the running interrupt program occurs, the factor is not memorized.

Therefore, the corresponding interrupt program will not be executed after the running interrupt program finishes. Also, if the I45 interrupt program for this cause cannot be executed, SM481 (Error flag for multiple CPU synchronous interrupt program (I45)) is turned on, and SD481 (Number of error occurrences for multiple CPU synchronous interrupt program (I45)) reaches its upper limit.



- For I28 to I31, I48, and I49

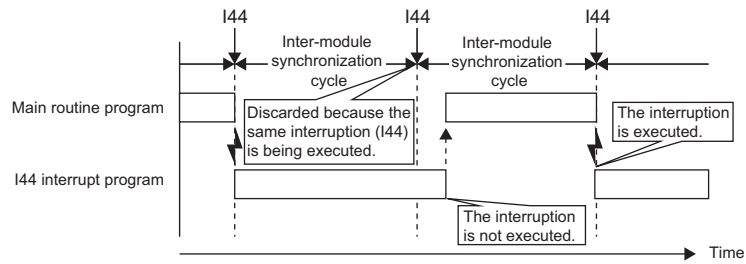
The occurred interrupt factor is memorized, and the interrupt program corresponding to the factor will be executed after the running interrupt program finishes. If the same interrupt factor occurs multiple times, it will be memorized once but operation at the second and later occurrences depends on setting of the fixed scan execution mode (Page 35 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.



- For I44

If an interrupt factor which is the same as that for the running interrupt program occurs, the factor is not memorized.

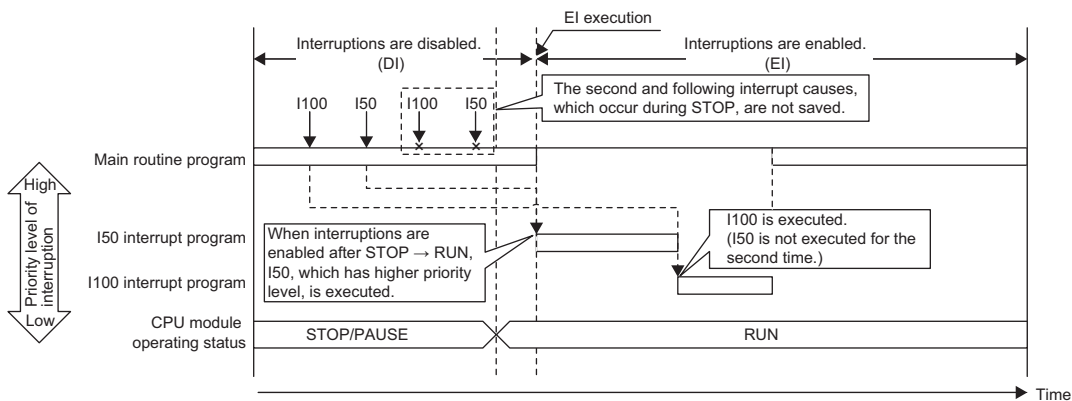
Therefore, the corresponding interrupt program will not be executed after the running interrupt program finishes. Also, if the I44 interrupt program for this cause cannot be executed, SM480 (Error flag for Inter-module synchronous interrupt program (I44)) is turned on, and SD480 (Number of error occurrences for inter-module synchronous interrupt program (I44)) reaches its upper limit.



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

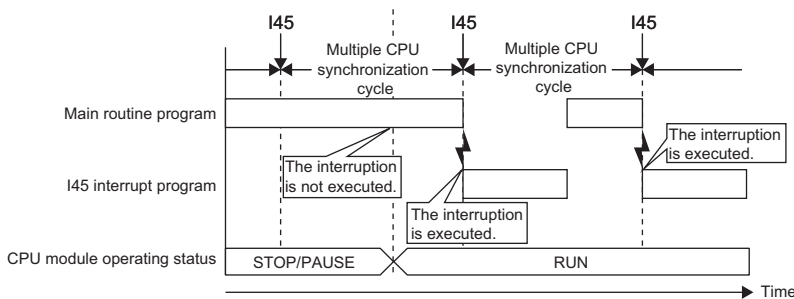
- For I0 to I15, I28 to I31, I48, I49, and I50 to I1023

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



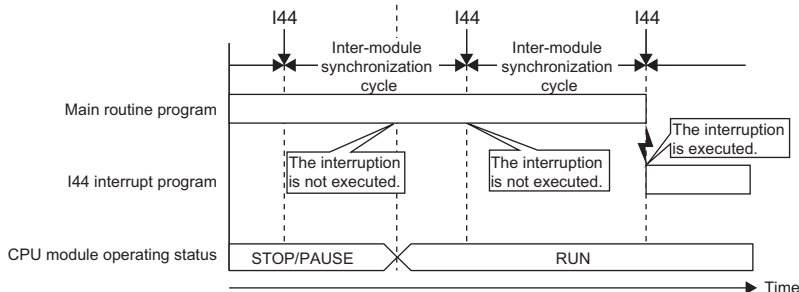
- For I45

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



- For I44

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

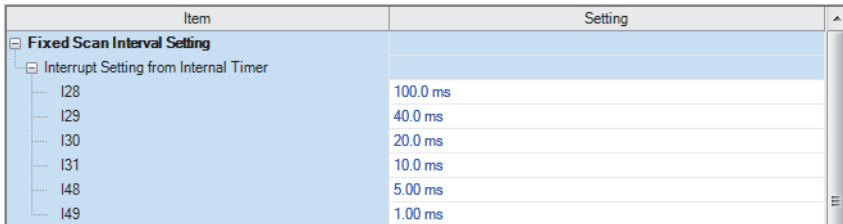


## Interrupt period setting

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

 [CPU Parameter] ⇒ "Interrupt Settings" ⇒ "Fixed Scan Interval Setting"

### Window




Item	Setting
Fixed Scan Interval Setting	
Interrupt Setting from Internal Timer	
I28	100.0 ms
I29	40.0 ms
I30	20.0 ms
I31	10.0 ms
I48	5.00 ms
I49	1.00 ms

### Displayed items

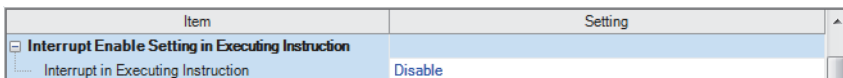
Item	Item	Description	Setting range	Default
Interrupt Setting from Internal Timer	I28	Sets the execution interval of I28.	0.5 to 1000ms (in units of 0.5ms)	100.0ms
	I29	Sets the execution interval of I29.		40.0ms
	I30	Sets the execution interval of I30.		20.0ms
	I31	Sets the execution interval of I31.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

## Interrupt enabled during instruction execution

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

 [CPU Parameter] ⇒ "Interrupt Settings" ⇒ "Interrupt Enable Setting in Executing Instruction"

### Window



Item	Setting
Interrupt Enable Setting in Executing Instruction	
Interrupt in Executing Instruction	Disable

### 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 56 Precautions)	<ul style="list-style-type: none"> <li>• Disable</li> <li>• Enable</li> </ul>	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

Item	Setting
<b>Block No. Save/Recovery Setting</b>	
File Register (R) Block No.	Save/Recover

### 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.	<ul style="list-style-type: none"><li>• Not Saved/Recovered</li><li>• Save/Recover</li></ul>	Save/Recover

### Point

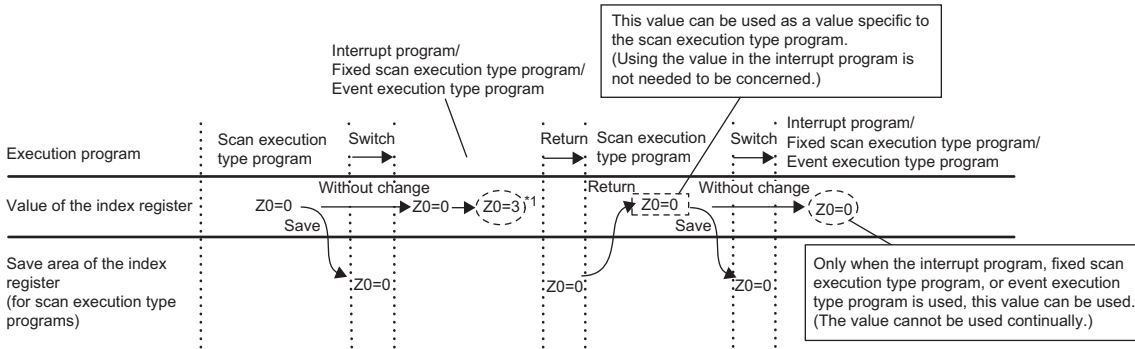
When the block number of the file register (R) is not changed during execution of an interrupt program, enabling "Not Saved/Recovered" can reduce the startup time and the termination time of the interrupt program ( Page 359 Overhead time when executing the interrupt program). To change the block number 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.



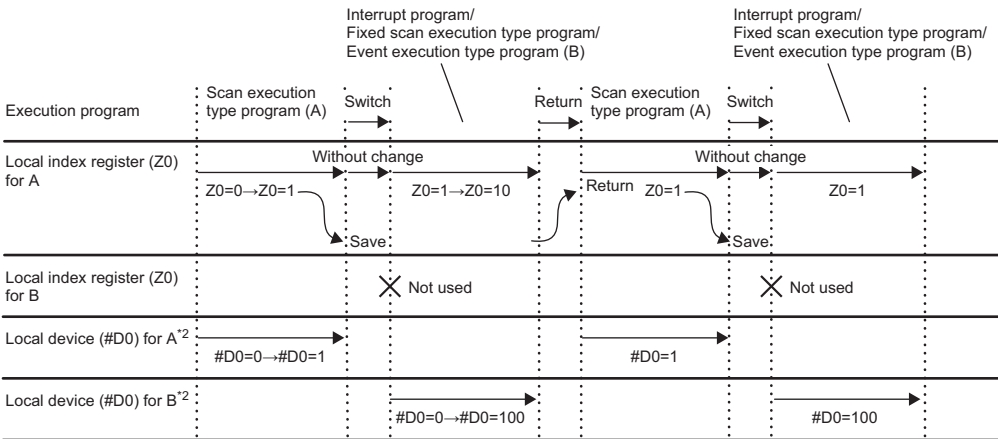
### ■ Saving/restoring of the index register (Z, LZ)

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 interruption 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



- Operations of the local index register and local device

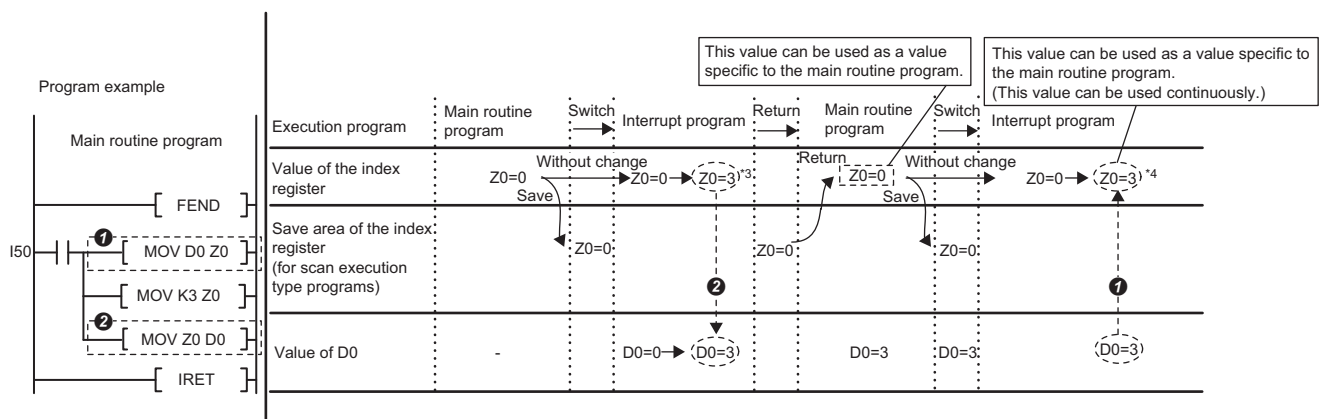


\*1 The value of Z0 is changed by the interrupt program, fixed scan execution type program, or event execution type program.

\*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 ① and ② are not added.

## Precautions

The precautions for the interrupt program are mentioned below.

### ■Restrictions on program creation

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

### ■Data inconsistency

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


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

### ■If the interrupt accuracy does not increase


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

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


### ■Monitoring the interrupt program execution time

The execution time of the interrupt program can be monitored with the interrupt program in which the inter-module synchronous interrupt (I44) and multiple CPU synchronous interrupt (I45) are used (  Page 179 Error Detection Setting)

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

If the execution interval of the interrupt program is short, the start-up of the interrupt program may be delayed when the instruction with a long processing time is executed, the online ladder block change is performed, or a Q series module is accessed simultaneously from multiple CPU modules using instructions in the multiple CPU system configuration. Therefore, when the execution time is monitored for the interrupt program that uses the inter-module synchronous interrupt (I44) and multiple CPU synchronous interrupt (I45), an error may be detected in the CPU module (  Page 179 Error Detection Setting)

### ■Consideration when using the data logging function

To use occurrence of an interrupt as a data collection condition, the user must specify the fixed scan interval of the interrupt program, taking into consideration the processing time of the data logging function (  Page 165 Using together with interrupt programs)

# 2 OPERATION PROCESSING ACCORDING TO OPERATING STATUS OF THE CPU MODULE

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

- RUN state
- STOP state
- PAUSE state

## 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 114 OUTPUT MODE AT OPERATING STATUS CHANGE (STOP TO RUN))

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

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

## Operation processing in STOP state

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

### ■Output at the time of entering STOP state

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

## Operation processing in PAUSE state

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



# 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 watchdog timer error. (☞ Page 73 Scan Time Monitoring Time Setting)

## **Power failure exceeding the allowable momentary power failure time**

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

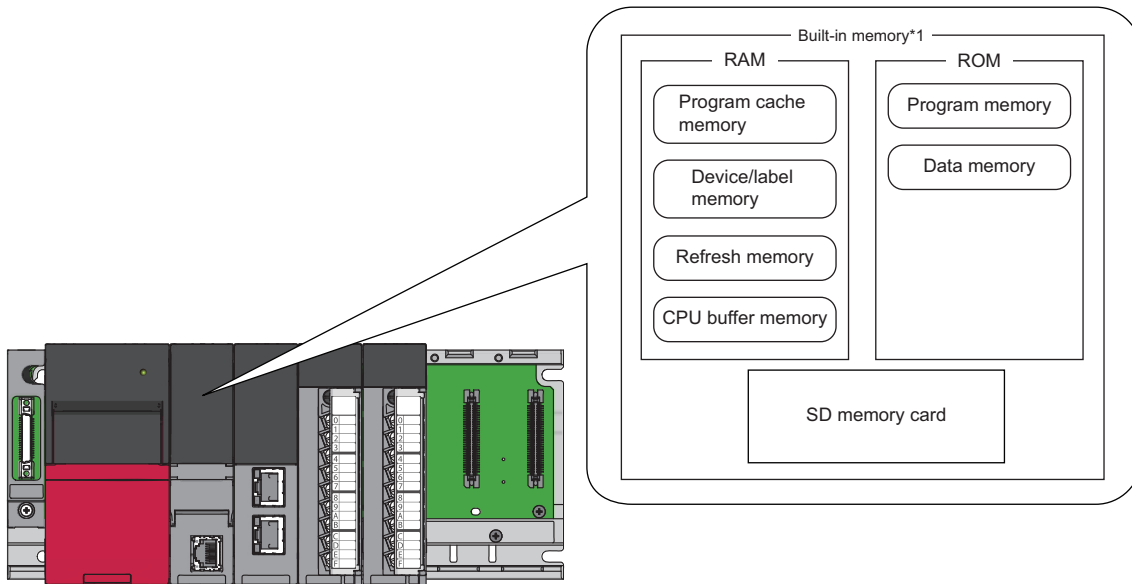
# MEMO

---

# 4 MEMORY CONFIGURATION OF THE CPU MODULE

## 4.1 Memory Configuration

The following shows the memory configuration of the CPU module.



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



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

GX Works3 Operating Manual

# Program memory/program cache memory

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

\*1 This memory is used for program operations.

- Powered-on
- Reset operation

## Application

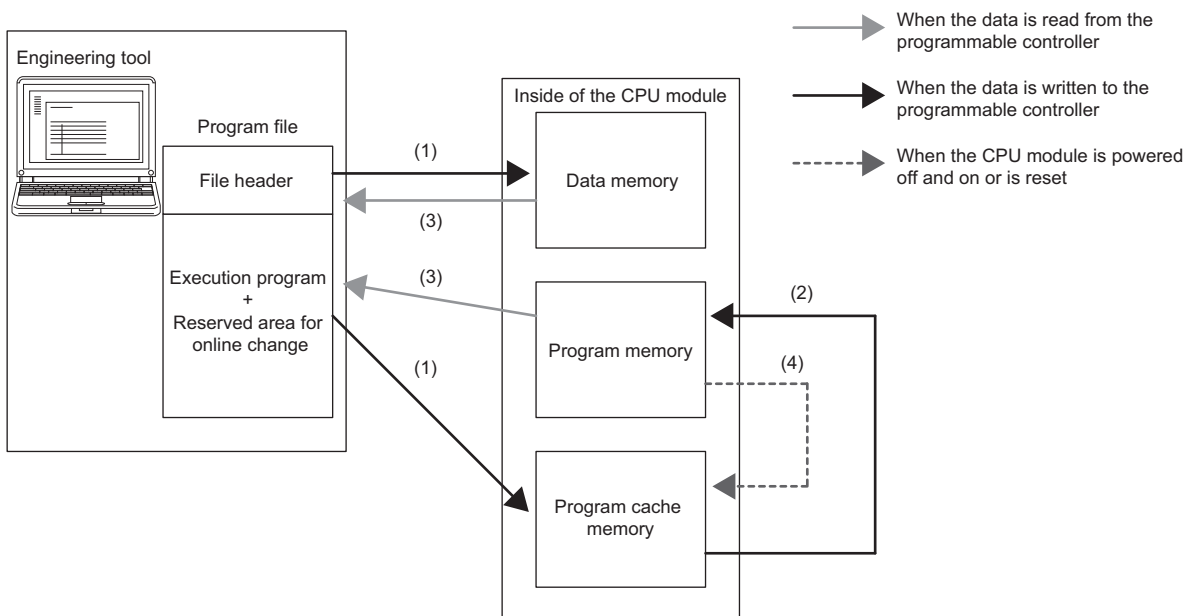
The program memory and program cache memory store execution programs of program files and FB (function block) files. Execution programs include instruction codes, statements, and notes.

## Data to be allocated

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

## Data allocation and procedure of read/write operations

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



- (1) At writing to the programmable controller, the execution program and steps reserved for online program change are written to the program cache memory, and the file header is written to the data memory.
- (2) After being written into the program cache memory, data is automatically transferred to the program memory.
- (3) At reading from the programmable controller, the execution program and steps reserved for online program change are read from the program memory, and the file header is read from the data memory.
- (4) At powering off and on or resetting the CPU module, the data in the program memory is transferred to the program cache memory and operations are executed.

### Point

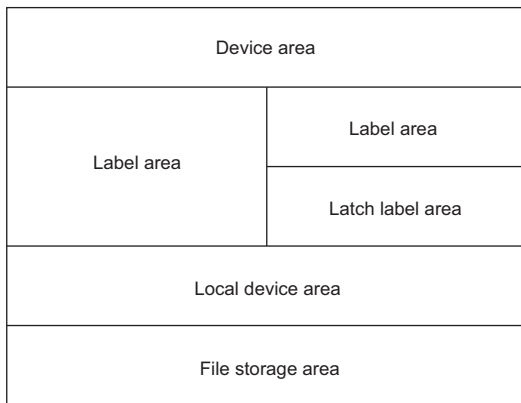
For read/write operations from/to the programmable controller, refer to the following.

GX Works3 Operating Manual



# Device/label memory

The device/label memory has the following areas.



## Data to be allocated

The following table lists the data allocated to each area.

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

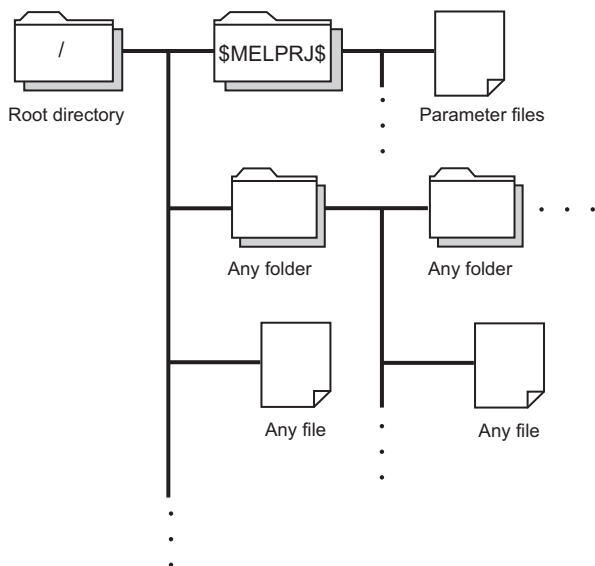
\*1 File register files which are stored in the area for storing file register files can be written or read in file unit.



Free space of areas can be checked in "Device/Label Memory Area Capacity Setting". (Page 108 Device/Label Memory Area Setting).

## Data memory

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




### Point


For details on how to create and delete user folders, refer to the following.

 GX Works3 Operating Manual

## Refresh memory

This memory is used to store data used to refresh communication with the intelligent function module. ( Page 251 Refresh Data Register (RD)).

## CPU buffer memory

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

## SD memory card

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

## 4.2 File Size Unit in Memory

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

CPU module	CPU built-in memory			SD memory card
	Program memory	Device/label memory	Data memory	
R04CPU	4 bytes	512 bytes	1024 bytes	32K bytes
R08CPU			2048 bytes	
R16CPU			4096 bytes	
R32CPU			8192 bytes	
R120CPU			16384 bytes	
R08PCPU	128 bytes		2048 bytes	
R16PCPU			4096 bytes	
R32PCPU			8192 bytes	
R120PCPU			16384 bytes	


4

### Point

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

## 4.3 Memory Operation

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

 GX Works3 Operating Manual

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

### Point

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

# 4.4 Files

This section lists the files used by the CPU module.

## File types and storage memory

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

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

File type		CPU built-in memory			SD memory card	File name and extension
		Program memory	Device/label memory	Data memory		
		Drive 0	Drive 3	Drive 4	Drive 2	
Program		◎ <sup>*4</sup>	×	◎ <sup>*4</sup>	○	ANY_STRING.PRG
FB file		○ <sup>*4</sup>	×	○ <sup>*4</sup>	○	ANY_STRING.PFB
CPU parameter		×	×	◎	○	CPU.PRM
System parameter		×	×	◎	○	SYSTEM.PRM
Module parameter		×	×	○	○	UNIT.PRM
Module extension parameter		×	×	○	○	• UEXmmmnn.PRM <sup>*1</sup> • UEXmmm00.PPR <sup>*5</sup>
Module-specific backup parameter		×	×	○	○	UBPmmmmnn.BPR <sup>*1</sup>
Memory card parameter		×	×	×	○	MEMCARD.PRM
Device comment		×	×	○	○	ANY_STRING.DCM
Initial device value		×	×	○	○	ANY_STRING.DID
Global label setting file		×	×	○	○	GLBLINF.IFG
Initial label value file	Initial global label value file	×	×	○	○	GLBLINF.LID
	Initial local label value file	×	×	○	○	PROGRAM_NAME.LID
File register		×	○	×	○ <sup>*3</sup>	ANY_STRING.QDR
Event history		×	×	○	○	EVENT.LOG
Device data storage file		×	×	○	○ <sup>*3</sup>	DEVSTORE.QST
General-purpose data		×	×	○	○	ANY_STRING.CSV/BIN
Data logging setting file	Common setting file	×	×	×	○	LOGCOM.LCS
	Individual setting file	×	×	○	○	LOGnn.LIS <sup>*2</sup>
Memory dump setting file		×	×	○	○ <sup>*3</sup>	MEMDUMP.DPS
Remote password		×	×	○	○	00000001.SYP

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

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

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

\*4 When a program or a FB (function) program is stored in the built-in memory of the CPU module, it is divided for the program memory and the data memory.

\*5 Module extension parameter for the protocol setting, storing protocol setting information in the predefined protocol support function

# File operation available

The following lists file operations which are available for each file type.

○: Available, —: N/A

File type		Operation from engineering tool			Operation with SLMP and FTP server function			Operation via instruction	
		Write	Read	Delete	Write	Read	Delete	Write	Read
Program		○*1	○	○*3	○*2*5	○	○*2*5	—	—
FB file		○*3	○	○*3	○*2*5	○	○*2*5	—	—
Parameter		○*3	○	○*3	○*2*5	○	○*2	—	—
Device comment		○	○	○*3	○*2*5	○	○*2*5	—	—
Initial device value		○	○	○*3	○*2*5	○	○*2*5	—	—
Global label setting file		○*3	○	○*3	○*2*5	○	○*2*5	—	—
Initial label value file	Initial global label value file	○	○	○*3	○*2	○	○*2*5	—	—
	Initial local label value file	○	○	○*3	○*2	○	○*2*5	—	—
File register		○	○	○*3	○	○	○*2	—	—
Event history		—	—	—	○	○	○	—	—
Device data storage file		—	—	—	○*2*5	○	○*2*5	○	○
General-purpose data		○	○	○	○	○	○	○	○
Data logging setting file	Common setting file	○*4	○*4	○*4	○	○	○	—	—
	Individual setting file	○*4	○*4	○*4	○	○	○	—	—
Memory dump setting file		○	○	○	○	○	○	—	—
Remote password		○*3	○	○*3	○*2*5	○	○*2*5	—	—
Module-specific backup parameter		—	—	○*2	—	—	○*2	—	—

- \*1 Done during write to the programmable controller when the CPU module operation status is STOP, and during online program change when the status is RUN. In addition, when the operating status of the CPU module is RUN, only the program that is registered with the parameter can perform write operation.
- \*2 Available only when the CPU module operation status is STOP/PAUSE. A communication error occurs when operated in the RUN state.
- \*3 Available only when the CPU module operation status is STOP/PAUSE. When a program tries to perform memory operation while the CPU module is RUN, the operation is continued after the operation status is changed through remote STOP.
- \*4 Operation on CPU Module Logging Configuration Tool.
- \*5 When the operation target is the SD memory card, the operation can be performed even while the CPU module is RUN.

## File size

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

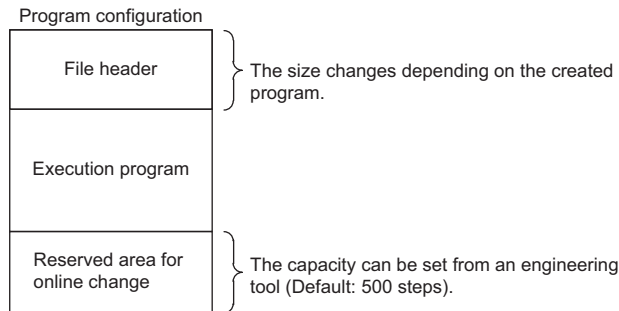
File type		File size
Program		Approx. 4050 bytes minimum (only END instruction + 500 steps reserved for online program change)
FB file		Approx. 4900 bytes minimum (non-processing FB + 500 steps reserved for online program change)
CPU parameter		744 bytes minimum
System parameter		112 bytes minimum
Module parameter		The size differs depending on the module used. For example, the size is 1036 bytes at minimum when the Ethernet function is used.
Module extension parameter		The size differs depending on the module used. For example, the size is 65572 bytes when the predefined protocol support function is used with the Ethernet function.
Module-specific backup parameter		The size differs depending on the module used. For details, refer to the 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. 140 bytes minimum (When the initial value of a word device is set)
Global label setting file		Approx. 660 bytes minimum (When a word-type global label is set)
Initial label value file	Initial global label value file	Approx. 140 bytes minimum (When the initial value of a word type global label is set)
	Initial local label value file	
File register		2048 bytes minimum (when the setting capacity is 1K words)
Event history		1024 bytes minimum (when the setting capacity is 1K words (default setting is 131072 bytes))
Device data storage file		2048 bytes minimum (when the setting capacity is 1K words)
General-purpose data		The size differs depending on the size of the file to be written.
Data logging setting file	Common setting file	142 bytes (When the setting is configured so that one-word data is collected in a binary file at each scan in continuous logging)
	Individual setting file	1192 bytes (When the setting is configured so that one-word data is collected in a binary file at each scan in continuous logging only for setting No.1)
Memory dump setting file		1224 bytes (when the save file name is "MEMDUMP_00")
Remote password		224 bytes minimum

## Configuration of a program file

The following figure shows the configuration of a program.

### ■ Configuration of a program

This FB file includes the file header, execution program, and steps reserved for online program change.

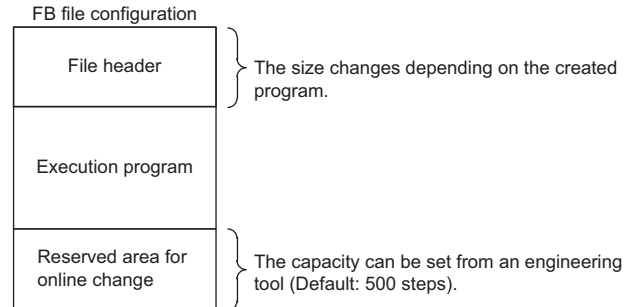


When the program is stored in the CPU module, execution program and steps reserved for online program change are allocated to the program memory. The file header is allocated to the data memory. The following table shows the destination memory to which each area is allocated when a program is stored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

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

### ■ FB file

This FB file includes the file header, execution program, and steps reserved for online program change.



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





# PART 2 FUNCTIONS

Part 2 consists of the following chapters.

5 SCAN MONITORING FUNCTION

---

6 CLOCK FUNCTION

---

7 ONLINE CHANGE

---

8 INTERRUPT FUNCTION

---

9 PID CONTROL FUNCTION

---

10 PROCESS CONTROL FUNCTION

---

11 CONSTANT SCAN

---

12 DATABASE FUNCTION

---

13 REMOTE OPERATION

---

14 DEVICE/LABEL MEMORY AREA SETTING

---

15 INTERNAL BUFFER CAPACITY SETTING

---

16 DEVICE/LABEL INITIAL VALUE SETTINGS

---

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

---

18 LABEL ACCESS SETTING FROM EXTERNAL DEVICE

---

19 ROUTING SETTING

---

20 LATCH FUNCTION

---

21 MEMORY CARD FUNCTION

---

22 DEVICE/LABEL ACCESS SERVICE PROCESSING SETTING

---

23 DATA LOGGING FUNCTION

---

24 MEMORY DUMP FUNCTION

---

25 REAL-TIME MONITOR FUNCTION

---

26 RAS FUNCTIONS

---

27 MULTIPLE CPU SYSTEM FUNCTION

---

28 SECURITY FUNCTION

---


# 5 SCAN MONITORING FUNCTION

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

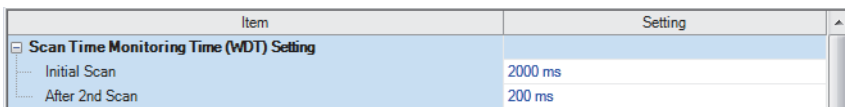
- Initial scan (first scan)
- Second and later scans

## 5.1 Scan Time Monitoring Time Setting

Sets the scan time monitoring time.

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

### Window



Item	Setting
Scan Time Monitoring Time (WDT) Setting	
Initial Scan	2000 ms
After 2nd Scan	200 ms

### 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

## 5.2 Watchdog Timer Reset

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

## 5.3 Precautions

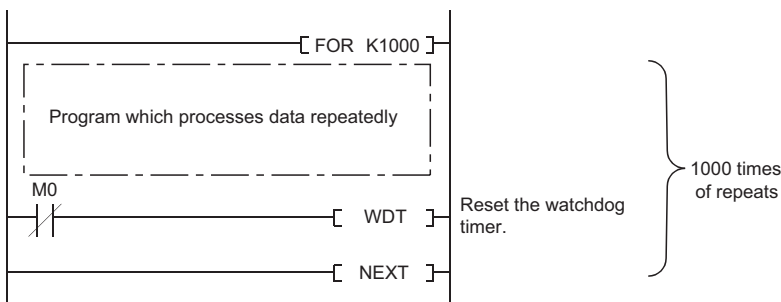
The following lists the precautions on the scan monitoring function.

### Measurement error of watchdog timer

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

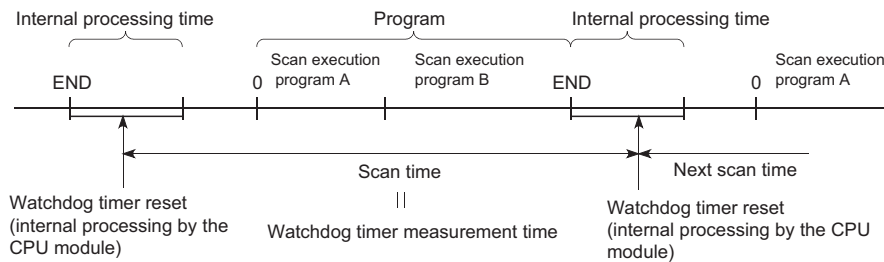
### Resetting the watchdog timer when repeatedly executing a program

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



### Scan time when the WDT reset instruction is used

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



# 6 CLOCK FUNCTION

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

## 6.1 Time Setting

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

### Changing the clock data

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


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

#### Point

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

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


#### Using the engineering tool

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

#### Using SM/SD

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

#### Using the instruction

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

### Reading the clock data


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

- Using SM/SD
- Using the instruction

#### Using SM/SD

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

#### Using the instruction

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

## Precautions on the clock data

The following lists the precautions on the clock data.

### When using the product for the first time

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

### Modifying the clock data

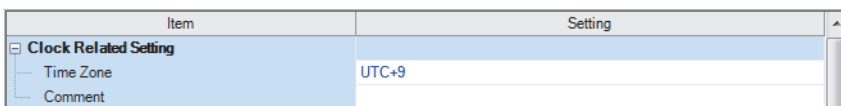
Even if a portion of the clock data is changed, be sure to write all the data to the CPU module again.

## 6.2 Setting Time Zone

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

 [CPU Parameter] ⇒ "Operation Related Setting" ⇒ "Clock Related Setting"

### Window



### Displayed items


Item	Description	Setting range	Default
Time Zone	Sets the time zone used by the CPU module.	<ul style="list-style-type: none"> <li>• UTC+13</li> <li>• UTC+12</li> <li>• UTC+11</li> <li>• UTC+10</li> <li>• UTC+9:30</li> <li>• UTC+9</li> <li>• UTC+8</li> <li>• UTC+7</li> <li>• UTC+6:30</li> <li>• UTC+6</li> <li>• UTC+5:45</li> <li>• UTC+5:30</li> <li>• UTC+5</li> <li>• UTC+4:30</li> <li>• UTC+4</li> <li>• UTC+3:30</li> <li>• UTC+3</li> <li>• UTC+2</li> <li>• UTC+1</li> <li>• UTC</li> <li>• UTC-1</li> <li>• UTC-2</li> <li>• UTC-3</li> <li>• UTC-3:30</li> <li>• UTC-4</li> <li>• UTC-4:30</li> <li>• UTC-5</li> <li>• UTC-6</li> <li>• UTC-7</li> <li>• UTC-8</li> <li>• UTC-9</li> <li>• UTC-10</li> <li>• UTC-11</li> <li>• UTC-12</li> </ul>	UTC+9
Comment	Enters a comment for a time zone (e.g., name of the city).	32 characters or less	—

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

## 6.3 System Clock


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

### Special relay used for system clock

Special relay used for system clock are as follows (  Page 322 System clock)

SM number	Name
SM400	Always On
SM401	Always Off
SM402	After RUN, On for one scan only
SM403	After RUN, Off for one 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 (ms) clock
SM420	User timing clock No.0
SM421	User timing clock No.1
SM422	User timing clock No.2
SM423	User timing clock No.3
SM424	User timing clock No.4
SM440	On only initial I44 execution after RUN
SM441	On only initial I45 execution after RUN

### Special register used for system clock

Special register used for system clock are as follows (  Page 346 System clock)

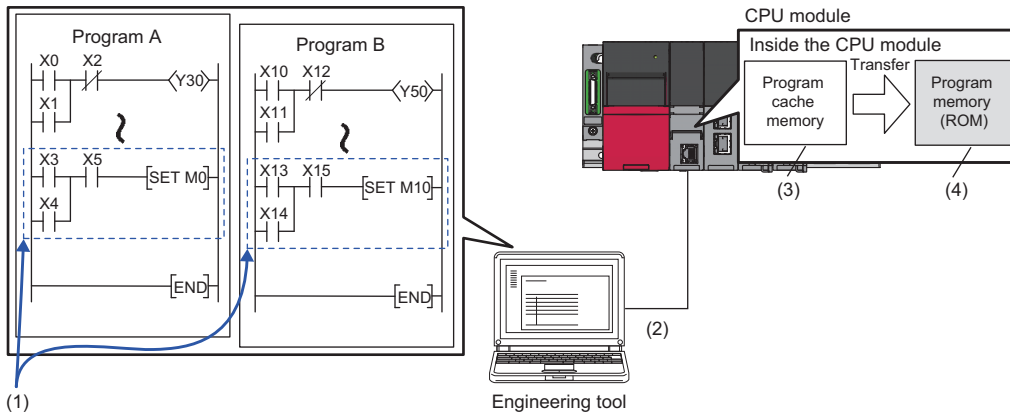
SD number	Name
SD412	One second counter
SD414	2n second clock setting
SD415	2n (ms) clock setting
SD420	Scan counter

# 7 ONLINE CHANGE

This chapter describes online change.

## 7.1 Online Ladder Block Change

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



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

### Point

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

GX Works3 Operating Manual

## Editable contents

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

### Editable contents within a program block

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

### Editable contents within FB definition

- To add local label (VAR) definition, 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.
- 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\*1, 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 Here, the subroutine-type FB and FUN interface information means:

- All FB definition, FUN definition, and their definition numbers contained in FB files
- Definition names
- The number of all input labels, and their data types
- The number of all output labels, and their data types
- The number of all I/O labels, and their data types
- The number of public local labels in FB definition and their data types
- The implementation method of FB definition



## Editable contents within FUN definition

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

## Editable contents for each program component

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

## Range changeable in a single session

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

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

## Steps reserved for online ladder block change

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

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

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

## Online ladder block change during the boot operation

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

# Setting the initial value for registering/changing label definition

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

## Initial value setting availability

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

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

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

\*1 By changing the data type, the initial value can be set only when a new area is assigned again. The initial value can be set for the following cases:

- Changing the data type
- Changing the latch setting
- Changing the constant setting (changing from constant to variable)

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

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

### Point

When using GX Works3 version 1.000A

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

## Precautions

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

### Prohibited operation at online ladder block change and writing data

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

### Operation from engineering tools

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

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

### When deleting OUT instruction which is on

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

### Program file not registered in program setting

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

## Instructions which do not operate correctly

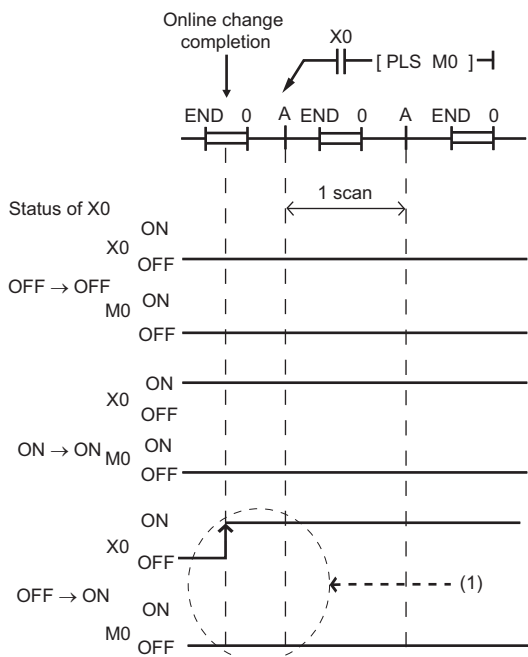
This section describes instructions which do not work correctly during online ladder block change or write to the programmable controller.

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

### ■Rising instruction

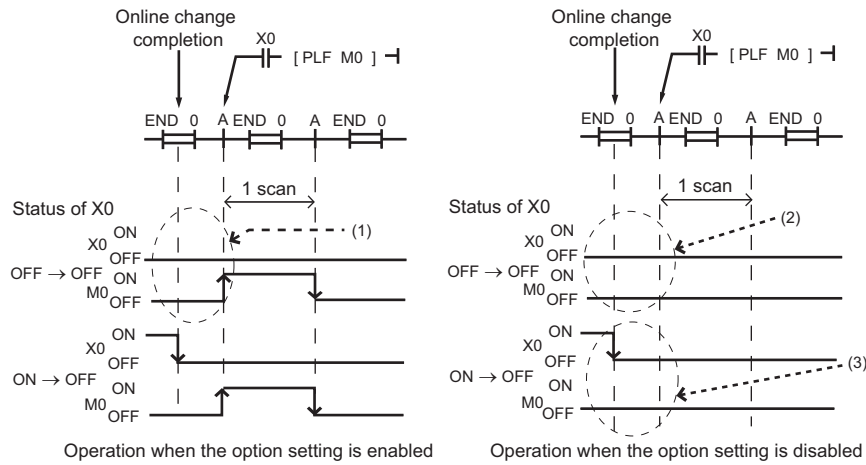
When a rising instruction exists within the range to be changed, the rising instruction will not be executed even if the execution condition (OFF to ON) is satisfied after completion of online ladder block change or write to the programmable controller.

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



## ■Falling instruction

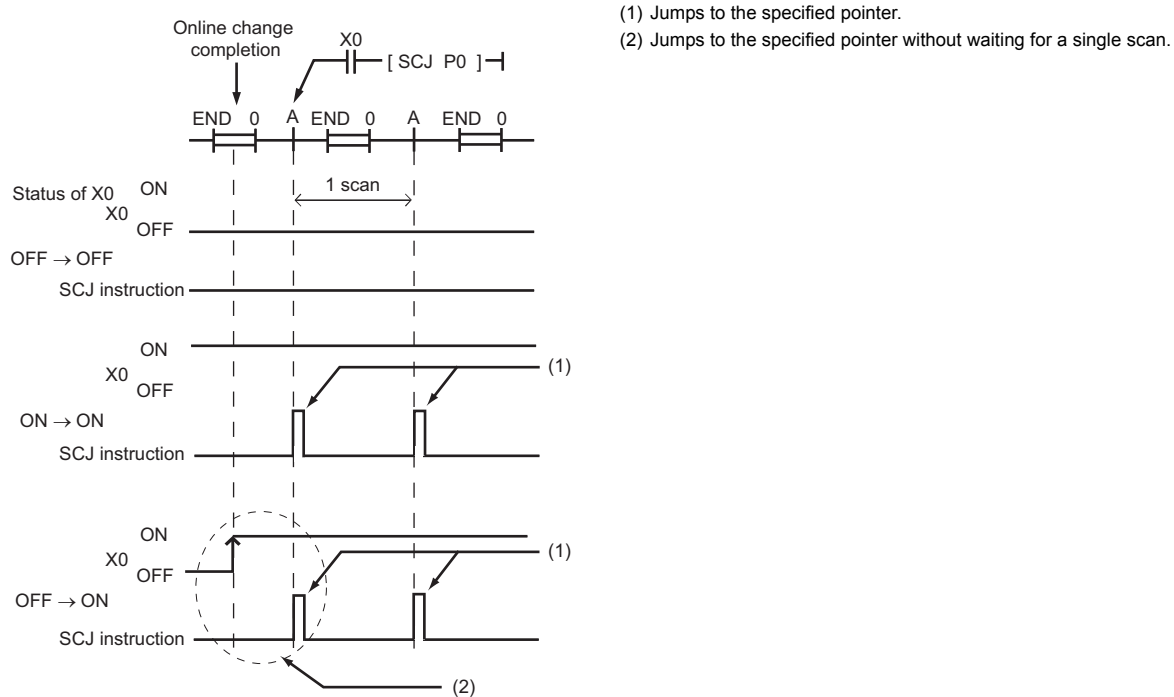
When a falling instruction exists within the range to be changed, the falling instruction will not be executed even if the execution condition (ON to OFF) is satisfied after completion of online ladder block change or write to the programmable controller.



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

## ■SCJ instruction

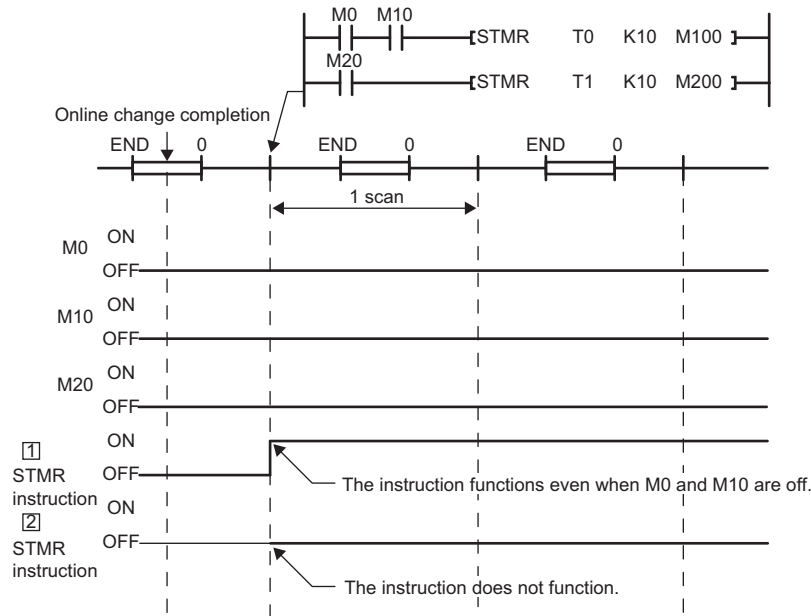
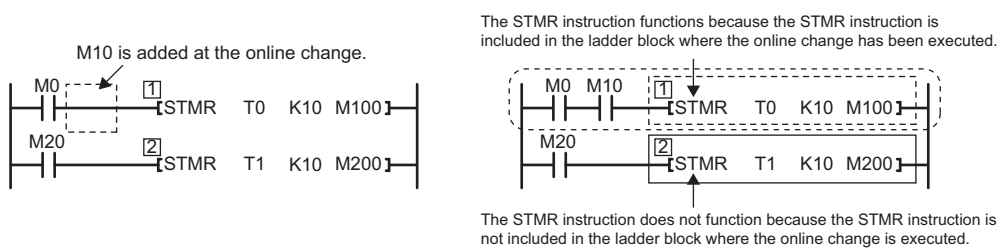
If an SCJ instruction exists within the range to be changed and the execution condition is taken at the completion of online ladder block change or write to the programmable controller completion, the program will jump without waiting for a single scan.



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

## ■STMR instruction

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



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

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

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

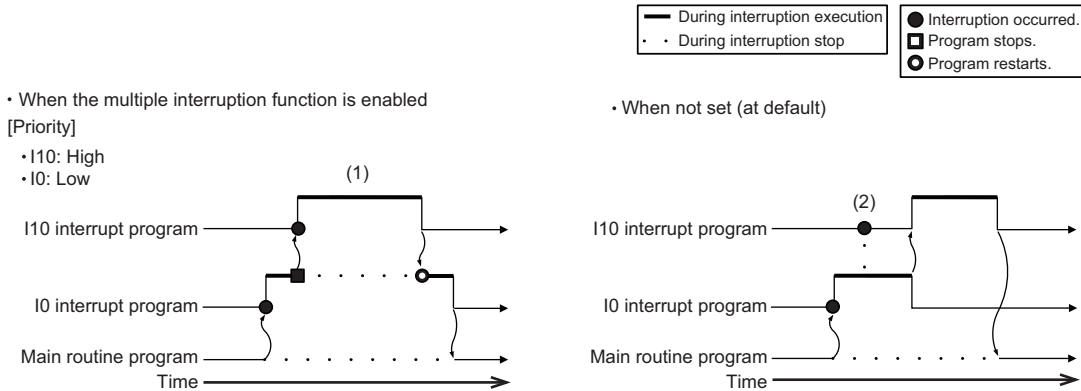
When the online ladder block change is performed, the start-up of the interrupt program may be delayed. Therefore, when the execution time is monitored for the interrupt program that uses the inter-module synchronous interrupt (I44) and multiple CPU synchronous interrupt (I45), an error may be detected in the CPU module. (Page 179 Error Detection Setting)

# 8 INTERRUPT FUNCTION

This chapter describes the interrupt function.

## 8.1 Multiple Interrupt Function

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



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

### Interrupt priority

If the interrupt priority of a program for which its execution condition has been satisfied is higher than that of the running program, the programs are executed in accordance with their interrupt priority. If the interrupt priority of the new program is the same or lower, it enters the waiting status until the running program finishes. The interrupt priorities 1 to 4 listed below cannot be changed, whereas the interrupt priorities 5 to 8 can. (👉 Page 85 Interrupt priority setting)

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

## Interrupt priority setting

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

[CPU Parameter] ⇒ "Interrupt Settings" ⇒ "Interrupt Priority Setting from Module"

### Operating procedure

"Interrupt Settings" window

Item	Setting
Interrupt Priority Setting from Module	
Multiple Interrupt	Enable
Interrupt Priority	<Detailed Setting>

"Detailed Setting" window

Interrupt Pointer	Priority
I0	8
I1	8
I2	8
I3	8
I4	8
I5	8
I6	8
I7	8
I8	8
I9	8
I10	8
I11	8
I12	8
I13	8
I14	8
I15	8

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.	<ul style="list-style-type: none"> <li>• Disable</li> <li>• Enable</li> </ul>	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.

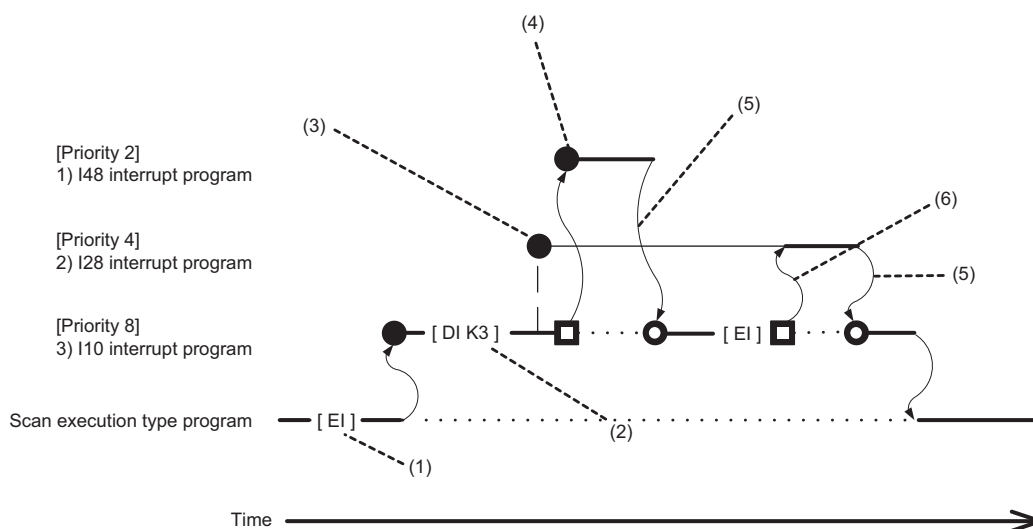
Ex.

Order of interrupt occurrence: ③ → ② → ①

Order of interrupt execution: ③ → ① → ②

Order of interrupt completion: ③ → ① → ②

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



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

### Point

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



# Multiple interrupt execution sequence

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

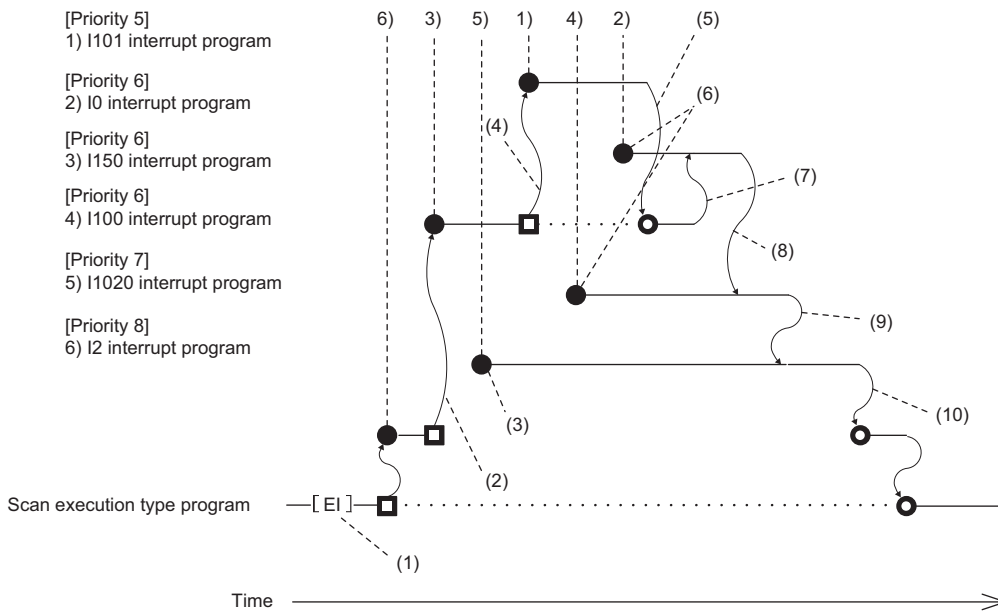
Ex.

Order of interrupt occurrence: 6 → 3 → 5 → 1 → 4 → 2,

Order of interrupt execution: 6 → 3 → 1 → 2 → 4 → 5

Order of interrupt completion: 1 → 3 → 2 → 4 → 5 → 6

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



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

# Precautions

---

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

## Delay in interrupt programs

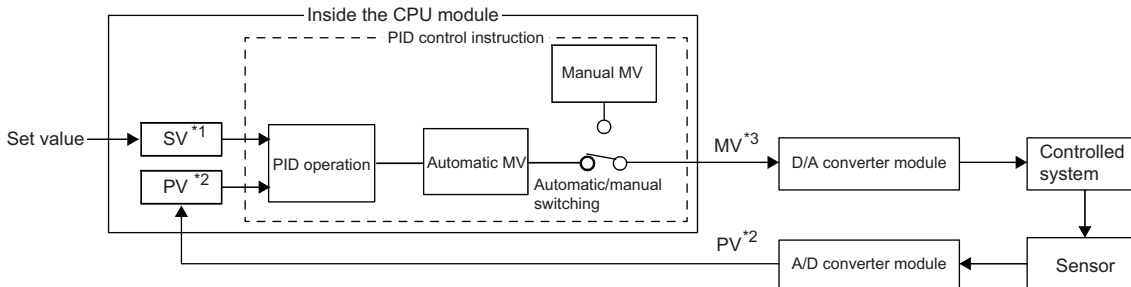
Completing low-priority interrupt programs may take time if high-priority interrupt programs are executed frequently. Adjust the processing of high-priority interrupt programs to allow execution of low-priority interrupts.

## Devices and labels used by interrupt programs

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

# 9 PID CONTROL FUNCTION

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



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

In the PID control, the PID operation is executed to compare the value set beforehand (SV) with the digital value (process value (PV)) in which the analog value measured by a sensor is read from the A/D converter module.

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

## Point

PID control is performed by using the PID control instruction. For details on the instruction specifications, PID control, and programming, refer to the following.

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

## Point

Exact differential is a PID control in which the input derivative term is used as it is. Inexact differential is a PID control in which a primary delay filter is added to the input derivative term, and is useful for the following cases:

- The control is easily affected by high-frequency noise.
- For a step-like change with the exact differential method, there is not enough energy to operate the final control element.

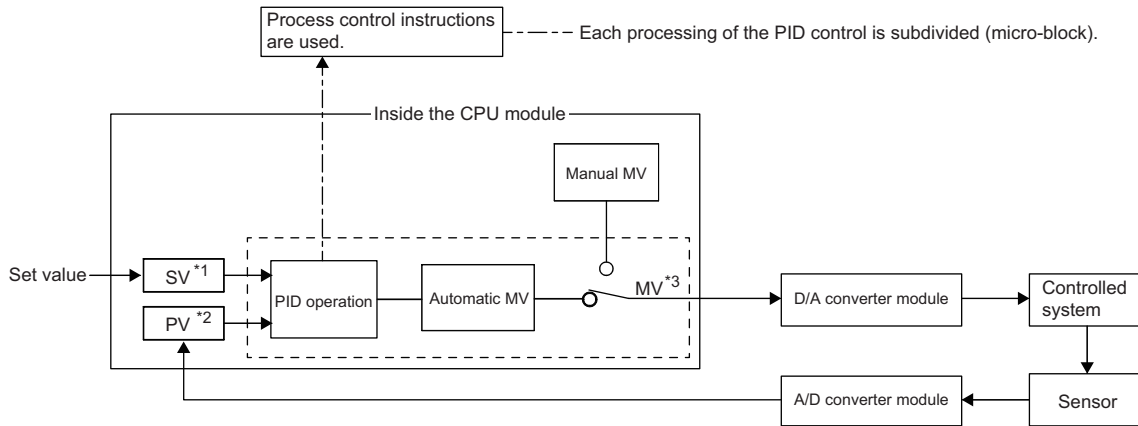
# MEMO

---

# 10 PROCESS CONTROL FUNCTION

This function performs various types of process control by using process control instructions, such as two-degree-of-freedom PID control, sample PI, and auto tuning, in combination.

Since each processing of the PID control is subdivided and multiple process control instructions are used in combination, this function performs more accurate and sensitive control than the PID control function.



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

Process control instructions have the following features:

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

**Restriction**

The programmable controller CPU does not support the use of process control instructions.

**Point**

Process control is performed by process control instructions. For the specifications of each instruction, basic loop types, and programming details, refer to the following.

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

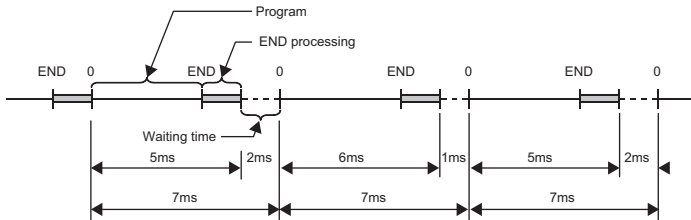
# MEMO

---

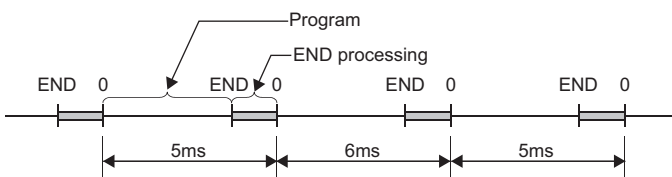
# 11 CONSTANT SCAN

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

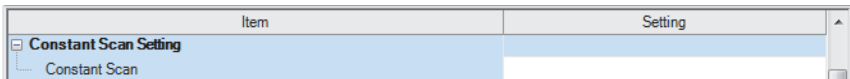


## 11.1 Setting Constant Scan

Constant scan can be set.

[CPU Parameter] ⇒ "RAS Setting" ⇒ "Constant Scan Setting"

### Window



### Displayed items

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

### Point

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

"WDT setting time" > "Constant scan setting time" > "Maximum scan time of the program"

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

## 11.2 Accuracy of Constant Scan

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

(Page 94 Precautions)

## 11.3 Precautions

---

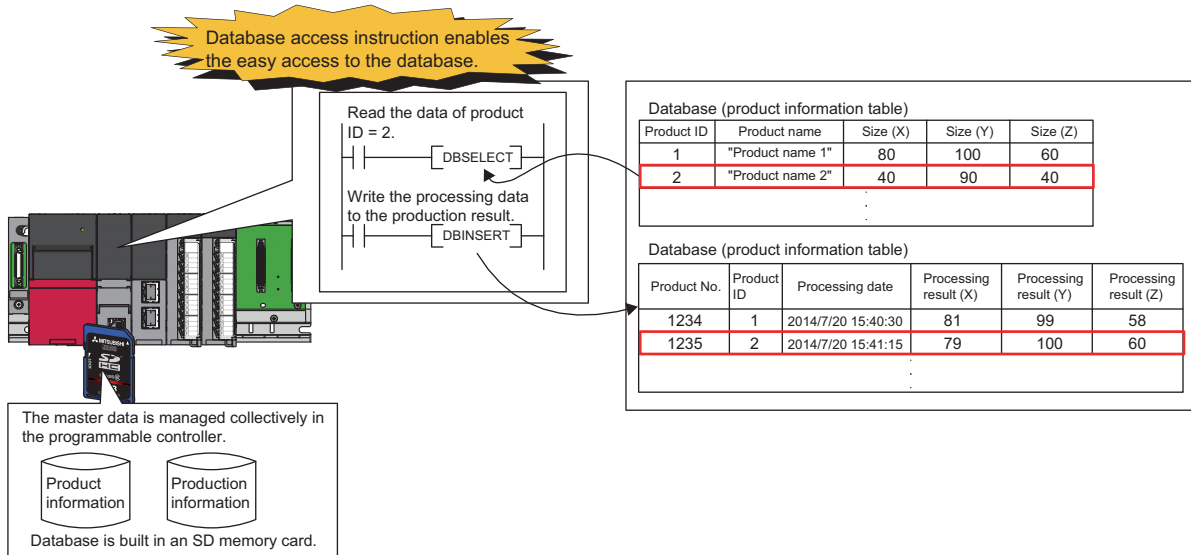
Processing of the program is stopped during the waiting time from the time the END processing is completed for the program, until the next scan starts. If any of the following processing requests is received during the waiting time, the corresponding processing is executed.

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



# 12 DATABASE FUNCTION

This function manages tabular data such as product information or production information as a database on the programmable controller, and accesses the database using data access instructions. It allows for easily adding, updating, searching, and deleting data regarding product information or production information.



**Point**

For details on instructions used for this function, refer to the following.  
MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

## 12.1 Usage Procedure

This section describes the procedure to use the database function.

### Creating databases

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

1. Create a Unicode text file which defines the configuration of a database and its tables. (Page 97 Creating Unicode Text Files)
2. Write the created Unicode text file to the SD memory card. For details on how to Unicode text files (write of user data), refer to the GX Works3 Operating Manual.
3. By executing the DBIMPORT(P) instruction with the created Unicode text file specified, the database folder is created on the same layer as the Unicode text file and the database is constructed. (Page 101 Folder Configuration of Databases)

**Point**

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

# Operating databases

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

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

## Point

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

2. Once the connection to the database is established, perform operation (addition, update, search, or deletion) of the database. To perform database operation, the user must specify the "database ID number" of the target database in each instruction. (☞ Page 96 List of Database Access Instructions)
3. When database operation is finished, execute the DBCLOSE(P) instruction to disconnect from the database.

## Point

The user can use a transaction when the user wants to perform multiple operations for a database as a set and update the database at once (☞ Page 101 Transactions for Databases)

## 12.2 List of Database Access Instructions

The following table lists instructions which are used for the database function.

Instruction name	Notation	Description
Database connection	DBOPEN(P)	Connects to the specified database.
Database disconnection	DBCLOSE(P)	Disconnects from the specified database.
Database record addition	DBINSERT(P)	Adds a record to the specified table of the database.
Database record update	DBUPDATE(P)	Updates all records that satisfy the specified conditions in the specified table of the database.
Database record search	DBSELECT(P)	Search all records that satisfy the specified conditions from the specified table of the database.
Database record deletion	DBDELETE(P)	Deletes all records that satisfy the specified conditions from the specified table of the database.
Database import	DBIMPORT(P)	Imports the specified data from the specified Unicode text file to create a database.
Database export	DBEXPORT(P)	Exports the specified data in the specified database into a Unicode text file.
Database transaction start	DBTRANS(P)	Starts a transaction for the specified database.
Database commitment	DBCOMMIT(P)	Commits a transaction for the specified database.
Database rollback	DBROLBAK(P)	Performs rollback on the specified database.

## Restriction

The user cannot execute multiple database access instructions simultaneously. If they are executed simultaneously, instructions other than the first one will be completed with an error.

## 12.3 Creating Unicode Text Files

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

### Database specifications

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

Item	Description
The maximum number of fields	16 per table
The maximum number of records	100,000 per table
The maximum number of characters of a field name	32 characters
The maximum number of tables	16 per database
The maximum number of characters of a table name	32 characters
Corresponding data type	<ul style="list-style-type: none"><li>• BOOL: Bit</li><li>• WORD: Unsigned 16-bit numerical value</li><li>• DWORD: Unsigned 32-bit numerical value</li><li>• INT: Signed 16-bit numerical value</li><li>• DINT: Signed 32-bit numerical value</li><li>• REAL: Single-precision real number</li><li>• LREAL: Double-precision real number</li><li>• STRING: Character string (Shift-JIS code)</li><li>• WSTRING: Character string (Unicode)</li></ul>
The maximum number of databases	No limitation (they can be created up to the capacity of the SD memory card)
The maximum number of concurrently connectable databases	4

# Setting details of Unicode text file

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

Item	Description
Database name	Specify a database name, using single-byte alpha-numeric characters. <ul style="list-style-type: none"> <li>• Within 32 characters</li> <li>• Not case-sensitive</li> <li>• The following characters cannot be used: ;, /, \, *, ?, ", ', &lt;, &gt;,  , #, {, }, %, &amp;, ~, (, )</li> </ul>
Table definition start tag	Specify the tag, <table>, using a single-byte lower-case characters. When defining more than one table, the user must enclose each table with <table> and </table>.
Table name	Specify a table name, using single-byte alpha-numeric characters. <ul style="list-style-type: none"> <li>• Within 32 characters</li> <li>• Not case-sensitive</li> <li>• The following characters cannot be used: ;, /, \, *, ?, ", ', &lt;, &gt;,  , #, {, }, %, &amp;, ~, (, )</li> </ul>
Field name	Specify the name of each field of each record, using single-byte alpha-numeric characters, and in the tab-delimited format. <ul style="list-style-type: none"> <li>• The maximum number of characters which can be used for each field name is 32.</li> <li>• Case-sensitive</li> <li>• The following characters cannot be used: " and '</li> <li>• The maximum number of fields is 16.</li> </ul>
Data type	Specify the data type of each field in the tab-delimited format. The data types which can be specified are as follows. They must be specified using upper-case characters. <ul style="list-style-type: none"> <li>• BOOL</li> <li>• WORD</li> <li>• DWORD</li> <li>• INT</li> <li>• DINT</li> <li>• REAL</li> <li>• LREAL</li> <li>• STRING: Specify the number of characters after a colon (:). (For example, when the number of characters is 16, the user can specify like that: "STRING: 16"). The range of the number of characters is 1 to 124.*1 *2</li> <li>• WSTRING: Specify the number of characters after the colon (:). The range of the number of characters is 1 to 124.*1</li> </ul>
Key constraint	Specify the key constraint on each field, using single-byte alphabetical characters. Between key constraints, a tab must be inserted as a delimiter. One of the following can be specified: <p>0: None</p> <p>1: Primary key (used to uniquely identify each record) When a primary key is specified, its key name also must be specified after a colon (:). (e.g. "1: pk1"). Key names must be up to 16 single-byte alphabetical characters and are case sensitive.</p> <p>2: NOT NULL constraint (When this setting is selected, setting values cannot be left unspecified by skipping specification of primary keys or foreign keys. (This value is used to set a restriction.)</p> <p>3: Foreign key (used to refer to the value of a field of another table.) When a foreign key is specified, a table name, which is to be associated with the key name, also must be specified after a colon (:), like that: "3: Key Name: Table Name" (e.g. "3: fk1: tb2"). Key names must be up to 16 single-byte alphabetical characters and are case sensitive.</p>
Index	When the user don't want to specify any indexes, the user can specify 0 (single-byte numeric character). When an index is specified, the name of the index and a field name must be specified using a colon (:). between them, like that: "Index Name: Field Name" (e.g. "id1: field1"). <ul style="list-style-type: none"> <li>• Index names must be up to 16 single-byte alphabetical characters and are case sensitive.</li> <li>• The maximum number of indexes is 16.</li> <li>• For fields with a primary or foreign key constraint, indexes are automatically set, and it is necessary to specify indexes by this setting (if specified, the DBIMPORT(P) instruction is completed with an error).</li> </ul>
Setting value	Specify the setting value of each field in the tab-delimited format. <ul style="list-style-type: none"> <li>• Characters or numbers which will be set to each field must match with the specified data type (☞ Page 97 Database specifications, ☞ Page 100 Format of setting values in Unicode text).</li> <li>• Individual records are set by row.</li> <li>• The maximum 100,000 records can be stored in a single table.</li> </ul>
Table definition end tag	Specify the tag, </table>, using a single-byte lower-case characters.

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

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

## Point

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


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

## Configuration of Unicode text files

```

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

```

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

Item	Description
Line break	<ul style="list-style-type: none"> <li>The line break code which can be recognized is CR+LF.</li> <li>Lines consisting of only line breaks are ignored.</li> </ul>
Comment	The block beginning with "/" and ending with a line break code is ignored as a comment.
Tab	Tabs cannot be used for database names, table names, field names, key constrains, and setting values. Tabs are recognized as delimiters.
Space	<ul style="list-style-type: none"> <li>Spaces between tabs and printable characters are not recognized as a part of settings.</li> <li>Spaces between printable characters are recognized as spaces.</li> </ul>

## ■ Format of setting values in Unicode text

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

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

\*2 The exponent part must always have a sign.

**Ex.**

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

proid1	proinfo1	proinfo2	.....	proinfoN
1	abcd	efgh		pqrs
1001	abcd	efgh		pqrs tuv
.				
.				
1010	abcd	efgh		pqrs

Data type: INT  
Key restriction: Main key

Data type: STRING, Number of characters: 124  
Key restriction: None

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

### Point

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

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

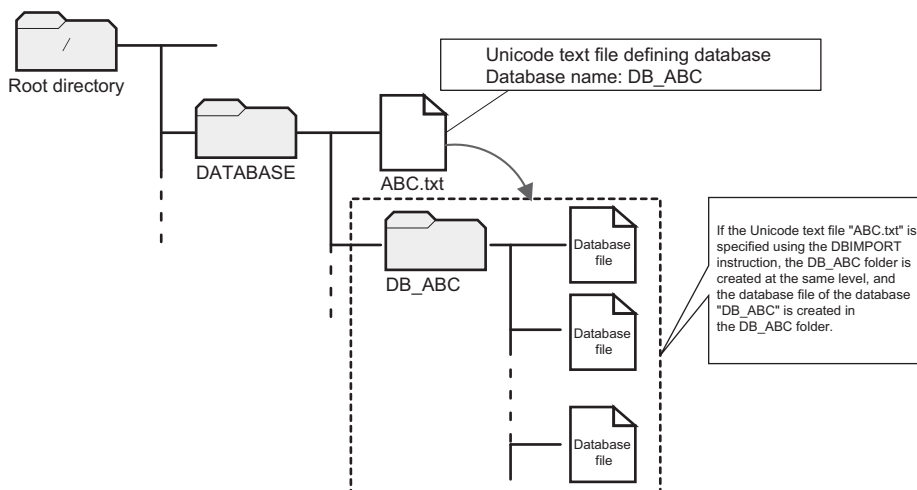
## 12.4 Transactions for Databases

The user can use a transaction (the DBTRANS(P) instruction and the DBCOMMIT(P) instruction) when the user wants to perform multiple operations for a database as a set and update the database at once. The DBTRANS(P) instruction starts a transaction, and the DBCOMMIT(P) instruction groups the results of the following instructions together to update the target database at once. The target of a transaction must be a single database. If a transaction is executed for more than one database, transaction instructions for the second and later databases are completed with errors. When the power is turned off or reset in the period from the start of a transaction (execution of the DBTRANS(P) instruction) to the confirmation of the transaction (execution of the DBCOMMIT(P) instruction), the database is automatically rolled back to the status before the start of the transaction. However, if a database access instruction is completed with an error, the database is not roll backed to the previous status. Also, to cancel the running transaction process and reset the database to the status before the start of the transaction, the user can perform rollback (DBROLBAK(P) instruction). Rollback is an operation which resets the target database to the previous status by executing the DBROLBAK(P) instruction in the period from the start of the transaction to the commitment of the transaction. When the DBCLOSE(P) instruction is executed without the DBCOMMIT(P) instruction or the DBROLBAK(P) instruction, the transaction is confirmed during execution of the DBCLOSE(P) instruction.

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

## 12.5 Folder Configuration of Databases

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



### Operation of files/folders in the database folder

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

## 12.6 Timing of Database Update

---

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

## 12.7 Precautions

---

This section describes precautions on using the database function.

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

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

### **When the SD memory card is write-protected**

To use database access instructions, the user must ensure that write protection of the SD memory card is off. When a database access instruction is performed and if the SD memory card is write-protected, the instruction is completed with an error.



# 13 REMOTE OPERATION

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

- Remote RUN/STOP
- Remote PAUSE
- Remote RESET

## 13.1 Remote RUN/STOP

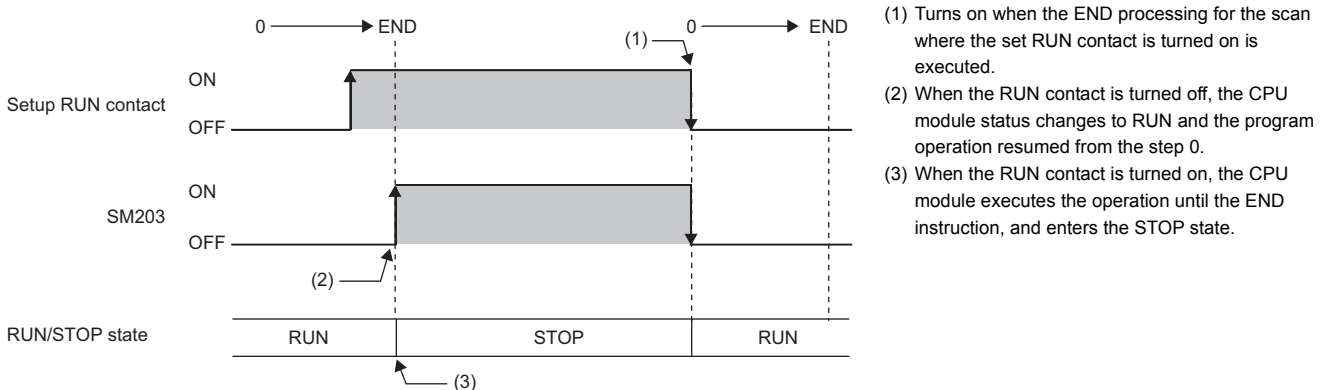
Externally changes the CPU module status to RUN or STOP with the RUN/STOP/RESET switch of the CPU module set to RUN. This function is useful to RUN or STOP a CPU module located in an inaccessible place, or to RUN or STOP a CPU module located on the control panel by using an external signal.

### Executing method

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

#### Using a contact

Set RUN contact in the RUN-PAUSE contact setting. (Page 105 Setting RUN-PAUSE contacts) When the END processing for the scan where the RUN contact is turned on is executed, SM203 (STOP contact) turns on to change the CPU module status to STOP and stop the operation. When the RUN contact is turned off, the CPU module exits the STOP state and resumes the program operation from the step 0.



#### Using an engineering tool

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

#### By an external device using SLMP

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

#### Using the module dedicated instructions

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

## Precautions

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

- When remote RUN is performed during execution of the data logging function, it may fail. In that case, wait for a while and retry remote RUN. If remote RUN still cannot be executed, check whether remote RUN is acceptable and retry remote RUN (📖 Page 165 About remote operation)
- When the RUN contact, which is specified in the RUN-PAUSE contact setting, is turned off during execution of the data logging function, it may take time to return to the RUN state.

## 13.2 Remote PAUSE

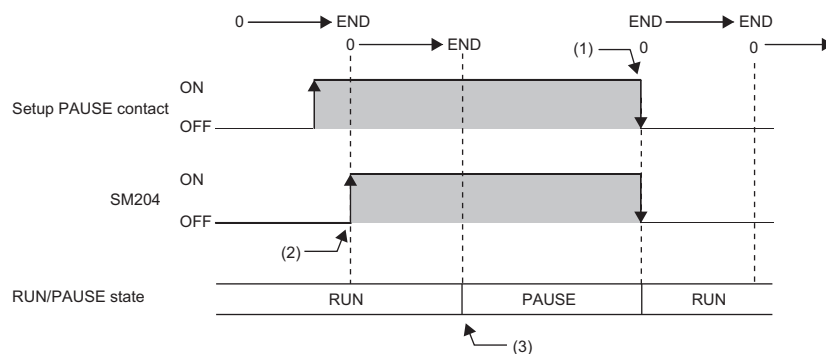
Externally changes the CPU module status to PAUSE with the RUN/STOP/RESET switch of the CPU module set to RUN. Use this in a process control and other situations to keep the CPU module status in RUN even when changing the status of the output (Y) to STOP.

### Executing method

The following methods are available to execute remote PAUSE:

#### Using a contact

Set PAUSE contact in "RUN-PAUSE Contact Setting". (📖 Page 105 Setting RUN-PAUSE contacts) When the END processing for the scan where the PAUSE contact is turned on is executed, SM204 (PAUSE contact) turns on. When the CPU module executes until the END instruction for the scan next to that where the PAUSE contact is turned on, its status changes to PAUSE, and the operation stops. When the PAUSE contact is turned off, the CPU module exits the PAUSE status and the program operation resumed from the step 0.



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

#### Using an engineering tool

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

#### By an external device using SLMP

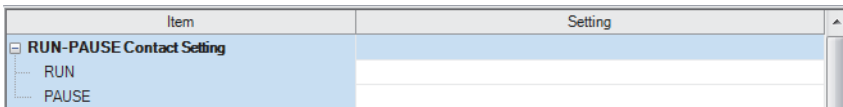
Perform remote PAUSE with the SLMP command. (📖 SLMP Reference Manual)

## 13.3 Setting RUN-PAUSE contacts

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

[CPU Parameter] ⇒ "Operation Related Setting" ⇒ "RUN-PAUSE Contact Setting"

### Window



### Displayed items

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

13

### Point

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

## 13.4 Remote RESET

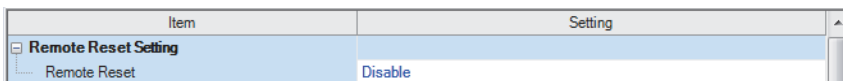
Externally resets a CPU module in the STOP state (including that stopped due to an error). Even when the RUN/STOP/RESET switch of a CPU module is set to RUN, the CPU module can be reset in the STOP state.

### Enabling remote RESET

Performing remote RESET requires setting for enabling remote reset.

[CPU Parameter] ⇒ "Operation Related Setting" ⇒ "Remote Reset Setting"

### Window



### Displayed items

Item	Description	Setting range	Default
Remote RESET	Set whether or not to enable the remote RESET.	<ul style="list-style-type: none"><li>• Disable</li><li>• Enable</li></ul>	Disable

### Executing method

#### Using an engineering tool

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

#### By an external device using SLMP

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

# 14 DEVICE/LABEL MEMORY AREA SETTING

The capacity of each area on the device/label memory can be specified (☞ Page 63 Device/label memory)

Item	Symbol	Device	
		Points	Range
Input	X	12K	0 to 2FFF
Output	Y	12K	0 to 2FFF
Internal Relay	M	12K	0 to 12287
Link Relay	B	8K	0 to 1FFF
Special Link Relay	SB	2K	0 to 7FF
Annunciator	F	2K	0 to 2047
Edge Relay	V	2K	0 to 2047
Timer	T	1K	0 to 1023
Long Timer	LT	1K	0 to 1023
Retentive Timer	ST	0	
Long Retentive Time LST		0	
Counter	C	512	0 to 511
Long Counter	LC	512	0 to 511
Data Register	D	18K	0 to 18431
Link Register	W	8K	0 to 1FFF
Link Special 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 108 Device/Label Memory Area Setting)

(2) The number of points of user devices can be changed (☞ Page 109 Device Setting)

## 14.1 Default Capacity of Each Area

The default capacity of each area is as follows.

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

### Point

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

- R04CPU: 40K words
- R08CPU, R08PCPU: 50K words
- R16CPU, R16PCPU: 60K words
- R32CPU, R32PCPU: 70K words
- R120CPU, R120PCPU: 90K words

# 14.2 The Setting Range of the Capacity of Each Area

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

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

## R04CPU

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

## R08CPU, R08PCPU

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

## R16CPU, R16PCPU

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

## R32CPU, R32PCPU

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

## R120CPU, R120PCPU

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

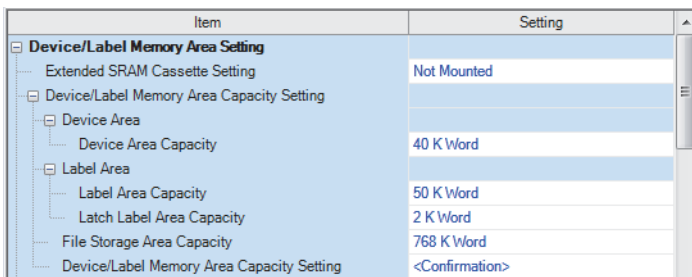
# 14.3 Device/Label Memory Area Setting

The capacity of each data area allocated within the device/label memory can be changed. (☞ Page 63 Device/label memory)

☞ [CPU Parameter] ⇒ "Memory/Device Setting" ⇒ "Device/Label Memory Area Setting"

### Operating procedure

"Device/Label Memory Area Setting" window



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

### Displayed items

Item	Description	Setting range	Default
Extended SRAM Cassette Setting	When an extended SRAM cassette is used, the capacity of the cassette must be selected.	<ul style="list-style-type: none"> <li>• Not Mounted</li> <li>• 1MB</li> <li>• 2MB</li> <li>• 4MB</li> <li>• 8MB</li> </ul>	Not Mounted
Device/Label Memory Area Capacity Setting	Device Area Capacity	☞ Page 107 The Setting Range of the Capacity of Each Area	☞ Page 106 Default Capacity of Each Area
	Label Area Capacity	☞ Page 107 The Setting Range of the Capacity of Each Area	
	Latch Label Area Capacity		
	File Storage Area Capacity		
Device/Label Memory Area Capacity Setting	Shows the device/label memory configuration.		

### Point

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

# 14.4 Device Setting

The number of points of each user device can be changed (☞ Page 226 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
Special Link Relay	SB	2K	0 to 7FF
Annunciator	F	2K	0 to 2047
Edge Relay	V	2K	0 to 2047
Timer	T	1K	0 to 1023
Long Timer	LT	1K	0 to 1023
Retentive Timer	ST	0	
Long Retentive Time	LST	0	
Counter	C	512	0 to 511
Long Counter	LC	512	0 to 511
Data Register	D	18K	0 to 18431
Link Register	W	8K	0 to 1FFF
Link Special 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

### 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 108 Device/Label Memory Area Setting)

## Range of use of device points

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

Type	Device name	Symbol	Range of use*1	Increment of setting
Bit	Input	X	X0 to X2FFF	—
Bit	Output	Y	Y0 to Y2FFF	—
Bit	Internal relay	M	M0 to M94773247	64 points
Bit	Latch relay	L	L0 to L32767	64 points
Bit	Link relay	B	B0 to B5A61FFF	64 points
Bit	Annunciator	F	F0 to F32767	64 points
Bit	Link special relay	SB	SB0 to SB5A61FFF	64 points
Bit	Edge relay	V	V0 to V32767	64 points
Word	Timer	T	T0 to T5265151	32 points
Word	Retentive timer	ST	ST0 to ST5265151	32 points
Word	Long timer	LT	LT0 to LT1480831	1 points
Word	Long retentive timer	LST	LST0 to LST1480831	1 points
Word	Counter	C	C0 to C5265151	32 points
Word	Long counter	LC	LC0 to LC2787391	32 points
Word	Data register	D	D0 to D5923327	4 points
Word	Link register	W	W0 to W5A61FF	4 points
Word	Link special register	SW	SW to SW5A61FF	4 points

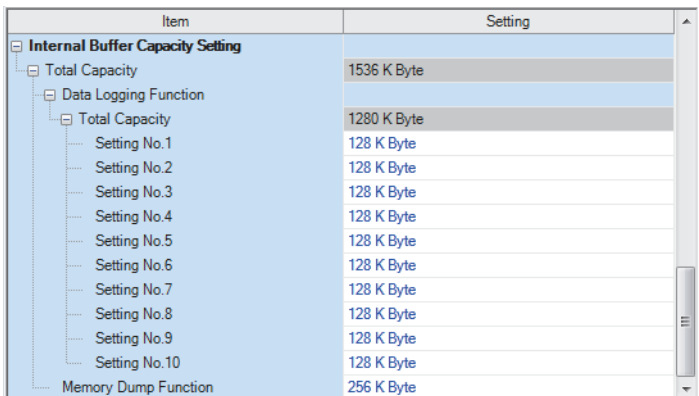
\*1 This is the maximum number of points for the R120CPU with an extended SRAM cassette (8MB). The number of points varies depending on the model of the CPU module used, whether to use an extended SRAM cassette, and the type of its cassette.

# 15 INTERNAL BUFFER CAPACITY SETTING

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

 [CPU Parameter] ⇒ "Memory/Device Setting" ⇒ "Internal Buffer Capacity Setting"

## Window




## Displayed items

Item	Description	Setting range	Default
Total Capacity	Shows the total of the internal buffer capacity set in the data logging function and the memory dump function.	60 to 3072K bytes	1536K bytes
Data Logging Function	Total Capacity	—	1280K bytes
	Setting No.1 to 10	<ul style="list-style-type: none"> <li>Each setting range: 32 to 3040K bytes (in 1K bytes)<sup>*1</sup></li> <li>Total setting range: 32 to 3040K bytes</li> </ul>	128K
Memory dump function	Set the internal buffer capacity used for the memory dump function.	32 to 3040K bytes (in 1K bytes) <sup>*2</sup>	256K bytes

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

\*2 A reduced capacity of the internal buffer decreases the extension of scan time, but it takes time to complete.

### Point

The internal buffer is also consumed in the real-time monitor function. Set the total of the internal buffer to no more than 3072K bytes including the internal buffer to be consumed in the real-time monitor function as well. The internal buffer for the real-time monitor function can be set with GX LogViewer. ( GX LogViewer Version1 Operating Manual)

### Restriction

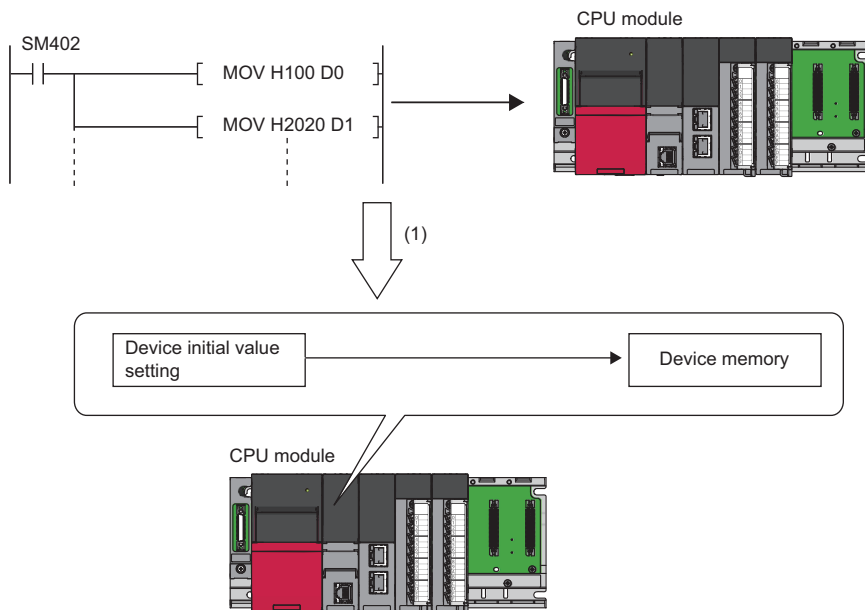
The internal buffer capacity setting cannot be set in the CPU module used or the function that is not supported by the engineering tool.

Check whether the engineering tool supports the function with availability for each model in the function list. ( MELSEC iQ-R CPU Module User's Manual (Startup))



# 16 DEVICE/LABEL INITIAL VALUE SETTINGS

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



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

## 16.1 Setting Initial Device/Label Values

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

### Setting initial device values

This section describes the settings of initial device values.

#### Setting procedure

The procedure for using initial device values is as follows.

1. First, the user must create an initial device value file. To set initial values to a local device, create an initial device value file with the same name as the program which sets these values, and specify the range of the values. To set initial values to a global device, create an initial device value file (with any name) which sets these initial values, and specify the range of the values.
2. On the device memory, set up initial device value data within the range specified in the initial device value file.  
( GX Works3 Operating Manual)
3. In the "Device Memory Register Diversion", select the device memory which was set up in Step 2. Setting "Device Memory Register Diversion" enables data set up on the device memory to be used as initial device values for the device which is specified in the initial device value file.  
( GX Works3 Operating Manual)
4. Configure CPU parameters. ( Page 112 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

Item	Setting
<input type="checkbox"/> Initial Value Setting	
Target Memory	Data Memory
Setting of Device Initial Value Use Or Not	Not Used
Global Device Initial Value File Name	

### 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.	<ul style="list-style-type: none"><li>Memory card</li><li>Data Memory</li></ul>	Data Memory
Setting of Device Initial Value Use Or Not	Selects whether or not to use initial device values.	<ul style="list-style-type: none"><li>Disable</li><li>Use</li></ul>	Not Use
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.

## 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	<ul style="list-style-type: none"><li>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.</li><li>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.</li></ul>

## 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 112 Initial value setting).
3. The CPU parameters, programs, initial global label value file, and initial local label value file are written to the CPU module.  
( GX Works3 Operating Manual)
4. The data in the specified initial global label value file and initial local label value file is automatically set to the specified label when the CPU module is powered off and on, reset, or the status changes from STOP to RUN.

## 16.2 Applicable Devices/Labels


For details on devices/labels to which initial device/label values can be set, refer to the following.

 GX Works3 Operating Manual

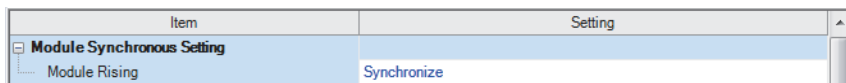
## 16.3 Precautions

This section describes the precautions when using the initial device/label value setting.

- When initial device values or initial label values are overlapped with the latch range, these initial values take precedence over the latch range. Therefore, even if a device or label is already latched, its values are changed to the specified initial values.
- The initial device value and initial label value are also reflected when the CPU module status changes from STOP to RUN. 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.
- To make the CPU module hold a device only when its values fall into the range of device initial setting (such as module access device and link direct device), the start-up time between the device and the CPU module must be synchronized using the CPU parameter, "Module Synchronous Setting".

 [CPU Parameter] ⇒ "Operation Related Setting" ⇒ "Module Synchronous Setting"

### Window



Item	Setting
Module Synchronous Setting	
Module Rising	Synchronize

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

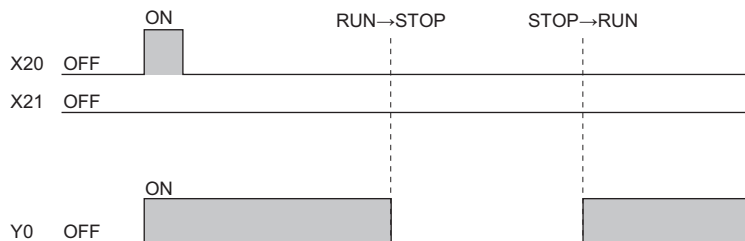
When the operating status changes from RUN to STOP, for example, the CPU module internally stores the status of the outputs (Y) to turn them all off.

## 17.1 Operation When the Operating Status Is Changed from STOP to RUN

Selects whether or not to resume from the previous output status when the CPU module status is changed from STOP to RUN by using a holding circuit, for example.

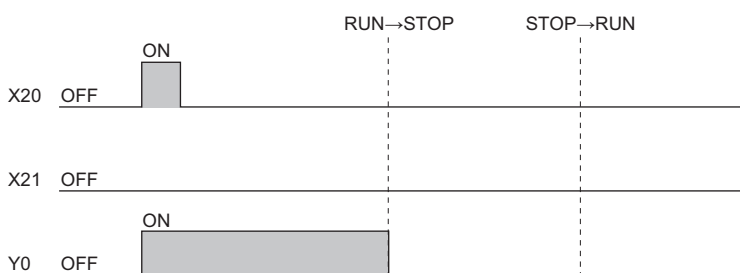
### Outputting the output (Y) status before STOP

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



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

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



## 17.2 Setting the Output Mode

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

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

### Window

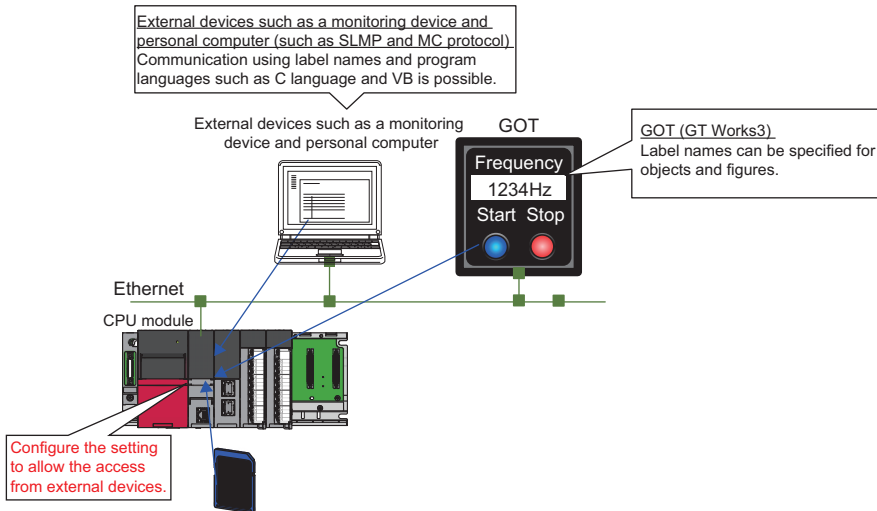
Item	Setting
<input type="checkbox"/> <b>Output Mode Setting of STOP to RUN</b> Output Mode of STOP to RUN	Output the Output (Y) Status before STOP

### 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.	<ul style="list-style-type: none"> <li>• Output the Output (Y) Status before STOP</li> <li>• Clear (Output is 1 scan later) the Output (Y)</li> </ul>	Output the Output (Y) Status before STOP

# 18 LABEL ACCESS SETTING FROM EXTERNAL DEVICE

This setting enables external devices to communicate with the CPU module through GOT, SLMP, or other method by specifying global label names.

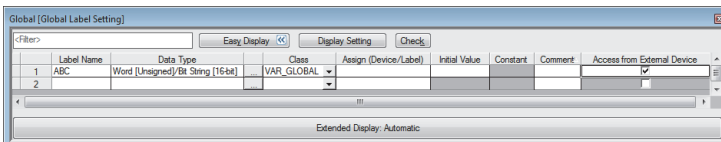


## 18.1 Configuration Procedure

This section describes the configuration procedure to enable access by specifying the global label from external devices.

### Operating procedure

"Global Label Setting" window



"Setting of File/Data Use or Not in Memory Card windows

Item	Setting
<input type="checkbox"/> <b>Setting of File/Data Use or Not in Memory Card</b>	
<input type="checkbox"/> Data for Label Communication	Not Used
<input type="checkbox"/> Module Extended Parameter	Not Used

1. Set the label in "Global Label Setting" and select the "Access from External Device" checkbox.
2. Check the capacity of the label communication data.
3. Enable "Data for Label Communication" to use the label communication data stored in the SD memory card (or store the label communication data in the SD memory card).  
 ⌨ [Memory Card Parameters] ⇒ "Setting of File/Data Use or Not in Memory Card" ⇒ "Data for Label Communication"
4. Write parameters, global label setting, and global label assignment information files to the CPU module.



Data storage to the data memory, not using the SD memory card, does not need the setting of procedure 3.

## 18.2 Label Communication Data Storage Location and Operation at the Time of Each Setting Operation

This section describes the storage location for label communication data and the accessibility according to the setting of whether to use the file/data in a memory card.

○: Access from external device possible (Communication memory indicated in parentheses) ×: Access from external device not possible (Generates communication error)

Label communication data storage location	When "Enable" is set in the setting for the usage status of the file/data in a memory card	When "Disable" is set in the setting for the usage status of the file/data in a memory card
Data Memory	× <sup>*1</sup>	○ (Data memory)
SD memory card	○ (SD memory card)	×
Data memory and SD memory card	○ (SD memory card)	○ (Data memory)

\*1 An error occurs in the CPU module.

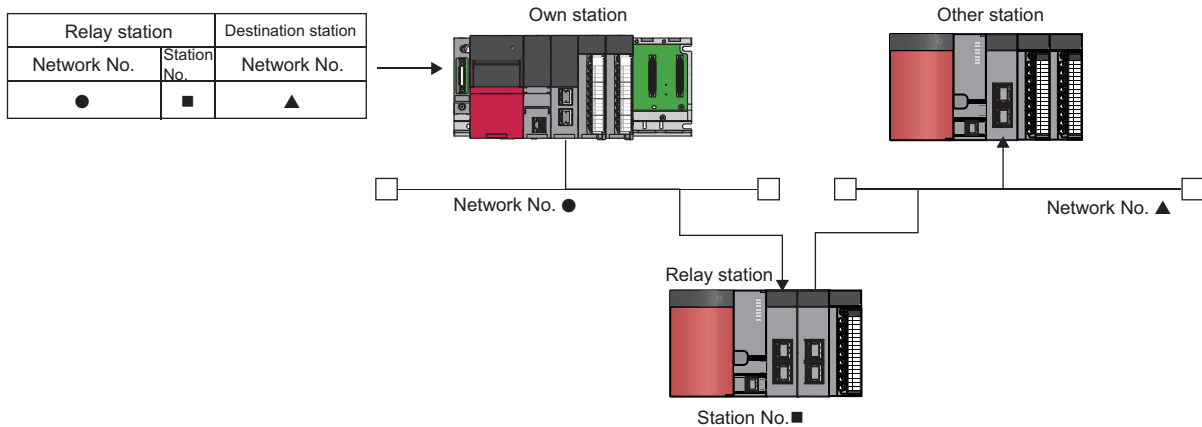
## 18.3 File Operation

The label communication data can be written from an engineering tool. Reading and deleting<sup>\*1</sup> are not possible.

\*1 Deleting an entire folder is possible.

# 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

No.	Relay Station			Target Station
	Network No.	Station No.		Network No.
1				
2				
3				
4				

### Displayed items

Item		Description	Setting range	Default
Relay Station	Network Number	Sets the network number of the fist relay station to pass through to the destination station.	1 to 239	—
	Station No.	Sets the station number of the fist relay station to pass through to the destination station.	0 to 120	—
Target Station	Network Number	Sets the network number of the final arrival network.	1 to 239	—

### Point

The S(P).RTWRITE instruction can be used to temporary change or add a routing setting during operation (setting made by the S(P).RTWRITE instruction is cleared when the CPU module is powered off or reset). Also, the S(P).RTREAD instruction can be used to read setting details of the parameters. For details on these instructions, refer to the following.

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

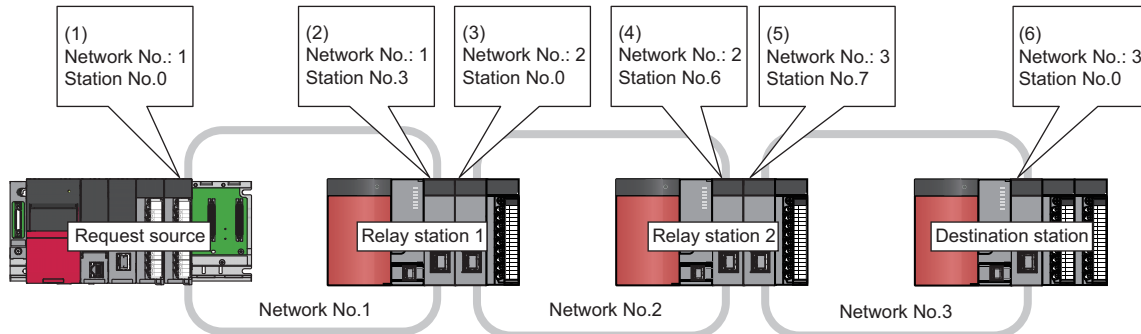


# 19.2 Setting Example

The following is an example of the routing setting.

**Ex.**

Transient transmission from the request source (Network No.1) to the target (Network No.3) via Network No.2.



Station	Request route	Response route									
Request source	<p>(1)</p> <table border="1"> <thead> <tr> <th colspan="2">Relay station</th> <th>Destination station</th> </tr> <tr> <th>Network No.</th> <th>Station No.</th> <th>Network No.</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>3</td> <td>3</td> </tr> </tbody> </table> <p>The data is transferred to the relay station (2) of the own network to go to the network No.3.</p>	Relay station		Destination station	Network No.	Station No.	Network No.	1	3	3	<p>(1)</p>
Relay station		Destination station									
Network No.	Station No.	Network No.									
1	3	3									
Relay station 1	<p>(2)</p> <p>(3)</p> <table border="1"> <thead> <tr> <th colspan="2">Relay station</th> <th>Destination station</th> </tr> <tr> <th>Network No.</th> <th>Station No.</th> <th>Network No.</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>6</td> <td>3</td> </tr> </tbody> </table> <p>The data is transferred to the relay station (4) of the own network to go to the network No.3.</p>	Relay station		Destination station	Network No.	Station No.	Network No.	2	6	3	<p>(2)</p> <p>(3)</p> <p>The data has been transferred to the network No.1, and is to be transferred to (1).</p>
Relay station		Destination station									
Network No.	Station No.	Network No.									
2	6	3									
Relay station 2	<p>(4)</p> <p>(5)</p> <p>The data has been transferred to the network No.3, and is to be transferred to (6).</p>	<p>(4)</p> <p>(5)</p> <table border="1"> <thead> <tr> <th colspan="2">Relay station</th> <th>Destination station</th> </tr> <tr> <th>Network No.</th> <th>Station No.</th> <th>Network No.</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>0</td> <td>1</td> </tr> </tbody> </table> <p>The data is transferred to the relay station (3) of the own network to go to the network No.1.</p>	Relay station		Destination station	Network No.	Station No.	Network No.	2	0	1
Relay station		Destination station									
Network No.	Station No.	Network No.									
2	0	1									
Request destination	<p>(6)</p>	<table border="1"> <thead> <tr> <th colspan="2">Relay station</th> <th>Destination station</th> </tr> <tr> <th>Network No.</th> <th>Station No.</th> <th>Network No.</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>7</td> <td>1</td> </tr> </tbody> </table> <p>The data is transferred to the relay station (5) of the own network to go to the network No.1.</p>	Relay station		Destination station	Network No.	Station No.	Network No.	3	7	1
Relay station		Destination station									
Network No.	Station No.	Network No.									
3	7	1									

# 19.3 Precautions

The precautions on the routing setting are as follows:

- For the multiple CPU system configuration, the same routing setting must be used for all CPUs.

# 20 LATCH FUNCTION

The data in each device/label of the CPU module is cleared to its default value in the following cases:

- When the CPU module is powered off and on
- When the CPU module is reset
- When a power failure exceeding the allowable momentary power failure time occurs

The data in each device/label with latch setting is retained even under the above described conditions. For this reason, control can be continued with the retained data even if the power of the CPU module is turned off or a power failure exceeding the allowable momentary power failure time occurs while the data is managed by a sequential control.

## 20.1 Types of Latch

The following types of latch are available:

- Latch (1): Data in this range can be cleared by a latch clear operation
- Latch (2): Data in this range cannot be cleared by a latch clear operation

## 20.2 Applicable Devices and Labels

This section describes the devices/labels that can be latched.

### Applicable devices

The devices below can be latched. However, local devices cannot.

Device	Specification method	Applicable latch type
Internal relay (M)	Specify the latch range.	Latch (1) or Latch (2)
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
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)
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	Attribute	Data type
Global label	VAR_GLOBAL	RETAIN	Basic data type, array, structure
Local label of a program block	VAR		
Local label of a function block	VAR		
	VAR_INPUT*1		
	VAR_OUTPUT		
	VAR_PUBLIC		

\*1 Only the subroutine-type function block is available.

## 20.3 Latch Setting

This section describes the latch setting.

### Setting latch on devices

Multiple latch ranges can be set for a device type. A total of 32 latch ranges between latch (1) and latch (2) can be set. However, the ranges of latch (1) and latch (2) must not overlap.


#### Setting a latch range

Set the device to latch, its range, and the latch type.

#### Operating procedure

"Device Setting" window

Item	Symbol	Latch (1)	Latch (2)
Input	X		
Output	Y		
Internal Relay	M	No Setting	No Setting
Link Relay	B	No Setting	No Setting
Special Link Relay	SB		
Annunciator	F	No Setting	No Setting
Edge Relay	V	No Setting	No Setting
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 Register	SW		
Latch Relay	L		No Setting
Total Device			
Total Word Device			
Total Bit Device			


1. Click "Detailed Setting" on the "Device Setting" window.  
 [CPU Parameter] ⇒ "Memory/Device Setting" ⇒ "Device/Label Memory Area Detailed Setting" ⇒ "Device Setting" ⇒ "Detail Setting"
2. On the "Device Setting" window, select the type of latch for the target device. Then, the "Latch Range Setting" window is displayed.

"Latch Range Setting" window


Latch (1)		Latch (2)	
No.	Device	Points (Decimal)	Start End
1			
2			
3			
4			

3. Check the tab for the latch type, select the device to set and set the latch range (Start, End).

## Setting the latch interval

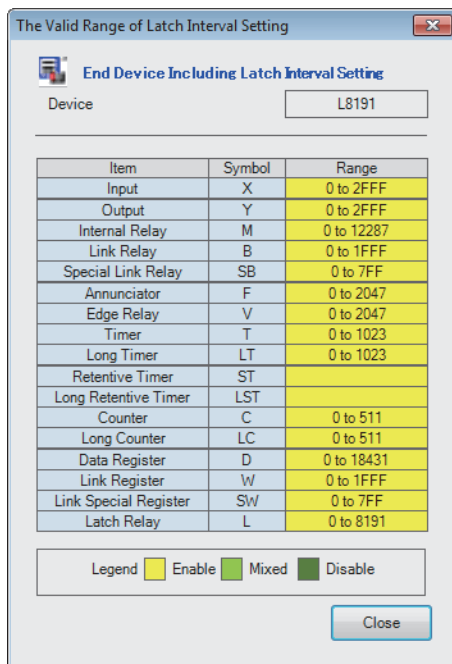
The user can specify the operation which should be performed at a latch interval (  Page 124 Timing of the latch processing) within the effective range of the latch timing setting\*1.

\*1 The range of devices for which the latch timing setting is enabled means the range enabled on the "The Valid Range of Latch Interval Setting" window. For other ranges (those which are not enabled on the "The Valid Range of Latch Interval Setting" window), this setting is not applied, and latching is performed in real time. The effective range of the latch timing means the range of devices which is enabled on the "The Valid Range of Latch Interval Setting" window. For other ranges (ranges of devices which are not enabled on the "The Valid Range of Latch Interval Setting" window), this setting is not applied, the END processing is accelerated, and latching is performed in real time.

 [CPU Parameter] ⇨ "Memory/Device Setting" ⇨ "Device Latch Interval Setting"

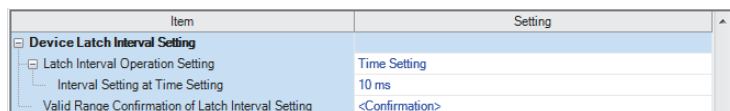
## Operating procedure

"The Valid Range of Latch Interval Setting" window



1. On the "Valid Range Confirmation of Latch Interval Setting" window, check the effective range for the latch timing setting. Specify the latch interval operation within the effective range.

"Device Latch Interval Setting" window



2. In "Latch Interval Operation Setting", select the timing of the latch processing. When "Time Setting" is selected, specify the time interval of latching.

## Displayed items

Item	Description	Setting range	Default
Latch Interval Operation Setting	Sets the timing of the latch processing.	<ul style="list-style-type: none"> <li>Per Scan</li> <li>Time Setting</li> </ul>	Time Setting
Interval Setting at Time Setting	Sets the time interval of latching (when "Time Setting" is selected).	1 to 2000ms (in units of 1ms)	10ms
Valid Range Confirmation of Latch Interval Setting	Shows the effective range for the latch timing setting.	—	—

For device latching, increasing the device range in the device setting of CPU parameters eliminates the latch processing from the END processing for the devices and enables real-time latching. For example, assume the following configuration for R04CPU: (1) 0K word is specified for the file storage area and 168K words for the device area in "Device/Label Memory Area Capacity Setting", (2) 100K points is specified for the data register (D) on the "Device Setting" window, and (3) D33151 and later are specified as the latch target devices. In this configuration, D33152 and later data registers (D) can be latched in real time as well as the devices (W, SW, and L) which are out of the effective range.

Device setting

Item	Symbol	Local Device			Latch (1)	Latch (2)
		Start	End	Points		
Link Relay	B				No Setting	No Setting
Special Link Relay	SB				No Setting	No Setting
Annunciator	F				No Setting	No Setting
Edge Relay	V				No Setting	No Setting
Timer	T				No Setting	No Setting
Long Timer	LT				No Setting	No Setting
Retentive Timer	ST				No Setting	No Setting
Long Retentive Time LST	LST				No Setting	No Setting
Counter	C				No Setting	No Setting

Latch Range Setting					
No.	Device	Points (Decimal)	Start	End	
1	D	69249	33151	102399	
2					
3					

Effective range of the latch timing setting

The Valid Range of Latch Interval Setting

Device: D33151

Item	Symbol	Range
Input	X	0 to 2FFF
Output	Y	0 to 2FFF
Internal Relay	M	0 to 12287
Link Relay	B	0 to 1FFF
Special Link Relay	SB	0 to 7FF
Annunciator	F	0 to 2047
Edge Relay	V	0 to 2047
Timer	T	0 to 1023
Long Timer	LT	0 to 1023
Retentive Timer	ST	
Long Retentive Timer	LST	
Counter	C	0 to 511
Long Counter	LC	0 to 511
Data Register	D	0 to 102399
Link Register	W	0 to 1FFF
Link Special Register	SW	0 to 7FF
Latch Relay	L	0 to 8191

Legend: ■ Enable ■ Mixed ■ Disable

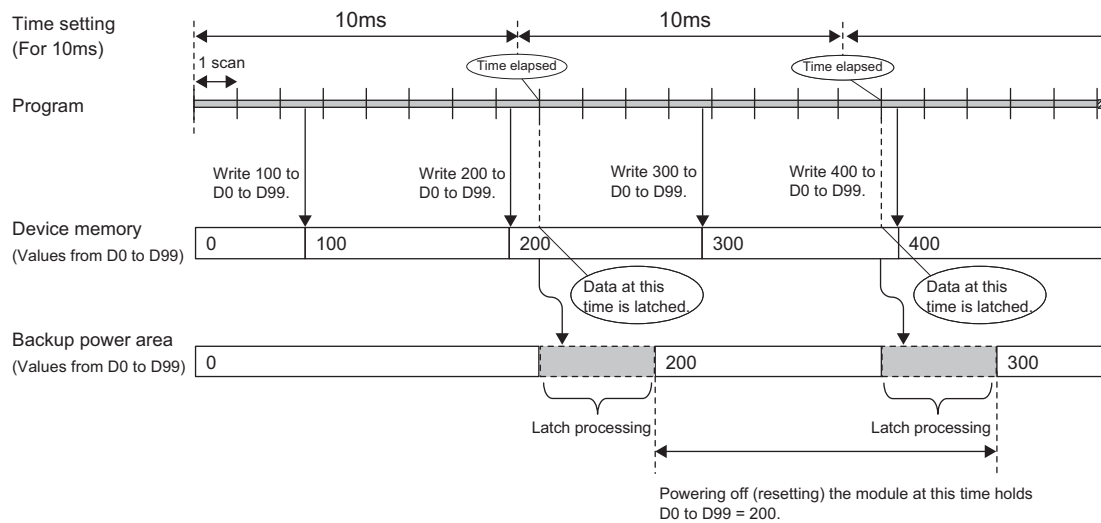
The data registers (D) D33152 and later and the device with the disabled range (W, SW, and L) can be latched in real time.

## ■ Timing of the latch processing

The timing of the latch processing is determined based on the effective range of the latch timing setting and the operation setting for the specified latch interval (🔗 Page 122 Setting the latch interval)

- When set to "Time Setting"

Latch processing is started in the END processing executed after the set time.

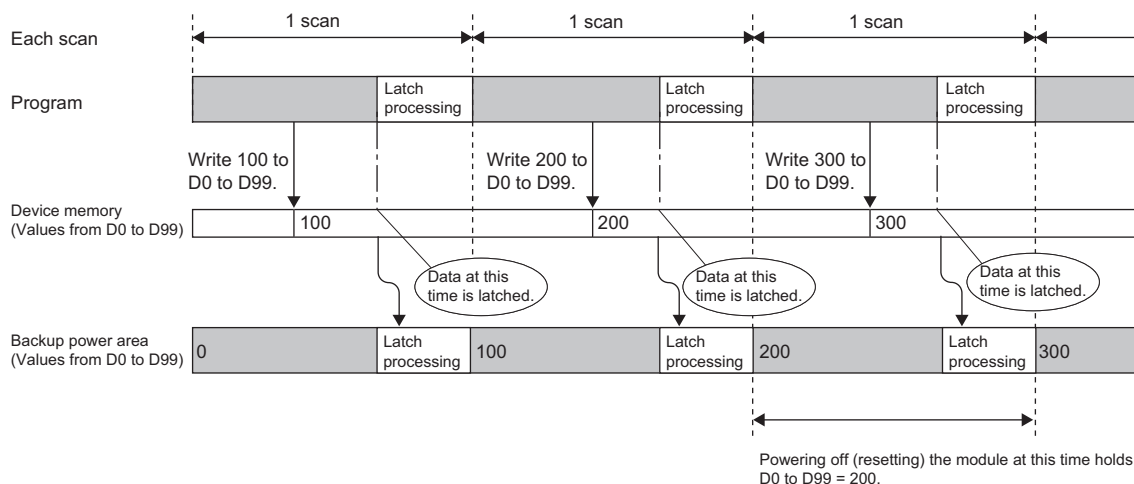


### Point

- The more the latch points, the more it takes to complete the latch processing. If the set time elapses while latch processing is still executing, the next one is executed in the END processing that follows the completion of the previous latch processing. Set a latch interval longer than the latch processing time so that the specified time does not elapse during latch processing.
- If the latch interval is shorter than the scan time, latch timing occurs more than once within one scan. However, latch processing is executed once during the END processing.

- When set to "Per Scan"

Latch processing is executed in the END processing of every scan. This setting allows to always retain the device of the previous scan because the devices are latched at every scan. However, it may increase the scan time.

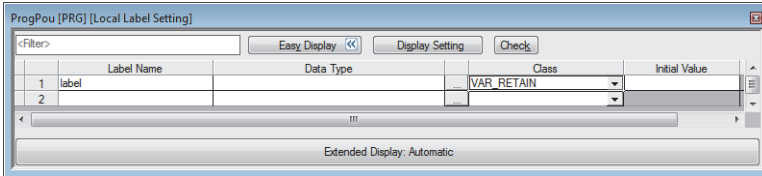


## Setting latch on labels

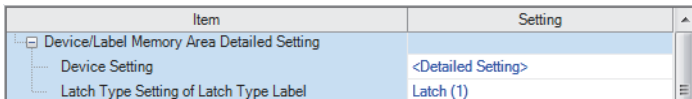
This section describes latch setting on labels.

### Operating procedure

Label edit window




"Device/Label Memory Area Detailed Setting" window




1. In the label edit window, specify "RETAIN" for label attribute.

2. There are two types of latch for labels: latch (1) and latch (2). Select one. The selected latch type is applied to labels of all latch attributes.

 [CPU Parameter] ⇒ "Memory/Device Setting"  
⇒ "Device/Label Memory Area Detailed Setting" ⇒ "Latch Type Setting of Latch Type Label"

## 20.4 Clearing Latch Range Data

Latch range data is cleared by either of the operations below. (  Page 65 Memory Operation)

- Latch clear: Performed from the engineering tool. (  GX Works3 Operating Manual)
- Latch clear by program: Execute RST instruction for latched devices, or clear by transferring K0 by using the MOV or FMOV instruction.

## 20.5 Precautions

This section describes the precautions when using the latch function.

- The data in a device/label within the latch range is retained by using the battery on the CPU module unit. Therefore, data cannot be retained during power failure if this battery runs out.
- If the latch range and number of device points are modified by using parameters, latch is performed on the modified latch range. However, if the value of the parameter setting the latch range is different between the previous and current operations because the CPU module was powered off and on or reset, and the latch range has been increased, the device range in the increased portion is not latched.

# 21 MEMORY CARD FUNCTION

This section describes functions that use the SD memory card.

## 21.1 SD Memory Card Forced Disable

Disables the SD memory card without turning the power off and on even while a function using the SD memory card, such as the data logging function is running.

### How to forcibly disable the SD memory card

The SD memory card is forcibly disabled by the following operations.

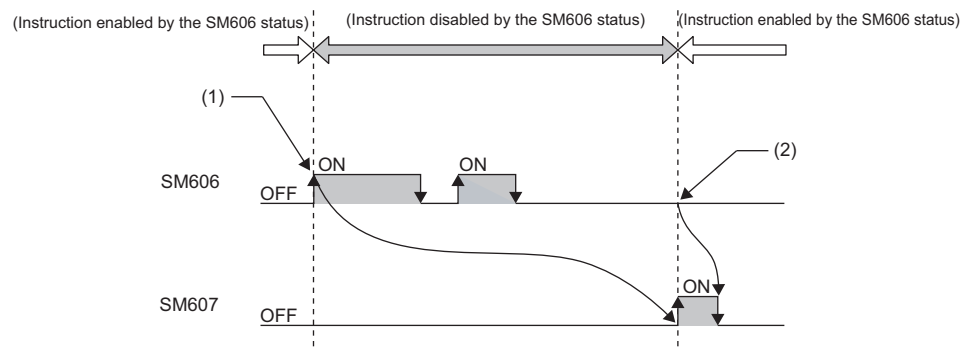
#### Operation of SD memory card lock switch

1. Press the SD memory card lock switch for one second or more.
2. The CARD READY LED flashes and goes out. \*1
3. Remove the SD memory card.

\*1 If the SD memory card was being accessed by a function, CARD READY LED goes out after access by the function is finished. Therefore, the time required for the LED to go out varies depending on the function.

#### Operation of special relay

1. Turn on SM606 (SD memory card forced disable instruction). Operation by turning SM606 on and off is disabled until SM607 turns on after SM606 is turned on. However, it is enabled and operates according to ON/OFF state of SM606 after the status of SM607 changes.



- (1) Forced disable instruction by ON state  
(2) Forced disable cancellation instruction enabled by OFF state

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



# Operation of functions accessing the SD memory card

Disabling the access to SD memory card affects the operation of functions that access the SD memory card. This section describes the operation of the functions listed below when SD memory card forced disable is executed during an access to the SD memory card or when they access the SD memory card after it is forcibly disabled.

Running function	When SD memory card forced disable is executed during access to the SD memory card	When SD memory card is accessed after it is disabled
Boot Operation	—	—
<ul style="list-style-type: none"><li>• Access to the device comment and label in the SD memory card</li><li>• Operation for device/label initial value at the time of STOP → RUN</li></ul>	The SD memory card is disabled after the running function finishes.	An error occurs in the CPU module.*1
Access to SD memory card by an engineering tool, SLMP, and FTP function	An error response is returned.	An error response is returned.
FREAD/FWRITE instruction	The SD memory card is disabled after the execution of the instruction finishes.	The instruction is completed with an error.

\*1 Same operation as when the SD memory card is not attached.

## Canceling SD memory card forced disable

The SD memory card forced disable is canceled by the following operation:

1. Install the SD memory card again. \*1
2. Power off and on, or reset the CPU module.

\*1 The CARD READY LED flashes and goes on.

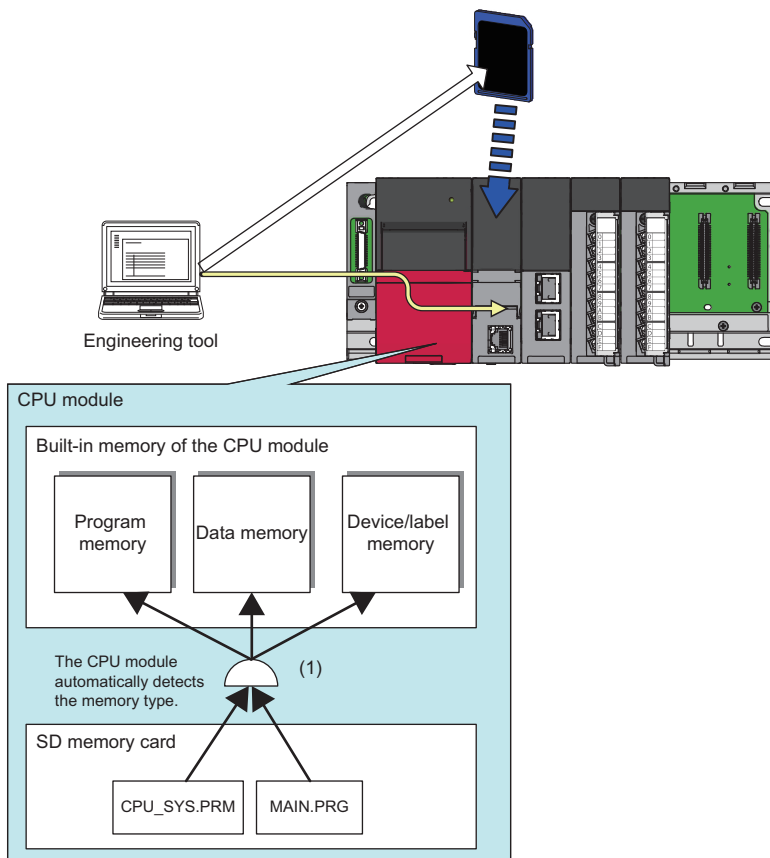
## Precautions

This section describes the precautions on using SD memory card forced disable.

- When the SD memory card is forcibly disabled by operating both the SD memory card lock switch and SM606, the operation executed first is valid and the operation executed afterwards, invalid. For example, turning SM606 on and off without removing the SD memory card after forcibly disabling it by the SD memory card lock switch cancels the SD memory card forced disable. However, if SM606 is turned on after forcibly disabling the SD memory card by the SD memory card lock switch and removing the SD memory card, the removed SD memory card will not be available even when it is re-inserted. To enable the SD memory card, SM606 must be turned off after re-insertion.
- If this function is executed while a file is written from an external device to the SD memory card, the writing may fail. Write the file again after enabling the disabled SD memory card.

## 21.2 Boot Operation

The files stored on the SD memory card are transferred to the storage memory which is automatically determined by the CPU module when the CPU module is powered off and on or reset.



(1) Executes the boot operation in accordance with the boot file setting when the CPU module is powered off and on or reset.

### Boot operation procedure

The boot operation procedure is as follows.

1. Configure the boot setting.
2. Attach an SD memory card.
3. Write the boot setting and boot file to the SD memory card.
4. Power off and on, or reset the CPU module.

## Specifiable file types

---

The files which can be booted are as follows.

- System parameter
- CPU parameter
- Module parameter
- Module extension parameter
- Module extension parameter for protocol setting
- Remote password
- Global label setting file
- Initial global label value file
- Program (each program)
- Initial local label value file
- FB file (each file)
- Device comment line
- Initial device value file

## Allowed maximum number of boot files

---

The maximum number of boot file settings which can be specified is 512. However, because more than one file is bootable for a single setting, the maximum number of boot files is the same as the number of files which can be stored in the transfer destination memory.

# Configuring the boot setting

Configure the necessary settings for the boot operation.

 [Memory Card Parameters] ⇒ [Boot Setting]

## Operating procedure

"Boot Setting" window

Item	Setting
<b>Boot Setting</b>	
Operation Setting at CPU Built-in Memory Boot	Do Not Clear
Boot File Setting	<Detailed Setting>

"Boot File Setting" window

No.	Type	Data Name
1		
2		
3		

"Add Type" window

"Boot File Setting" window

No.	Type	Data Name
1	System Parameter	SYSTEM
2	CPU Parameter	CPU
3	Module Extended Parameter for Protocol Setting(R16CPU)	UEX3FF00
4	Module Extended Parameter for Protocol Setting(0000:RJ71)	UEX00000
5	Remote Password	00000001
6	Global Label	GLBLINF
7	Global Label Initial Value	GLBLINF
8	Label Initial Value by Program	MAIN
9	Program File	MAIN
10	FB/FUN File	FBFILE
11	Device Comment	COMMENT
12	Device Initial Value	MAIN

1. Click "Detailed Setting" on the "Boot File Setting" window.
2. Click the "Type" column. The maximum number of boot files that can be specified is the same as the number of files that can be stored in the storage memory.
3. Select type for the boot file. (Multiple selection possible)
4. Set the data name (file name).

## Displayed items

Item	Description	Setting range	Default
Operation Setting at CPU Built-in Memory Boot	Sets whether or not to clear the CPU built-in memory upon file transfer from the SD memory card.	<ul style="list-style-type: none"> <li>• Do Not Clear</li> <li>• Clear</li> </ul>	Do Not Clear
Boot File Setting	Sets the file used for boot operation from the SD memory card.	—	—

## Operation when security functions are enabled

This section describes the operation when security functions are enabled.

### When a security key is set

When a security key is set to the boot target program file and the security of the program file does not match with that of the CPU module, a boot error occurs. Also, when no security key is written to the CPU module, a boot error occurs as well.

Security key of boot target program file	Security key of CPU module	Security key match/mismatch	Boot program execution
Set	Written	Match	Execute
	Written	Not match	Not execute (boot error)
	Not written	—	Not execute (boot error)

### When a file password is set

If a file password is set on both the source boot file and destination file, the file can be transferred only when the passwords match. Furthermore, the file transfer does not work if a file password is set only on either one.

Transferring boot file		Transferred boot file		Password match/mismatch	Transfer	
File	File password setting	File	File password setting			
Existing	Set	Existing	Set	Match	Yes	
			Not set	Mismatch	No	
		Not set	—	—	No	
	Not set	Existing	Set	—	—	Yes
			Not set	—	—	No
		Not set	Not set	—	—	Yes

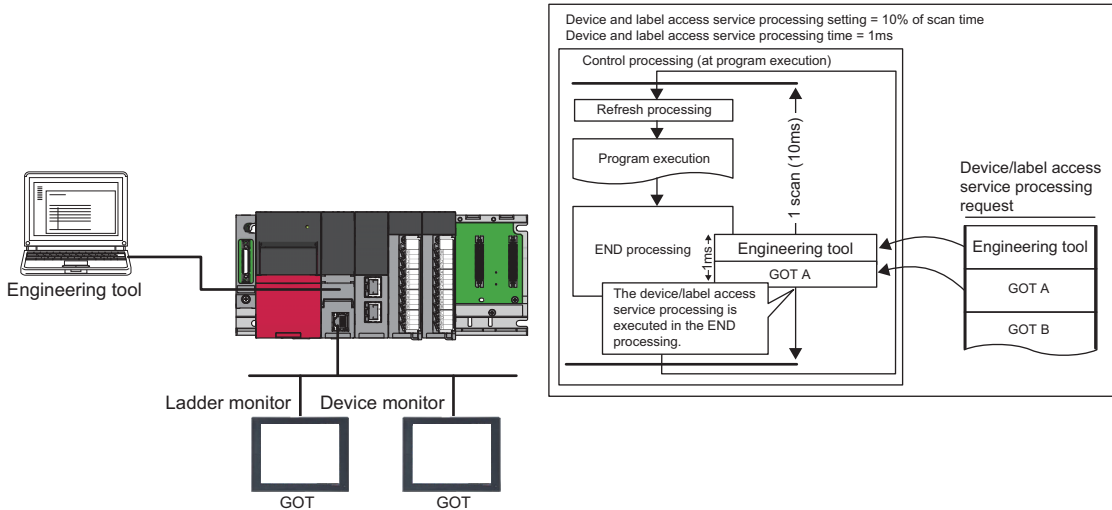
## Precautions

This section describes the precautions when using the boot operation.

- When a parameter file is set as boot file, it overwrites the parameter file inside the transfer destination CPU module. In addition, if a parameter file is not set as boot file even when stored on the SD memory card, the CPU module operates in accordance with the settings in the parameter file inside it.
- If a program in the program memory is changed online during boot operation, the change is also reflected in the transfer source program on the SD memory card.
- Note the model of the program written on the SD memory card (program specified in the boot file settings) and the model of the CPU module must be the same.

# 22 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.



## 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 374 List of the Targets of Device/Label Access Service Processing Setting)

# 22.1 Operations Enabled by Setting Details

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

Item	Scan performance		Device/label access service process performance		Inter-program monitoring *5	Application
	Increase*1	Stability*2	Response time*3	Stability*4		
Execute the process as scan time proceeds	Medium	Medium	Medium	Medium	None	This setting is useful to execute the device/label access service processing in a way commensurate with the system size. It allows the system to be designed without considering the device/label access service processing time because it is determined as a function of the scan time.
Set processing time	Medium	High	Medium	High	None	This setting is useful to give priority to the device/label access service processing. It allows for stable communication because the CPU module can always process a constant amount of the device/label access service processing without affecting the scan time.
Set processing counts	Large	Low	Medium	High	None	This setting is useful to stably execute the device/label access service processing in a system where requests come from multiple peripherals. It provides stable communication in a system where multiple peripherals exist because the CPU module can execute the device/label access service processing based on the number of request sources.
Execute END processing between programs	Large	Medium	Quick	High	Yes	This setting is useful to give priority to the device/label access service processing in a system with a large number of programs. It improves the response performance as it enables the device/label access service processing to be executed multiple times during a single scan.

\*1 Indicates how much the scan time will be extended at its maximum due to the device/label access service processing.

\*2 Indicates how much the scan time will be varied due to the device/label access service processing.

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

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

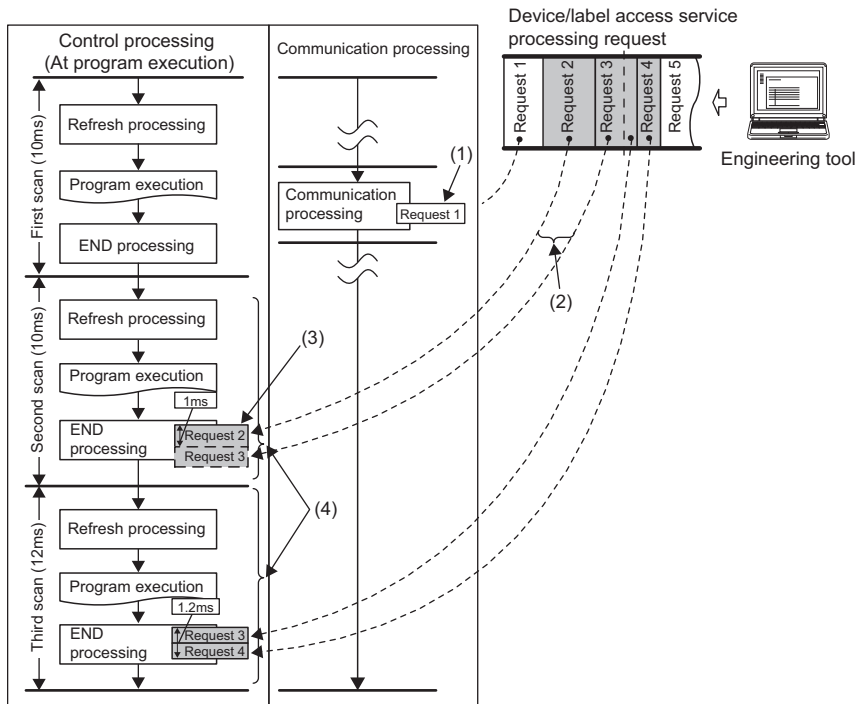
\*5 When this function is enabled, monitoring is performed between execution of programs, and therefore values of ongoing operation may be read out (☞ Page 138 When "Execute END Processing between Programs" is enabled)

## Execute the Process as Scan Time Proceeds

This setting is useful to execute the device/label access service processing in a way commensurate with the system size. It allows the system to be designed without considering the device/label access service processing time because it is determined as a function of the scan time.

**Ex.**

When "Scan time ratio = 10%" is set



- (1) Synchronization with program is unnecessary.
- (2) Multiple requests are processed until the specified device/label access service processing ratio (10%) is exceeded. When the specified device/label access service processing ratio is exceeded, the remaining requests are processed in the END processing of the next scan. Also, for scan which has the device/label access service processing time shorter than 0.1ms, it is assumed as "The device/label access service processing time per scan = 0.1ms".
- (3) Operations such as access to device, which are synchronized with the program, are processed in the END processing.
- (4) The maximum device/label access service processing time available varies because the scan time (program execution time) varies.

### Point

For operations such as access to devices, which are synchronized with the program, adjust the time by this setting because they are processed during the END processing. If no request data for the device/label access service processing exists, the scan time is shortened by the specified ratio as the CPU module proceeds to the next scan without waiting for requests.

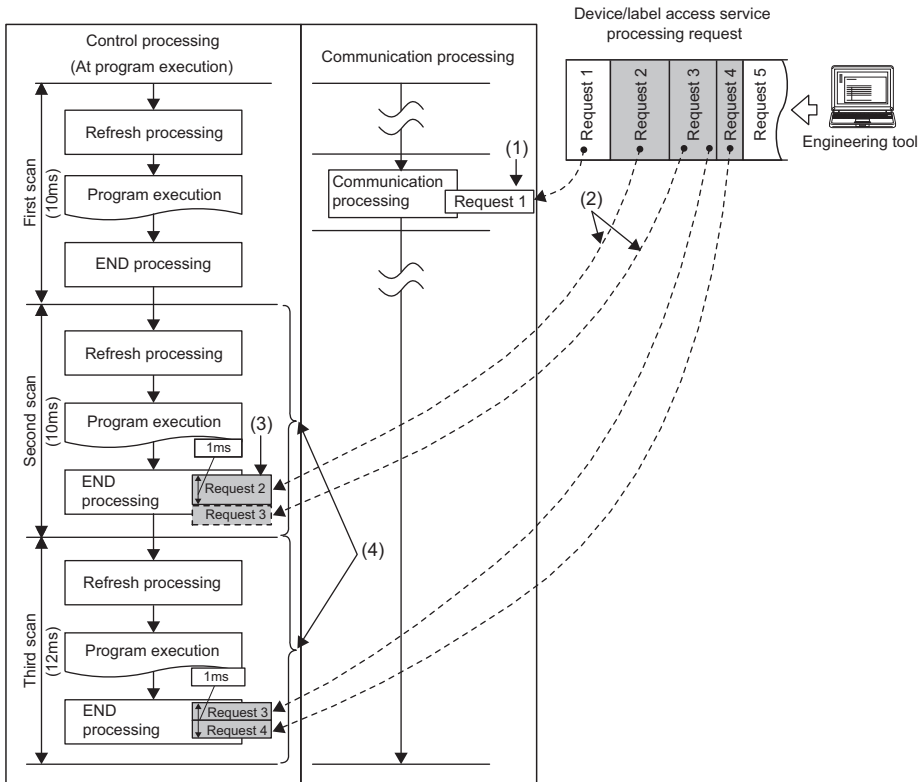


# Set Processing Time

This setting is useful to give priority to the device/label access service processing. It allows for stable communication because the CPU module can always process a constant amount of the device/label access service processing without affecting the scan time.

**Ex.**

When "Processing time = 1ms" is set



- (1) Synchronization with program is unnecessary.
- (2) Multiple requests are processed until the specified processing time (1ms) is exceeded. If the specified processing time is exceeded, the remaining requests are processed in the END processing of the next scan.
- (3) Operations such as access to device, which are synchronized with the program, are processed in the END processing.
- (4) The maximum device/label access service processing time available is the same even when the scan time (program execution time) varies.

**Point**

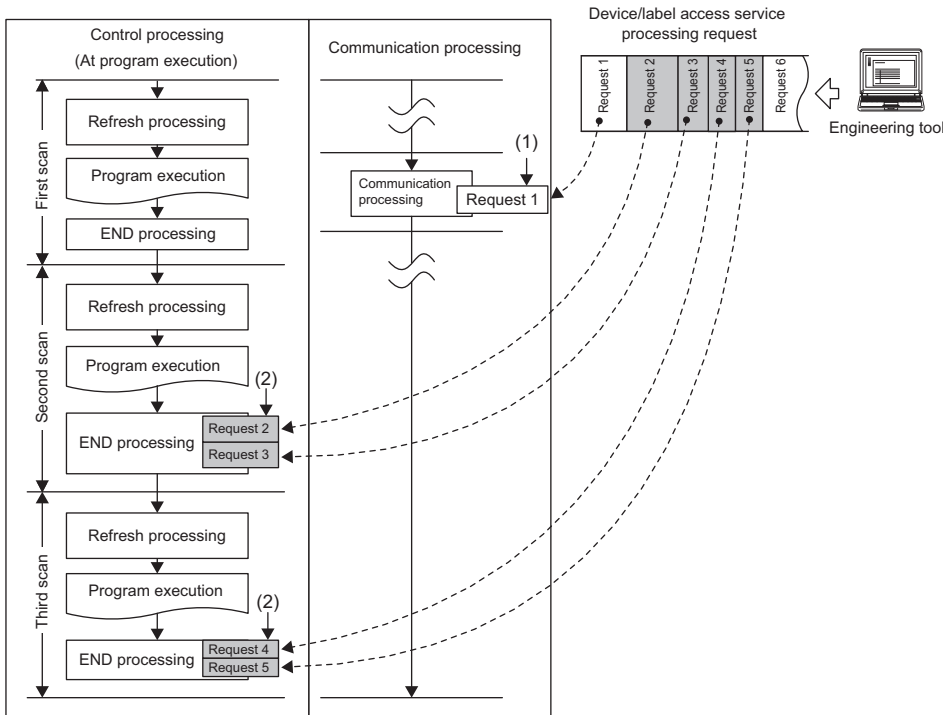
For operations such as access to devices, which are synchronized with the program, adjust the time by this setting because they are processed during the END processing. If no request data for the device/label access service processing exists, the scan time is shortened by the specified ratio as the CPU module proceeds to the next scan without waiting for requests.

# Set Processing Counts

This setting is useful to stably execute the device/label access service processing in a system where requests come from multiple peripherals. It provides stable communication in a system where multiple peripherals exist because the CPU module can execute the device/label access service processing based on the number of request sources.

**Ex.**

When "Processing counts = 2" is set



(1) Synchronization with program is unnecessary.

(2) Two requests are processed in one END processing independently of the requested processing time.

**Point**

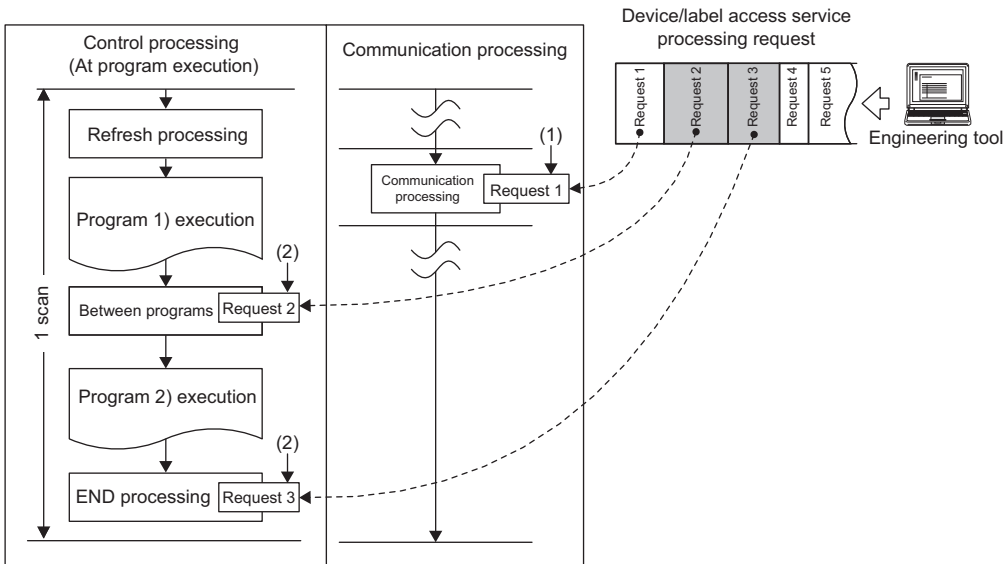
For operations such as access to devices, which are synchronized with the program, adjust the number of executions by this setting because they are processed during the END processing. If no request data for the device/label access service processing exists, the CPU module proceeds to the next scan without waiting for requests.

# Execute END processing between programs

This setting is useful to give priority to the device/label access service processing in a system with a large number of programs. Because operations such as access to devices are performed between program executions and during the END processing, requests as many as the number of programs can be processed during a single scan. As a result, it can reduce the response for the device/label access service processing.

**Ex.**

When "Execute END Processing between Programs" is enabled




(1) Synchronization with program is unnecessary.

(2) Requests for operations such as access to devices are processed between program executions and during the END processing.

## 22.2 Setting Method

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

 [CPU Parameter] ⇒ "Service Processing Setting" ⇒ "Device/Label Access Service Processing Setting"

### Window

Item	Setting
Device/Label Access Service Processing Setting	
Specifying Method	Execute the Process as Scan Time Proceeds
Rate	10 %
Time	0.1 ms
Counts	1 Times

### Displayed items

Item	Description	Setting range	Default
Specifying Method	Set the operation of the output (Y) when the operating status is changed from STOP to RUN.	<ul style="list-style-type: none"> <li>Execute the Process as Scan Time Proceeds</li> <li>Set Processing Time</li> <li>Set Processing Counts</li> <li>Execute END Processing between Programs</li> </ul>	Execute the Process as Scan Time Proceeds
Rate	Set the ratio of the device/label access service processing to execute during a single scan when "Execute the Process as Scan Time Proceeds" is enabled. The device/label access service processing time is determined based on the scan time.	1 to 99% (unit: 1%)	10%
Time	Set the device/label access service processing time per scan when "Set Processing Time" is enabled	0.1 to 1000ms (unit: 0.1ms)	0.1ms
Counts	Set the number of executions of the device/label access service processing per scan when "Set Processing Counts" is enabled.	1 to 10 times (unit: 1 time)	1 time

## 22.3 Precautions

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

### Functions that may prolong the scan time

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


- Online change
- Device/buffer memory batch monitor
- Data logging function
- File register write or read\*<sup>1</sup>

\*<sup>1</sup> The scan time becomes longer when the size of data to write or read is large

### When constant scan is set

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

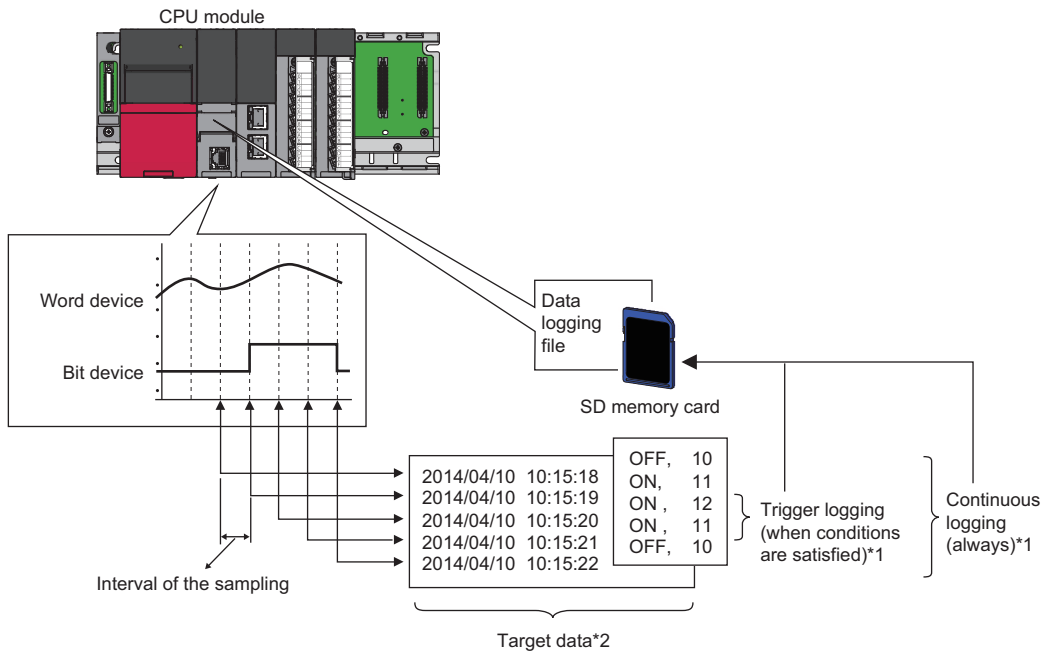
### When "Execute END Processing between Programs" is enabled

The device/label access service processing, such as access to devices, is performed between program executions and/or during the END processing. Therefore, when monitoring is performed in the situation where a device value is processed across programs, values of ongoing operation may be read out\*<sup>1</sup> Also, when "Use File Register of Each Program" is enabled and monitoring is performed for the file registers, unintended data may be obtained. ( GX Works3 Operating Manual)

\*<sup>1</sup> Because the timing of monitoring (during execution of a program or the END processing) cannot be specified, values of ongoing operation may be read out.

# 23 DATA LOGGING FUNCTION

This function collects data at a specified interval or a desired timing and stores them as a file on the SD memory card. The entire data logging function allows the entry of up to 10 data logging settings.



\*1 For details on logging types, refer to the following. (👉 Page 146 Logging Type)

\*2 For details on the data to be collected, refer to the following. (👉 Page 140 Data To be Collected)

## Point

For system configuration, specifications, the procedure for operating and configuring this function in the CPU Module Logging Configuration Tool, which are not covered here, refer to the CPU Module Logging Configuration Tool. (👉 Page 376 How to Use CPU Module Logging Configuration Tool)

# 23.1 Data To be Collected

This section describes the data to be collected by data logging.

## Number of points of data

The data logging function can collect up to 1280 data records. (10 settings × 128 records)<sup>\*1</sup>

\*1 Duplicate data records are counted as distinct.

## Data type

The following table shows the number of data records for each data type.

Data type	Number of data records
Bit	1
Word (signed)	1
Double word (signed)	2
Word (unsigned)	1
Double word (unsigned)	2
Single-precision real number	2
Double-precision real number	4
Time	2
Character string <sup>*1</sup>	Specified size/2 <sup>*2</sup>
Numeric string	Specified size/2 <sup>*2</sup>

\*1 Outputs the entered character code.

\*2 The specified size can be 1 to 256. If the specified size is an odd number, the number of data records is rounded to the next higher integer. Example: The number of data records is 3 if the specified size is 5.

## Data to be collected

The data for the following devices can be specified to be collected.

Type	Device <sup>*1</sup>
Bit device <sup>*2</sup>	X, DX, Y, DY, M, L, F, SM, V, B, SB, T (contact) <sup>*4</sup> , T (coil) <sup>*4</sup> , ST (contact) <sup>*4</sup> , ST (coil) <sup>*4</sup> , C (contact) <sup>*4</sup> , C (coil) <sup>*4</sup> , LT (contact) <sup>*4</sup> , LT (coil) <sup>*4</sup> , LST (contact) <sup>*4</sup> , LST (coil) <sup>*4</sup> , LC (contact) <sup>*4</sup> , LC (coil) <sup>*4</sup> , FX, FY, Jn\X, Jn\Y, Jn\SB, Jn\B
Word device <sup>*3</sup>	T (current value), ST (current value), C (current value), D, SD, W, SW, RD, R, ZR, Z, FD, Un\G, Jn\W, Jn\SW, U3En\HG
Double-word device	LT (current value), LST (current value), LC (current value), LZ

\*1 A local device, index modification, and indirect specification cannot be specified.

\*2 For bit devices, digit specification is allowed for K1 to K8 only.

\*3 For word devices, bit specification is allowed.

\*4 To specify these devices with the CPU Module Logging Configuration Tool, use T (contact): TS, T (coil): TC, ST (contact): STS, ST (coil): STC, C (contact): CS, C (coil): CC, LT (contact): LTS, LT (coil): LTC, LST (contact): LSTS, LST (coil): LSTC, LC (contact): LCS, and LC (coil): LCC.

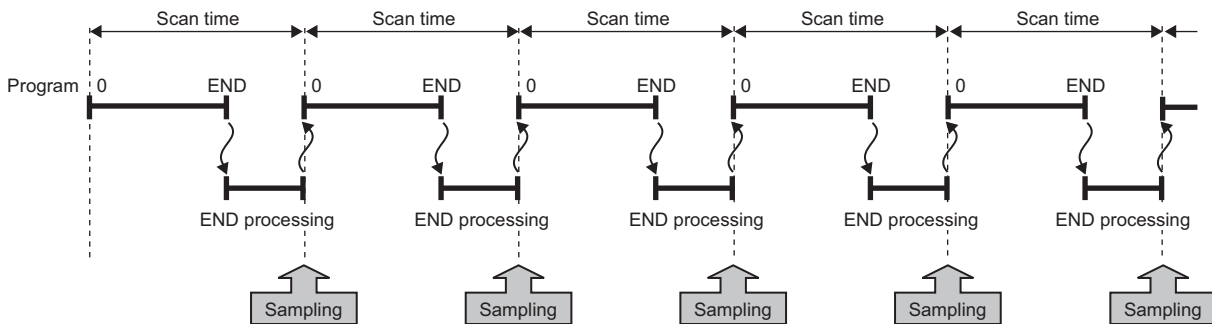
# 23.2 Data Collection Conditions

This section describes the timing when data is collected and the conditions under which data is collected.

Data Collection Conditions		Description
Each scan		Collects data during the END processing of each scan.
Time specification	Data collection at specified time interval	Collects data at specified time interval.
	Data collection during the END processing after specified time interval	Collects data during the END processing after specified time interval.
Interruption occurrence		Collects data at specified time interval after the execution of an interrupt program.
Condition specification	Device specification	Collects data when the monitored data meets the specified condition during the END processing.
	Step No. specification	Collects data when the specified condition is met immediately before the execution of the specified step.

## Each scan

Collects data during the END processing of each scan.



## Time specification

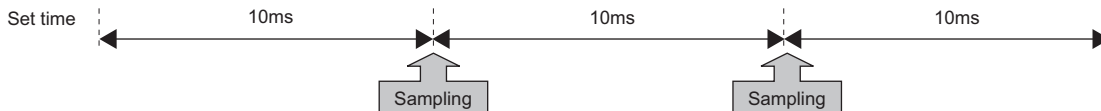
Lets the user specify the collection time interval. The user can also specify the timing of data collection using advanced settings.

### Data collection at specified time interval

Collects data at specified time interval.

**Ex.**

When the time interval is set to 10 milliseconds

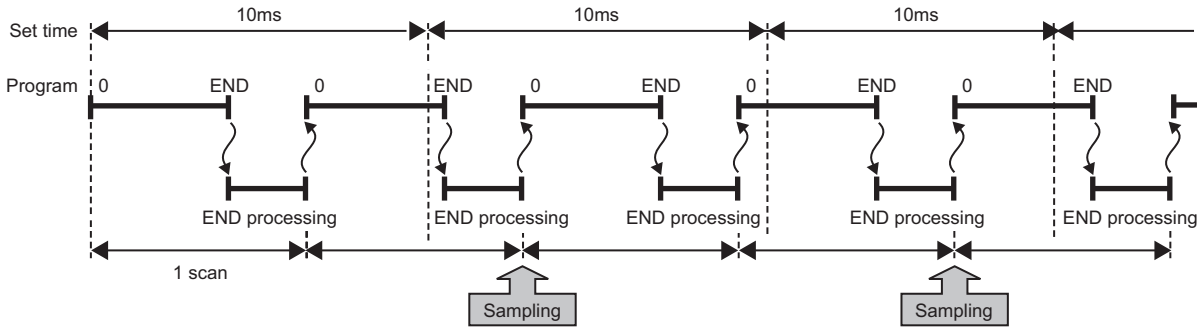


## Data collection during the END processing after specified time interval

This option causes data collection to be performed at the timing of the END processing rather than during the course of program execution. Ensure that the "Scan time" is less than "Time specification." If the scan time is longer than the specified time and the collection interval or the collection timing occurs more than once during the same scan, data is collected only once during the END processing. Data collection is performed on a scan by scan basis, which is the same behavior as when "Each scan" is used.

**Ex.**

When the time interval is set to 10 milliseconds

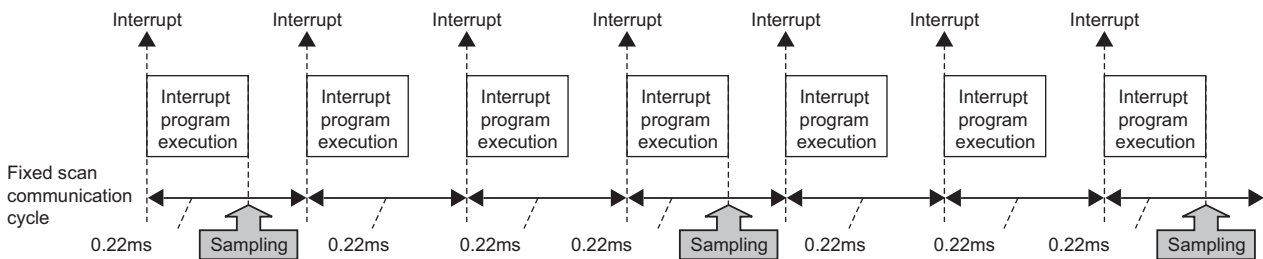


## Interrupt occurrence

Collects data at specified time interval after the execution of an interrupt program. To specify when to collect data, use "Interrupt cycle specified sampling intervals" × "Interval." Interrupt pointers which can be specified are I28 to I31, I44, I45, I48, and I49.

**Ex.**

When multiple CPU synchronous interrupt (I45) is used (multiple CPU synchronous interrupt is set to 0.22 milliseconds and interval to 3):



### Point

Since data collection is performed at the timing of the execution of the interrupt program, data is collected only when all of the following conditions are met:

- The specified interrupt meets the operating condition.
- The specified interrupt pointer exists in the program.
- The current state is an EI state and the interrupt mask of the interrupt pointer has been reset.



# Condition specification

This option lets the user set the data collection timing by specifying the device data conditions and step number. The AND condition using a combination of "Device specification" and "Step No. specification" results in the collection of data at the time when both conditions are established.

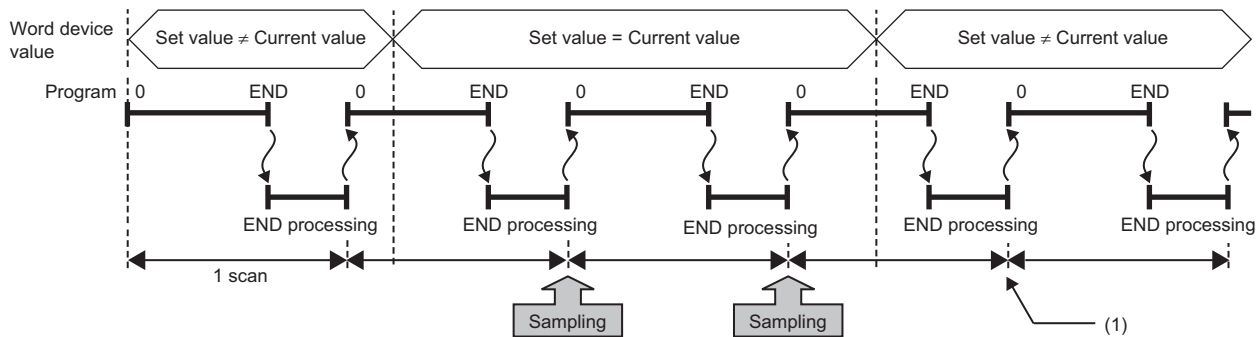
## Device specification

Collects data when the monitored data meets the specified condition during the END processing.

### ■To collect data continuously while the conditions are met

The following conditional formula causes the data logging function to collect data continuously while the conditions are met:

- =: When the current value of the monitored data is equal to the comparison value
- ≠: When the current value of the monitored data is not equal to the comparison value
- ≥: When the current value of the monitored data is equal to or larger than the comparison value
- >: When the current value of the monitored data is larger than the comparison value
- ≤: When the current value of the monitored data is equal to or smaller than the comparison value
- <: When the current value of the monitored data is smaller than the comparison value

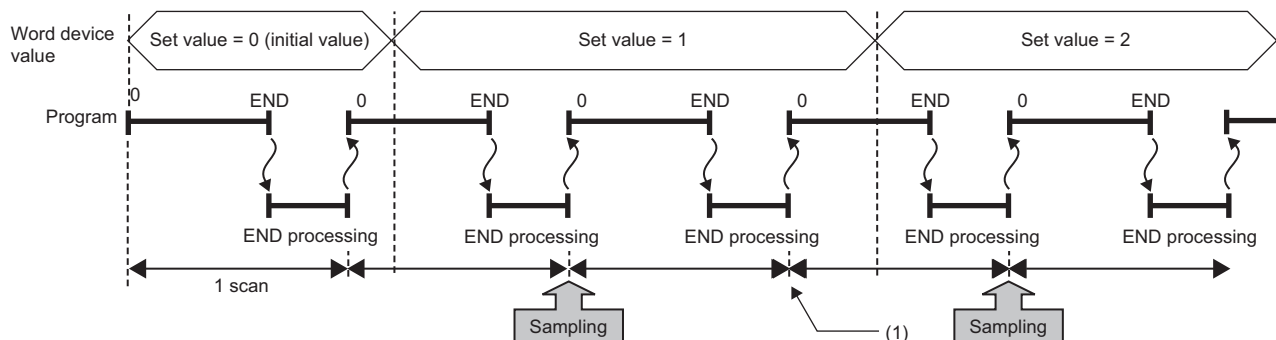


(1) During the END processing, the data logging function does not collect data because the conditions are not met.

### ■To collect data only when the state changes

The following conditional formula causes the data logging function to collect data only during the END processing for the scans where the conditional formula is met. It does not collect data for any single scan where the conditional formula is not met during the END processing (even if the conditional formula is met before the END processing is initiated).

- ↑: When the specified data turns off and on
- ↓: When the specified data turns on and off
- At change: When the current value of the specified data changes



(1) The data logging function does not collect data because there has been no change in state since the last scan.

## ■Specifying the monitored data

For monitored data, the following devices can be specified. The data types that can be selected include bit/word (unsigned), word (signed), double word (unsigned), and double word (signed).

Type	Device <sup>*1</sup>
Bit device <sup>*2</sup>	X, Y, M, L, F, SM, V, B, SB, T (contact) <sup>*4</sup> , ST (contact) <sup>*4</sup> , C (contact) <sup>*4</sup> , LT (contact) <sup>*4</sup> , LST (contact) <sup>*4</sup> , LC (contact) <sup>*4</sup> , FX, FY, Jn\X, Jn\Y, Jn\SB, Jn\B
Word device <sup>*3</sup>	T (current value), ST (current value), C (current value), D, SD, W, SW, RD, R, ZR, Z, FD, Un\G, Jn\W, Jn\SW, U3En\G, U3En\HG
Double-word device	LT (current value), LST (current value), LC (current value), LZ

\*1 A local device, index modification, and indirect specification cannot be specified.

\*2 For bit devices, digit specification is allowed for K1 to K8 only.

\*3 For word devices, bit specification is allowed.

\*4 To specify these devices with the CPU Module Logging Configuration Tool, use T (contact): TS, ST (contact): STS, C (contact): CS, LT (contact): LTS, LST (contact): LSTS, and LC (contact): LCS.

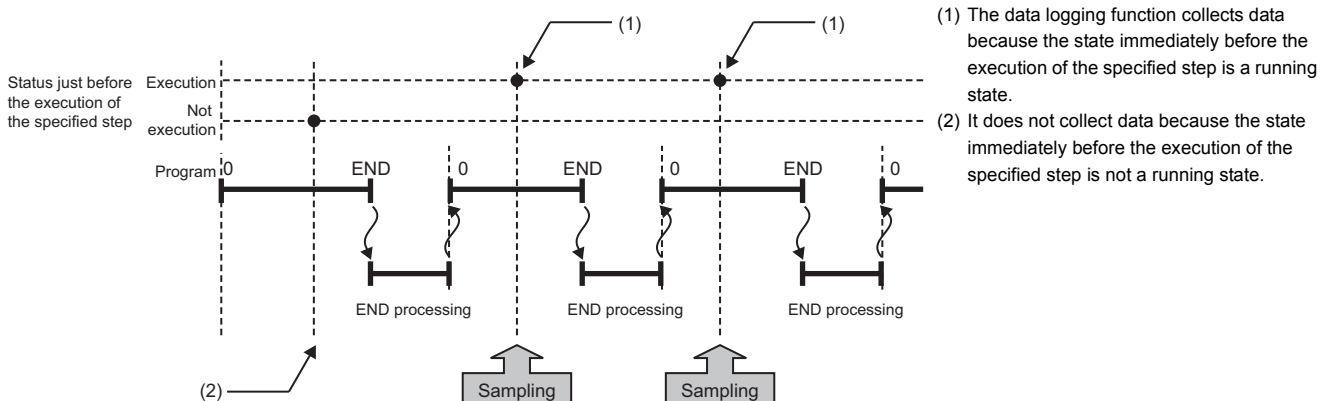
## Step No. specification

Collects data when the specified condition is met immediately before the execution of the specified step.

### ■To collect data continuously while the execution conditions are met

The following execution conditions cause the data logging function to collect data continuously while the execution condition are met:

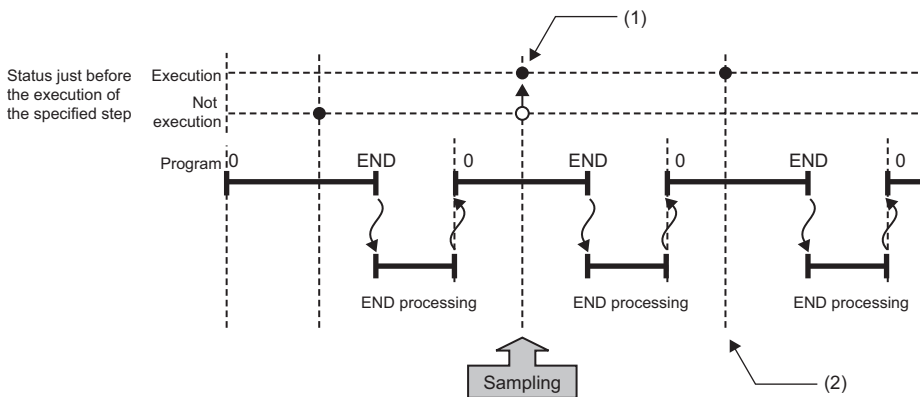
- Always: Executes the specified step regardless of the state immediately before the execution of it.
- In the specified condition satisfied: Executes the specified step if the state immediately before the execution is a running state.
- In the specified condition not satisfied: Executes the specified step if the state immediately before the execution is not a running state.



## ■ To collect data only when the execution conditions are met

The following execution conditions cause the data logging function to collect data only when the execution conditions are met:

- On the rising edge of the specified condition: Executes the specified step if the state changes from non-running to running immediately before the execution of it.
- On the falling edge of the specified condition: Executes the specified step if the state changes from running to non-running immediately before the execution of it.



- (1) The data logging function collects data because the state changes from running to non-running immediately before the execution of the specified step.
- (2) The data logging function does not collect data because there has been no change in state since the last scan.

### Point

If the specified step is contained between FOR and NEXT loop of instructions, the data logging function collects the data for only the first iteration of the loop where the specified conditions are met.

### Restriction

Since the step number cannot be checked in the following programs, the step number cannot be specified.

- Program having multiple parts
- ST program
- FB program

## 23.3 Logging Type

The following table describes available methods of data collection:

Logging type	Data collection method	Application
Continuous logging	Continuously collects specified data at specified interval or timing.	Allows the user to continuously monitor the content of specified data.
Trigger logging	Collects specified data at specified interval or timing and extracts a specified number of data records before and after the satisfaction of a trigger condition.	Allows the user to monitor the content of specified data before and after the satisfaction of a trigger condition.

### Operating procedure for continuous logging

In continuous logging, the CPU module stores specified data in its internal buffer at a specified collection interval or timing and, at the time of a file save operation, it saves the data in a data logging file residing in the storage memory.

1. Write the settings into data memory or SD memory card using the CPU Module Logging Configuration Tool.
2. Instruct the CPU Module Logging Configuration Tool to start data logging. The data logging settings are registered and continuous logging begins. (The special relay (data logging start) turns on.)
3. Data collection finishes upon reaching "Number of files to be saved" specified as part of the "Stop" setting configured in "Operation when the number of files exceeds the set value."
4. Specify the desired file in the storage memory to read the results of data logging.

#### Point

In continuous logging, data logging files are continuously created, thus allowing the user to read the results of data logging any time without having to wait for the completion of collection.

### To stop continuous logging

The user can completely stop data logging by instructing the CPU Module Logging Configuration Tool to stop data logging and unregister the data logging settings stored in the CPU module. (The special relay (data logging start) turns off.)

### To suspend/resume continuous logging

The user can suspend data logging with the data logging settings remaining intact by doing either of the following:

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

To resume continuous logging from suspension, do either of the following:

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

# Operating procedure for trigger logging

In trigger logging, the CPU module stores specified data in its internal buffer at a specified collection interval or timing; it extracts a specified number of data records before and after the satisfaction of a trigger condition and saves the extracted data in a data logging file residing in the storage memory. Note that data collection is performed not only at the specified collection interval or timing but also when a trigger condition is met. In addition, once a trigger condition is met, any subsequent trigger conditions are ignored.

1. Write the settings into data memory or SD memory card using the CPU Module Logging Configuration Tool.
2. Instruct the CPU Module Logging Configuration Tool to start data logging. The data logging settings are registered and trigger logging begins. (The special relay (data logging start) turns on.)
3. Wait until the trigger condition is met. (Trigger standby)
4. The data specified in CPU Module Logging Configuration Tool is collected. (Trigger condition met)
5. Data collection is completed by collecting as much data as the number of records specified in the CPU Module Logging Configuration Tool and writing the collected data into the storage memory.
6. Specify the desired file in the storage memory to read the results of data logging.

## To stop trigger logging

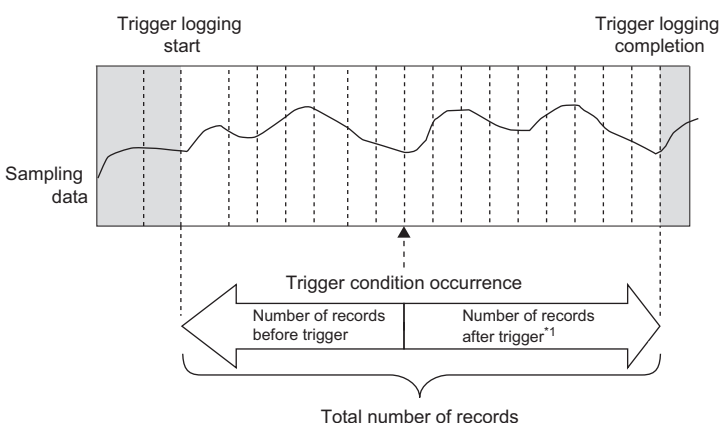
The user can completely stop data logging by instructing the CPU Module Logging Configuration Tool to stop data logging and unregister the data logging settings stored in the CPU module. (The special relay (data logging start) turns off.)

## To suspend/resume trigger logging

The user can suspend data logging with the data logging settings remaining intact by instructing the CPU Module Logging Configuration Tool to suspend data logging. (The special relay (data logging start) turns off.) Trigger logging can be resumed from suspension by instructing the CPU Module Logging Configuration Tool to resume data logging. (The special relay (data logging start) turns on.)

## Number of records

Specify the number of records to be collected before and after the satisfaction of a trigger condition. (☞ Page 400 Number of records)



\*1 This number includes the record exactly at the time when the trigger condition is met.

### Point

After starting data logging, if the trigger condition is met before data collection of the specified number of records (before trigger) is completed, the number of sampled records will be less than the specified.

## 23.4 Trigger Condition

The following table lists the conditions to be used as a trigger.

Trigger condition		Description
Condition specification	Device change specification	A trigger occurs when the monitored data meets the specified condition.
	Step No. specification	A trigger occurs when the specified condition is met immediately before the execution of the specified step.
When trigger instruction executed		A trigger occurs when the LOGTRG instruction is executed.

### Condition specification

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

#### Device change specification

A trigger occurs when the monitored data meets the specified condition.

- ↑: When the specified data turns off and on
- ↓: When the specified data turns on and off
- =: When the monitored data is equal to the comparison value, regardless of whether or not its current value is equal.
- ≠: When the monitored data is not equal to the comparison value, regardless of whether or not its current value is equal.
- ≥: When the monitored data is equal to or larger than the comparison value, regardless of whether or not its current value is equal.
- >: When the monitored data is larger than the comparison value, regardless of whether or not its current value is equal.
- ≤: When the monitored data is equal to or smaller than the comparison value, regardless of whether or not its current value is equal.
- <: When the monitored data is smaller than the comparison value, regardless of whether or not its current value is equal.
- At change: When the current value of the specified data changes

#### ■Specifying the monitored data

For the device change specification, monitored data can be configured to be collected from the devices listed in the following table. The data types that can be selected include bit/word (unsigned), word (signed), double word (unsigned), and double word (signed). If double word (unsigned) or double word (signed) is specified, a trigger occurs only when as much data as one double word is written. No trigger occurs when only the upper or lower word of a double word is written.

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

\*1 A local device, index modification, and indirect specification cannot be specified.

\*2 For bit devices, digit specification is not supported.

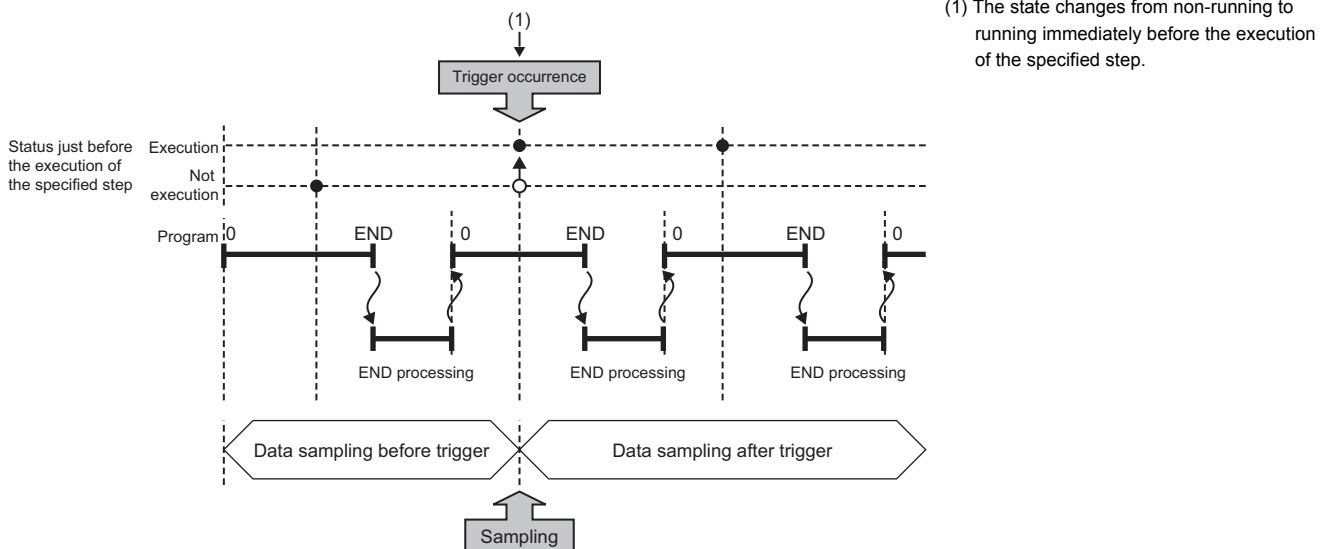
\*3 For word devices, bit specification is allowed.

\*4 To specify these devices with the CPU Module Logging Configuration Tool, use T (contact): TS, ST (contact): STS, C (contact): CS, LT (contact): LTS, LST (contact): LSTS, and LC (contact): LCS.

## Step No. specification

A trigger occurs when the specified condition is met immediately before the execution of the specified step.

- Always: Executes the specified step regardless of the state immediately before the execution of it.
- In the specified condition satisfied: Executes the specified step if the state immediately before the execution is a running state.
- In the specified condition not satisfied: Executes the specified step if the state immediately before the execution is not a running state.
- On the rising edge of the specified condition: Executes the specified step if the state changes from non-running to running immediately before the execution of it.
- On the falling edge of the specified condition: Executes the specified step if the state changes from running to non-running immediately before the execution of it.



### Point

If the specified step is contained between FOR and NEXT loop of instructions, only the first iteration of the loop where the specified conditions are met is handled as a trigger.

### Restriction


Since the step number cannot be checked in the following programs, the step number cannot be specified.

- Program having multiple parts
- ST program
- FB program

## When trigger instruction executed

A trigger occurs when the LOGTRG instruction is executed. MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

# 23.5 States of the Data Logging Function

The data logging function has various states that can be classified into data logging and storage. The data logging state and storage state can be checked by CPU Module Logging Configuration Tool. (  Page 411 Logging status and operation)

## Data logging states

The following table lists all the possible data logging states.

Data logging states	Description
Stop	No data logging settings are registered and data collection is inactive.
Stop (Execute other functions after completing collection)	Transition from "Collection completed" to "Stop" has occurred due to the execution of another function* <sup>1</sup> .
Stop (Execute other functions after error)	Transition from "Error" to "Stop" has occurred due to the execution of another function* <sup>1</sup> .
Waiting RUN Not collected	Data collection has not yet begun because the CPU module is not in the RUN mode.
Waiting start Not collected	Data collection is inactive, waiting for the start command.
Pause	Data logging is suspended and data collection is inactive.
Waiting to establish collection conditions Not collected	Waiting for the first collection timing after the start command.
Collecting	Continuous logging is active and collecting data.
Waiting trigger Collecting before trigger	Trigger logging is active and collecting data, waiting until the trigger condition is met.
Collecting after trigger	Trigger logging is active and collecting data after the trigger condition is met.
Collection completed	<ul style="list-style-type: none"><li>• Continuous logging: Data collection has finished upon reaching "Number of files to be saved" specified as part of the "Stop" setting configured in "Operation when the number of files exceeds the set value."</li><li>• Trigger logging: Has finished collecting as much data as the specified number of records.</li></ul>
Error	Data logging has failed due to the occurrence of an error.

\*1 The execution of another function includes:

- Execution of data logging with the same trigger conditions (trigger conditions = the specified conditions)
- Auto logging
- Online change

## Storage states

The following table lists all the possible storage states.

Storage states	Description
Unsaved	Has not yet stored the collected data into the SD memory card.
Saving in progress	Has begun but not yet finished storing the collected data into the SD memory card.
Save completed	Has finished storing the collected data as much as the specified number of records into the SD memory card.* <sup>1</sup>

\*1 If the data logging function has not yet collected and stored as much data as the specified number of records (i.e., either data logging has been stopped or suspended before collecting or storing as the specified number of records or the CPU module has been stopped), it completes the storage operation by storing all the data that has been collected into the internal buffer. It does not store data, however, before the trigger condition is met.



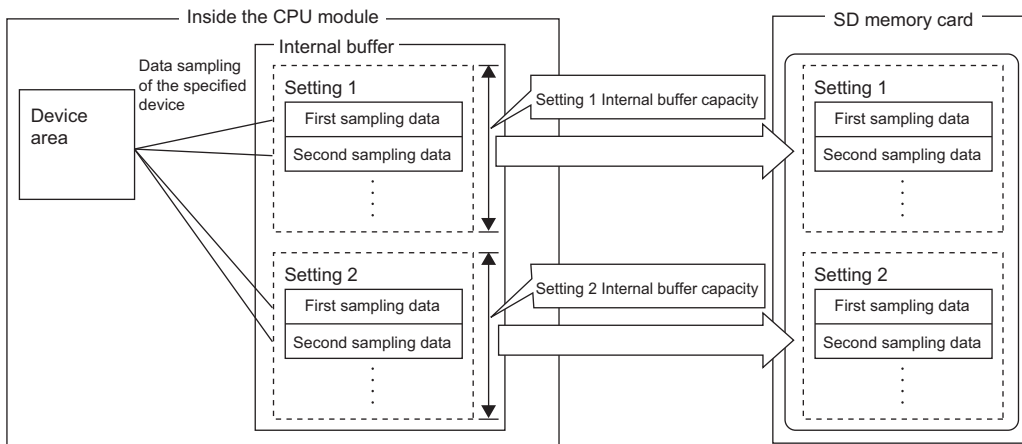
# LED status

Whether the data logging function is active or not can be checked by the LED of the CPU module.

States of the Data Logging Function		LED status
At registration	The CPU module is now populated with an SD card that contains the settings to enable the auto logging function after the user instructs the CPU Module Logging Configuration Tool to start data logging.	<ul style="list-style-type: none"> <li>• CARD READY LED: On</li> <li>• CARD ACCESS LED: Off</li> <li>• FUNCTION LED: On</li> </ul>
Execution in progress	All of the registered data logging sessions are in the state "Waiting RUN Not collected", "Waiting start Not collected", "Pause", "Waiting to establish collection conditions Not collected", or "Waiting trigger Collecting before trigger".	
	One or more of the registered data logging sessions are in the state "Collecting" (including the data being saved) or "Collecting after trigger" (including the data being saved).	<ul style="list-style-type: none"> <li>• CARD READY LED: On</li> <li>• CARD ACCESS LED: Turns on when the SD memory card is accessed</li> <li>• FUNCTION LED: Flashes slowly (every one second)</li> </ul>
At completion	All of the registered data logging sessions have finished (or failed due to an error).	<ul style="list-style-type: none"> <li>• CARD READY LED: On</li> <li>• CARD ACCESS LED: Off</li> <li>• FUNCTION LED: Flashes at normal rate (every 200 milliseconds)</li> </ul>

## 23.6 Steps Until the Collected Data is Saved

The collected data is temporarily stored in the specified internal buffer. (☞ Page 154 Internal Buffer) The data stored in the internal buffer is stored into the SD memory card at the time of a file save operation.



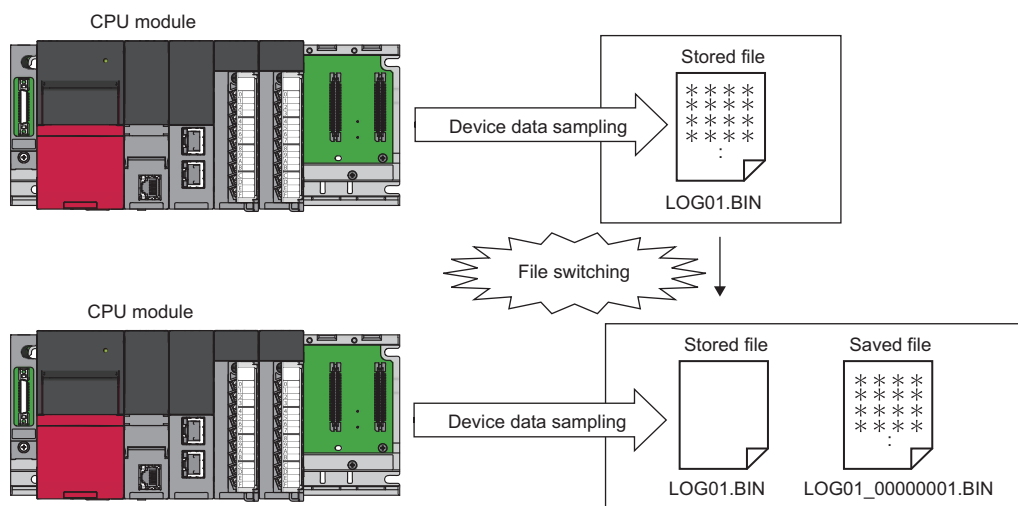
### Switching to a storage file

The data collected by data logging is temporarily stored in a stack file that resides in the SD memory card. The stack file can be switched to a storage file to free the space in the SD memory card.

#### How file switching (log rotation) works

File switching works as follows:

1. The CPU module writes collected data into a stack file (such as LOG01.BIN).
2. It changes the file name when the storage file switching condition is met. \*1\*2
3. It creates a new stack file.
4. It continues to write collected data into the newly created stack file.



\*1 The file name format can be customized. (☞ Page 403 Save)

\*2 The file number of the most recent storage file is stored in the special register (Latest storage file number).

## File switching condition

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

Setting item	Description
Number of records	Specify the number of records within the range of 1 to 65500.
File size <sup>*1</sup>	Specify the number of kilobytes within the range of 10 to 16384KB.

<sup>\*1</sup> File switching occurs before the file grows beyond the specified size.

However, file switching occurs regardless of the setting when:

- the number of records has reached 65500;
- the file size has reached 16MB;
- the CPU module is stopped or suspended/resumed: or
- data logging is started and there is an existing stack file.

### Point

Trigger logging does not require the configuration of these settings because the stack file is automatically switched to a storage file after as much data as the specified number of records is written into the stack file. SM1218 (logging data storage file switching in progress) can be used to check if storage file switching is in progress.

## Storage file

The CPU module creates a subfolder ("storage file container folder") under the file storage folder and writes storage files to that storage file container folder. One storage file container folder can contain up to 256 storage files. When the files contained in the current storage file container folder reach the maximum number, the CPU module creates a new storage file container folder at the time of next storage file switching and begins writing storage files to that new folder. The number of files that can be contained in one file storage folder is configurable within the range of 1 to 65535.

### Point

The base folder name of a storage file container folder is an eight-digit (hexadecimal) number. This number matches the lowest of the serial numbers of the files contained in the directory. Date and time stamps can be appended to the folder name.

## Action to take when the maximum number of storage files is exceeded

Either "Overwrite" or "Stop"<sup>\*1</sup> can be selected as the action to take when the maximum number of storage files is exceeded.

\*1 This settings is not configurable for trigger logging.

### ■When "Overwrite" is selected

When the storage file switching condition is met after the specified maximum number of storage files is exceeded, the CPU module deletes the file with the lowest serial number and creates a new file that has a serial number incremented by one from the highest serial number, allowing data logging to continue. In addition, if deleting the file with the lowest serial number results in an empty folder, the CPU module deletes that folder as well.

### ■When "Stop" is selected

As described in the following table, the action differs depending on when the specified maximum number of storage files is exceeded.

Occurrence timing	Occurrence condition	Behavior
When data logging is started	There exist more storage files than the specified maximum number when data logging is started.	<ul style="list-style-type: none"><li>• If an attempt is made to register the data logging settings from within the CPU Module Logging Configuration Tool, an error is generated, resulting in failure to run data logging.</li><li>• If an attempt is made to register<sup>*1</sup> the data logging settings from outside the CPU Module Logging Configuration Tool, the special relay (data logging error) turns on and the special register (data logging error cause) stores the cause of the error, resulting in failure to run data logging.</li></ul>
While data logging is running	The specified maximum number of storage files is reached due to file switching upon the satisfaction of the storage file switching condition.	Data logging stops and enters into the completion state. The special relay (data logging completed) turns on to indicate that data logging is completed.

\*1 When an attempt is made to register the data logging settings again, the CPU module enters into the data logging completed state. The special relay (data logging completed) turns on to indicate that data logging is completed.

## 23.7 Internal Buffer

The internal buffer is a system area used to temporarily store collected data.

### Internal buffer capacity setting

This setting can be specified using an engineering tool (🔧 Page 110 INTERNAL BUFFER CAPACITY SETTING). For trigger logging, increasing the internal buffer capacity allows for a larger number of data records collected before a trigger and also helps to prevent processing overflow. If the free space in the internal buffer is still insufficient after increasing the internal buffer capacity, use the following workarounds:

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

### Amount of internal buffer consumed

This value can be calculated by multiplying "Number of data points" by 2 bytes. Note, however, that additional space is consumed by columns configured for output, as indicated below:

- Date/time column: 10 bytes
- Data collection interval column: 8 bytes
- Execution step No. column: 10 bytes
- Execution program No. column: 2 bytes
- Index column: 4 bytes

**Ex.**

When data logging is configured to collect as much data as one setting x 128 records and output all of the columns (i.e., maximum allowable configuration):

$$128 \times 2 + 2 + (10 + 8 + 10 + 2 + 4) = 290 \text{ bytes}$$

## 23.8 Data Logging File

This section describes data logging files.

### Storage format of data logging files

The following storage formats are available for data logging files. For details on the output format, format specifications, and output contents of each file, refer to the data output format. (☞ Page 382 Data output type)

File format	Application
Unicode text file format	This is a file format which can be open in generic-purpose application programs such as Excel and Notepad. GX LogViewer is also available for displaying data.
Binary file format	Comparing the Unicode text file format, the size of files is small and therefore quicker access to files is provided. GX LogViewer is also available for displaying data.

## 23.9 Missing Data

The term "missing data" means that some of the collected data is missing, resulting in data discontinuity.

### Conditions under which missing data occurs

Missing data occurs under the following conditions:

Item	Description
Processing overflow	Processing overflow has occurred due to failure to keep up with the specified collection interval/timing.*1
Operations for the CPU module	The CPU module has been stopped and run with "Operation at transition to RUN" set to "Auto Start."
	The CPU module has been turned off and on with "Operation at transition to RUN" set to "Auto Start."
	The CPU module has been reset and run with "Operation at transition to RUN" set to "Auto Start."
Operation from engineering tools, CPU Module Logging Configuration Tool, and external devices via protocols such as FTP, SLMP, and MC	<ul style="list-style-type: none"><li>• When the CPU module is suspended and restarted, and operation for displaying the logging state is performed from CPU Module Logging Configuration Tool</li><li>• File read*2, write, delete, or verification</li></ul>

\*1 Failure to collect data at the specified collection interval/timing due to the execution of a long-running instruction (such as FMOV) does not cause processing overflow or missing data.

\*2 The following operation also is included:

- Online operation which displays data by operation such as read from the programmable controller performed from an engineering tool (retrieval and display of a list of files on the CPU module)
- View of the event history (retrieval of the event history from the CPU module)

### Processing overflow

In normal cases when the usage of the internal buffer reaches the specified maximum capacity, the CPU module overwrites the data stored in the storage memory on a first-in first-out basis. If the internal buffer becomes full before all of the data stored in it is saved to the storage memory, however, the CPU module does not overwrite the existing data and stops storing data in the internal buffer, thus resulting in missing data. This situation is referred to as processing overflow. Upon the occurrence of overflow, the special register (Number of processing overflow occurrences) stores the number of times when processing overflow occurred.

## 23.10 Special Relay and Special Register Used by the Data Logging Function

For details on the special relays and special registers used by the data logging function, refer to the following:

- Special relay: Special relay related with the data logging function (☞ Page 330 Data logging function)
- Special register: Special register related with the data logging function (☞ Page 353 Data logging function)

# 23.11 Other Functions

This section describes some useful functions available in addition to the basic features of the data logging function.

## Setting the behavior at the time of a transition to RUN

This function configures the behavior of data logging that occurs when the user performs the following operations (transition to RUN) after the data logging setting are registered. (🔍 Page 405 Logging operation)

- Turning off and on the CPU module and switching to the RUN mode
- Resetting and running the CPU module
- Stopping and running the CPU module

### Point 🔍

The behavior at the time of a transition to RUN can be set individually for each setting number (1 to 10).

### Behavior at the time of a transition to RUN

The behavior at the time of a transition to RUN can be set to either of the following:

#### ■Auto start

After the user performs one of the operations listed above, data logging automatically starts when the operating status of the CPU module changes from STOP to RUN.

### Point 🔍

To start data logging, first, the user must instruct the CPU Module Logging Configuration Tool to start data logging.

#### ■Start by user operation

After the user performs one of the operations listed above, the data logging state is switched to "Waiting start Not collected" when the operating status of the CPU module changes from STOP to RUN. To start data logging again, the user must instruct the CPU Module Logging Configuration Tool to start data logging.

### Data logging behavior that occurs after operating status of CPU module has changed

Data logging does not continue when the operating state of the CPU module changes from RUN to STOP or PAUSE after it has been started. The data logging state changes to "Waiting RUN Not collected" and data collection is stopped.

# Auto logging

When inserting an SD memory card, which holds data logging setting, into the CPU module, the data logging automatically starts based on the data logging setting information on the SD memory card.

## How to use auto logging

This section describes how to use auto logging.

1. Prepare an SD memory card that contains data logging settings as well as common settings (auto logging common settings) in which auto logging is enabled.
2. Insert the SD memory card prepared in step 1 into the CPU module while it is running.

## Conditions for auto logging completion

Auto logging completes when a completion condition is met as described in the following table. These completion conditions can also be configured in combination with each other. When they are configured in combination, auto logging completes as soon as one of the conditions is met.

Completion condition	Description
Data logging stop	Choose one of the following: <ul style="list-style-type: none"><li>• When all data loggings stop</li><li>• When any of the data loggings stops</li></ul>
Complete with timer	Auto logging is stopped when the specified time has elapsed after the start of data logging. Elapsed time setting range: 1 second to 86400 seconds (unit: second)

23

### ■When "Data logging stop" is selected

Auto logging is assumed to be completed if all the settings or any of auto logging stops\*1.

- \*1 Here the term "stop" means one of the following:
- Continuous logging: When the "Number of files to be saved" setting configured in the save setting has been exceeded and data logging has been completed.
  - Trigger logging: When as much data as the number of records specified in "Number of records" has been collected, the collected data has been written to the SD memory card, and data logging has been completed.
  - When the user has instructed the CPU Module Logging Configuration Tool to stop data logging.

#### Point

- If "When any of the data loggings stops" is selected, the behavior of any other data logging sessions than stopped is the same as the behavior that occurs upon the elapse of the time configured using "Complete with timer."
- When completing auto logging for continuous logging, do not select "Overwrite" for the operation at the time when the number of files exceeds the save setting because doing so results in failure to stop data logging; instead, select "Stop."

### ■When "Complete with timer" is selected

When the operating time since the start of data logging reaches the specified time, the CPU module completes auto logging by moving all the data collected so far from the internal buffer to the SD memory card. If trigger logging has not yet collected as much data as the number of records specified "Number of logging lines", however, the CPU module does not store any data including the collected data.

#### Point

The timer is cleared to zero when auto logging is suspended and resumed by stopping and running the CPU module or turning off and on the CPU module or resetting it before the completion of auto logging. If auto logging is suspended by the turning on and off of the special relay (Data logging suspend/resume flag), the timer continues to run even while data logging is suspended.

## Conditions under which auto logging does not start

Doing any of the following operations once auto logging is completed does not start auto logging:

- Turn off and on the power
- Reset
- Stop and run

## Behavior of auto logging that occurs when entering into the RUN mode

If some, not all, of data logging sessions configured using auto logging have been completed when entering into the RUN mode, only the uncompleted data logging sessions are started.

## Behavior of auto logging that occurs when an error is generated

When an error is generated, auto logging behaves as follows:

### ■When registration has failed at the start of auto logging

If auto logging cannot be started due to failure of registration at the start of auto logging, it behaves in the same way as when it is completed.

### ■When an error has occurred during the execution of auto logging

An error that occurs during the execution of auto logging does not constitute the auto logging completion condition since it does not prevent data logging from being resumed. If such an error is resulting from an online change during the execution of auto logging, however, it constitutes the auto logging completion condition because it prevents data logging from being resumed. \*1

\*1 Applies only when "When all data loggings stop" is selected as part of the auto logging completion condition.

## SD memory card replacement

SD memory cards can be replaced using the SD memory card forced disable function even while data logging is in progress. (☞ Page 126 SD Memory Card Forced Disable) This function works by disabling data writes to the SD memory card while allowing data collection to continue. (Data collection continues in accordance with the settings registered when data logging is started.)

### Point

If SD memory card replacement causes processing overflow, make adjustments by changing the collection interval, internal buffer capacity, or other settings.

## Behavior during SD memory card replacement

If the internal buffer becomes full during the time between SD memory card replacement and the resumption of data writes to the SD memory card, processing overflow occurs resulting in missing data.

## Storage file numbers after SD memory card replacement

The numbering of the first storage file created after SD memory card replacement differs depending on the storage file switching condition, as described in the following table.

Storage file switching condition	Storage file numbers after SD memory card replacement
Overwrite	Numbering continues from the number of the last storage file contained in the replaced SD memory card.
Stop	Numbering begins at 00000001.

### Point

If the new SD memory card contains a "LOGGING" folder and its subfolders, folder deletion takes time, possibly resulting in missing data. Ensure that the new SD memory card does not contain a "LOGGING" folder.



## Logging state during SD memory card replacement

SD memory cards can be replaced without depending on the current data logging state. SD memory card replacement results in the deletion of the LOGGING folder if the data logging state is other than below:

- Stop
- Stop (Execute other functions after completing collection)
- Stop (Execute other functions after error)

## Operations during SD memory card replacement

If one of the following operations is performed during the time between the removal and installation of SD memory cards, any data collected during that time will not be stored in the new SD memory card.

- Stop and run<sup>\*1</sup>
- Power off and on<sup>\*1</sup>
- Reset<sup>\*1</sup>
- Suspend data logging
- Stop data logging

\*1 An error is generated if data logging was previously running based on the setting file contained in the replaced SD memory card.

## Operations after SD memory card replacement

If the SD memory card was replaced while data logging was running based on the data logging setting file contained in the SD memory card, the data logging setting file contained in the new SD memory card is used when data logging is started by one of the following operations. If the new SD memory card does not contain the data logging setting file, data logging is not started.

- Stop and run
- Power on and run
- Reset and run
- Data logging start

## Stack file remaining in the replaced SD memory card

Replacing an SD memory card that contains a stack file may result in the storage file remaining in the replaced SD memory card along with storage files. If the stack file is remaining in the replaced SD memory card, recover the latest data contained in the stack file by doing the following:

- Retrieve the data from the stack file and combine the data with a storage file.
- Save the stack file as a storage file.

# 23.12 SD Memory Card Life When the Data Logging Function is Used

An SD memory card has a life (restriction on writing data). The following shows the calculation method of an SD memory card life when the data logging function is used. Note that the actual life of the card varies depending on the use conditions and environment. Therefore, use the calculated life as a rough standard for the replacement of the card.


## Calculation formula of SD memory card life

SD memory card life (year) = Total size of data that can be written (G bytes) ÷ Size of data to be written per year (G bytes/year)

## Total size of data that can be written

Capacity × Number of writes<sup>\*1</sup>

<sup>\*1</sup> For the capacity of applicable SD memory cards and the number of writes, refer to the following.

 MELSEC iQ-R Module Configuration Manual

## Size of data to be written per year



The size of data to be written per year is obtained by the following formula.

Size of data to be written per year (G bytes/year) =  $((DS1^{*1} + 6144) \times DN1 + \dots + (DSn^{*1} + 6144) \times DNn + (DCS1^{*1} + 6144) \times DCN1 + \dots + (DCSn^{*1} + 6144) \times DCNn) \div 1073741824$

<sup>\*1</sup> Round up DS<sub>n</sub> and DCS<sub>n</sub> to a multiple of 512.

DS<sub>n</sub>, DN<sub>n</sub>, DCS<sub>n</sub>, and DCN<sub>n</sub> are obtained as follows.

### ■Data logging data size per record (DS<sub>n</sub>)

Binary file output format: Refer to the data. ( Page 386 Binary file output format) Unicode text file format: Refer to the data row. ( Page 382 Unicode text file output type)

### ■Number of records for data logging per year (DN<sub>n</sub>)

Continuous logging:  $DNn = 60 \times 60 \times 24 \times 365 \div \text{Collection interval and timing (seconds)}^{*1} \times \text{Operating rate}^{*2}$



Trigger logging:  $DNn = \text{Total number of records}^{*3}$

<sup>\*1</sup> The value that is determined depending on the condition set in "Sampling" when "Continuous logging" is selected for the logging type. (When the value is determined in milliseconds, convert the value into seconds.)

<sup>\*2</sup> Calculate the ratio using the operating time per year of the CPU module. For example, if the operating time per year is 5000 hours, the operating rate is calculated as follows:  $5000 \div (24 \times 365) = 0.57$ .

<sup>\*3</sup> The value set in "Number of logging lines" when "Trigger logging" is selected for the logging type.

### ■Header size of data logging (DCS<sub>n</sub>)

Binary file output format: Refer to the header. ( Page 386 Binary file output format) Unicode text file format: Refer to the file information row to device comment row. ( Page 382 Unicode text file output type)

### ■Number of file switching times for the data logging per year (DCN<sub>n</sub>)

Calculate this number with an estimated number according to the save setting of the data logging and system operations. For example, when 1000 records are set in "Number of records" of "File switching timing" in the save setting and "Each scanning cycle" is specified for "Sampling interval" in the sampling setting, the time interval of the file switching is obtained by multiplying the scan time by 1000. Therefore, the number of file switching times for the data logging per year is obtained by the following formula:  $60 \times 60 \times 24 \times 365 \div (\text{Scan time (second)} \times 1000)$

# 23.13 Precautions to Take When Using the Data Logging Function

This section describes precautions to take when using the data logging function.

## Mutual exclusion of the data logging function

This section describes the mutual exclusion of the data logging function.

### ■When another function is executed during the execution of the data logging function

The following table lists the cases when another function is executed during the execution of the data logging function.

Function that has been already executed	Function to be executed later	Behavior
Data logging function	Data logging function	When the data logging is started using the CPU Module Logging Configuration Tool to the same data logging setting number via another route, the data logging to be executed later cannot be executed. However, the data logging to be executed later can be executed to a data logging setting number different from the data logging setting number currently being executed.  The data conditions cannot be specified as the trigger conditions for multiple trigger conditions at the same time.  For the execution of multiple data loggings, the data logging settings stored in different target memory areas cannot be executed at the same time.
	Auto logging	The auto logging cannot be executed during the execution of the data logging. (Even though an SD memory card where the auto logging setting is written is inserted, the auto logging does not start.)
	Online change	When a step number is set in the condition specification of the collection or trigger condition, the data logging function and online change may not be executed at the same time.
	Function specified in the internal buffer capacity setting *1	If the internal buffer capacity setting is changed to execute the subsequent function, attempting to start the subsequent function results in an error. The data logging continues to function normally.
	Function not specified in the internal buffer capacity setting	If the condition "Total capacity that is set in the internal buffer capacity setting + Internal buffer capacity that is set in other than the internal buffer capacity setting > 3072K bytes" is satisfied, attempting to start the subsequent function results in an error. The data logging continues to function normally.  If the internal buffer capacity setting is changed to execute the subsequent function, attempting to start the subsequent function results in an error. The data logging continues to function normally.
Auto logging	Data logging function	Another data logging cannot be executed during the execution of the auto logging.

\*1 The data logging function is not included here.

### ■When the data logging function is executed during the execution of another function

The following table lists the cases when the data logging function is executed during the execution of another function.

Function that has been already executed	Function to be executed later	Behavior
Online change	Data logging function	When a step number is set in the condition specification of the collection or trigger condition, the data logging function and online change may not be executed at the same time.
Function specified in the internal buffer capacity setting *1		If the internal buffer capacity setting is changed to execute data logging, attempting to start data logging results in an error. The function already in execution continues to function normally.
Function not specified in the internal buffer capacity setting		If the internal buffer capacity setting is changed to execute data logging, attempting to start data logging results in an error. The function already in execution continues to function normally.

\*1 The data logging function is not included here.

## Locations from which data logging can be performed

Data logging cannot be performed from multiple locations to the same setting number. The CPU module supports data logging performed concurrently at a maximum of 10 locations assigned to setting numbers 1 to 10.

## Retention and clearance of data logging settings

The data logging settings registered in the CPU module are latched and thus survive across a power cycle (power off and on) or reset of the CPU module in normal cases. In the following cases, however, the data logging status is cleared to the unregistered state and therefore the setting data must be written again:

- The CPU module is turned off and on or reset without an SD memory card that contains the data logging setting file.
- The replacement SD memory card does not contain the data logging setting file and the CPU module is turned off and on or reset. \*1

\*1 If the data logging setting file contained in the replacement (new) SD memory card is different from that contained in the replaced (old) SD memory card, register the data logging setting file contained in the replacement SD memory card.

## Behavior that occurs when trigger logging is resumed

If data logging is stopped or collection is suspended before the completion of trigger logging and subsequently data logging is run again, data collection begins from the initial state before trigger logging, rather than continuing from the last time.

## Stopping/suspending data logging using CPU Module Logging Configuration Tool

After data logging is stopped or suspended from within the CPU Module Logging Configuration Tool, all the data in the internal buffer is written into the target memory. If a small number of records or a small file size is specified as part of the storage file switching condition, writes to the target memory may take a longer time.

## Behavior against the failure to register data logging setting files

When an attempt to register multiple data logging setting files at the same time is made and fails for some of them, the CPU module behaves differently depending on whether the registration is via the CPU Module Logging Configuration Tool and program or via auto logging:

- If the registration is via the CPU Module Logging Configuration Tool and program, the CPU module runs data logging for the setting files that have been successfully registered.
- If the registration is via auto logging, the CPU module does not run any data logging session.

## Behavior upon change of the internal buffer capacity

When the internal buffer capacity is changed during execution of the data logging function, please note that:

- If the internal buffer capacity for the setting number of the running data logging is left empty to disable the capacity, an error occurs when the data logging is stopped and restarted (write to the programmable controller does not cause an error).
- If the internal buffer capacity of the setting number of the running data logging is changed to a smaller value, data may be lost when the data logging is stopped and restarted.

## Trigger condition should not be met during data logging registration

Ensure that the trigger condition is not met during data logging registration. If the trigger condition is met, the data logging settings cannot be registered.

## When file register is specified as specification device in condition specification

If a file register is specified as the specification device in the condition specification for the collection and trigger condition settings, do not change the file register file name or the block number of the file register once the data logging settings are registered. Doing so may possibly result in failure to successfully collect data logging results.

## When the data collection condition is set to "Time specification"

If "Time specification" is selected for the data collection condition, data collection is performed as an interrupt processing operation and therefore special care should be taken when configuring the data collection interval, the data collection timing, and the data logging processing time per scan. In the following cases, scans may take a longer time, possibly resulting in a watchdog timer error:

- The collection interval and the collection timing are so short that data logging is frequently performed during a single scan.
- The data to be collected is so much that the data logging processing time per scan is long.

## Numbering of the storage files used during data logging

If one or more numbered storage files already exist in the specified file storage folder and a new file is written, the new file is given a file name that uses a number incremented by one from the highest number among the existing files. If the file storage folder has no storage files but one or more storage file container folders exist, the new file is stored under the folder with the lowest number and it is given the same number as the folder. However, if there are 258 or more folders under the said conditions, a new folder is created and the file is given the same number as the new folder.

## Behavior that occurs while collected data is stored in the target memory

If one of the following operations is performed while collected data is stored in the target memory, any unsaved data is cleared and not reflected to the results:

- Turning power of the CPU module off and on
- Reset

If one of the following operations is performed, unsaved data continues to be stored in the target memory:

- Stopping the CPU module
- Stopping/suspending data logging from within the CPU Module Logging Configuration Tool
- Issuing the LOGTRGR instruction

## Creating files and folders

Under the "LOGGING" folder that contains data logging setting files and data logging files, do not attempt to create files or folders using a personal computer or other device. Doing so may result in deletion of files and folders.

## When collection is performed at the specified time

When "Time specification" is selected for the data collection condition and the collection at the specified time is selected rather than data collection during the END processing, check the collection interval by referring to the information reported in the collection interval column. Do not rely on the information reported in the date/time column because it may be incorrect due to clock accuracy error.

## Changing the clock data

Whatever changes, such as advancing or reverting the clock, are made to the clock data of the CPU module during data logging, the CPU module performs data collection at the specified collection interval/timing, but the date/time column in the output file reports the changed clock data.

## Events that are not recognized as a trigger condition

For trigger logging, the following events are not recognized as a trigger condition:

- A second trigger condition is met after the first trigger condition is met.
- The data condition specified as part of the trigger condition is met within the I49 interrupt program.

### Point

If a data condition is specified as part of the trigger condition, any trigger condition met during the execution of the I49 interrupt program will not be recognized as a trigger condition. In this case, using I48 instead of I49 allows for avoiding the timing when the condition is not met.

## Access to the SD memory card

If data logging is performed with a small setting of the data collection interval/timing or with a large number of records to be collected, access (read/write) to the SD memory card occurs so frequently that a delay occurs in completing the access. To avoid such a delay, use the following workarounds:

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

## Time-out error in a communication or dedicated instruction from a peripheral

Under the following conditions, a time-out error may occur in a communication or dedicated instruction issued from a peripheral to the CPU module while the data logging function is running:

### ■ Multiple data logging sessions are started concurrently

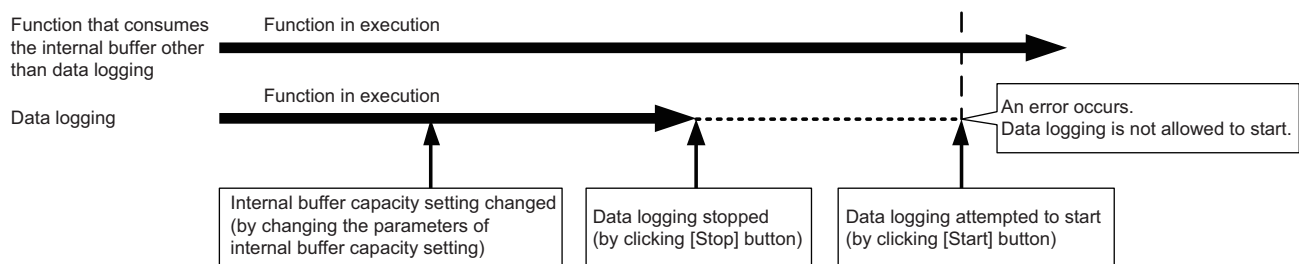
To fix a time-out error, review and correct the device/label access service processing settings, try to start the data logging sessions one after another, or take other appropriate measures.

### ■ Multiple data logging sessions' settings are registered to be started when the CPU module is stopped and run

To fix a time-out error, increase the time-out time period of the affected peripheral or reduce the number of data logging sessions or take other appropriate measures.

## Behavior at parameter change when functions consuming the internal buffer are active

If the internal buffer capacity setting is changed during the execution of functions that consume the internal buffer, attempting to start data logging results in an error, where the data logging fails to start.



## Changing the operating status of the CPU module

The operating state of the CPU module should not be changed until the completion of the following operations and registrations:

- Save of the data in the internal buffer by changing the state of the CPU module from RUN to STOP or instructing the CPU Module Logging Configuration Tool to stop or suspend the CPU module
- Registering multiple data logging sessions' settings<sup>\*1</sup>
- Registering data logging settings with any unused folders remaining in the storage memory<sup>\*1</sup>

<sup>\*1</sup> Data logging settings are also registered when the CPU module is stopped and run.

### Point

- To shorten the time required to register multiple data logging sessions' settings, reduce the number of data logging sessions.
- To shorten the time required to register data logging settings with any unused folders remaining in the storage memory, delete the unused folders before registration.

## Unicode text file

To view the data, use a software application with Unicode support. The data may be displayed incorrectly in a software application with no Unicode support. To open a Unicode text file with multiple lines of comments in Microsoft<sup>®</sup> Excel, be sure to drag and drop the file on the Microsoft<sup>®</sup> Excel window; otherwise, the content of the file may be displayed incorrectly in Microsoft<sup>®</sup> Excel.

## File Operation during Execution of Data Logging

This section describes file operation during execution of data logging.

Target file	File operation	Behavior
Data logging setting file	Write	During execution of data logging, it is not possible to write/read data to/from the data logging setting file being used.
	Delete	
	Initialize	During execution of data logging, it is not possible to initialize the memory storing the data logging setting file being executed.
	Folder delete	Folders cannot be deleted from the \$MELPRJ\$ folder in which the data logging setting file is stored.
Data Logging File	Write	During execution of data logging, it is not possible to perform write/read data to/from and delete folders from the data logging setting file being used.
	Delete	
	Folder delete	
	Initialize	During execution of data logging, it is not possible to initialize the memory storing the data logging setting file being executed.

## About remote operation

When remote RUN is performed while the data logging function is in the following execution status, the remote RUN may fail. In that case, wait for a while and retry remote RUN. If remote RUN still cannot be executed, check whether remote RUN is acceptable and retry remote RUN.


Execution state of data logging function	The situation to accept remote RUN
Data saving into memory card in progress	No special relay (Data logging data saving into memory card in progress) is on.
Registration of the data logging setting from CPU Module Logging Configuration Tool in progress	The special relay (data logging preparation) and the special relay (data logging start) corresponding to the setting number of the data logging setting, which is being registered in the way shown in the left column, are on.
Registration of the data logging setting by auto logging in progress	M1200 (Auto logging setting file and registration status) is on.

23

## RUN operation through switching operation or the RUN contact

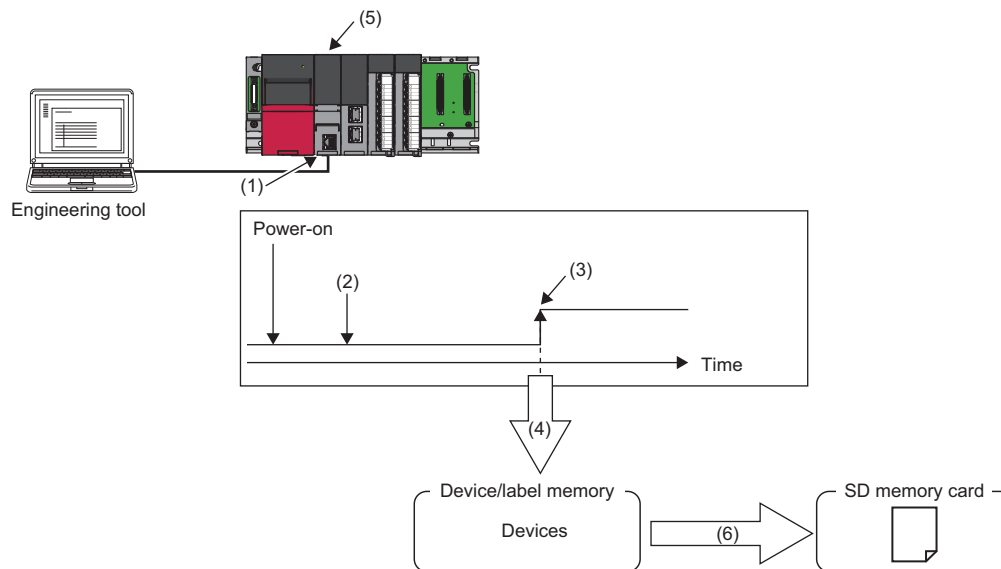
During execution of data logging, when STOP and RUN is performed using the RUN, STOP, and RESET switches, or when the RUN contact, which is specified in the RUN-PAUSE contact setting, is turned off, it may take time to return to the RUN state.

## Using together with interrupt programs

When occurrence of an interrupt is specified as the data collection condition for data logging, the processing time of the data logging function is added and, consequently, the processing time of the interrupt program is increased. Therefore, when I49, an interrupt by timer, is specified with a shorter interval (such as 0.05ms) in the internal timer for an interrupt program and occurrence of I49 is specified as the data collection condition for data logging, the processing time of the interrupt program may be too long to fit in the interval specified in the fixed scan interval setting, which causes an END instruction executed error (Excessive WDT error) due to continuous execution of the interrupt program. For the method of reducing the processing time of interrupt programs, refer to Saving/restoring of the file register (R) block number. (  Page 54 Saving/restoring of the file register (R) block number)

# 24 MEMORY DUMP FUNCTION

This function stores device values of the CPU module at any given timing. Checking data at the desired timing through the function facilitates the analysis of troubles, the occurrence of which depends on a particular condition.



- (1) Perform memory dump settings.
- (2) Enters a wait state for the trigger after the memory dump setting file has been written.
- (3) Establishment of the trigger condition
- (4) Start of data collection
- (5) The LED on the front of the CPU module allows memory dump status to be checked. (☞ Page 171 LED status)
- (6) The memory dump file is stored in the SD memory card. (☞ Page 171 Memory Dump File)

## Restriction

- Before executing the function, check the versions of the CPU module and engineering tool used. (☞ Page 417 Added and Enhanced Functions)
- The Process CPU does not support the function.

## 24.1 Object Data

This section describes the data to be collected by memory dump.

### Data to be collected

Of the devices listed below, all devices that are within the range specified in the device settings are subject to the collection.

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

\*1 Includes local devices as well. Note that to the memory dump file, the program names of collected local devices are output.

\*2 All file registers that exist in the device/label memory are collected.



## 24.2 Trigger Condition

The following table lists the conditions to be used as a trigger. Set the trigger condition in the memory dump settings. (GX Works3 Operating Manual)

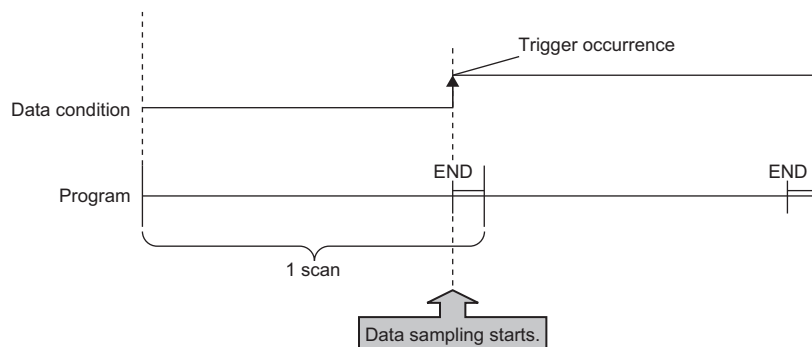
Trigger condition	Description
Device specification	Data are collected when the specified monitoring target data (bit data) turns on during the END processing.
Error code specification	Data are collected when an error occurs in the specified CPU module.

### Point

- On the occurrence of consecutive triggers, if the status of data save due to the previous occurrence of trigger is "Save complete" of collected data, the next trigger is recognized as a trigger again. Note that events other than the above are not recognized as a trigger condition.
- A trigger can be generated with trigger conditions combined. (Page 169 Combining trigger conditions)

### Device specification

Data are collected when the specified monitoring target data (bit data) turns on during the END processing. Even though the value of the monitoring target data changes during a single scan, if the value during the END processing is same as that during the last END processing, it is not recognized as a trigger. Completion bits (including error completion bits) used in dedicated instructions cannot be used as a trigger.



For monitoring data, the following devices can be specified.

Type	Device <sup>*1</sup>
Bit device	X, Y, M, L, F, SM, V, B, SB, T (contact) <sup>*3</sup> , ST (contact) <sup>*3</sup> , C (contact) <sup>*3</sup> , LT (contact) <sup>*3</sup> , LST (contact) <sup>*3</sup> , LC (contact) <sup>*3</sup> , FX, FY
Word device <sup>*2</sup>	D, SD, W, SW, R, ZR, FD, RD

\*1 A local device, index modification, and indirect specification cannot be specified.

\*2 Word devices allow bit specification only.

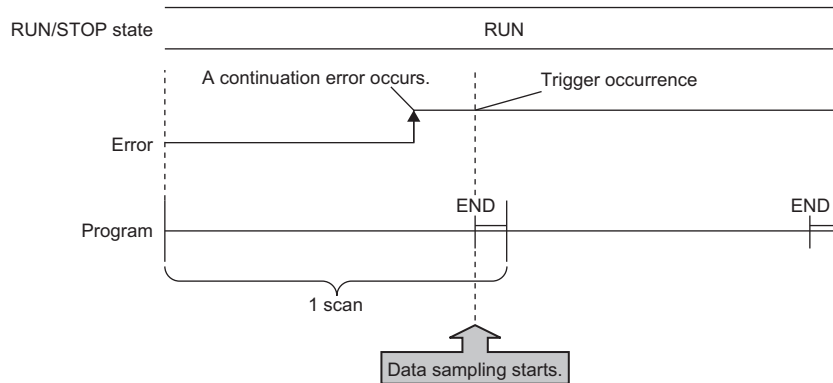
\*3 In the engineering tool, specify TS as T (contact), STS as ST (contact), CS as C (contact), LTS as LT (contact), LSTS as LST (contact), and LCS as LC (contact).

# Error code specification

With a specified error code of the CPU module as a trigger, data is to be collected. The occurrence timing of trigger varies depending on the error type: continuation error or stop error.

## At the occurrence of a continuation error

The occurrence timing of trigger is at the time of END processing of the scan where an error has occurred.



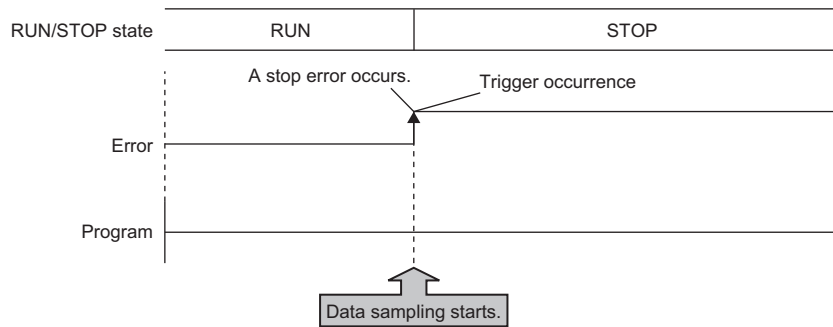
In the following situations, the occurrence of a continuation error that is specified as a trigger condition is not recognized as a trigger, with no data collection:

- After the occurrence of a continuation error that is specified as a trigger condition, the same error, or a continuation error, has occurred again.
- In the situation that a continuation error of 15 or more has occurred, a continuation error that is specified as a trigger condition has occurred.

Being recognized as a trigger requires the error to be cleared.

## At the occurrence of a stop error

The occurrence timing of trigger is at the time of occurrence of an error.

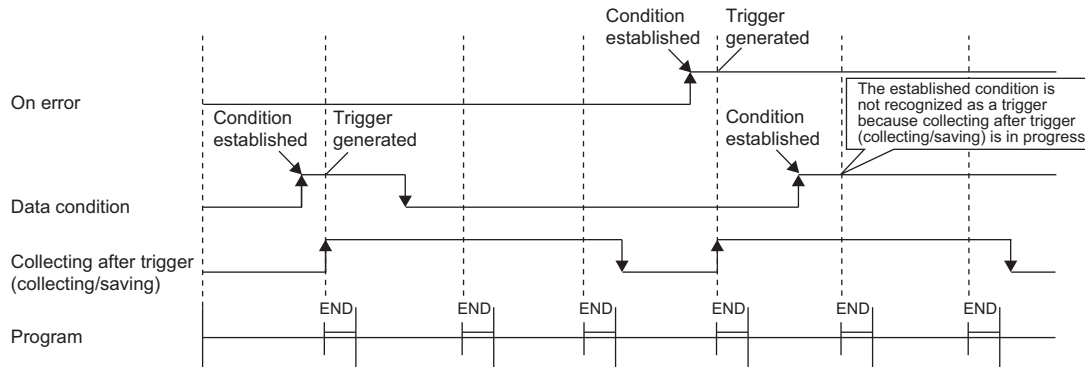


### Point

To specify an annunciator number as a trigger condition, employ device specification to specify any desired annunciator number.

## Combining trigger conditions

A trigger can be generated with trigger conditions combined. This combination is based on an OR condition. The establishment of a condition, either device specification or error code specification, results in data collection.



## 24.3 Procedure for Memory Dump

This section describes the procedure for memory dump. Note that each operation of the memory dump function is performed with the engineering tool.

[Debug] ⇒ [Memory Dump]

For how to view and operate the window, refer to the following.

GX Works3 Operating Manual

1. Configure the memory dump settings by the menu operation in the engineering tool.
2. Writing the memory dump setting file results in a wait state for the trigger. Whether the CPU module is in a RUN state, STOP state (a stop error also included\*1), or PAUSE state, a wait state for the trigger results.

\*1 Limited to where the trigger condition is device specification.

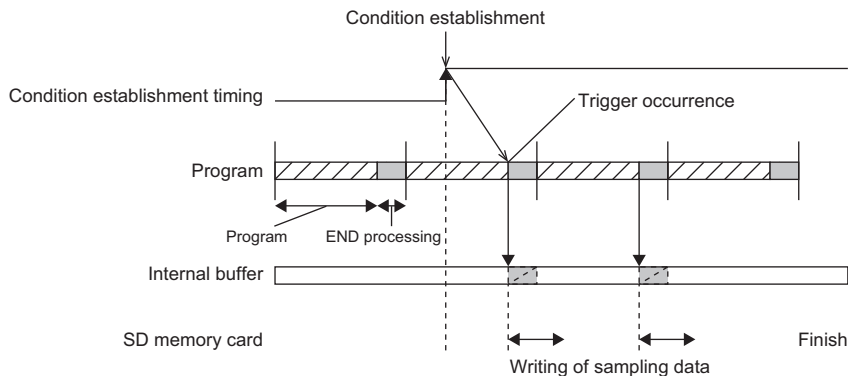
### Point

- If the memory dump setting file is stored, putting the CPU module back into operation (powering off and on, or resetting) results in a wait state for the trigger.
- The engineering tool allows the memory dump status to be checked. The event history also allows checking whether or not in a wait state for the trigger.

3. Establishment of the trigger condition initiates data collection, saving the memory dump file to the SD memory card.
4. The contents of the memory dump file (collected device data) can be checked with the engineering tool.

## 24.4 Flow of Data Collection

Collected data is stored in the internal buffer, where the data is partitioned at END processing and saved in the SD memory card.



The size of data to be collected per END processing is determined according to the capacity setting of the internal buffer and the individual area size of devices to be collected. Data is collected and partitioned into the groups per END processing, as in the table below. If the capacity of an individual area exceeds the internal buffer capacity, collected data in the individual area is further partitioned. The first 1K byte in the internal buffer is used as the system area, and thus the collection size per END processing is determined to a value of the internal buffer capacity minus 1K byte.

Partition area	Remarks
Device area	If the total value of devices exceeds the internal buffer capacity, the amount by which the internal buffer capacity is exceeded is collected at the next END processing in order from the top device, not collected at the current END processing.
Local device (at each program)	If the total value of local devices of programs concerned exceeds the internal buffer capacity, the amount by which the internal buffer capacity is exceeded is collected at the next END processing in order from the top device, not collected at the current END processing.
File register (at each file)	—
Refresh memory	Refresh memory area
	—

### Effect on the scan time

The greater the number of collected points, the longer the scan time. For the increase in scan time due to the collected points, refer to the processing time. (☞ Page 366 Memory dump function)

## 24.5 Memory Dump File

This file stores data that is collected through memory dump (collection result by memory dump). Data collected by one execution is saved in one file. The memory dump file is saved in a binary format and stored under the "MEMDUMP" folder.

### Save file name

The file name can be arbitrarily set within a range of 64 characters (extension and period included) together with an auto-assigned number (00 to 99). Specify the save file name in the memory dump settings. (📖 GX Works3 Operating Manual)

**Ex.**  
MEMDUMP\_00\*1

\*1 Between a specified file name (MEMDUMP) and an auto-assigned number (00), the single-byte underbar ( \_ ) is added. When the memory dump function is registered, the debug folder (DEBUG (fixed)) and the memory dump folder (MEMDUMP (fixed)) are created in the SD memory card. The memory dump file (result file) is stored in the memory dump folder. One folder can contain a maximum of 100 files. If any file does not exist in creating a save file, the file with the number 00 is created. If any file already exists in creating a save file, the behavior is as follows:

Number of files	Behavior
For less than 100	Creates a file*2 assigning the number obtained by adding 1 to the number of the file where the creation date and time is the latest.
For 100 (maximum)	Deletes the file where the creation date and time is the oldest and creates a new file using the deleted number as it is.

\*2 If the corresponding file number is 99, a file with file number 00 is created.

## 24.6 States of the Memory Dump Function

The state of the memory dump function is reflected in the memory dump status. The engineering tool allows the memory dump status to be checked. (📖 GX Works3 Operating Manual)

### Memory dump status

The following table lists the memory dump status.

Memory dump status	Description
Trigger-wait not collected	A state that data is not yet collected and establishment of the trigger condition is being waited
Collecting after trigger	A state that collection of the data after trigger is in progress (includes a state that collected data is being saved in the target memory)
Collection completed	A state that collection of a specified data is completed
Error	A state that a memory dump error occurs and memory dump fails

### LED status

Whether the memory dump function is active or not can be checked by the LED of the CPU module. To let the FUNCTION LED display the status while the memory dump function is used, setting parameters is required. (📖 Page 181 LED display setting)

LED status	Description
Trigger-wait not collected	<ul style="list-style-type: none"><li>CARD READY LED: On</li><li>CARD ACCESS LED: Off</li><li>FUNCTION LED: On</li></ul>
Collecting after trigger	<ul style="list-style-type: none"><li>CARD READY LED: On</li><li>CARD ACCESS LED: Turns on when the SD memory card is accessed</li><li>FUNCTION LED: Flashes slowly (every one second)</li></ul>
Collection completed	<ul style="list-style-type: none"><li>CARD READY LED: On</li><li>CARD ACCESS LED: Off</li><li>FUNCTION LED: Flashes at normal rate (every 200 milliseconds)</li></ul>

# 24.7 Sizes of Files Used for the Memory Dump Function

This section shows the sizes of files used for the memory dump function.

## Capacity of the memory dump setting file

The capacity of the memory dump setting file varies depending on the length of the save file name. The following formula is used for the calculation:

Capacity of memory dump setting file =  $((\text{Number of characters of save file name}^{*1} \times 2 \text{ bytes} + 1201 \text{ bytes (fixed)}) + 3) \div 4)^{*2} \times 4$

\*1 Except for the period and extension.

\*2 The remainder is discarded.

## Capacity of the memory dump file

The capacity of the memory dump file is given by the total of the following items:

Capacity of memory dump file = Volume of header + Volume of data of program file name + Volume of device data + Volume of local device data + Volume of data of file register file name + Volume of file register data

### Volume of header

The volume of header is given by:

Volume of header = 1088 bytes (fixed)

### Volume of data of program file name

The volume of data of program file name is given by the total of the following items. Note that this data is always created in the memory dump file regardless of the settings of CPU parameters.

Volume of data of program file name = 16 bytes (fixed) + (Number of programs  $\times$  (2 bytes (length of program file name) + 130 bytes (program file name)))

### Volume of device data

The volume of device data is given by the total of the following items. Note that this data is always created in the memory dump file regardless of the settings of CPU parameters.

Volume of device data = 682 bytes (fixed) + Volume of collected device data

#### ■Volume of collected device data

The volume of collected device data is given by the following:

Volume of collected device data = (Total number of points of bit devices  $\div$  8) + (Total number of points of word devices  $\times$  2) + (Total number of points of word devices  $\times$  4)

### Volume of local device data

The volume of local device data is given by the total of the following items. Note that this data is not created in the memory dump file unless local devices are set in the CPU parameters.

Volume of local device data = 16 bytes (fixed) + (Number of programs  $\times$  4 bytes) + (Number of programs  $\times$  Volume of local device contents)

#### ■Volume of local device contents

The volume of local device contents is given by the following:

Volume of local device contents = 418 bytes (fixed) + Volume of collected local device data

#### ■Volume of collected local device data

The volume of collected local device data is given by the following:

Volume of collected local device data = (Total number of points of bit devices  $\div$  8) + (Total number of points of word devices  $\times$  2) + (Total number of points of word devices  $\times$  4)

## Volume of data of file register file name

The volume of data of file register file name is given by the total of the following items. Note that this data is not created in the memory dump file unless a file register file exists in the device/label memory.

Volume of data of file register file name = 16 bytes (fixed) + (Number of file register files × (2 bytes (length of file register file name) + 130 bytes (file register file name))

## Volume of file register data

The volume of file register data is given by the total of the following items. Note that this data is not created in the memory dump file unless a file register file exists in the device/label memory.

Volume of file register data = Number of file register files × (148 bytes (fixed) + (Number of file registers × 2))

# 24.8 Special Relay and Special Register Used in the Memory Dump Function

For details on the special relay and special register used in the memory dump function, refer to the following:

- Special relay: Special relay relating to the memory dump function (☞ Page 332 Memory dump function)
- Special register: Special register relating to the memory dump function (☞ Page 354 Memory dump function)

# 24.9 Precautions for the Memory Dump Function

This section describes precautions to take when using the memory dump function.

## Mutual exclusion of the memory dump function

The mutual exclusion of the memory dump function is as follows.

### ■When another function is executed

The following table lists the cases when another function is executed during the execution of the memory dump function\*<sup>1</sup>.

Function that has been already executed	Function to be executed later	Behavior
Memory dump function	Function specified in the internal buffer capacity setting* <sup>2</sup>	If the internal buffer capacity setting is changed to execute the subsequent function, attempting to start the subsequent function results in an error. The memory dump continues to function normally.
	Function not specified in the internal buffer capacity setting	If the condition "Total capacity that is set in the internal buffer capacity setting + Internal buffer capacity that is set in other than the internal buffer capacity setting > 3072K bytes" is satisfied, attempting to start the subsequent function results in an error. The memory dump continues to function normally.
		If the internal buffer capacity setting is changed to execute the subsequent function, attempting to start the subsequent function results in an error. The memory dump continues to function normally.

\*<sup>1</sup> Indicates the state that the memory dump status is "Collecting after trigger" or the save status is "Saving in progress".

\*<sup>2</sup> The memory dump function is not included here.

### ■When the memory dump function is executed during the execution of another function

The following table lists the cases when the memory dump function is executed during the execution of another function.

Function that has been already executed	Function to be executed later	Behavior
Function specified in the internal buffer capacity setting* <sup>1</sup>	Memory dump function	If the internal buffer capacity setting is changed to execute memory dump, attempting to start memory dump results in an error. The function already in execution continues to function normally.
Function not specified in the internal buffer capacity setting		If the internal buffer capacity setting is changed to execute memory dump, attempting to start memory dump results in an error. The function already in execution continues to function normally.

\*<sup>1</sup> The memory dump function is not included here.

## ■When the memory dump function is executed

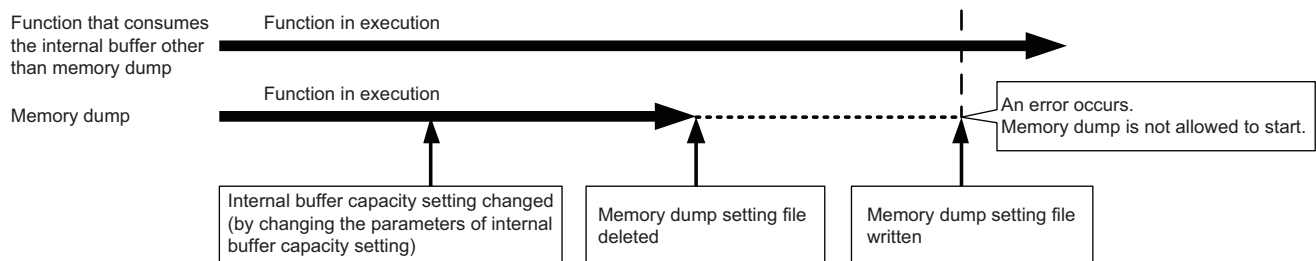
The following table shows the cases where the file operation related to the memory dump function is executed while the memory dump function is in execution.\*1

Target file	File operation	Behavior
Memory dump setting file	Write	Settings that are subsequently written during the execution of the memory dump function are reflected after the completion of save, not reflected immediately.
	Delete	If the memory dump setting file is subsequently deleted during the execution of the memory dump function, the memory dump settings are cleared after the completion of save.
	Initialize	Initialization fails on the memory dump setting file during the execution of the memory dump function.
	Folder delete	Folder delete fails on the folder in which the memory dump setting file is stored.
Memory dump file	Write, read, delete, initialize, and folder delete	Write, read, delete, initialize, and folder delete fails on the memory dump file during the execution of the memory dump function.

\*1 Indicates the state that the memory dump status is "Collecting after trigger" or the save status is "Saving in progress".

## Behavior at parameter change when functions consuming the internal buffer are active

If the internal buffer capacity setting is changed during the execution of functions that consume the internal buffer, attempting to start memory dump results in an error, where the memory dump fails to start.



## Operation on each individual file

Write, read, and delete are possible on each file. In addition, folder/file all delete and folder delete are possible on the memory or folder in which files are stored. The following table shows whether each operation is possible or not depending on the execution status of memory dump.

○: Operation possible, ×: Operation not possible

File type	Operation to be performed							
	Read		Write/delete		Folder/file all deletion		Folder delete	
	Not during execution*1	During execution*1	Not during execution*1	During execution*1	Not during execution*1	During execution*1	Not during execution*1	During execution*1
Memory dump setting file	○	○	○	○	○	×	×	×
Memory dump file	○	×	○	×	○	×	○	×

\*1 Indicates the state that the memory dump status is "Collecting after trigger".

## Where to carry out memory dump

Concurrent execution from multiple sources is not allowed. In the CPU module, execution at a time from only one source is possible.

## Trigger condition during the registration of memory dump

If a trigger condition is established during the registration of memory dump, the memory dump settings can be registered. A second establishment of the trigger condition is recognized as a trigger condition there.

## When file register is specified as specification device in the condition specification

After registering memory dump, do not change the file name of the file registers and the block number of the file registers. Doing so may result in a failure to successfully collect the memory dump results.



## Creating files and folders

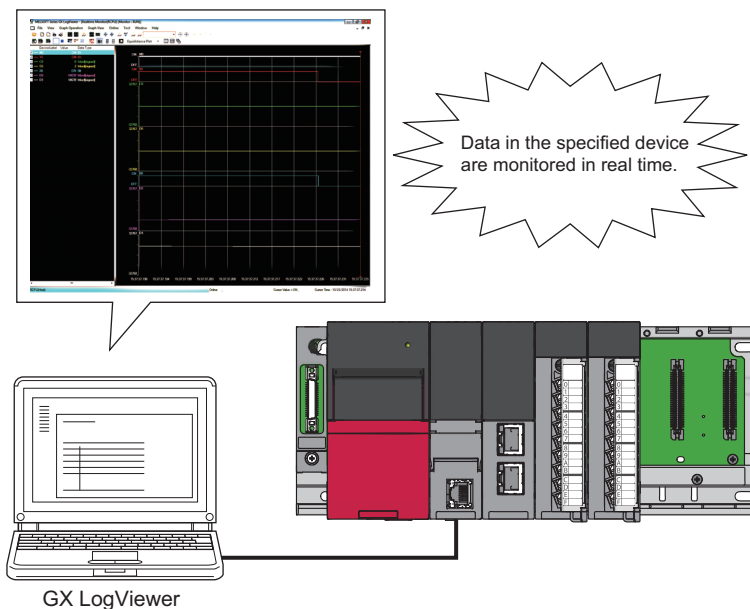
Under the "MEMDUMP" folder containing memory dump files, do not create any files or folders using a personal computer or other device. Doing so may result in deletion of files and folders.

## Access to the SD memory card

The SD memory card is so frequently accessed that a delay occurs in completing the access to the SD memory card (read/write).

# 25 REAL-TIME MONITOR FUNCTION

This function monitors the contents of a specified device of the CPU module in real time with a specified interval or a desired timing. The function can be set with GX LogViewer, where the value changes of a specified device can be shown graphically. Saving the set data and displayed graphs makes it possible to simplify the settings and check the graphs at a later time.



## Restriction

- Before executing the function, check the versions of the CPU module and GX LogViewer used. (☞ Page 417 Added and Enhanced Functions)
- The Process CPU does not support the function.

For details on the function, refer to the following.

📖 GX LogViewer Version1 Operating Manual



# 26 RAS FUNCTIONS

## 26.1 Self-Diagnostics Function

Checks if a problem exists with the CPU module.

### Self-diagnostics timing

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

### Check method of error

This section describes the check methods when error occurs.

#### Check method using the special relay and special register

When the CPU module detects an error, it turns SM0 (Latest Self-diagnostics error (annunciator on included)) and SM1 (Latest Self-diagnostics error (annunciator on not included)) on and stores the error code corresponding to the error definition in SD0 (diagnostics error). 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 (Self-diagnostics error code) to SD25 (Self-diagnostics error code). (The error code for the error content of 17th piece on and after will not be stored.)

#### Check method using LED

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

#### Check method using the engineering tool

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

#### ■Existing errors

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

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

#### ■Error history

Occurred errors is logged in the event history (📖 Page 184 Event History Function). The error log 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 occurrence of a stop error, the information on the existing errors is not refreshed and only the error log is updated.

# CPU Module Operation Upon Error Detection Setting

Configure each CPU Module Operation setting when an error is detected.

## Mode when an error is detected

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

### ■ Mode for stopping the operation of CPU module

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

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

#### Point

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

### ■ Mode for continuing the operation of CPU module

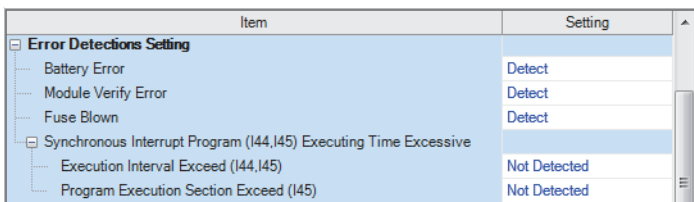
If an error has been detected, the program other than programs (instruction) where an error was generated is executed.

## Error Detection Setting

Sets whether or not to detect errors.

[CPU Parameter] ⇒ "RAS Setting" ⇒ "Error Detections Setting"

### Window



Item	Setting
Error Detections Setting	
Battery Error	Detect
Module Verify Error	Detect
Fuse Blown	Detect
Synchronous Interrupt Program (I44,I45) Executing Time Excessive	Not Detected
Execution Interval Exceed (I44,I45)	Not Detected
Program Execution Section Exceed (I45)	Not Detected

### Displayed items

Item	Description	Setting range	Default
Battery Error	Sets whether or not to detect the battery error.	• Detect • Not Detected	Detect
Module Verify Error <sup>*1</sup>	Sets whether or not to detect the module verification error.		
Fuse Blown	Sets whether or not to detect the fuse blown of the controlled module.		
Synchronous Interrupt Program (I44, I45)	Execution Interval Exceed (I44, I45)	Not Detected	Not Detected
Executing Time Excessive	Program execution section excess (I45)		

\*1 When "Not Detected" is enabled and the operating CPU module is removed, a verify module error is not detected, but a stop error may occur if an program tries to access the removed CPU module. Also, the removed CPU module will not be accessible even when it is re-installed. Therefore, if access is tried to the re-installed CPU module through the STOP and RUN operation, it may result in a stop error.

## CPU Module Operation Upon Error Detection Setting

Sets the CPU module operation upon error detection.

[CPU Parameter] ⇒ "RAS Setting" ⇒ "CPU Module Operation Setting at Error Detected"

### Window


Item	Setting
<b>CPU Module Operation Setting at Error Detected</b>	
Instruction Execution Error	
I/O No./Network No. Specification Incorrect	Stop
Device/Label/Buffer Memory Specification Incorrect	Stop
File Name Specification Incorrect	Stop
Operation Error	Stop
Memory Card Error	Stop
Module Verify Error	Stop
Fuse Blown	Stop
Synchronous Interrupt Execution Interval Error (CPU Module)	Stop

### Displayed items

Item	Description	Setting range	Default
Instruction Execution Error	I/O numbers/Network numbers specification error	<ul style="list-style-type: none"> <li>Stops</li> <li>Continues</li> </ul>	Stops
	Device/Label/Buffer memory specification error		
	File Name Specification Incorrect		
	Operation Error		
Memory Card Error	Sets the CPU module operation upon a memory card error.		
Module Verify Error	Sets the CPU module operation upon a module verification error.		
Fuse Blown	Sets the CPU module operation upon fuse blown error of controlled module.		
Synchronous Interrupt Execution Interval Error (CPU Module)	Sets the CPU module operation upon a detection of a synchronization signal error on the CPU module.		

## CPU Module Operation Setting

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

 [System Parameter] ⇒ [I/O Assignment Setting] ⇒ "I/O Assignment Setting"


### Window

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

### Displayed items

Item	Description	Setting range	Default
CPU Module Operation Upon Error Detection Setting	Sets the CPU module operation upon the detection of Major or Moderate errors in the configured module.	<ul style="list-style-type: none"> <li>• Critical: Stop, Moderate: Continue</li> <li>• Critical: Stop, Moderate: Stop</li> <li>• Critical: Continue, Moderate: Continue</li> </ul>	Critical: Stop, Moderate: Continue

## Stop Setting

For the multiple CPU system configuration, it can be specified whether or not to stop all CPU modules when a major or moderate error occurs on a CPU module. ( Page 193 Stop setting)

## LED display setting

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


 [CPU Parameter] ⇒ "RAS Setting" ⇒ "LED Display Setting"

### Window

Item	Setting
LED Display Setting	
ERROR LED	
Minor Error (Continue Error)	Display
USER LED	
Annunciator ON	Display
BATTERY LED	
Battery Error	Display
FUNCTION LED	
Function to Use	Data Logging Function

### Displayed items

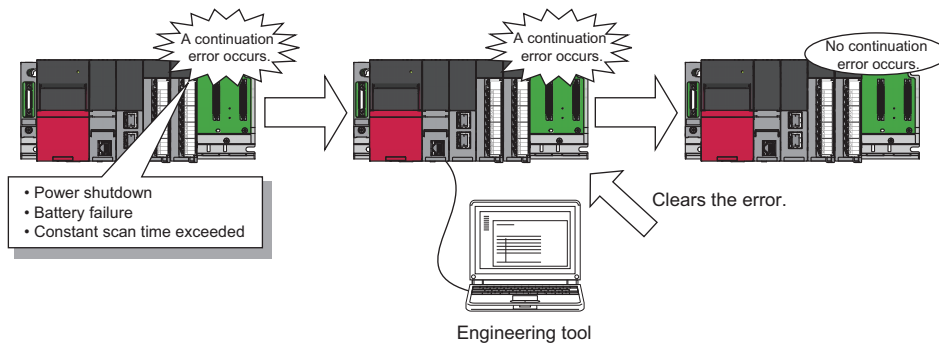
Item	Description	Setting range	Default
ERROR LED	Minor Error (Continue Error)	<ul style="list-style-type: none"> <li>• Display</li> <li>• Do Not Display</li> </ul>	Display
USER LED	Annunciator ON		
BATTERY LED	Battery Error		
FUNCTION LED*1*2	Display Function	<ul style="list-style-type: none"> <li>• Data logging function</li> <li>• Memory dump function</li> <li>• Do Not Display</li> </ul>	Data logging function

\*1 To specify whether to display or hide the FUNCTION LED, verify the versions of the CPU module and the engineering tool. ( Page 417 Added and Enhanced Functions)

\*2 The Process CPU does not support the display/hide of the FUNCTION LED.

## 26.2 Error Clear

This function clears all the existing continuation errors at once.



### Errors that can be cleared

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

Error code	Error name
1000H	Power shutoff
1080H	ROM write count error
1090H	Battery error
1100H	Memory card access error
1120H	SNTP clock setting error
1124H	Default gateway/gateway IP address error
1128H	Own node port number error
1129H	Open specification port number error
112DH	Specified IP address error
112EH	Connection establishment failed
1130H	IP address duplication error
1133H	Socket communications response send error
1134H	TCP connection timeout
1152H	IP address error
1155H	Connection number acquisition error
1157H	Receive buffer securement error
1165H	UDP/IP send failed
1166H	TCP/IP send failed
1167H	Unsend data send error
1200H, 1210H	Module moderate error
1220H	Another CPU module moderate error
1240H, 1241H	Inter-module synchronization processing error
1260H, 1262H	Multiple CPU synchronization processing error
1800H	Annunciator ON
1830H	Receive queue full
1831H	Receive processing error
1832H	Transient data error
1900H	Constant scan time error
2120H, 2121H	Memory card error
2400H, 2401H	Module verification error
2420H	Fuse blown error
2441H, 2442H	Major error in module
2450H	Detection of major error in module
2461H, 2462H	Another CPU module major error
2470H	Another CPU module major error



Error code	Error name
2610H	Inter-module synchronization signal error
2630H	Multiple CPU synchronization signal error
2800H, 2801H, 2802H, 2803H, 2804H, 2805H, 2806H, 2807H, 2810H	I/O numbers/Network numbers specification error
2820H, 2821H, 2822H, 2823H, 2824H	Device/Label/Buffer memory specification error
2840H, 2841H	File name specification error
3400H, 3401H, 3402H, 3403H, 3404H, 3405H, 3406H, 3420H, 3421H, 3422H, 3440H, 3441H, 34A0	Operation error

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

### How to clear errors

Errors can be cleared in two ways:

#### ■Using the engineering tool

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

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

#### ■Using SM/SD

Clear errors by operating SM/SD.

1. Check SD0 (Latest self-diagnostics error code) to identify what errors are detected.
2. Clear the cause of each of the currently detected continuation errors.
3. Turn off and on SM50 (error reset).

### Precautions

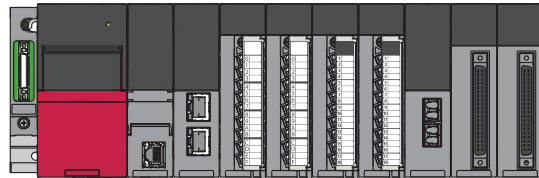
This section describes some precautions to take when using the error clear function:

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

## 26.3 Event History Function

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

The information of events that occurred in the host CPU module and modules which are managed by the host CPU module are sampled by the CPU module in a batch and saved.



The event information that is held in the CPU module is displayed with an engineering tool.

### Point

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

## Event history settings

Under normal circumstances, the event history function can be used with its default settings and need not be manually configured. The storage memory and size settings for event history files can be changed as needed. (Page 185 Event history file)

[CPU Parameter] ⇒ "RAS Setting" ⇒ "Event History Setting"

### Window

Item	Setting
Event History Setting	
Save Destination	Data Memory
Set Save Volume of Per File	128 K Byte

### Displayed items

Item	Description	Setting range	Default
Save to	Specify the storage location of event history files.	<ul style="list-style-type: none"> <li>Data Memory</li> <li>Memory Card</li> </ul>	Data Memory
Set Save Volume of Per File	Specify the storage capacity per event history file.	1 to 2048K bytes (in 1K bytes)	128K Byte


# Logging of the event history

This section describes events saving for the event history.


## Modules from which event history information is collected

Event history information is collected from the CPU module and other modules installed on the same base unit (i.e., the main base unit plus any additional extension base units). Event history information may or may not be collected from devices on the network depending on the specifications of the network modules used to connect to them. Refer to the manuals of the respective modules for more information including the coverage of event history collection regarding devices on the network. For a multi-CPU system, note that each CPU module logs only events detected on the modules under its control.

## Events logged by the CPU module

Information logged in the event history includes operation initiator and other detailed information for troubleshooting purposes. For events that are logged in the event history on the CPU module, refer to the event history ( Page 312 Event List)

## Event history file

The storage memory and file size for event history files can be changed in event history setting. ( Page 184 Event history settings)

### ■Storage memory

Choose either the data memory or SD memory card. If the storage memory is the SD memory card: when the light 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 Light 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 Light protect switch is enabled. (At the time immediately after an error is generated, the module diagnostics of engineering tool can detect it, however, after Power is turned off→on or after resetting operation, the generated error will not be stored into the event history.)

### Point

For a system where file writes occur frequently or the communication conditions are unstable and frequently change, the event history file size should be made large enough to accommodate a greater number of events. If this is the case, the SD memory card is highly recommended as the storage memory.

## ■File size

The size for event history files can be changed in event history setting (📄 Page 184 Event history settings). If the specified size is exceeded, the records are deleted 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 <sup>*1</sup>

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

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

[Calculation formula]

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

### Ex.

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

### Operating procedure

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

Element	Size (byte)	
File header	20	
Event history management information	12	
Event to be saved	Power-on and reset	40
	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)<sup>\*1</sup>
- Write of parameters (when no event history file exists, or after an event history setting is changed).

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

### Point

When a new event history file is created, that operation is logged into the event file as an event.

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

Operation	Operation for the event history
Memory initialization	When this event occurs, the event history is stored into the internal memory. If the internal memory reaches the maximum number of event history records it can store, all subsequent events are lost (☞ Page 187 Loss of event history information)
Event history creation	The event history, which has been stored in the internal memory during absence of the event history file, is stored into the data memory or the SD memory card (If any event was lost, it is logged as "**HST LOSS**").

Indicates the operation of the event history for the SD memory which was removed and mounted in case that the save destination memory is the SD memory.

Operation	Operation for the event history
Removal of the SD memory card	When this event occurs, the event history is stored into the internal memory. If the internal memory reaches the maximum number of event history records it can store, all subsequent events are lost (☞ Page 187 Loss of event history information)
Installation of the SD memory card	The event history, which have been stored in the internal memory during absence of the SD memory card, is stored to the SD memory card. If the re-inserted SD memory card contains an event history file of the same file size, the CPU module continues to store the event history information. If the file size is different, the CPU module removes the existing event history file and creates a new event history file.

## ■When parameters take effect

Any changed parameters take effect when:

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

### Point

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

## Loss of event history information

If many events are detected frequently, some events may be lost without being collected. When event loss occurs, "\*\*HST LOSS\*\*" appears in the "Event Code" field of the engineering tool.

## Viewing the event history

The event history can be viewed using the menus of the engineering tool. For operating procedures and how to interpret the displayed information, refer to the following:

☞ GX Works3 Operating Manual

## Clearing the event history

The event history can be cleared using the event history window. Once the event history is cleared, the CPU module deletes all the event history information stored in the specified storage memory. For operating procedures and other details, refer to the following:

☞ GX Works3 Operating Manual

## 26.4 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.

# 27 MULTIPLE CPU SYSTEM FUNCTION

With multiple CPU modules mounted on the base unit, each of the CPU modules controls their own assigned I/O modules and intelligent function modules. In addition, the CPU modules communicate with each other.

## Point

- For details on the concept of the multiple CPU system configuration specification (System configuration specifications such as the mounting position of the CPU modules and assignment of CPU number/IO number), refer to the MELSEC iQ-R Module Configuration Manual.
- For the start-up (Setting, operating procedures, etc) of the Multiple CPU system, refer to the GX Works3 Operating Manual.

## Restriction

- The startup time of the multiple CPU system may be slowed, depending on the configuration of installed modules, boot operation, functions performed before CPU modules enter into the RUN state (e.g. setting of initial device/label values), the configuration of system parameters and CPU parameters (e.g. the number of programs to run).
- When diagnostics of the SD memory card is performed due to operation such as power-off during access to the SD memory card, the startup time of the multiple CPU system may be slowed
- Create a program so that only one CPU module accesses the Q series module, if possible. If multiple CPU modules access the Q series module simultaneously, the scan time (including the execution processing of interrupt programs) may be extended due to access waiting time.

# 27.1 Out-of-group I/O Fetch

The access range to the controlled module is different from that to the non-controlled module. To fetch the data which cannot be accessed, use the out-of-group I/O fetch function.

## Accessing controlled module

The access range to the controlled module of the CPU module is the same as that to the single CPU system, and I/O refresh for the controlled module and/or reading/writing to buffer memory of the intelligent function module are enabled.

## Accessing non-controlled module

Access to the non-controlled module of the CPU module is limited to reading input (DX) and buffer memory of the intelligent function module. Note that On/Off data of input (X)/output (Y) of the non-controlled module can be fetched using the out-of-group I/O Fetch function. (☞ Page 191 Out-of-group I/O Fetch setting)

○: Accessible, ×: Not accessible

Communication method	Access target		Access range		
			When "Import" is selected in the settings for out-of-group I/O fetch	When "Not Imported" is selected in the settings for out-of-group I/O fetch	
Communication through refresh	Input (X)	Read to another CPU module	○*1	× (Non-processing)	
		Output (Y)	Write to another CPU module		× (Non-processing)
	The buffer memory of the intelligent function module	Read	× (Cannot be specified from engineering tools)	× (Cannot be specified from engineering tools)	
		Write			
	Link direct device	Read	× (Cannot be specified from engineering tools)	× (Cannot be specified from engineering tools)	
Write					
Communication through direct access	Input (DX)	Read to another CPU module	○	○	
		Output (DY)	Write to another CPU module	× (Non-processing)	× (Non-processing)
	The buffer memory of the intelligent function module	Read	○	× (An error occurs in the CPU module.)	× (An error occurs in the CPU module.)
		Write			
	Link direct device	Read	× (An error occurs.)	× (An error occurs.)	
		Write			

\*1 Cannot read from CPU modules which are synchronized through the inter-module synchronization function.

### Precautions

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



## Out-of-group I/O Fetch setting

In this menu item, whether or not out-of-group I/O status is fetched can be specified.

 [System Parameter] ⇒ "Multiple CPU Setting" ⇒ "Other PLC Control Module Setting" ⇒ "I/O Setting Outside Group"

### Window

Item	Setting
Other PLC Control Module Setting	
I/O Setting Outside Group	
Input Status Outside Group	Not Imported
Output Status Outside Group	Not Imported

27

### Displayed items

Item	Description	Setting range	Default
Input Status Outside Group	Specify whether or not out-of-group input status is fetched.	<ul style="list-style-type: none"> <li>• Not Imported</li> <li>• Import</li> </ul>	Not Imported
Output Status Outside Group	Specify whether or not out-of-group output status is fetched.	<ul style="list-style-type: none"> <li>• Not Imported</li> <li>• Import</li> </ul>	Not Imported

## Fetching input (X)

When "Import" is selected for "Input Status Outside Group" in "I/O Setting Outside Group", the input refresh before the start of the program operation fetches On/Off data from the input module and/or intelligent function module controlled by other CPU modules.

### ■Modules from which input (X) can be fetched

Input (X) can be fetched from the following modules mounted on the base/extension base unit.

- Input module
- I/O combined module\*<sup>1</sup>
- Intelligent function module

\*<sup>1</sup> Data in the areas assigned to the output portion (area not used) are not fetched. On/Off state is maintained.

#### Restriction

Input (X) cannot be fetched from the Q series modules.

## Fetching output (Y)

When "Import" is selected for "Output Status Outside Group" in "I/O Setting Outside Group", the output refresh before the start of the program operation fetches On/Off data output to the output module and/or intelligent function module controlled by other CPU modules into output (Y) for the host CPU module.

### ■Modules from which output (Y) can be fetched

Output (Y) can be fetched from the following modules mounted on the base/extension base unit.

- Output module
- I/O combined module
- Intelligent function module

#### Restriction

Input (X) cannot be fetched from the Q series modules.

## Output to output/intelligent function module

On/off data cannot be output to non-controlled modules. When turning on or off output of the output module and/or intelligent function module controlled by other CPU modules by the program or others, the output is turned on or off within the CPU module. However, it is not output to output/intelligent function module.

## Accessing buffer memory of intelligent function module

### ■Reading data on buffer memory

The following instructions can be used to read data stored in the buffer memory of the intelligent function module.

- FROM instruction
- Instruction using the CPU module access device (Un\Gn)

### ■Writing data to buffer memory

The following instructions cannot be used to write data to the buffer memory of the intelligent function module.

- TO instruction
- Instruction using the CPU module access device (Un\Gn)

## Accessing a module by using link direct devices

The CPU module can access only to modules under its control by using link direct devices. Non-controlled modules cannot be accessed.

## 27.2 Operation Settings

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

### Stop setting

An operating status, whether to stop the operation of all the CPU modules or not, if a major or moderate error occurs in any of the CPU modules is set.

 [System Parameter] ⇒ "Multiple CPU Setting" ⇒ "Operation Mode Setting" ⇒ "Stop Setting"

27

#### Window

Item	Setting
Operation Mode Setting	
Stop Setting	
PLC No. 1	Major: All Station Stop, Moderate: All Stat
PLC No. 2	Major: All Station Stop, Moderate: All Stat
PLC No. 3	Major: All Station Stop, Moderate: All Stat
PLC No. 4	Major: All Station Stop, Moderate: All Stat

#### Displayed items

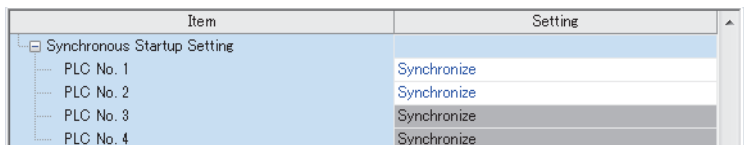
Item	Description	Setting range	Default
PLC No. 1	Set whether to stop the operation of all the CPU modules or not if a major or moderate error occurs in CPU No.1.	• Major: All Station Stop, Moderate: All Station Stop	Major: All Station Stop, Moderate: All Station Stop
PLC No. 2	Set whether to stop the operation of all the CPU modules or not if a major or moderate error occurs in CPU No.2.	• Major: All Station Stop, Moderate: All Station Continue	
PLC No. 3	Set whether to stop the operation of all the CPU modules or not if a major or moderate error occurs in CPU No.3.	• Major: All Station Continue, Moderate: All Station Continue	
PLC No. 4	Set whether to stop the operation of all the CPU modules or not if a major or moderate error occurs in CPU No.4.	• Major: All Station Continue, Moderate: All Station Continue	

# Synchronous startup setting

Startup time is synchronized among the CPU modules. This setting eliminates the need for an interlock program that monitors the startup time of another CPU module when accessing it. Note, however, that the startup of the entire system delays because the system starts up with the last CPU module.

 [System Parameter] ⇒ "Multiple CPU Setting" ⇒ "Operation Mode Setting" ⇒ "Synchronous Startup Setting"

## Window



Item	Setting
Synchronous Startup Setting	
PLC No. 1	Synchronize
PLC No. 2	Synchronize
PLC No. 3	Synchronize
PLC No. 4	Synchronize

## Displayed items

Item	Description	Setting range	Default
PLC No. 1	Set the CPU modules whose startup time is synchronized in the multiple CPU system.	• Synchronize • Do not Synchronize	Synchronize
PLC No. 2			
PLC No. 3			
PLC No. 4			

## Point

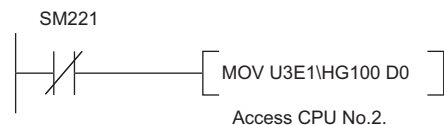
- Group setting for the synchronized start-up is available. For example, a setting in which only CPUs No.1 and No. 2 start synchronously within a multiple CPU system with four CPU modules is possible.
- If a reserved (empty) CPU is specified to synchronize, it is skipped and the other CPUs in the group start synchronously.
- This setting is designed to access to each CPU module without interlock in the multiple CPU system. It is not intended to be used for starting operation processing at the same time across CPU modules after startup.

## Program to check start-up of each CPU module

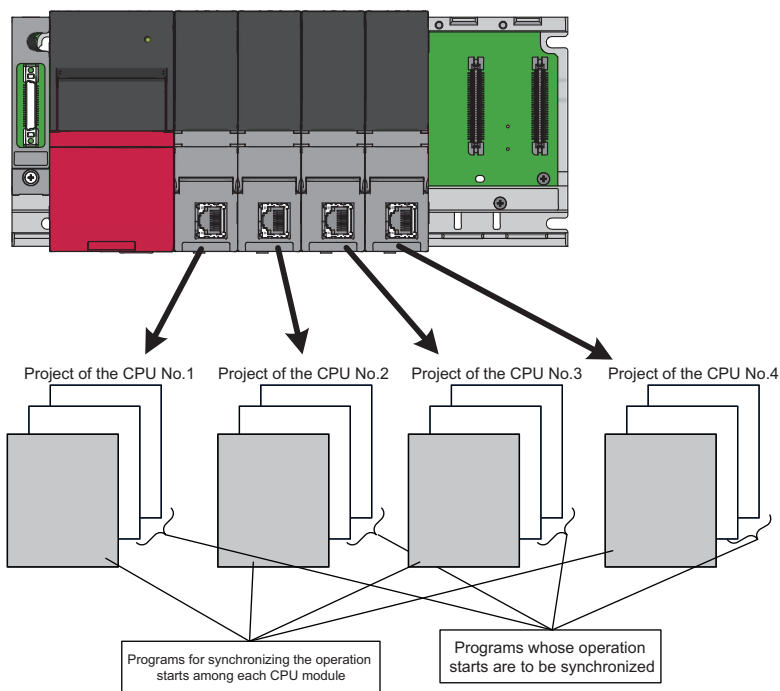
It is recommended to create a program that checks start-up of each CPU module using SM220 to SM223 (CPUs No.1 to No.4 preparation completed) when the multiple CPU synchronized startup is disabled. If a certain instruction is issued against a CPU module that has not started, the instruction executes no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

**Ex.**

Program to check start-up of CPU module of CPU No.2



To synchronize the start of operation processing across CPU modules, a program is required to check whether individual CPU modules are ready to start operation processing or not.



The following is an example of the configuration of a program which synchronizes the start of operation processing across CPU modules and programs which should start operation processing synchronously.

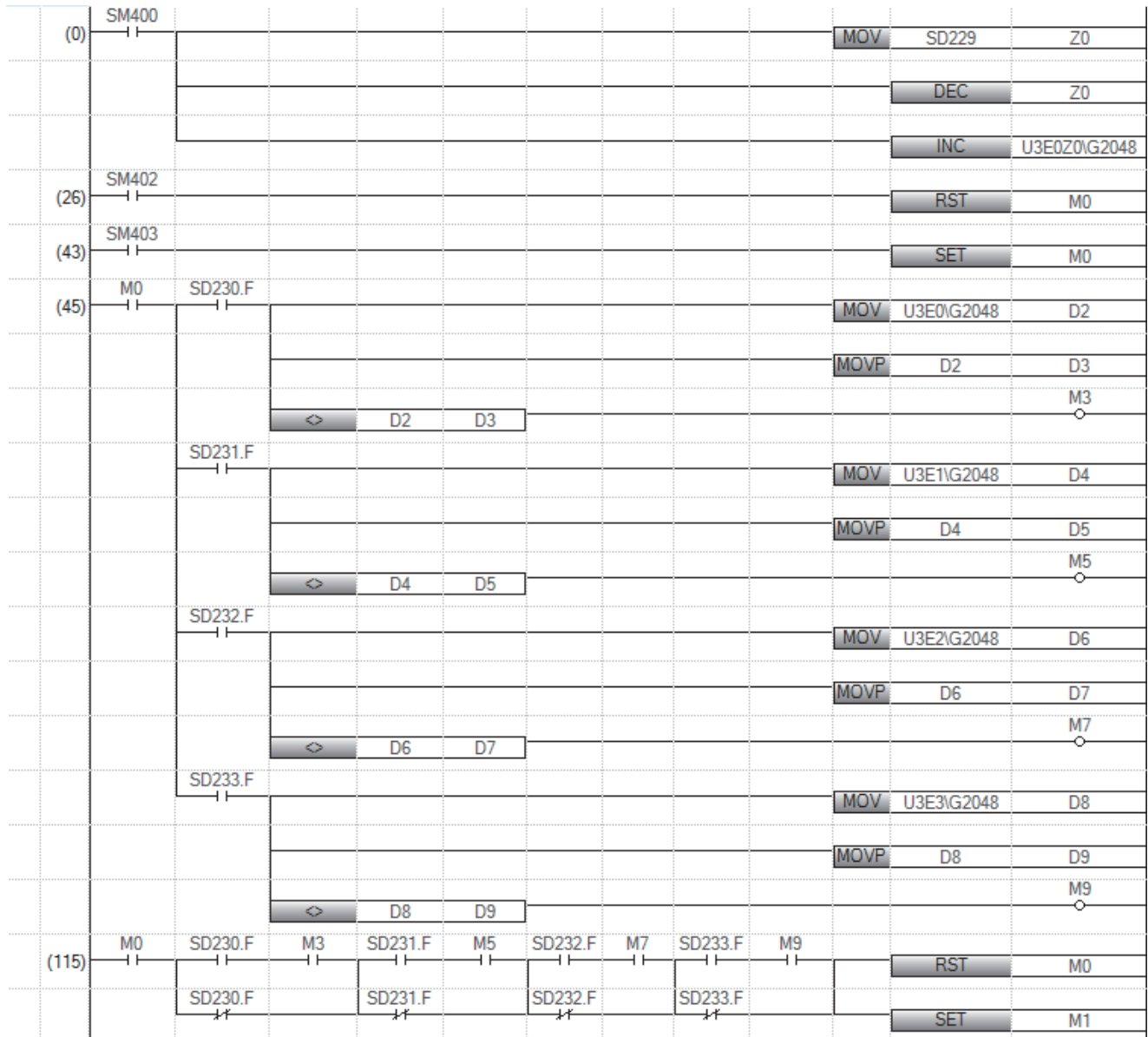
Program	Execution type	Description
Program to synchronize the start of operation processing	Scan	This is a program used to synchronize the start of operation processing across all CPU modules. This program must be specified as one for running at the beginning of the scan execution type of program. Also, when all the CPU modules are ready to start operation processing, the flag indicating an operation start turns on for only one scan.
Program to start operation processing synchronously	Event	This is a program which should be executed when all the CPU modules are ready to start operation processing. This program must be specified as an event execution type program which is triggered when the operation start flag is turned on.

## Program example

- Devices to be used for programs to start operation processing synchronously

Device to be used	Application
M0	Flag that indicates the operation processing is ready to be started (after a flag that indicates the operation start turns on, this flag turns off.)
M1	Flag that indicates an operation start (this flag turns on for only one scan.)
U3En\G2048	The synchronization counter of each CPU module (n=0: CPU No.1, n=1: CPU No.2, n=2: CPU No.3, n=3: CPU No.4)
D2 to D9	The storage location for values of the synchronization counter of each CPU module
M2 to M9	The operation processing start wait state of each CPU module

- Example of a program to synchronize the start of operation processing



## Clock data

---

CPUs No.2 to No.4 automatically synchronize their clock data to the one set for CPU No.1 (even if setting up clock data individually for each CPU, they will be overwritten). Therefore, simply setting up the clock data for CPU No.1 allows to manipulate a unified clock data across the entire multiple CPU system (☞ Page 75 Time Setting)

---

### **Point**

As with the clock data, the time zone setting for CPUs No.2 to No.4 follows the one configured on CPU No.1 (☞ Page 76 Setting Time Zone)

---

## 27.3 Multiple CPU Parameter Checking

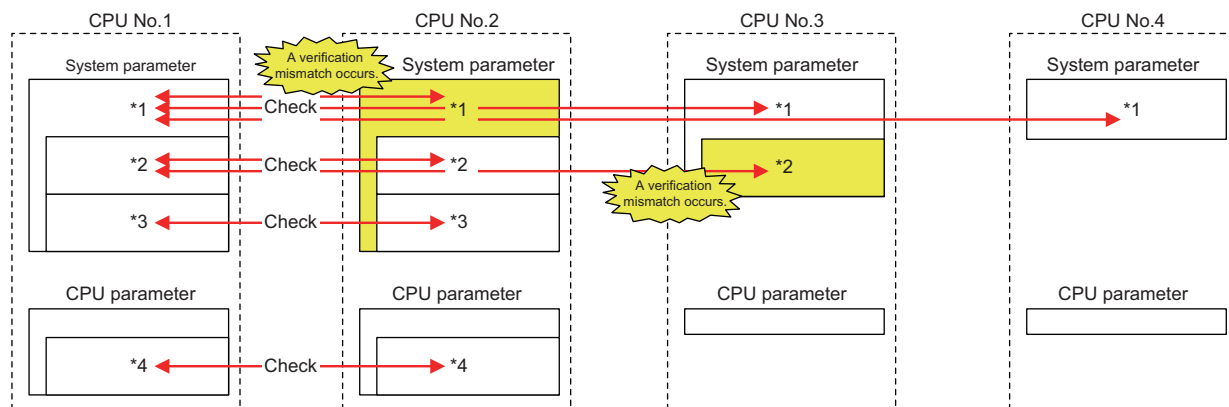
Whether the same setting is configured for between the system parameter of each CPU module and multiple CPU refresh number of points of CPU parameter is checked by the multiple CPU system at the timing shown below. However, as for the fixed scan communication setting and inter-module synchronization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

- Powered-on
- When the CPU No.1 is reset
- STOP→RUN state after parameter was changed

Timing	Parameters targeted for checking	Checking conditions for CPU No.1	Checking conditions for CPU No.2 and over
When power is on or the CPU No.1 is reset	System parameters (other than fixed scan communication setting and inter-module synchronous setting)	Checking is not conducted.	Compares with the parameters of the CPU of the lowest number.
	Settings of fixed scan communication	Checking is not conducted by the CPU module of the number for which the fixed scan communication setting is not configured. The CPU module of the number for which the fixed scan communication setting is configured will compare the parameters with those of the CPU of the lowest number.	
	Inter-module synchronization setting	Checking is not conducted by the CPU module of the number for which the Inter-module synchronization setting is not configured. The CPU module of the number for which the Inter-module synchronization setting has been configured will compare the parameters with those of the CPU of the lowest number.	
	CPU parameters (number of points of refresh settings)	Checking is not conducted by the CPU module of the number for which the fixed scan communication setting is not configured. The CPU module of the number for which the fixed scan communication setting is configured will compare the parameters with those of the CPU of the lowest number.	
STOP→RUN state after parameter was changed	—	Compares with the parameters of the host CPU module before parameters are changed.	

Ex.

An error occurs in the module of each CPU No 2 and 3, and the module of each CPU No 1 and 4 starts up normally when parameter checking. (Operation in error)



○: there is setting (if the number immediately next is the same number, it refers to the same parameter) , ×: there is no setting

Item		Setting conditions for the CPUs of each number			
		CPU No.1	CPU No.2	CPU No.3	CPU No.4
With or without the setting for each parameter	System parameters (other than fixed scan communication setting and inter-module synchronous setting) <sup>*1</sup>	○1	○5	○1	○1
	Settings of fixed scan communication <sup>*2</sup>	○2	○2	○6	×
	Inter-module synchronization setting <sup>*3</sup>	○3	○3	×	×
	CPU parameters (number of points of refresh settings) <sup>*4</sup>	○4	○4	×	×



## 27.4 Data Communication Between CPU Modules

CPU modules within a multiple CPU system can send and transfer data to each other. The refresh communication and direct access communication enable data writing or reading between CPU modules. Indicates the Data communication method.

Communication method	Application	Description
Data communication with CPU buffer memory	Used when data is sent or received at the timing of each CPU module.	The CPU module for sending the data writes data into the CPU Buffer memory of the host CPU module. The CPU module for receiving data reads data from the CPU Buffer memory of the sender CPU module (other CPU modules).
Data communication with fixed scan communication area	Used when data is sent or received through adjusting the timing between CPU modules.	The CPU module for sending the data writes data into the fixed scan communication area (send area) of the host CPU module. The CPU module for receiving data reads data from the fixed scan communication area (receive area) of the host CPU module of the send source CPU module.

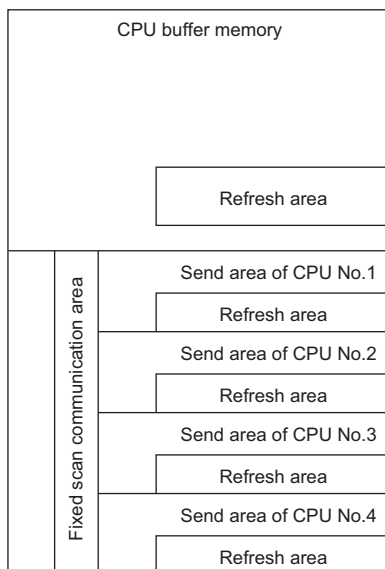
27

### Memory to be used

CPU buffer memory is utilized for data communication between the CPU modules.

#### Memory configuration of CPU buffer memory

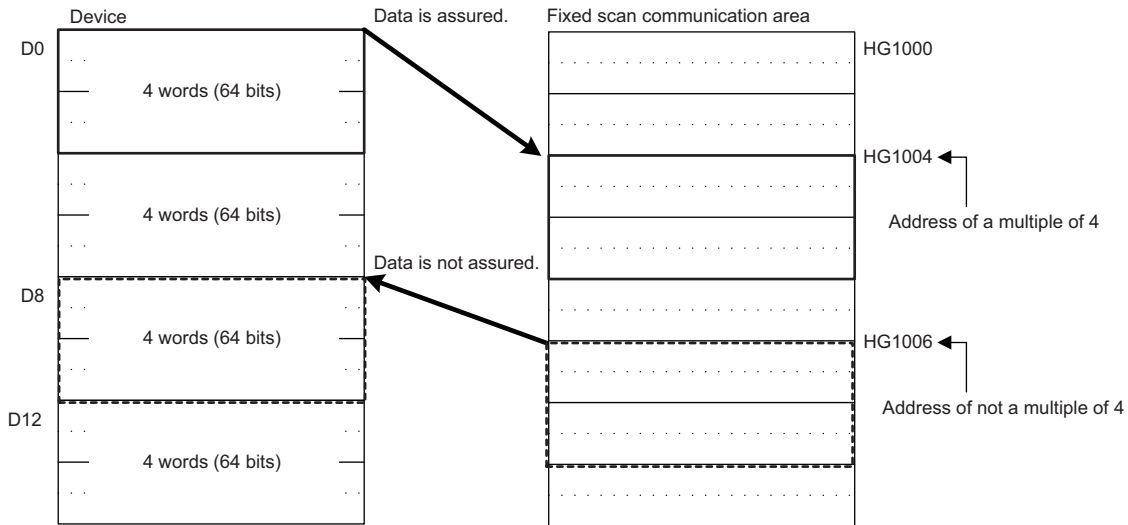
Indicates the memory configuration of CPU buffer memory.



Memory	Communication method	Description	Area size
CPU buffer memory	Communication through direct access	This area is used to read/write data to the areas of the host CPU module or other CPU modules.	512K words fixed
Refresh area	Communication through refresh	Data communication is conducted through refresh at END processing.	
Fixed scan communication area	Communication through direct access	This area communicates the data to the fixed scan communication area of the host CPU module, and is used when the host CPU module area and another CPU module area transfer the data at the fixed scan communication intervals.	It is possible to change within the range of 0 to 24K words in total. The send area as per unit can be set within the range of 0 to 12K words. (Page 202 Setting fixed scan communication area)
The Refresh Area in the Fixed Scan Communication Area	Communication through refresh	The refresh operation is conducted at the fixed scan communication cycle.	

## ■ Avoidance of 64-bit data inconsistency

To avoid 64-bit data inconsistency, access the CPU buffer memory by specifying the start address as a multiple of four similarly to the device to be specified.



(1) Data is guaranteed because the head of the specified address is a multiple of four. (DMOV D0 U3E0\HG1004)

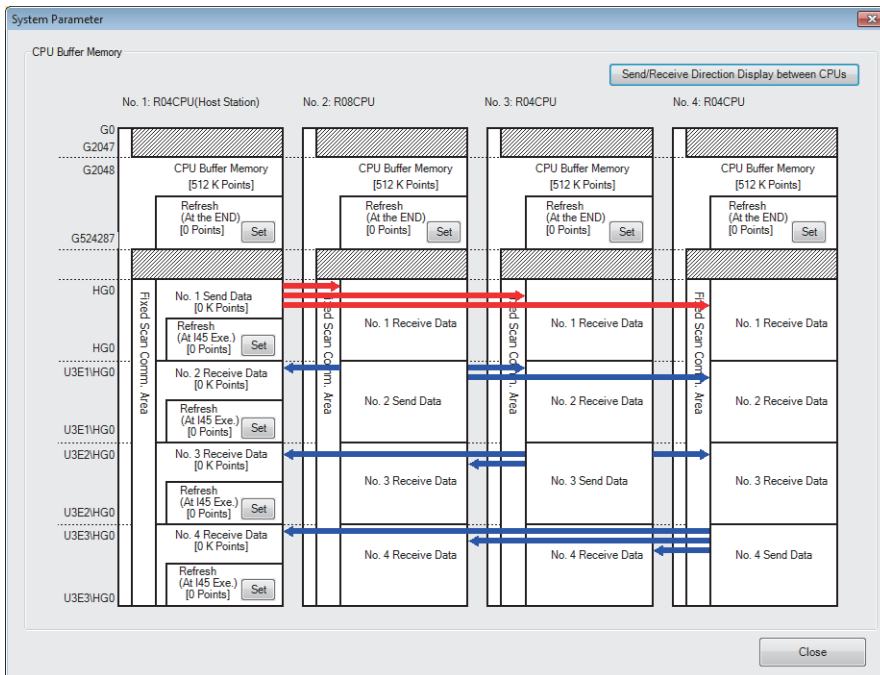
(2) Data is not guaranteed because the head of the specified address is not a multiple of four. (DMOV D0 U3E0\HG1006)

## Checking for the memory configuration

Indicates the CPU buffer memory configuration of each CPU No. Besides, the refresh setting, which is set by the CPU parameter, can also be set through this section. (☞ Page 210 Refresh settings)

☞ [System Parameter] ⇒ "Multiple CPU Setting" ⇒ "Communication Setting between CPU" ⇒ "CPU Buffer Memory Setting" ⇒ "Detailed Setting"

## Window



27

## Displayed items


Item	Description	Setting range	Default
[Setting] button of each refresh area	This menu item sets up the refresh settings used for data communication between CPU modules. (Linked with the refresh settings of CPU parameter)	—	0 points
[Inter-CPU send/receive direction display] button	Click the button, and the arrow indicates the send/receive direction.	—	—

## Setting the data communication with fixed scan communication area

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

### ■Setting whether or not it should be used

To communicate data with the fixed scan communication area, "Enable" must be set to "Fixed Scan Communication Function".

 [System Parameter] ⇒ "Multiple CPU Setting" ⇒ "Communication Setting between CPU" ⇒ "Fixed Scan Communication Function"

### Window


Item	Setting
Fixed Scan Communication Function	Not Use

### Displayed items

Item	Description	Setting range	Default
Fixed Scan Communication Function	Specify whether or not the fixed scan communication function should be used.	<ul style="list-style-type: none"> <li>• Not Use</li> <li>• Use</li> </ul>	Not Use

### ■Setting fixed scan communication area

This menu item sets up the range of send area for each CPU in the fixed scan communication area (the refresh area and the total of areas used for direct access communication). Only the fixed scan communication area can be changed with the parameter settings. The other areas are not configurable.

 [System Parameter] ⇒ "Multiple CPU Setting" ⇒ "Communication Setting between CPU" ⇒ "Fixed Scan Communication Area Setting"

### Window

Item	Setting
Fixed Scan Communication Area Setting	
Total [K Word]	0 K Word
PLC No. 1 [Start XY: U3E0]	0 K Word
PLC No. 2 [Start XY: U3E1]	0 K Word
PLC No. 3 [Start XY: U3E2]	0 K Word
PLC No. 4 [Start XY: U3E3]	0 K Word

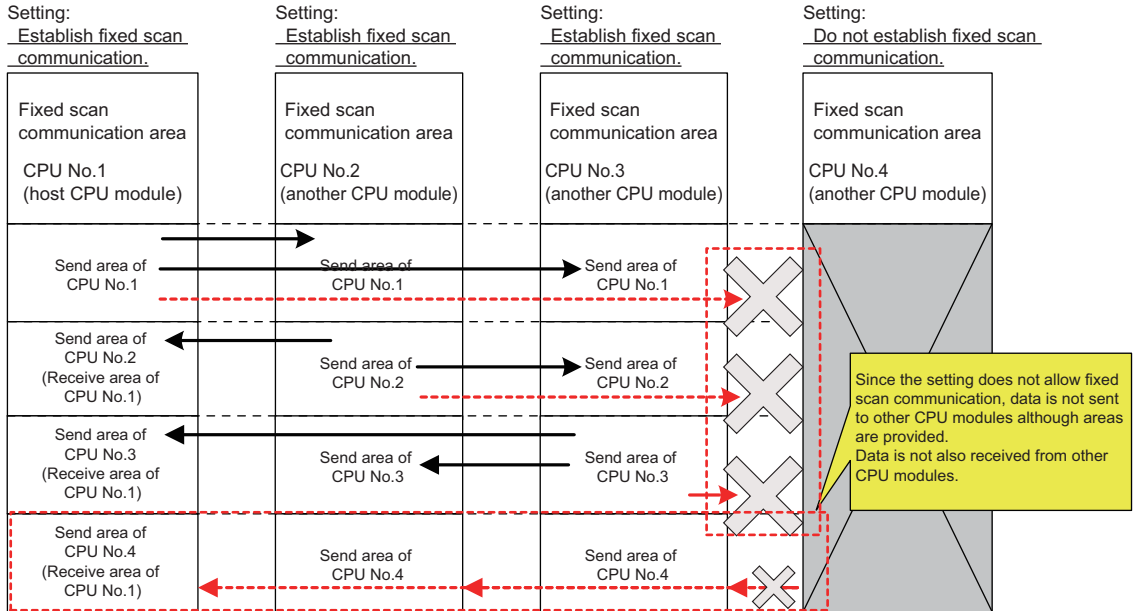
### Displayed items

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

\*1 The max area size that can be set through setting of the fixed scan communication area may differ depending on the fixed scan communication cycle.

Fixed scan communication cycle	Max area size that can be set through setting of the fixed scan communication area
0.10ms	12K points
0.15ms	20K points
Except for shown above	24K points

When there exists a CPU module for which "Disable" is set to "Fixed Scan Communication Function", if the send area of the fixed scan communication area is set to a CPU module for which "Disable" is set to "Fixed Scan Communication Function" (unspecified) in the parameter setting on the host CPU module, no error is generated because the unspecified CPU module is considered as a reserved one for future configuration. Example: in four module configuration with the host CPU module is set to CPU No.1 and the fixed scan communication function of CPU No.4 set to "Disable"

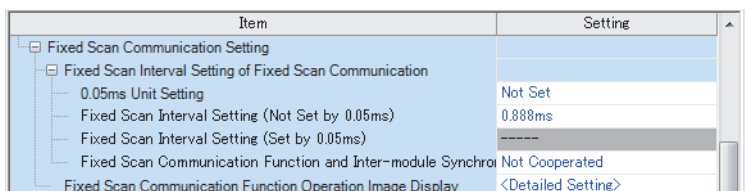


## Fixed scan communication setting

This menu item sets up the interval for data transfer between CPU modules. The data transfer interval can be synchronized with the timing for inter-module synchronization cycle (📖 MELSEC iQ-R Inter-Module Synchronization Function Reference Manual)

🔗 [System Parameter] ⇒ "Multiple CPU Setting" ⇒ "Fixed Scan Communication Setting" ⇒ "Fixed Scan Interval Setting of Fixed Scan Communication"

### Window



### Displayed items

Item	Description	Setting range	Default
0.05ms Unit Setting	Specify whether to set the fixed scan communication cycle in increments of 0.05ms.	<ul style="list-style-type: none"> <li>• Not Set</li> <li>• Set</li> </ul>	Not Set
Fixed Scan Interval Setting (Not Set by 0.05ms)	Select the fixed scan communication cycle to be set from the list items. The same option should be specified only for CPU modules which will use the fixed scan communication function.	<ul style="list-style-type: none"> <li>• 0.222ms</li> <li>• 0.444ms</li> <li>• 0.888ms</li> <li>• 1.777ms</li> <li>• 3.555ms</li> <li>• 7.111ms</li> </ul>	0.888ms
Fixed Scan Interval Setting (Set by 0.05ms)	Specify the fixed scan communication cycle freely in increments of 0.05ms.	0.10ms to 10.00ms	0.10ms
Fixed Scan Communication Function and Inter-module Synchronization Function	Set whether the fixed scan communication cycle cooperates with the inter-module synchronization cycle.	<ul style="list-style-type: none"> <li>• Not Cooperated</li> <li>• Cooperate</li> </ul>	Not Cooperated

### Point

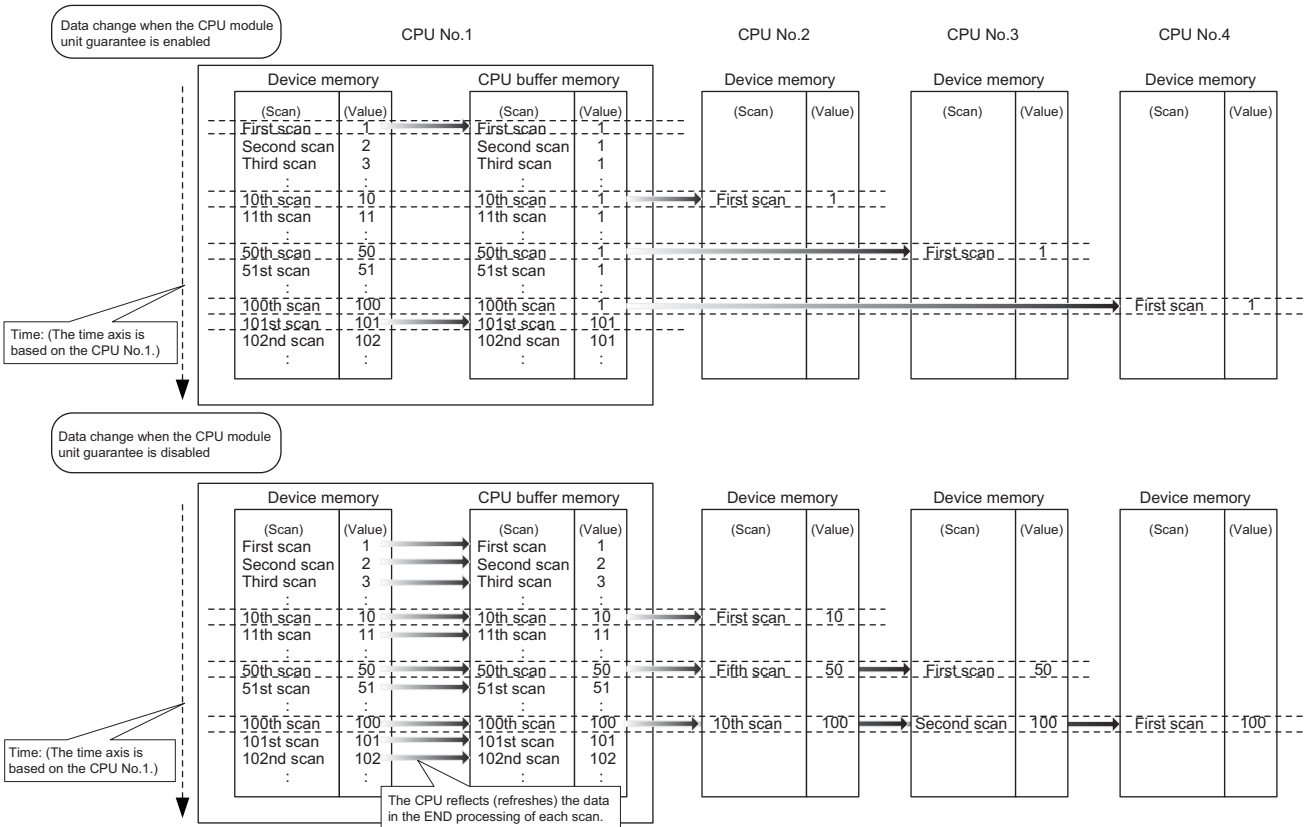
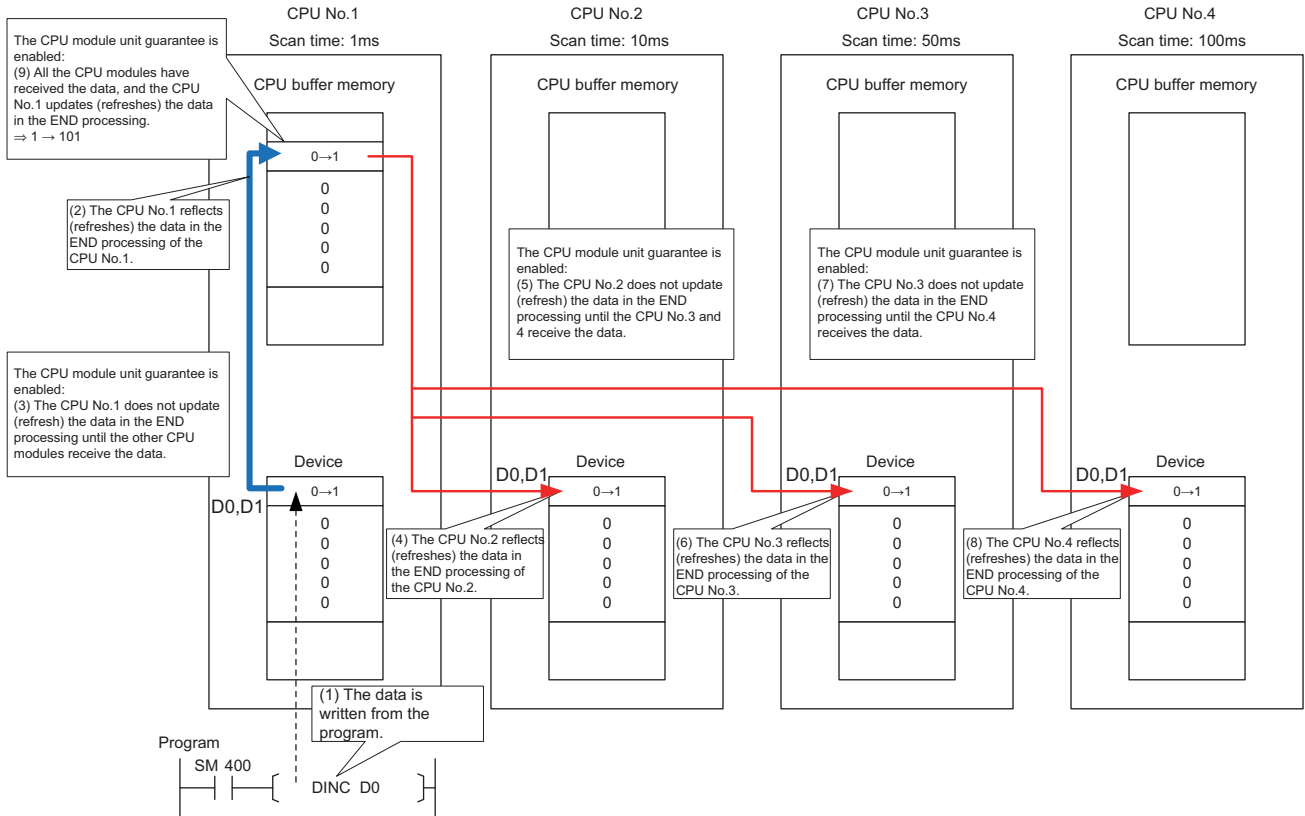
The send image for the fixed scan communication can be reviewed through the "Fixed Scan Communication Function Operation Image Display".

## Error detection setting

When data is communicated among the CPU modules in a multiple CPU system, some data writing timing may lead to a failure of the complete data communication in the fixed scan communication cycle after the data writing and data inconsistency may occur. In this setting, set whether to detect a continuation error or not in this case. (📖 Page 179 Error Detection Setting)

# Module-by-module data guarantee

In data communication, data is handled in units of 64 bits. Therefore, when data larger than 64 bits is handled, old and new data may be mixed for each CPU module depending on the timing between data reading by the host CPU module and data writing by other CPU modules/data receiving from other CPU modules. (Data inconsistency) To avoid this problem, the system conducts the refresh operation by exchanging handshake signals between the CPU modules for guaranteeing refresh data on a module-by-module basis.



## Prevention of data inconsistency by module-by-module data guarantee

The following table shows the preventive control against data inconsistency according to the presence or absence of module-by-module data guarantee.

○: With the preventive control against data inconsistency by the system, ×: Without the preventive control against data inconsistency by the system\*1

Communication method	CPU buffer memory		Fixed scan communication area	
	Module-by-module data guarantee enabled	Module-by-module data guarantee disabled	Module-by-module data guarantee enabled	Module-by-module data guarantee disabled
Communication through refresh	○	×	○	×
Communication through direct access	×	×	○*2	×

\*1 The countermeasure by the program is required. (☞ Page 215 Data guarantee by program)

\*2 Limited to the case of access within the multiple CPU synchronous interrupt program (I45). (☞ Page 216 When accessing fixed scan communication area)

## Module-by-module data guarantee

Set up the module-by-module data guarantee.

☞ [System Parameter] ⇒ "Multiple CPU Setting" ⇒ "Communication Setting between CPU" ⇒ "PLC Unit Data"

### Window

Item	Setting
PLC Unit Data	Disable

### Displayed items

Item	Description	Setting range	Default
PLC Unit Data	Specify whether data is guaranteed on a module-by-module basis.	<ul style="list-style-type: none"> <li>Disable</li> <li>Enable</li> </ul>	Disable

### Point

- For communications by the direct access with the CPU buffer memory other than the fixed scan communication area, data is not guaranteed on a module-by-module basis even when this setting is enabled, because the data on the CPU buffer memory of other CPU modules is directly read after execution of the read instruction.
- Data send/receive timing coincides with an update interval of the CPU module with the slowest scan time, because the next send/receive starts only after all CPU modules have completed data reception.



# Communication through refresh

The device data for each CPU module is written/read only by the parameter settings. Using refresh areas allows data communication between all or a part of the CPU modules in the multiple CPU system, thereby enabling devices of other CPU modules to be used by the host CPU module.

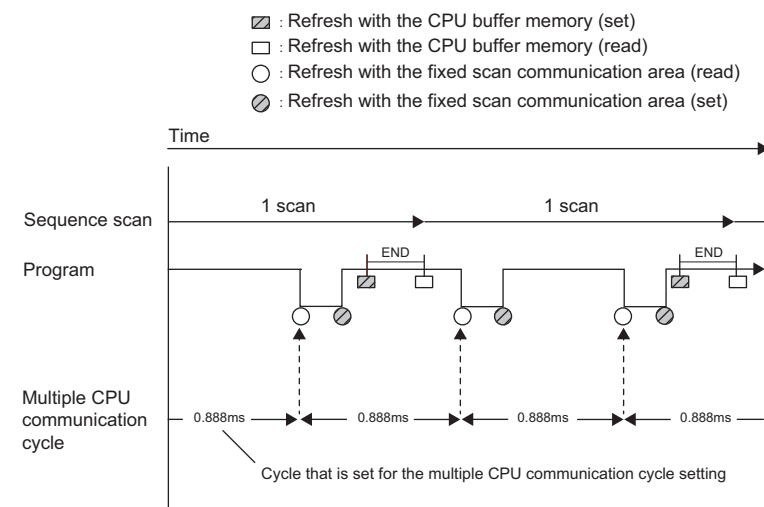
## Types of refresh

The following table shows the types of refresh.

Refresh timing	Application	Memory to be used
At the END processing	For data communication of devices in synchronization with the scan	CPU buffer memory
At the execution of multiple CPU synchronous interrupt program (I45)*1*2	For data communication of devices at the period specified in the fixed scan communication setting	Fixed scan communication area

- \*1 Executed at the period specified in the fixed scan communication setting.
- \*2 Refresh fails if the multiple CPU synchronous interrupt program (I45) does not exist on the program.

The following figure shows the timing of each refresh.

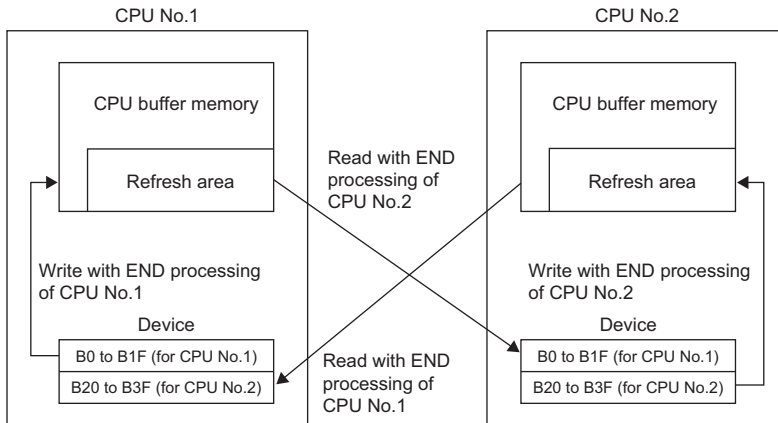


## Refresh using CPU buffer memory

At the END processing of the host CPU module, device data of the host CPU module is written to the refresh area within the CPU buffer memory on the host CPU module. The data written to the refresh area is transferred to the device of another CPU module at the END processing of another CPU module.

**Ex.**

When CPU No.1 refreshes 32 points (B0 to B1F) and CPU No.2 refreshes 32 points (B20 to B3F):

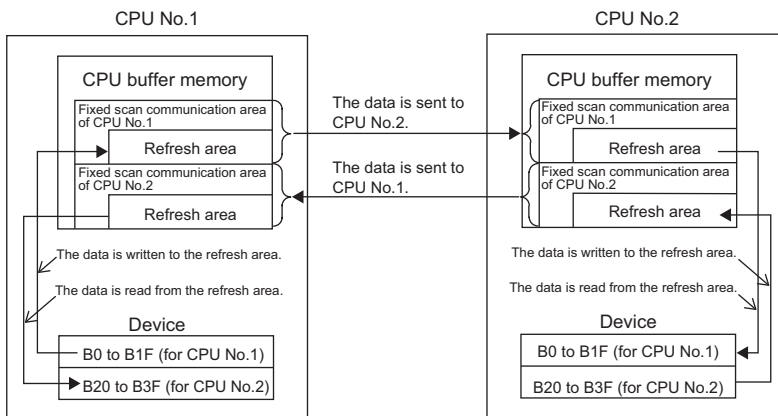


## Refresh using fixed scan communication areas

At the period specified in the fixed scan communication setting, the device data of the host CPU module is written to the refresh area within the fixed scan communication area of the host CPU module. The data written to the refresh area is sent to the refresh area within the fixed scan communication area of another CPU, which in turn reads the transferred data into the device.

**Ex.**

When CPU No.1 refreshes 32 points (B0 to B1F) and CPU No.2 refreshes 32 points (B20 to B3F):



## Executing refresh

Refresh is executed when the CPU module is in RUN and/or STOP (PAUSE) state. For details on the behavior when the CPU module is in stop error state, refer to CPU Module Operation Upon Error Detection Setting. (Page 180 CPU Module Operation Upon Error Detection Setting)

### ■ Behavior during the multiple CPU synchronous interrupt program (I45) execution

If refresh is set to be performed during the multiple CPU synchronous interrupt program (I45) execution, the refresh behavior when the CPU module is in RUN state varies depending on either of the following conditions. (Page 218 Multiple CPU Synchronous Interrupt)

- A program (I45 to IRET) which includes the multiple CPU synchronous interrupt program (I45) exists.
- Event execution type program exists.

The following table lists the refresh behaviors.

○: Execute refresh, ×: Not execute refresh

Execution type		Refresh behavior		Refresh behavior after the program control instruction is executed
		STOP state	RUN state	RUN state
Other than event execution type	The I45 interrupt pointer doesn't exist	○	×	×
	The I45 interrupt pointer exists	○	○	○
Event execution type (I45 interrupt specified)		○	○	× (stop after changing the execution type <sup>*1</sup> )

\*1 The PSCAN/PSTOP instruction changes after the next scan, and the POFF instruction changes after the next two scans.

○: Execute refresh, ×: Not execute refresh, —: Execution disabled

Execution type		Refresh behavior triggered by operating status change of the CPU module						
		RUN state	Upon addition of the I45 interrupt pointer during online program change	Upon removal of the I45 interrupt pointer during online program change	RUN to STOP state	STOP state	After the refresh setting is changed; when parameters are written	STOP to RUN state; turning power off and on; resetting
Other than event execution type	The I45 interrupt pointer doesn't exist	×	○	×	○	○	○ <sup>*3</sup>	×
	The I45 interrupt pointer exists	○	○	×	○ <sup>*2</sup>	○	○ <sup>*3</sup>	○ <sup>*4</sup>
Event execution type (I45 interrupt specified)		○	—	—	○ <sup>*2</sup>	○	○ <sup>*3</sup>	○ <sup>*4</sup>

\*2 Not affect to the behavior.

\*3 Operates with the pre-change parameters.

\*4 Operates with the post-change parameters.

## Configurable data

This section lists the configurable data with the refresh settings.

### ■Maximum number of settings

Maximum of 32 settings can be configured per CPU module for any refresh setting.

### ■Data number of points

Data number of points can be configured within the number of points assigned to the area (within the send range of the fixed scan communication area) in increments of two points. Device duplication between multiple setting numbers is not allowed.

### ■Data that can be specified

The device other than local device can be specified. However, when "Use File Register of Each Program" is enabled, file registers cannot be specified. Doing so may prevent the device from operating at file registers for each program depending on the CPU operating status.

Type	Devices that can be specified
Bit device*1	X, Y, M, L, B
Word Device	D, W, R, ZR, RD

\*1 This device can be specified only in units of 16 points (one word).

## Refresh settings

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

 [CPU Parameter] ⇒ "Refresh Setting between Multiple CPU"

### Operating procedure

"Refresh Setting between Multiple CPU" window

Item	Setting
<input type="checkbox"/> Refresh Setting (At the END)	
Refresh Setting (At the END)	<Detailed Setting>
<input type="checkbox"/> Refresh Setting (At I45 Exe.)	
Refresh Setting (At I45 Exe.)	<Detailed Setting>

"Detailed Setting" window

Setting No.	Device		
	Points	Start	End
<input type="checkbox"/> No. 1(Send)			
<input type="checkbox"/> Total	0/522240 Points		
1			
2			
3			

1. Click "Detailed Setting" at the execution timing for each refresh.
2. Enter the start/end of the device.

### Point

Set a blank column for "Head" and "End" and set only number of points so that the setting by which auto refresh is not conducted can be configured. (It is possible to perform setting which does not allow the refresh data of other CPU modules, which is unwanted for the host CPU module, to be obtained.)

### ■Enabling timing of refresh settings

The refresh settings are enabled at the following timing:

- When power is turned off and on or the system is reset:
- When the mode is switched from STOP to RUN after the parameters are written to the CPU module (STOP, RUN, STOP, and RUN):

## CPU module operating status and refresh behavior

When the refresh timing coincides with the period specified in the fixed scan communication setting, the refresh behavior depends on the operating status of the CPU module.

CPU module operating status	Refresh enabled/disabled	Refresh timing	
		CPU buffer memory	Fixed scan communication area
RUN (including a continuation error period)	Execute refresh	During the END processing	Before and after the multiple CPU synchronous interrupt program (I45) execution <sup>*1</sup>
STOP (including a stop error period due to a moderate error)	Execute refresh	During the END processing	During the END processing
STOP (major error period)	Not execute refresh	—	—
PAUSE	Execute refresh	During the END processing	During the END processing

\*1 In the disabling interrupt state by the DI instruction, the refresh is not operational, because the multiple CPU synchronous interrupt program (I45) does not work.

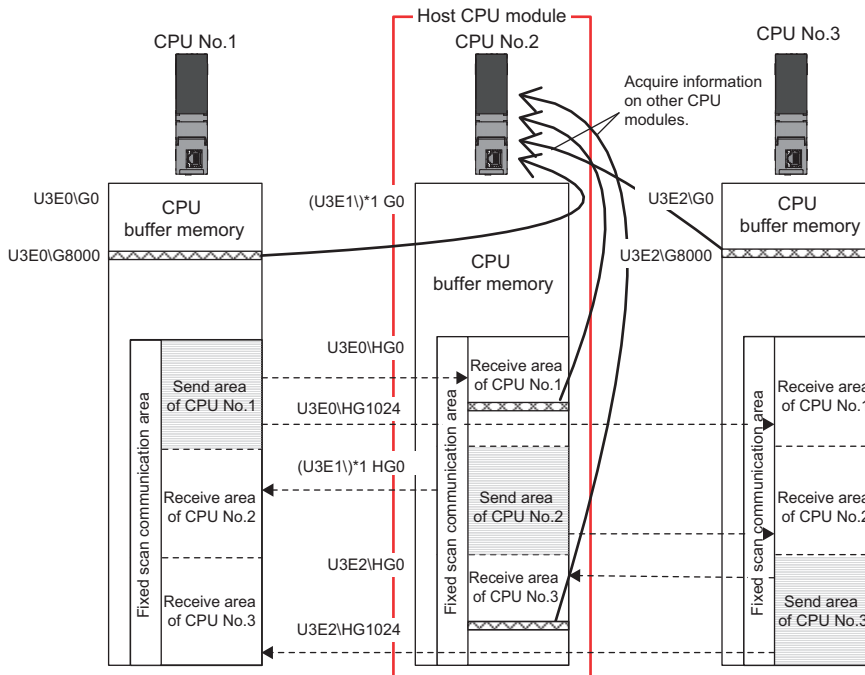
# Communication through direct access

This method uses programs to communicate with other CPUs. The following table lists the communications using the direct access method.

Communication method	Description	Instruction to be used
Communication using CPU buffer memory	Transfer data between CPU modules using any area on the CPU buffer memory.	<ul style="list-style-type: none"> <li>• FROM/TO instruction</li> <li>• DFROM/DTO instruction</li> <li>• Instruction using the CPU buffer memory access device (U3En\G□)</li> </ul>
Communication using the fixed scan communication area	Transfer data between CPU modules using the fixed scan communication area.	Instruction using the CPU buffer memory access device (U3En\HG□)
Communication by a dedicated instruction	Transfer data between CPU modules in the multi CPU system and/or sends control command to a CPU module of other CPUs using a dedicated instruction.	DDWR/DDRD instruction and others

## Specification method through CPU buffer memory access device

Specify the CPU buffer memory as "U3En\Gn," or the fixed scan communication area as "U3En\HGn" when accessing these memory/area (☞ Page 246 CPU Buffer Memory Access Device)



### ■ Obtaining information stored on the CPU buffer memory

Access the CPU buffer memory.

### ■ Obtaining information stored on the fixed scan communication area for other CPUs

Access the fixed scan communication area on the host CPU module. Obtain data sent to the fixed scan communication area on the host CPU module at the fixed scan communication cycle.

## Communication using CPU buffer memory and fixed scan communication area

This section describes the communication using CPU buffer memory and fixed scan communication area.

### Available area for communication

The following area can be used for communication.

Area	Description
CPU buffer memory	All the CPU buffer memory area except for the refresh area is available. The start address of the available area for each CPU module varies depending on the refresh settings. The end address of the area is fixed by CPU module models.
Fixed scan communication area	All the fixed scan communication area except for the refresh area is available. The start address of the available area for each CPU module is HG0 and the end address varies depending on the refresh settings.

27

### Instructions to be used for communication

Communication with each CPU module is enabled by issuing the following read/write instructions to each area.

- Write instruction: the instructions using the CPU buffer memory access device<sup>\*1</sup> and TO/DTO instruction
- Read instruction: the instructions using the CPU buffer memory access device<sup>\*1</sup> and FROM/DFROM instruction

\*1 Specify "U3En\G□" when accessing the CPU buffer memory and "U3En\HG□" when accessing the fixed scan communication area.

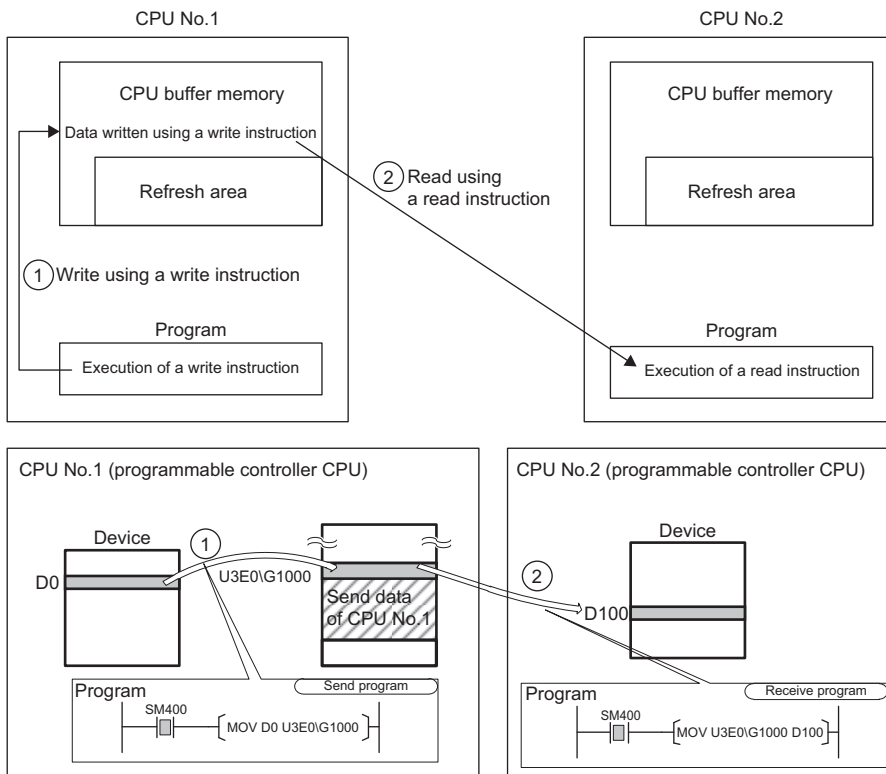
### Data communication behavior

- When using an area within the CPU buffer memory

Data written to the area within the CPU buffer memory on the host CPU module using the write instruction can be read by other CPU modules using the read instruction. Unlike the refresh, data registered during the instruction execution can be directly read.

Ex.

When data written to the CPU buffer memory on the CPU No.1 using the write instruction is read by CPU No.2 using the read instruction:

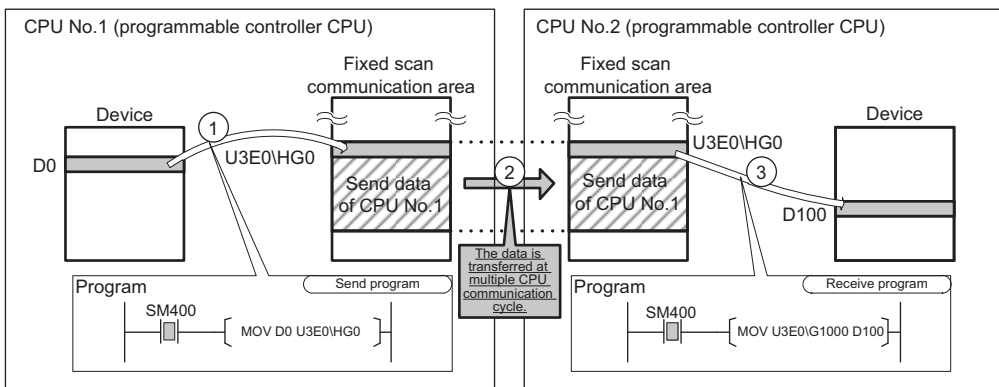
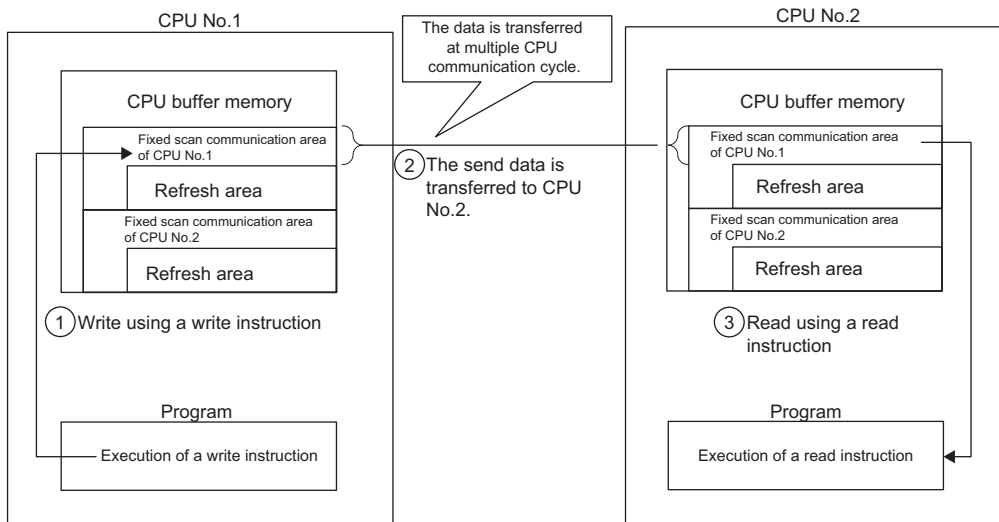


- When using an area within the fixed scan communication area

Data written to the area within the fixed scan communication area on the host CPU module using the write instruction is sent to other CPU modules at the period specified in the fixed scan communication setting. Other CPU modules read the received data using the read instruction. Unlike the refresh, data registered during the instruction execution can be directly read.

**Ex.**

When data written to the fixed scan communication area on the CPU No.1 using the write instruction is read by CPU No.2 using the read instruction:





# Data guarantee by program

This section describes how to avoid the inconsistency of data larger than 64 bits using the program. To set up the module-by-module data guarantee using the parameters, use the multiple CPU setting. (Page 206 Module-by-module data guarantee)

## Data guarantee in communication through the refresh

Inconsistency of transferred data can be avoided by setting the interlock device to a transfer number lower than the one for the transferred data, because data is transferred in descending order from the highest setting number in the refresh settings.

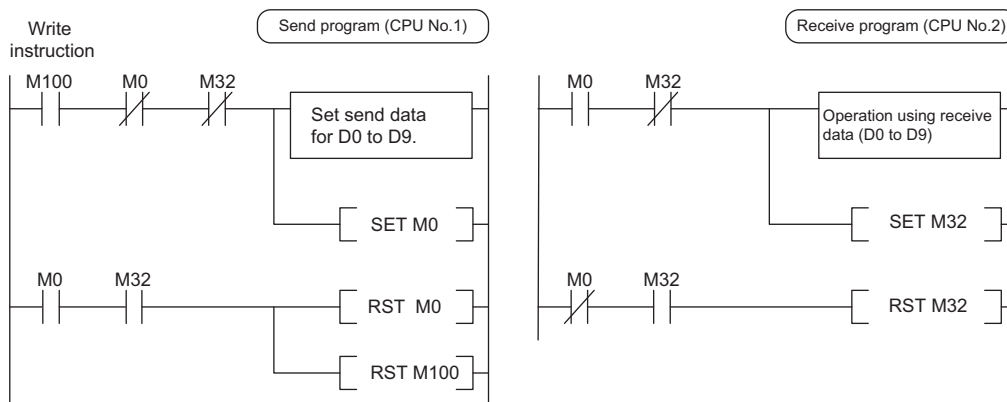
**Ex.**

Interlock program in communication by refresh

- Parameter settings

CPU No.1 refresh setting							Direction	CPU No.2 refresh setting						
CPU No.	Transfer No.	Send/receive range for each CPU module			Send/receive device setting			CPU No.	Transfer No.	Send/receive range for each CPU module			Send/receive device setting	
		Number of points	start	end	start	end				Number of points	start	end	start	end
CPU No.1	Transfer No.1	2	0	1	M0	M31	→	CPU No.1	Transfer No.1	2	0	1	M0	M31
	Transfer No.2	10	2	11	D0	D9			Transfer No.2	10	2	11	D100	D109
CPU No.2	Transfer No.1	2	0	1	M32	M63	←	CPU No.2	Transfer No.1	2	0	1	M32	M63

- Program example



## Data guarantee for communication through direct access

The behavior varies depending on the area to be accessed.

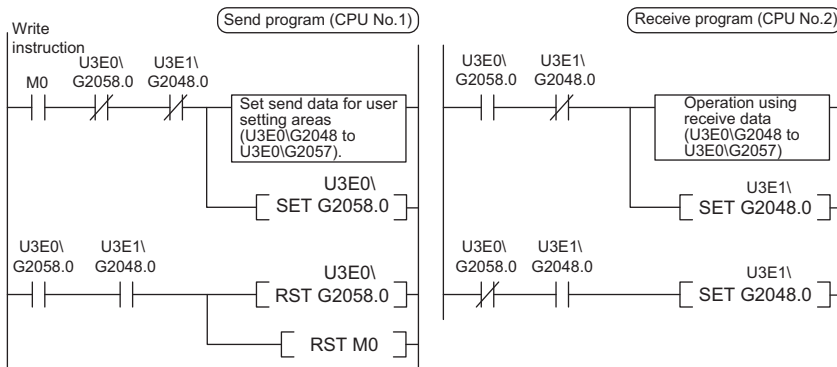
### ■When accessing CPU buffer memory:

The program reads data in ascending order from the start address of the CPU buffer memory other than the refresh area, and the write instruction writes send data in descending order from the end address of the CPU buffer memory other than the refresh area. Therefore data inconsistency can be avoided by setting an interlock device at the start position of data to be communicated.

**Ex.**

Interlock program in communication by direct access

- Program example



### ■When accessing fixed scan communication area

When accessing within the multiple CPU synchronous interrupt program (I45), enabling the setting of module-by-module data guarantee eliminates the need of an interlock circuit. When this refresh area is accessed within a program other than the above, or when the setting of module-by-module data guarantee is disabled, an interlock circuit is required, as with the access to the CPU buffer memory. (☞ Page 205 Module-by-module data guarantee)

# Communication between CPU modules in error state

The following section describes communication between CPU modules in an error state.

## Behavior in receive data error state

A CPU module receiving illegal data due to noise and/or failure discards the received data. If a received data is discarded, the receive-side CPU module keeps the last data received before discarding. When the CPU module receives the next correct data, it updates the received data.

## Refresh execution in an error state

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

Error		Refresh <sup>*1</sup>	Data communication between CPU modules <sup>*2</sup>
Minor error		○	○
Moderate error	Causes other than the following item	○	○
	Parameter error for fixed scan communication function (including the consistency check during start-up)	× <sup>*3*4</sup>	× <sup>*3*4</sup>
Major error		× <sup>*3</sup>	× <sup>*3</sup>

\*1 Indicates data transfer between user devices and the fixed scan communication area on the host CPU module.

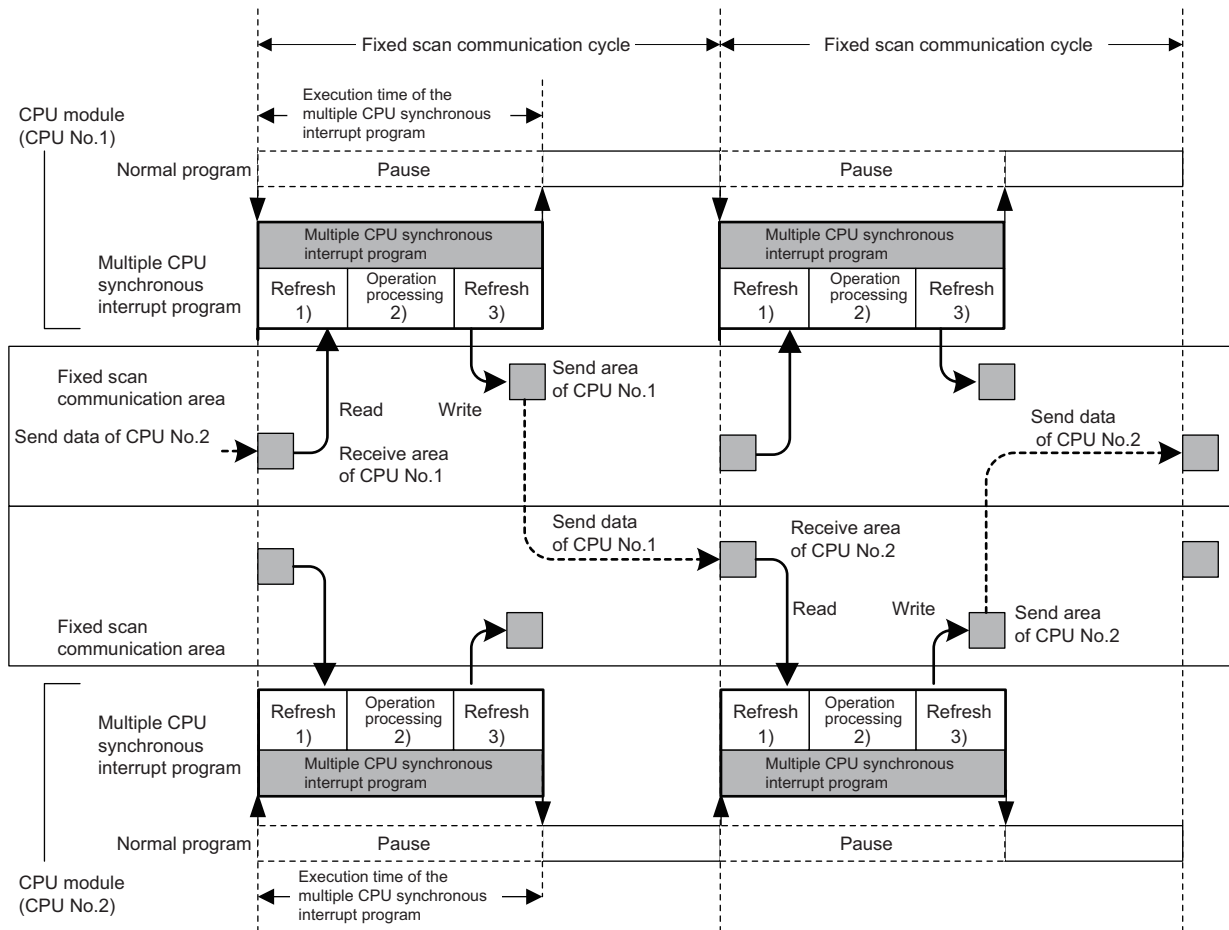
\*2 Indicates data communication between the fixed scan communication areas on the host CPU module and other CPU modules.

\*3 When an error occurs during normal operation, normal data generated immediately before the error occurs is continued to be sent between the fixed scan communication areas on the host CPU module and other CPU modules.

\*4 If the consistency check fails due to a parameter change in normal operation, refresh and data send/receive between the CPU modules are continued.

# 27.5 Multiple CPU Synchronous Interrupt

Triggers an interrupt program at the fixed communication timing set by a parameter. The interrupt program executed at the fixed scan communication cycle is called "Multiple CPU synchronous interrupt program". Using the multiple CPU synchronous interrupt enables synchronizing with the fixed scan communication cycle so that data communication between CPU modules can be made. (It allows synchronizing the data communication timing between CPU modules.)



- ❶ Data reading of other CPU modules (refresh): Data sent from other CPU modules is read to a device or a label. (Data is read from the receive area of the host CPU module.)
- ❷ Operation processing: The multiple CPU synchronous interrupt program is executed.
- ❸ Data sending of other CPU modules (refresh): Data to be sent to other CPU modules is written from a device or a label. (Data is written to the send area of the host CPU module.)

**Point**

The operation methods required when an interrupt factor occurs and the program creating methods are the same as those for normal interrupt program. (☞ Page 45 Interrupt program)

# Execution timing

The multiple CPU synchronous interrupt program (I45) is executed at the timing for the fixed scan communication cycle. The fixed scan communication cycle can be changed through the fixed scan communication setting. (☞ Page 204 Fixed scan communication setting)

**Point**

It is also possible to perform refresh during the multiple CPU synchronous interrupt program (I45) in execution. (☞ Page 207 Communication through refresh)

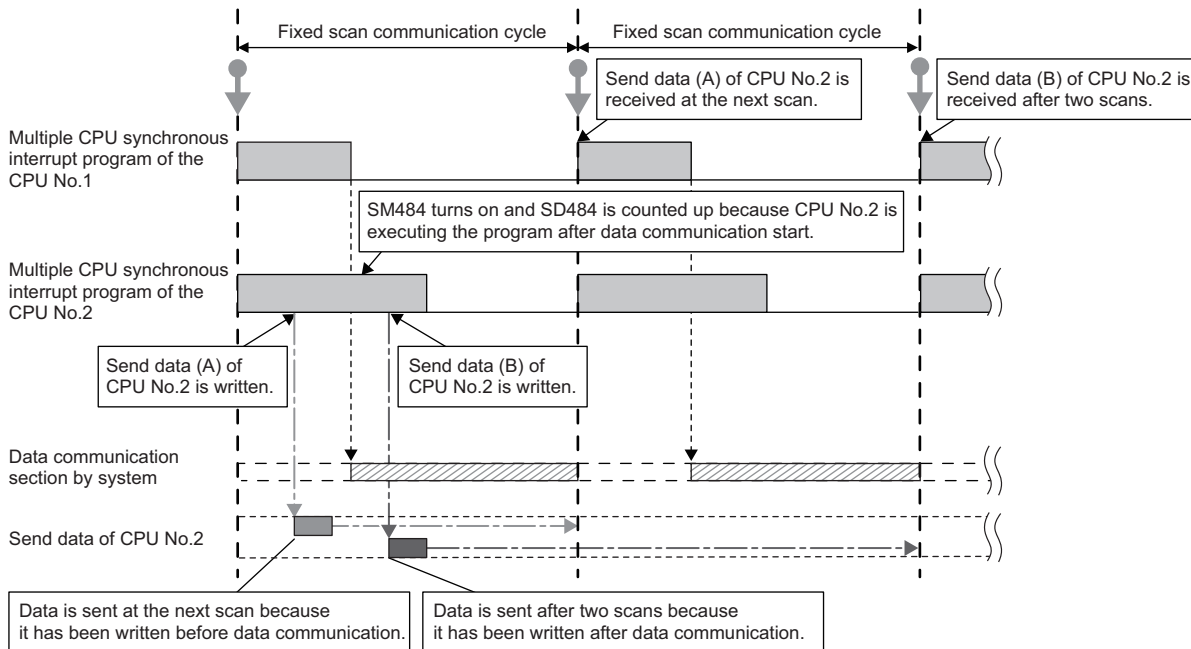
## Multiple interrupt

For the multiple interrupt of the multiple CPU synchronous interrupt program, refer to the multiple interrupt function. (☞ Page 84 Multiple Interrupt Function)

## Precautions

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

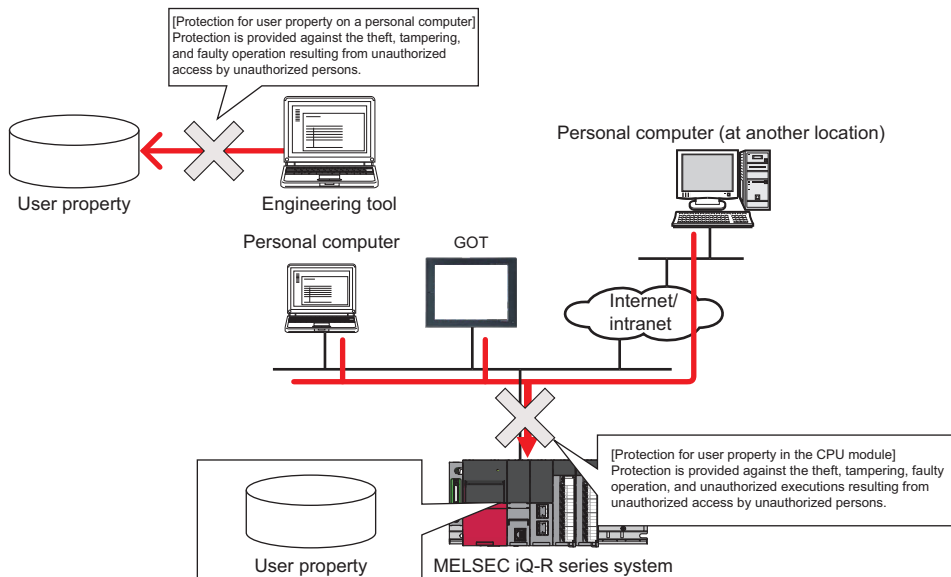
- Create a multiple CPU synchronous interrupt program so that it has the execution processing time shorter than the fixed scan communication cycle. If the interrupt program has the execution processing time equal to or longer than the cycle, the multiple CPU synchronous interrupt interval cannot be guaranteed. (☞ Page 46 Operation upon occurrence of an interrupt factor) The execution time of the multiple CPU synchronous interrupt program (I45) can be monitored using the RAS setting of the CPU parameter. (☞ Page 179 Error Detection Setting)
- To send data successfully in the next fixed scan communication cycle, select "Detect" for "Program Execution Section Exceed (I45)" in "RAS Setting" of [CPU Parameter]. (☞ Page 179 Error Detection Setting) With this setting, when data is written after the host CPU module starts data transfer in the multiple CPU synchronous interrupt program, SM484 (Execution section excess error flag for multiple CPU synchronization interrupt program) is turned on and the number of data sending errors in the next cycle is stored in SD484 (Number of execution section excess errors for multiple CPU synchronization interrupt program). In addition, when the data is written while the data cannot be sent in the fixed cycle communication cycle set by the parameter, an error can be detected. (The CPU module continues its operation.)





# 28 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:



Data to be protected	Purpose	Function	Reference
Project	Prevents unauthorized access to programs (in units of POU). (A password is used.)	Block password function	GX Works3 Operating Manual
	Prevents unauthorized access to programs (in units of program file). (A security key is used.)	Security key authentication function	
CPU module	Prevents unauthorized execution of programs. (A security key is used.)		
	Prevents unauthorized read and write of files. (A password is used.)	File password function	<ul style="list-style-type: none"> <li>• GX Works3 Operating Manual</li> <li>• MELSEC iQ-R Ethernet User's Manual (Application)</li> <li>• SLMP Reference Manual</li> <li>• MELSEC- iQ-R Serial Communication Module User's Manual (Application)</li> </ul>
	Blocks access from an invalid IP address by identifying the IP address of an external device via Ethernet.	IP filter function	MELSEC iQ-R Ethernet User's Manual (Application)
	Limits access to just specific communication routes via Ethernet. (A password is used.)	Remote password function	

## Point

If a personal computer with a security key registered is abused by an outsider, there is no way to prevent the outflow of the program property, and thus the user needs to take adequate measures as shown below:

- Preventive measures against the theft of a personal computer (for example, wire locking)
- Management of users of the personal computer (for example, deletion of unnecessary user accounts, strict control of login information, and implementation of fingerprint authentication)

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

- Import the registered security key to another personal computer.
- Export the registered security key to a file and store the file in a safe place.

# MEMO

---



# PART 3

# DEVICES AND CONSTANTS

Part 3 consists of the following chapters.

29 DEVICES

---

30 CONSTANTS

---

31 HOW TO USE DEVICES CONVENIENTLY

---


# 29 DEVICES

This chapter describes the devices.

## 29.1 Device List

This section lists the devices.

Classification	Type	Device name	Symbol	Number of points of Default	Range changeable through setting of the parameters	Decimal/hexadecimal
User device	Bit	Input	X	12K points	Unchangeable	Hexadecimal
	Bit	Output	Y	12K points		Hexadecimal
	Bit	Internal relay	M	12K points	Changeable (☞ Page 109 Device Setting)	Decimal
	Bit	Latch relay	L	8K points		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		
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	JnX	16384 points maximum <sup>*1</sup>	Unchangeable	Hexadecimal
	Bit	Link output	JnY	16384 points maximum <sup>*1</sup>		Hexadecimal
	Bit	Link relay	JnB	32768 points maximum <sup>*1</sup>		Hexadecimal
	Bit	Link special relay	JnSB	512 points maximum <sup>*1</sup>		Hexadecimal
	Word	Link register	JnW	131072 points maximum <sup>*1</sup>		Hexadecimal
	Word	Link special register	JnSW	512 points maximum <sup>*1</sup>		Hexadecimal
Module access device	Word	Module access device	UnG	268435456 points maximum <sup>*1</sup>	Unchangeable	Decimal
CPU buffer memory access device	Word	CPU buffer memory access device	U3EnG	268435456 points maximum <sup>*1</sup>	Unchangeable	Decimal
			U3EnHG	12288 points maximum	Changeable	Decimal
Index register	Word	Index register	Z	20 points	Changeable (☞ Page 248 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

Classification	Type	Device name	Symbol	Number of points of Default	Range changeable through setting of the parameters	Decimal/hexadecimal
Nesting	—	Nesting	N	15 points	Unchangeable	Decimal
Pointer	—	Pointer	P	8192 points <sup>*2</sup>	Changeable (  Page 255 Pointer setting)	Decimal
	—	Interrupt pointer	I	1024 points	Unchangeable	Decimal
Other devices	—	Network No. specification device	J	—	Unchangeable	Decimal
	—	I/O No. specification device	U	—		Hexadecimal

\*1 These are the maximum points that can be handled in the CPU module. The number of points actually used differs depending on the module used.

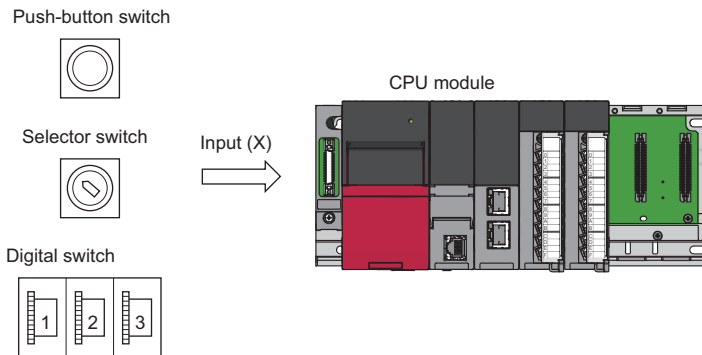
\*2 The default number of points is 16384 for the R120CPU and R120PCPU.

## 29.2 User Device

This chapter describes the user device.

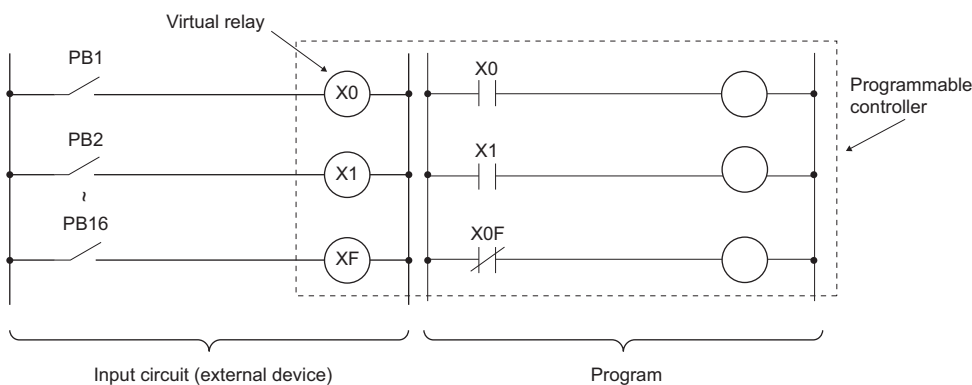
### Input (X)

This device provides the CPU module with commands and/or data using an external device, such as pushbutton, transfer switch, limit switch, and digital switch.



### Concept of input

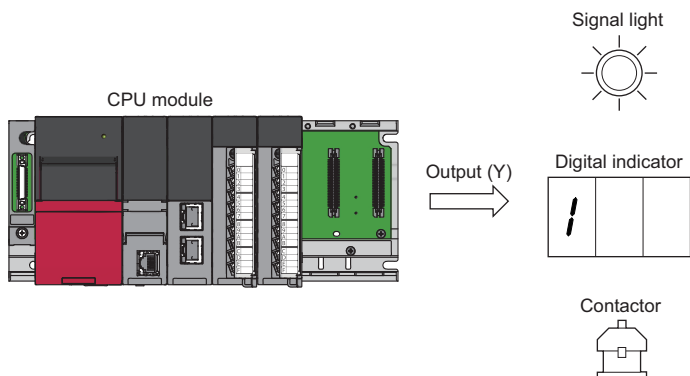
Assume that one virtual relay  $X_n$  is incorporated into the CPU module for each input point. In the program, a normally open contact and normally closed contact for the  $X_n$  are used.



Also, the input can be used as a target re-flesh (CPU module side) device of the remote input (RX), such as the CC-Link IE Field Network.

### Output (Y)

This device outputs the control results of the program to various devices, such as external signal light/digital HMI/ electromagnetic switch (contactor)/solenoid.



## Internal relay (M)

This device is used as an auxiliary relay within the CPU module. The following operations turn off all the internal relays.

- Turning power of the CPU module off and on
- Reset
- Latch clear

## Latch relay (L)

This device is an auxiliary relay which enables latching (data retention during power failure). This device is used within the CPU module. This device latches operation results (ON/OFF information) even after the following operations.

- Turning power of the CPU module off and on
- Reset

## Link relay (B)

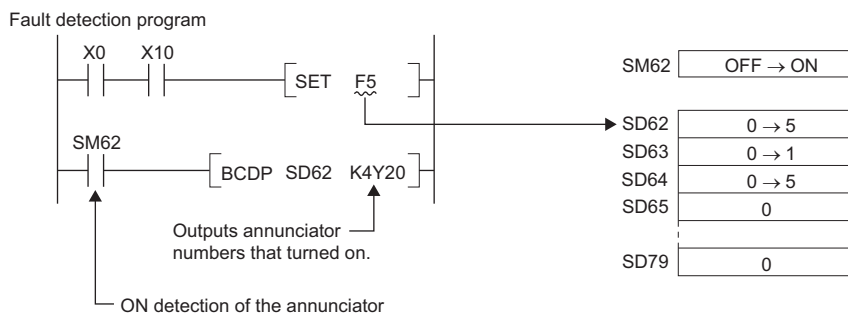
This device is used as a CPU side device when refreshing bit data between the network module, such as the CC-Link IE Controller Network module and the CPU module.

### Refreshing network modules using link relay

Data are transferred/received between the link relay (B) within the CPU module and the link relay (LB) of the network module, such as the CC-Link IE Controller Network module. The refresh range is specified using parameters on the network module. The part which is not used for refreshing can be used for other applications.

## Annunciator (F)

This device is an internal relay used for the program which detects malfunction/failure of customer-created equipment. When the annunciators are turned on, SM62 (annunciator detection) is turned on, the number of activated annunciators and their device numbers are stored in SD62 (annunciator number) to SD79 (annunciator detection number table).



Also the number of the annunciator turned on first (the number stored in SD62) is registered in the event history.

### Point

Only one annunciator number is registered in the event history while power is turned on.

## On/off method for annunciator

Annunciators are turned on by either the SET F□ instruction or the OUT F□ instruction. Annunciators are turned off by the RST F□ instruction or the LEDR instruction or the BKRST instruction.

### Point

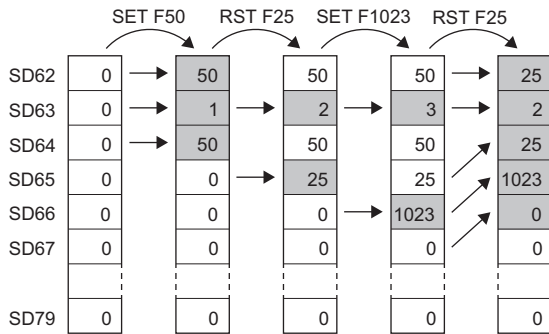
When the annunciators are turned on/off using any methods (e.g. the MOV instruction) other than shown above, the operation is the same as that of internal relays. As a result, SM62 is not turned on and annunciator numbers are not stored into SD62 and SD64 (annunciator detection number table) to SD79.

### ■ Operations when annunciators are turned on

1. The annunciator numbers turned on are stored sequentially into SD64 to SD79.
2. The annunciator number stored into SD64 is stored into SD62.
3. SD63 value (the number of annunciators) is incremented by one.

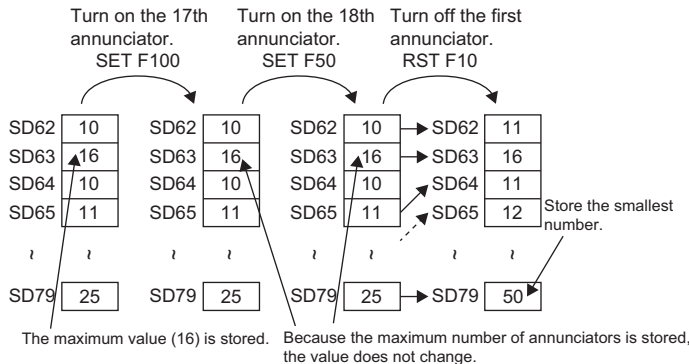
### ■ Operations when annunciators are turned off

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.



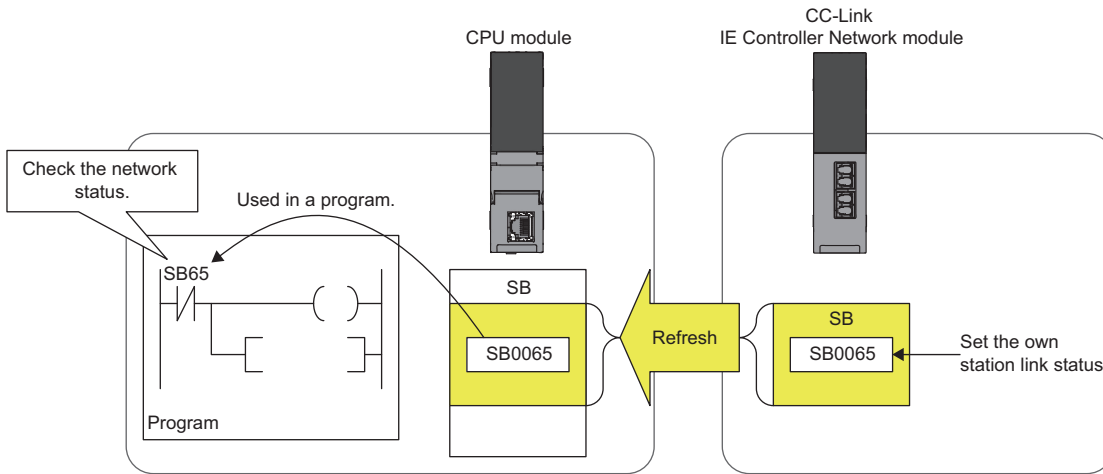
### Point

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.



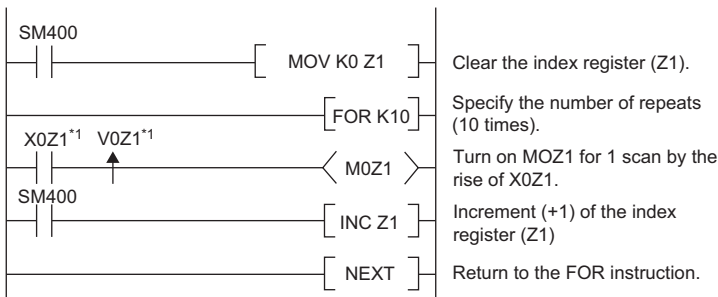
# Link special relay (SB)

Communication status and/or error detection status for network modules, such as the CC-Link IE Controller Network module are output into 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.

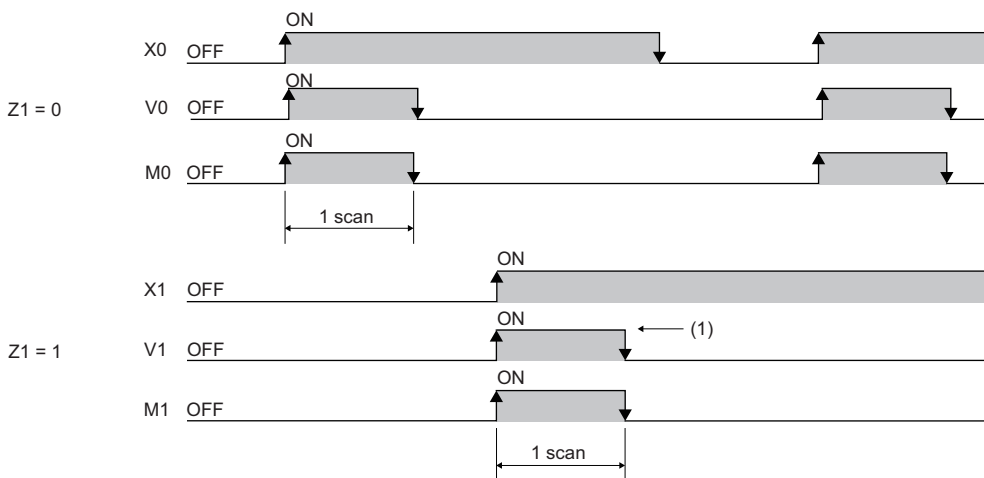


# Edge relay (V)

The edge relay is a device that memorizes operation results (on/off information) from the head of the ladder block, allowing its use only by the EGP/EGF instruction. This device is executed for various objectives such as the rising (off and on) detection in the structured programs by the index modification.



\*1 Edge relay V0Z1 memorizes on/off information of X0Z1.



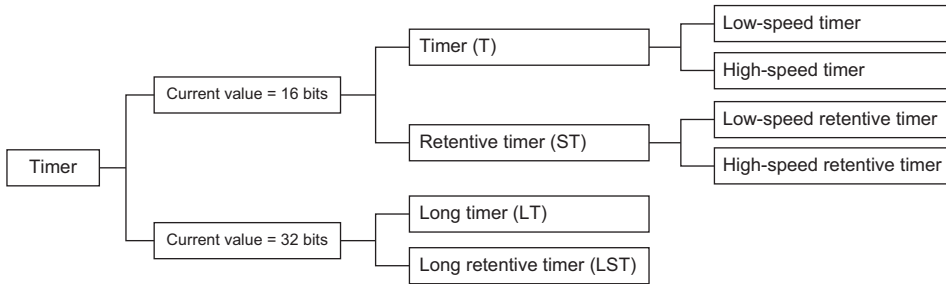
(1) When X1 is rising, this device is turned on for one scan duration.

# 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

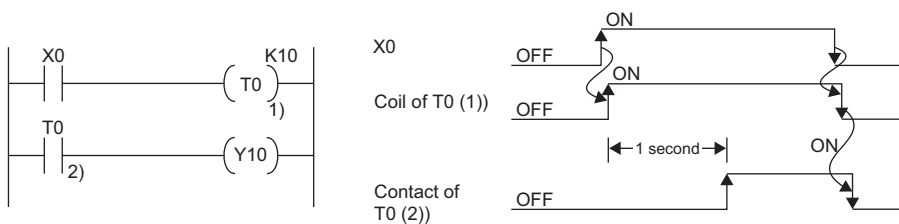
There are two types of timers: timer (T) which retains the current value in 16-bit units and long timer (LT) which retains it in 32-bit units. The timer (T) and the long timer (LT) are different devices and the number of device points can be set for each of them. In addition, there are the retentive timer (ST) and the long retentive timer (LST), both of which retain the current value even if the coil is turned off. \*1



\*1 For the timer (T)/long timer (LT), the current value returns to 0 when the coil is turned off.

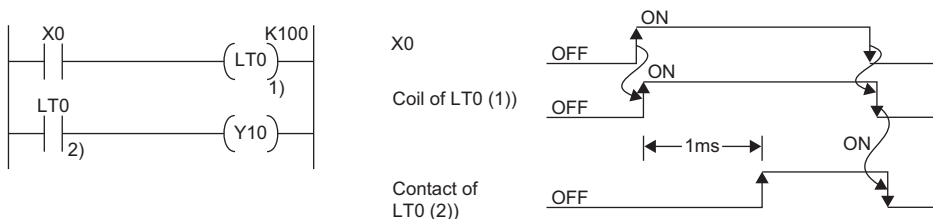
### ■Timer (T)

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 LTI 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.



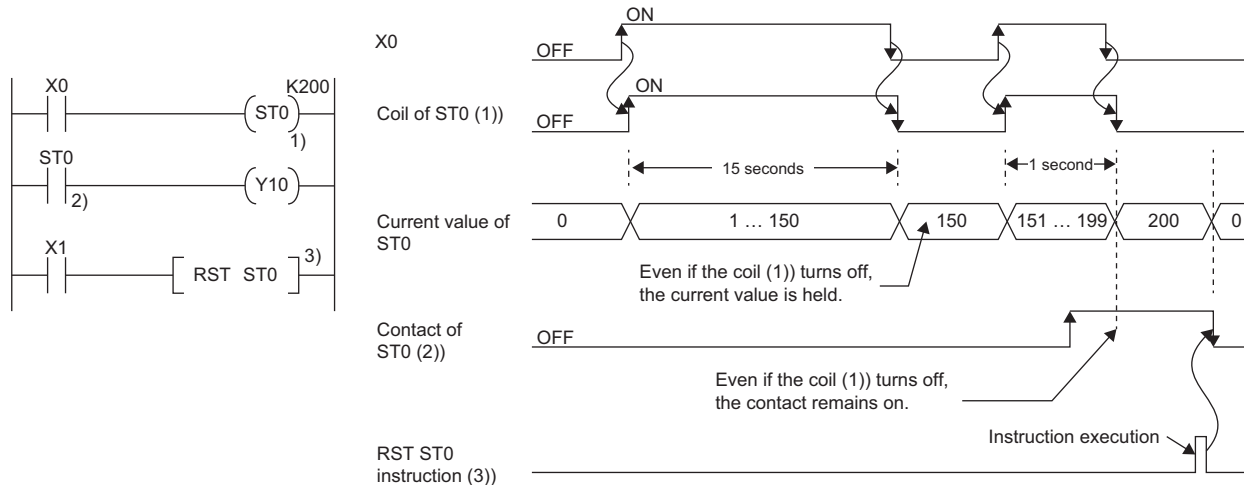
\*1 This figure shows the example when the long timer time limit value setting is 0.01ms.



- For the long timer contact to be turned on after the long timer coil is turned on, the next coil must be executed after the long timer coil turned off and on. The long timer contact is not turned on simultaneously when the long timer coil is off and on.
- The long timer (LT) can be used in interrupt programs. (☞ Page 45 Interrupt program)

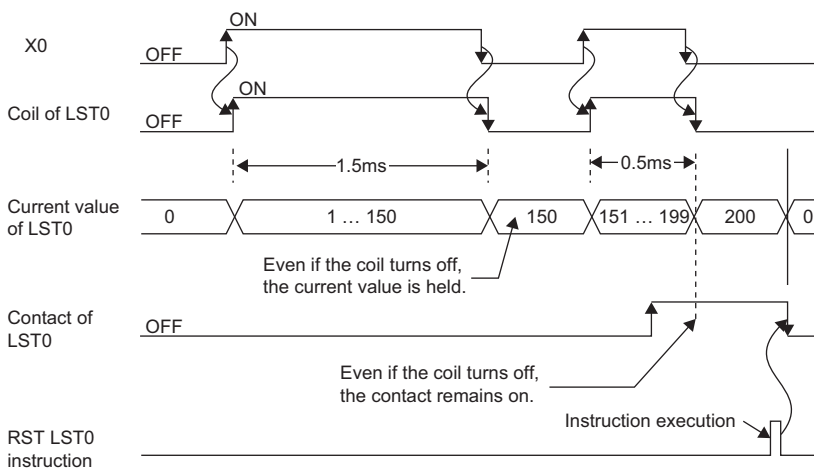
### Retentive timer (ST)

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.



### 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.



The long retentive timer (LST) can be used in interrupt programs. (☞ Page 45 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

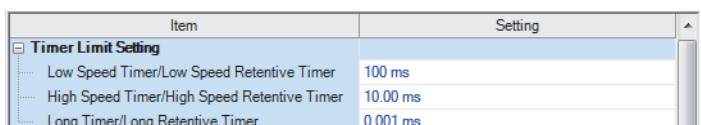
Although the low-speed timer and high-speed timer are the same device, timer limit value are different depending on how to specify the timer device (how to write the instruction). For example, specifying OUT T0 generates a low-speed timer and specifying OUT H T0 produces a high-speed timer even when 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



Item	Setting
Timer Limit Setting	
Low Speed Timer/Low Speed Retentive Timer	100 ms
High Speed Timer/High Speed Retentive Timer	10.00 ms
Long Timer/Long Retentive Timer	0.001 ms

### Displayed items

Item	Description	Setting range	Default
Low Speed Timer/Low Speed Retentive Timer	Set the timer time limit value of T and ST used for the low-speed timer and low-speed retentive timer.	1 to 1000ms (unit: 1ms)	100ms
High Speed Timer/High Speed Retentive Timer	Set the timer time limit value of T and ST used for the high-speed timer and high-speed retentive timer.	0.01 to 100ms (unit: 0.01ms)	10.00ms
Long timer/Long retentive timer	Set the timer time limit value of LT and LST used for the long timer and long retentive timer.	0.001 to 1000ms (unit: 0.001ms)	0.001ms

## Timer current value and the measurable range

Indicates the timer current value and the measurable range.

### ■Timer (T/ST)

The setting range of the current value is 0 to 65535, which is the same as the range of unsigned 16-bit integers. The measurable time range is from 0 to (timer time limit value × 65535).

### ■Long timer (LT/LST)

The setting range of the current value is 0 to 4294967295, which is the same as the range of unsigned 32-bit integers. The measurable time range is from long timer time limit value to (timer time limit value × 4294967295).

## Handling timers

When executing the timer coil (the OUT T□ instruction), the timer coil is turned on/off, the current value is updated, and the contact is turned on/off.

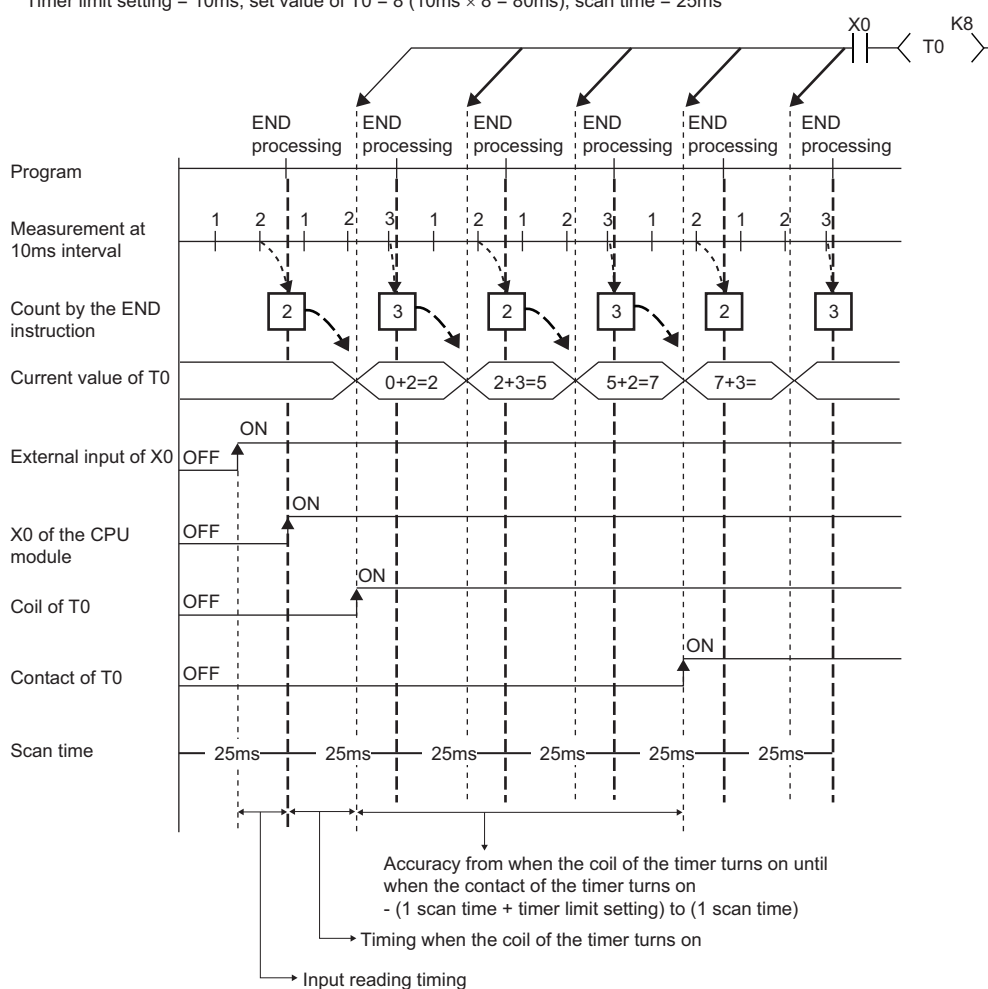
## Accuracy of timers

Indicates the accuracy of timers.

### ■Timer (T/ST)

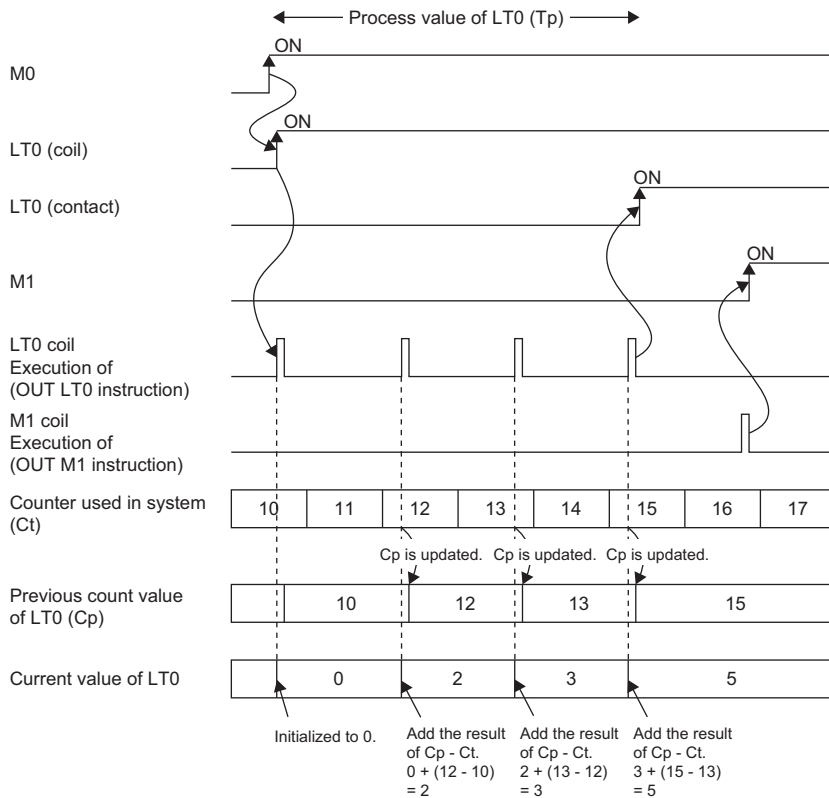
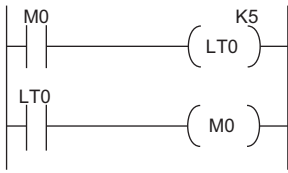
The scan time value measured by the END instruction is added to the current value when the OUT T□ instruction is executed. If the timer coil is turned off when the OUT T□ instruction is executed, the current value is not updated. The maximum response accuracy of the timer (the time duration from capture of an input (X) to output of it) is "2 scan time + timer time limit setting."

Timer limit setting = 10ms, set value of T0 = 8 (10ms × 8 = 80ms), scan time = 25ms



## ■ Long timer (LT/LST)

In the following program, the accuracy of  $T_p$  (the time duration from the long timer coil activation to long timer contact activation) is  $(T_s - T_u) \leq T_p < (T_s + T_u)$ .



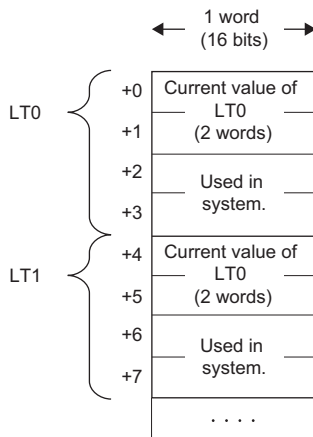
\*1  $T_p$ : time duration from the long timer coil activation to long timer contact activation

\*2  $T_s$ : setting value of the long timer

\*3  $T_u$ : 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.



29

### Point

The current value of the long timer (LT) and the long retentive timer (LST) is 32-bit data. It can be specified by the instructions which can specify signed or unsigned 32 bit data. (It cannot be specified with the BK + Instruction.)

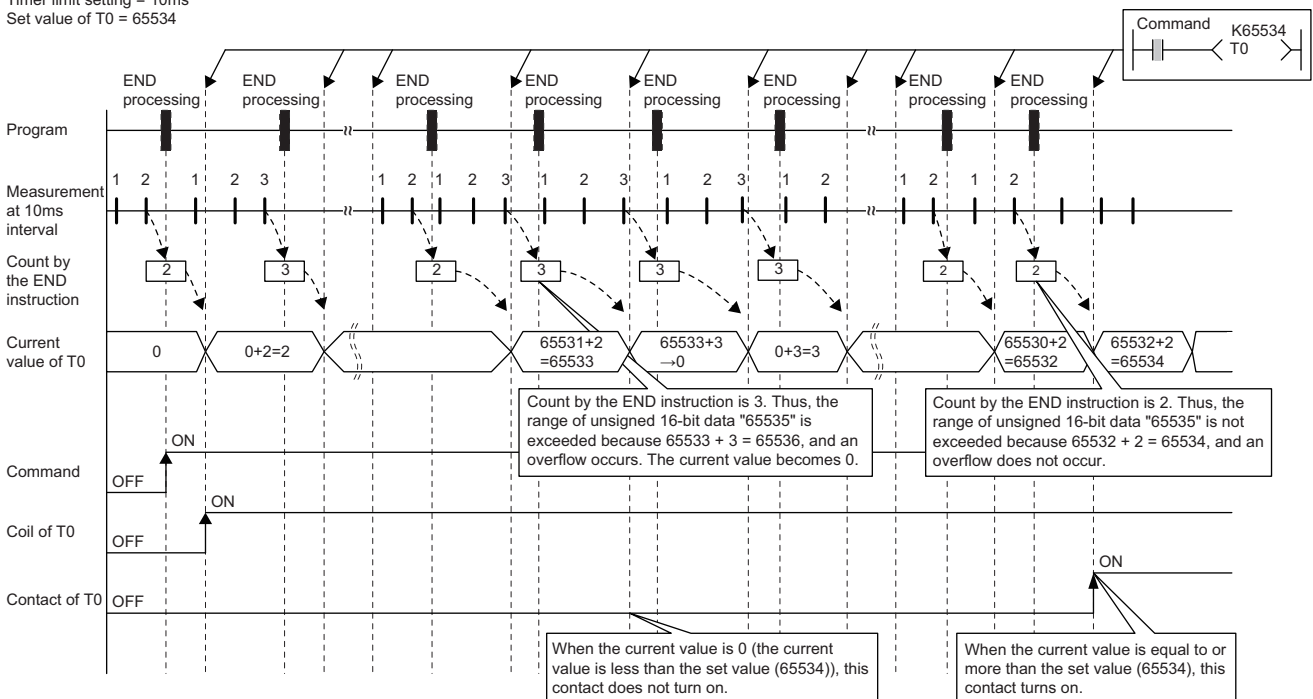
## Precautions

This section describes the precautions when using the timer and long timer.

### ■Precautions about timer usage

- Don't describe more than one coil (the OUT T□ instruction) on the same timer during a single scanning. Doing so results in improper measurement because the timer current value is updated when the coil for each timer is executed.
- When timer is not used for data collection for each scan: While the coil on a timer (e.g. T1) is turned on, the timer coil (the OUT T□ instruction) cannot be skipped by the instructions such as the C.J. When the timer coil is skipped, proper measurement is impossible because the timer current value is not updated. In addition, when the timer exists in a subroutine program, be sure to execute a subroutine call including T1 coil only once for each scanning operation while the coil of the timer (e.g. T1) is turned on. Otherwise proper measurement is impossible.
- The timer cannot be used in the initial execution type program, the fixed scan execution type program, or the event execution type program where the occurrence of an interrupt is set to be a trigger. The timer can be used in standby type programs if the coil of timer (OUT T□ instruction) is executed one time for one scan using a subroutine program.
- The timer cannot be used in interrupt programs. The timer can be used in subroutine programs or FB programs if the coil of timer (OUT T□ instruction) is executed one time for one scan.
- When setting value is 0: The contact is turned on when the OUT T□ instruction is executed.
- Even when the setting value is increased after the timer time is up, the timer status does not change (time is continues to be up) and the timer does not operate.
- When a setting value close to the upper limit value (65535) is used, the timing at which the contact turns on differs depending on the count measured by the END instruction. When using a setting value close to the upper limit (65535), using the long timer is advisable.

Timer limit setting = 10ms  
Set value of T0 = 65534

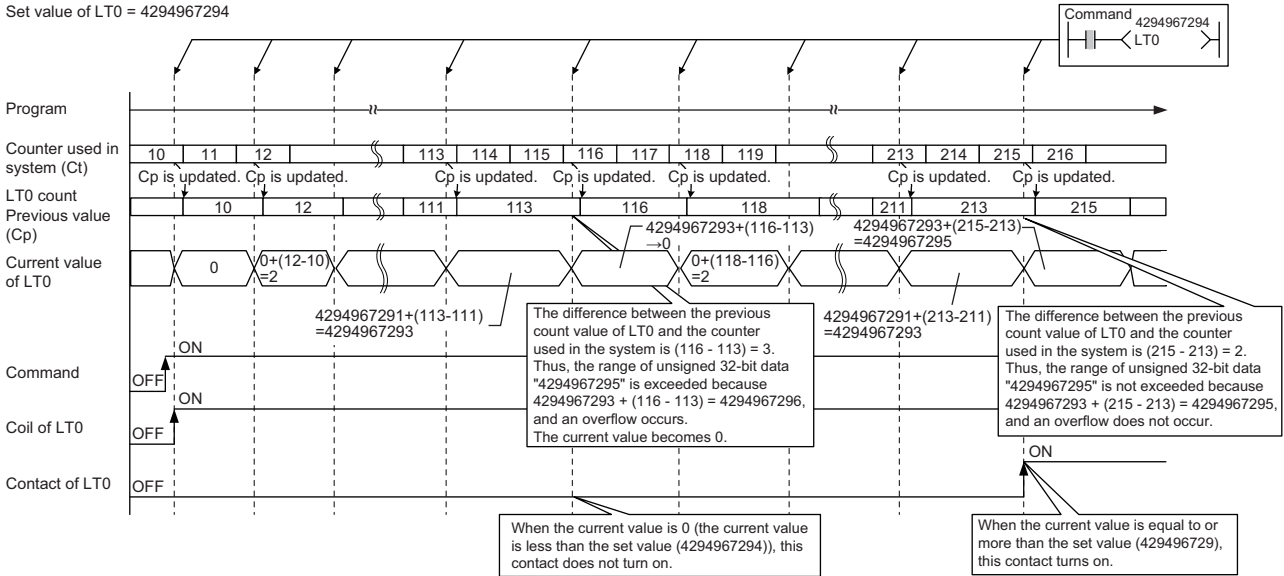


## ■Precautions about long timer usage

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.
- When a setting value close to the upper limit value (4294967295), the timing at which the contact turns on differs depending on the difference from the counter that is used in the system and counts up at the execution of the OUT LT□ instruction.

Set value of LT0 = 4294967294



# 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 45 Interrupt program)

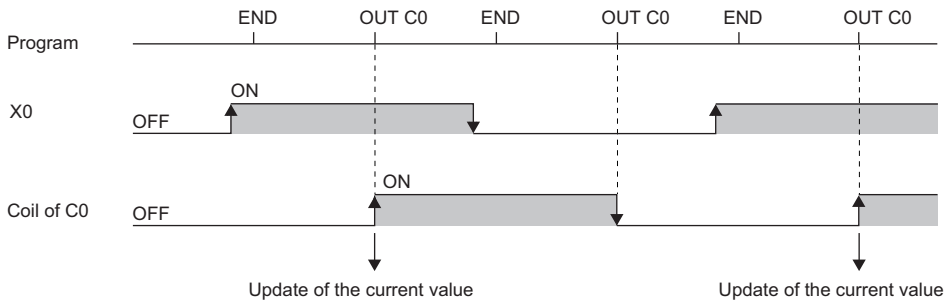
## Counting process

When executing the counter coil (OUT C□instruction / OUT LC□instruction), the counter coil is turned on/off, the current value is updated (count value +1), and the contact is turned on/off. The current value is updated (count value +1), when the counter coil input is rising (off and on). The current value is not updated when the coil input is turned off, on and on, and on and off.

[Ladder example]



[Update timing of the current value]





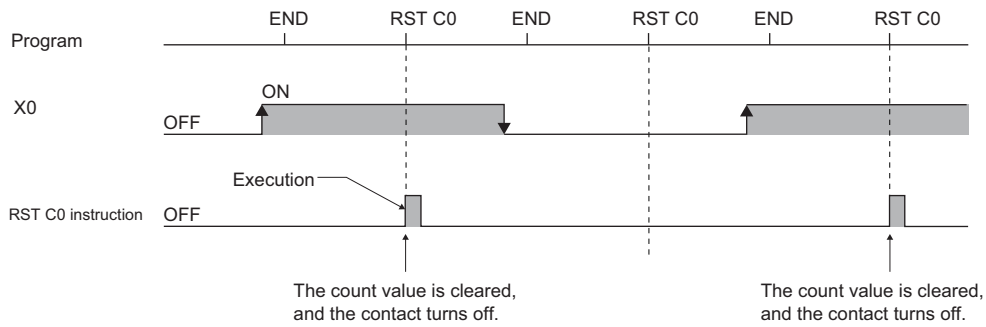
## Resetting counters

The counter current value is not cleared even when the counter coil input is turned off. To clear the counter current value (resetting) and turn off the contact, issue the RST C□/RST LC□ instruction. When executing the RST C□ instruction, the counter value is cleared and the contact is turned off.

[Ladder example]

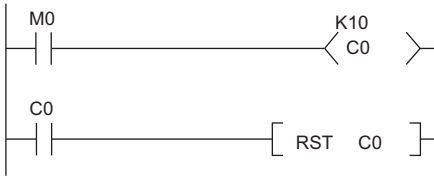


[Reset timing of the counter]

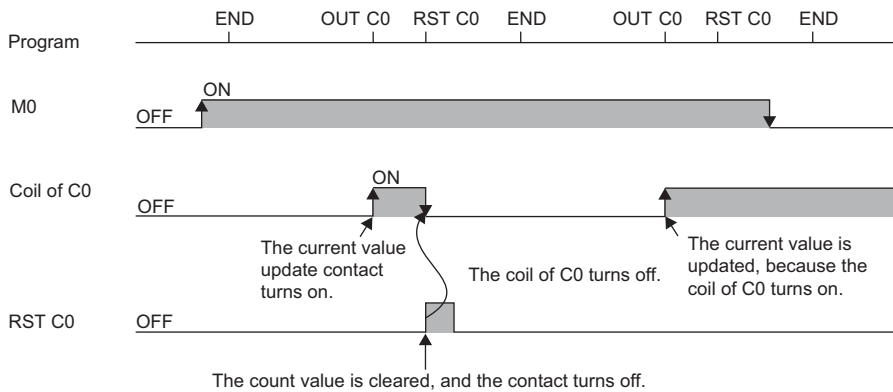


## ■Precautions about counter reset

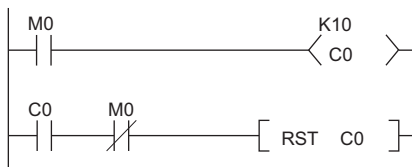
When executing the RST C□ instruction, the coil for C□ is also turned off. If the execution condition for the OUT C□ instruction is turned on after the RST C□ instruction is executed, the coil of C□ is turned on and the current value is updated (count value +1) when the OUT C□ instruction is executed.



In the above ladder example, the coil of C0 is turned on by turning off and on M0 and as a result the current value is updated. When C0 count reaches its upper limit, C0 contact is turned on and C0 current value is cleared by execution of the RST C0 instruction. At this time C0 coil is also turned off. When M0 is turned on at the next scanning, the current value is updated because C0 coil is turned off and on during the OUT C0 instruction execution (the current value is changed to 1).



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.

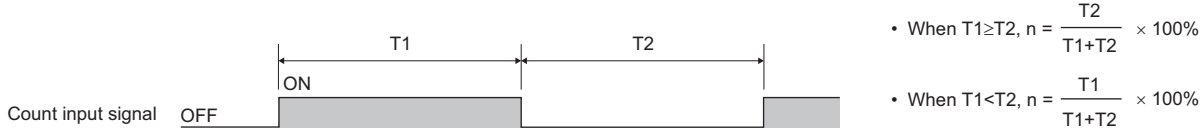


## Maximum counting speed for counters

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:

$$\text{Maximum counting speed } C_{\text{max}} = \frac{n}{100} \times \frac{1}{T} \quad [\text{times/s}]$$

\*1 Duty (n) is a value which expresses the ratio of on/off time of the count input signal as a percent (%) value.



## Data register (D)

This device can store numerical values.

## Link register (W)

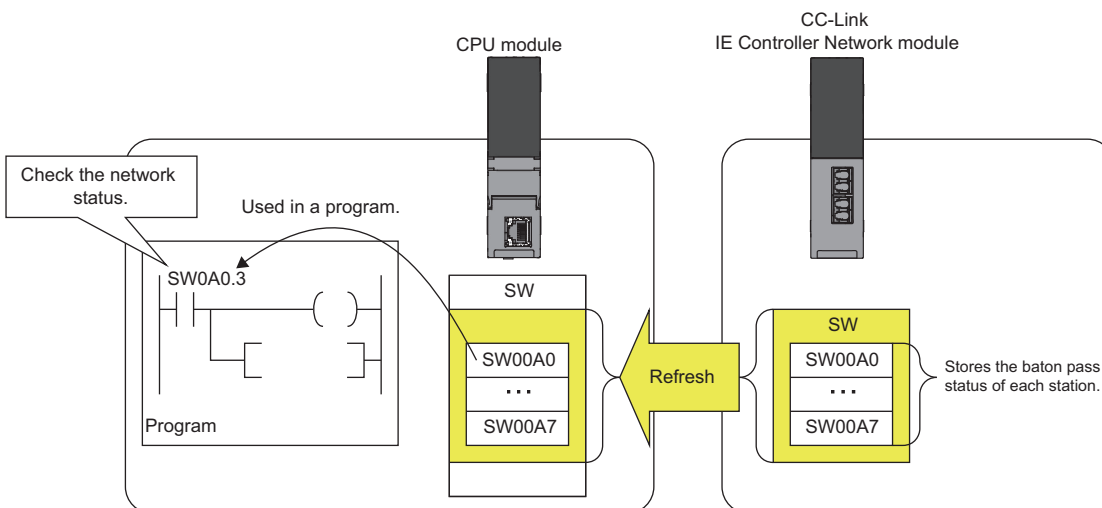
This device is used as a CPU module side device when refreshing word data between the network module, such as the CC-Link IE Controller Network module and the CPU module.

## 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 for communication status and/or error detection status for network, such as the 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.



## 29.3 System Device

The system device is used by the system. Assignment/capacity is fixed and cannot optionally be altered.

### Function device (FX/FY/FD)

This device is used for the subroutine programs with argument passing. Data is written/read between the subroutine call sources with argument passing and the subroutine programs with argument passing. When using the function device in a subroutine program, the device used in each subroutine program call source can be determined. As a result, when using the same subroutine program, it can be used without being aware of the call source of other subroutine programs.

#### Function input (FX)

This device is used when passing on/off data to a subroutine program. In the subroutine program, bit data specified in a subroutine call instruction with argument passing are captured and used for operation. All the bit data specification devices of the CPU module are available.

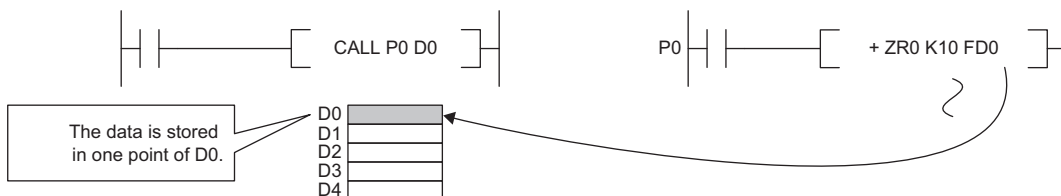
#### Function output (FY)

This device is used when passing operation results (on/off data) in a subroutine program to a subroutine program call source. Operation results are stored into the device specified in the subroutine program with argument passing. The bit data specification device other than the CPU module input (X) are available.

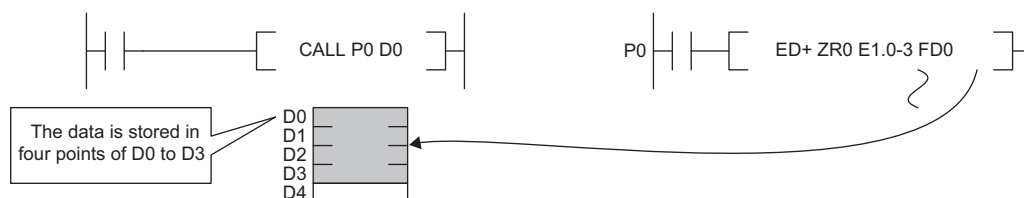
#### Function register (FD)

This device is used for writing/reading between the subroutine call source and the subroutine program. The input/output condition of the function register is automatically identified by the CPU module. "Source data" in a subroutine program means data inputted into the subroutine program. "Destination data" in a subroutine program means data outputted data from the subroutine program. One point of a function register occupies maximum of four words and can store 16-bit data, 32-bit data, 64-bit data, single-precision real number, and double-precision real number. However the number of words to be used depends on the instruction in the subroutine program.

For example, for the destination of addition instruction (+instruction) of 16-bit signed integer, one word is used.



Besides, for the destination of addition instruction (ED+instruction) of double-precision real number, four words are used.



### 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 319 List of Special Relay Areas)

### Special register (SD)

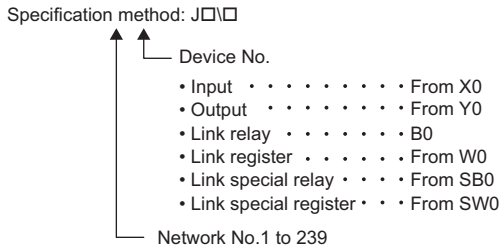
This is the internal register for which the specification is defined in the CPU module, where the status (diagnostics information, system information, etc) of the CPU module is stored. (☞ Page 335 List of Special Register Areas)

# 29.4 Link Direct Device

This device directly access link relays and/or link registers of the network module in the CC-Link IE Controller Network and/or CC-Link IE Field Network.

## Specification method

Specify as shown below: The link register 10 (W10) with the network number 2 can be specified as "J2W10."



**Point**

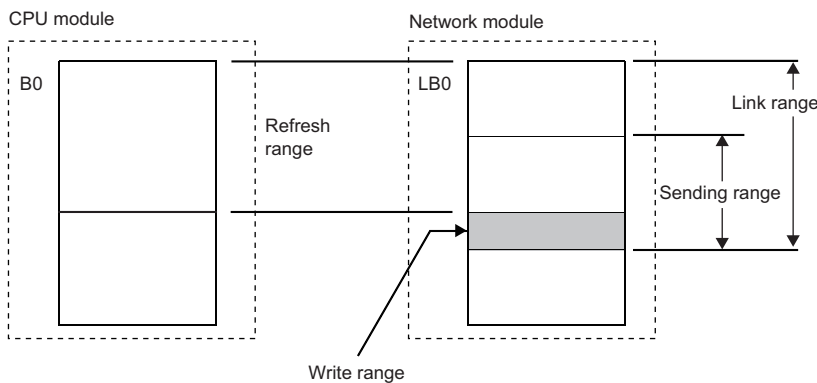
For bit devices, digit specification is allowed. (e.g. J1\K1X0, J10\K4B0)

## Specification range

All the link devices of the network module can be specified. The link devices which fall outside the range specified with the network refresh parameter can also be specified.

### Specification range for writing

Writing should be done for the range which is within the link device range specified as a sending range of network parameters and outside the range specified as a refresh range for the network refresh parameter. Writing for a range specified as the refresh range overwrites link device data of the link module during refresh process. When writing data to a writing range of another station using the link direct device, data is overwritten with received data during data reception from another station.



### Specification range for reading

Data can be read from all the range of link devices of the network module. There is only one network module for which writing/reading is allowed with the link direct device for each network number. When more than one network module with the same network number is mounted, the network module with the lowest slot number is the target for writing/reading by the link direct device.

## Difference from link refresh

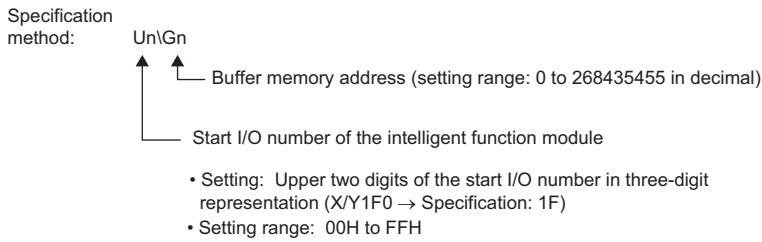
The following table shows the difference between the link direct device and link refresh.

Item		Link direct device	Link refresh
Description method in program	Input	Jn\K4X0...	X0...
	Output	Jn\K4Y0...	Y0...
	Link relay	Jn\K4B0...	B0...
	Link register	Jn\W0...	W0...
	Link special relay	Jn\K4SB0...	SB0...
	Link special register	Jn\SW0...	SW0...
Access range in relation to network module		All the link devices for each network module	The range specified in the refresh parameter
Guarantee range of access data		In word (16-bit) units	In word (16-bit) units

# 29.5 Module Access Device

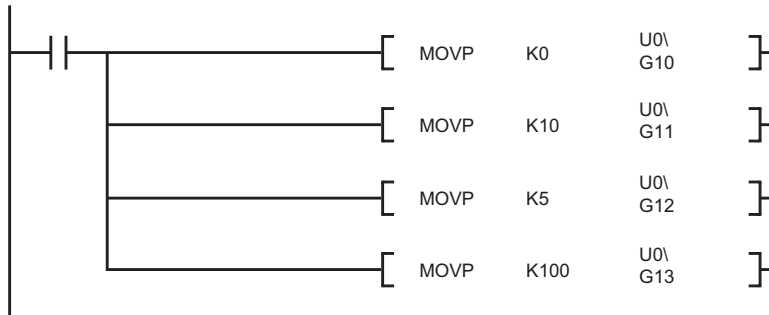
This device directly accesses from the CPU module to the buffer memory of the intelligent function module mounted on the main base unit and extension base unit.

Specify as shown below: (e.g. U5\G11)

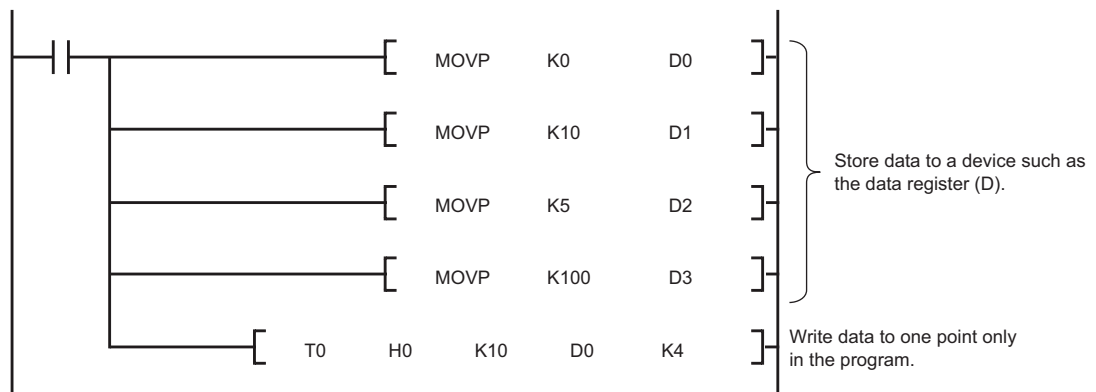


When reading/writing the buffer memory data using the module access device more than twice within a program, the processing speed can be increased by conducting read/write operations at a single point of the program using the FROM/TO instruction.

- When data is written using more than one module access device:



- When data is written at a single point of the program using the TO instruction:




## 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.

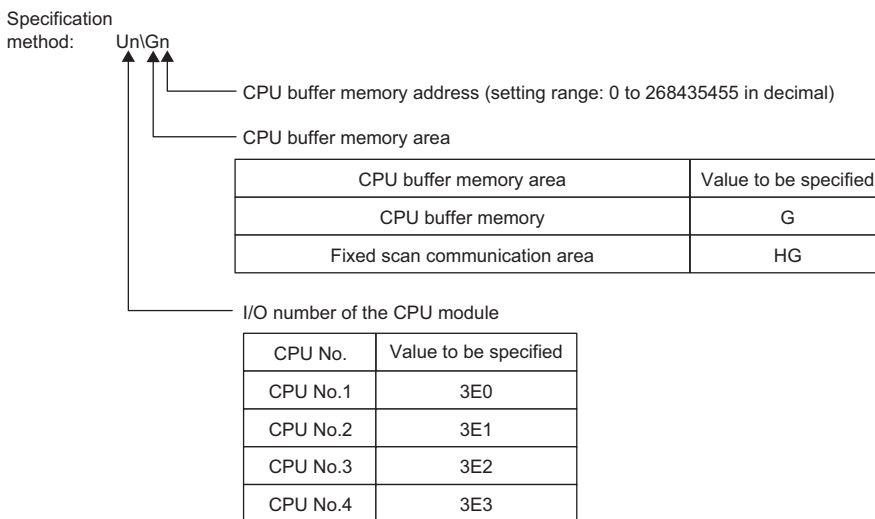
\*1 When the data is transferred from the CPU module to the module, the target memory is the buffer memory or link device. When the data is transferred from the module to the CPU module, the target memory is the specified device of the CPU module.

## 29.6 CPU Buffer Memory Access Device

This device accesses memory used by the built-in function of the CPU module, such as data writing/reading between CPU modules on the multiple CPU system and Ethernet function (  Page 212 Specification method through CPU buffer memory access device)

### Specification method

Specify as shown below: (e.g. U3E1\G4095, U3E2\HG1024).





# 29.7 Index Register (Z/LZ)

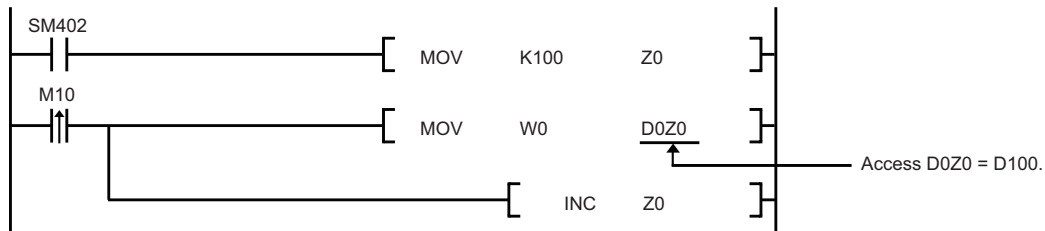
This device is used for the index modification of the device. The index modification is the indirect specification using the index register.

## Types of Index registers

There are two types of index registers: index register (Z) and long index register (LZ).

### Index register (Z)

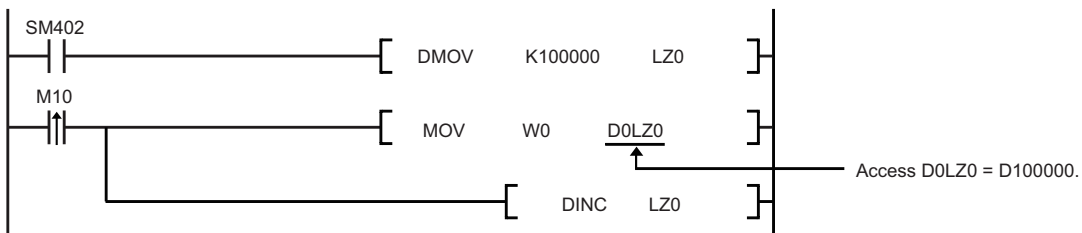
Used for 16-bit index modification



In addition, 32-bit index modification with ZZ expression using two index registers is also available.

### Long index register (LZ)

Used for 32-bit index modification

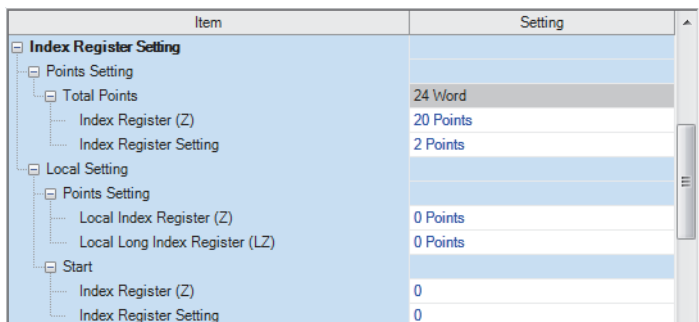


# Index register setting

The following window allows to specify the number of points for the index register (Z) and long index register (LZ) and the range where they are used as a local device. The total number of points of the index register (Z) and the long index register (LZ) must be set to 24 words.

 [CPU Parameter] ⇨ "Memory/Device Setting" ⇨ "Index Register Setting"

## Window



## Displayed items

Item		Description	Setting range	Default	
Number of points setting	Total Points	Show the total number of points for index register and long index register.	—	—	
	Index register (Z)	Set the number of points for the index registers.	0 to 24 points (in two-point increments)	20 points	
	Long index register (LZ)	Set the number of points for the long index registers.	0 to 12 points (in one-point increments)	2 points	
Local setting	Number of points setting	Local Index register (Z)	Set the number of points for index registers used as a local device. Set within the range of the index register.	0 to 24 points (in one-point increments)	0 points
		Local long index register (LZ)	Set the number of points for long index registers used as a local device. Set within the range of the long index register.	0 to 12 points (in one-point increments)	0 points
	Head	Index register (Z)	Set the head for the local index registers. Set within the range of the index register.	0 to 23	0
		Long index register (LZ)	Set the head for the local long index registers. Set within the range of the long index register.	0 to 11	0

# 29.8 File Register (R/ZR)

This device is a word device for extension. This device is specifically a file register file which exists in the file storage area on the device/label memory.

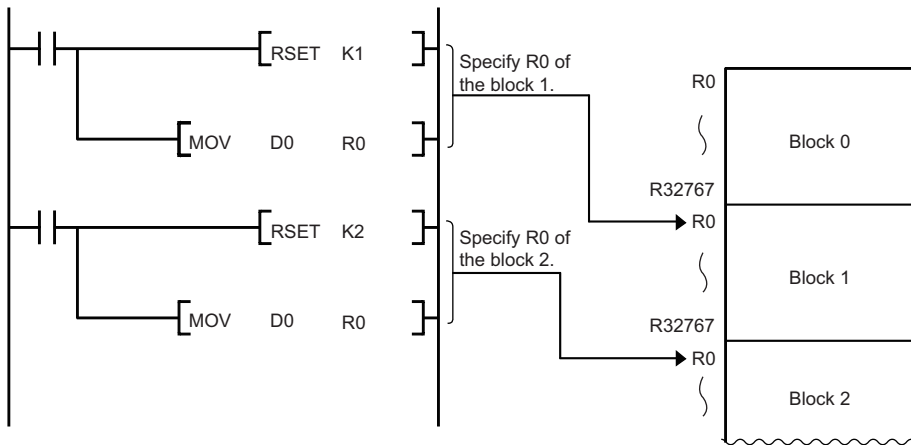
## Specification method

There are two types of the specification methods for the file register: block switching and serial number methods.

### Block switching method

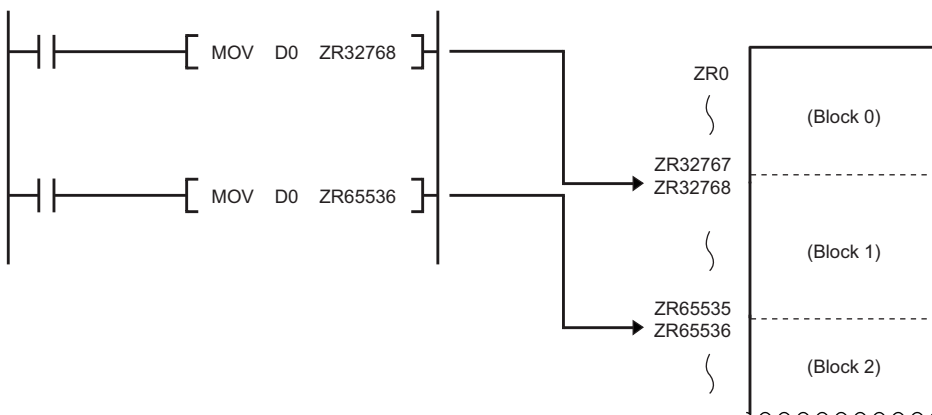
In this method the number of points of file register being used is specified by being divided in increments of 32K point (R0 to R32767). When using more than one block, specification is conducted by switching to the block number used by the RSET instruction. "R" is used as the device symbol. The range of "R" is from R0 to R32767. However in the following cases the upper limit of the device number is "block size (unit: word) -1."

- The file register file size is smaller than 64K bytes.
- The file register file size is not a multiple of 64K bytes and the end block is specified in the RSET instruction.



### Serial number method

In this method file registers having more than 32K points are specified using serial device numbers. File registers of the multiple blocks can be used as consecutive file registers. "ZR" is used as the device symbol. The range of ZR is from ZR0 to (file register file size (unit: word) -1).



## Setting file registers

This section describes the settings required to use the file registers.

### Configuration procedure

This section describes the procedure to use the file registers.

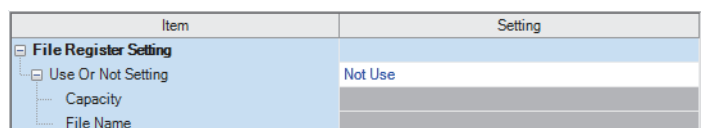
1. Set the file register usage with [CPU Parameter].
  2. To use the file register for each program, previously create the device memory which will become the file register file. (📖 GX Works3 Operating Manual)
  3. When using the file registers, which are common for all programs, the file registers of the file name and capacity, which have been set through setting of the file register, are created. \*1
- \*1 If the capacity is not set, it must be set when creating a file register file and writing it to the programmable controller in the same manner as the procedure 2.
4. Write parameters and file register files into the CPU module.

### File register setting

This setting must be completed before using the file registers.

🔗 [CPU Parameter] ⇨ "File Setting" ⇨ "File Register Setting"

#### Window



Item	Setting
File Register Setting	
Use Or Not Setting	Not Use
Capacity	
File Name	

#### Displayed items

Item	Description	Setting range	Default
Use Or Not Setting	Specify whether or not file registers should be used.	<ul style="list-style-type: none"><li>• Not Use</li><li>• Use File Register of Each Program</li><li>• Use Common File Register in All Programs</li></ul>	Not Use
Capacity	Specify the capacity of the file register in units of 1K words when "Use Common File Register" in All Programs is selected.	This value depends on whether or not the extended SRAM cassette is mounted and its capacity.	—
File Name	Assign a file name to the file register when "Use Common File Register in All Programs" is selected.	1 to 60 characters	—

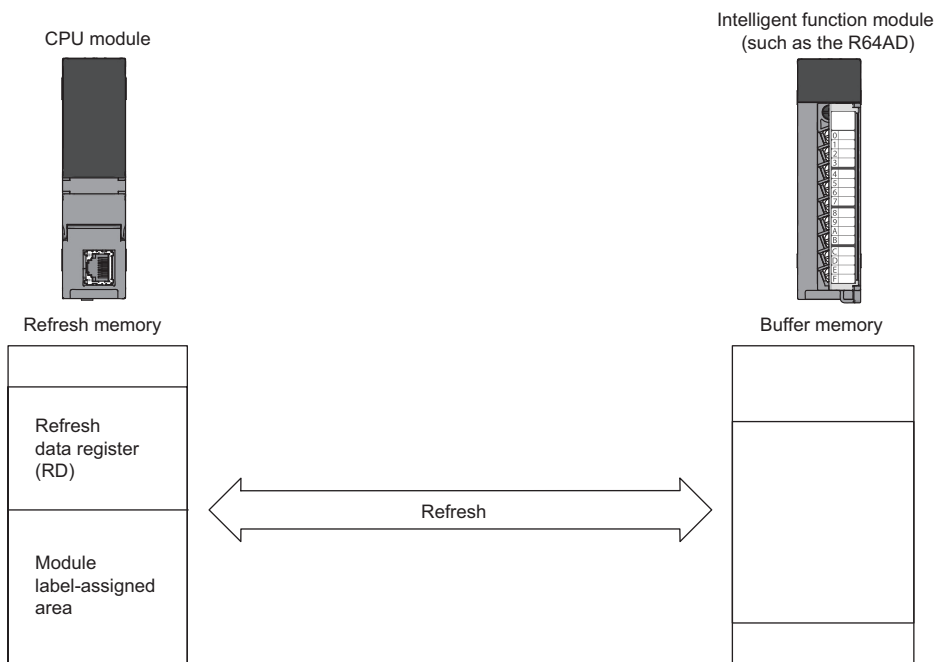
## Clearing file registers

To clear the file registers, use the following methods (🔗 Page 65 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)

# 29.9 Refresh Data Register (RD)

This device is provided for using as a refreshing target of buffer memory on the various devices, such as an intelligent function module. Refresh Data Register (RD) is assigned into the refresh memory area. ( Page 64 Refresh memory)



## Refresh memory setting

[CPU Parameter] ⇒ "Memory/Device Setting" ⇒ "Refresh Memory Setting"

### Window

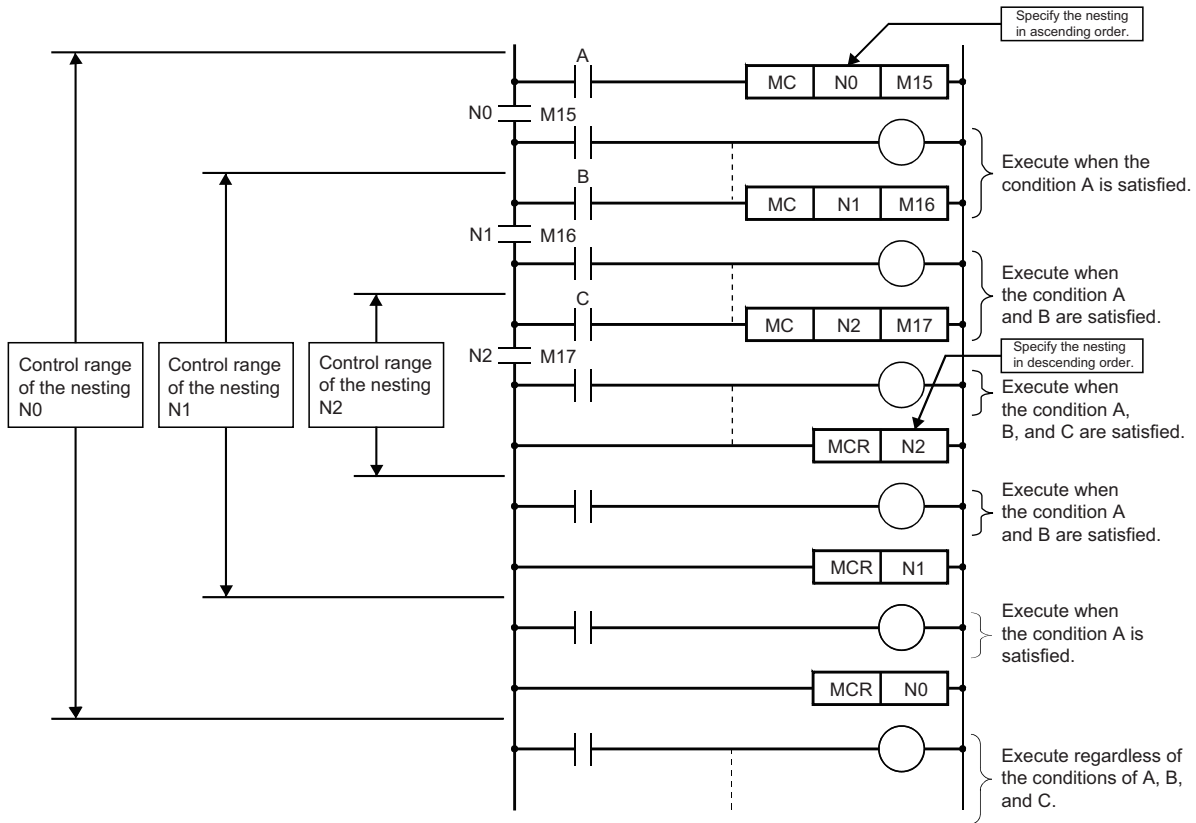
Item	Setting
Refresh Memory Setting	
Total Points	1024 K Point
Refresh Data Register (RD) Area	512 K Point
Module Label Assignment Area	512 K Point

### Displayed items

Item	Description	Setting range	Default
Total Points	Show the total number of points for the refresh data register and the assigned area.	—	1024K Point
Refresh Data Register (RD) area	Set the number of points of refresh data register.	0 to 1024K points (in one-point increments)	512K Point
Unit label assigned area	Set the number of points for the module label assigned area.	0 to 1024K points (in one-point increments)	512K Point

# 29.10 Nesting (N)

This device is used in the master control instructions (the MC/MCR instruction)\*1 and enables the programming of operation conditions in a nesting structure. Specify this device from outside the nesting structure starting with the lowest number (in ascending order from N0 to N14).



\*1 This instruction creates an effective ladder-switching program using opening/closing the common rail of the ladders.

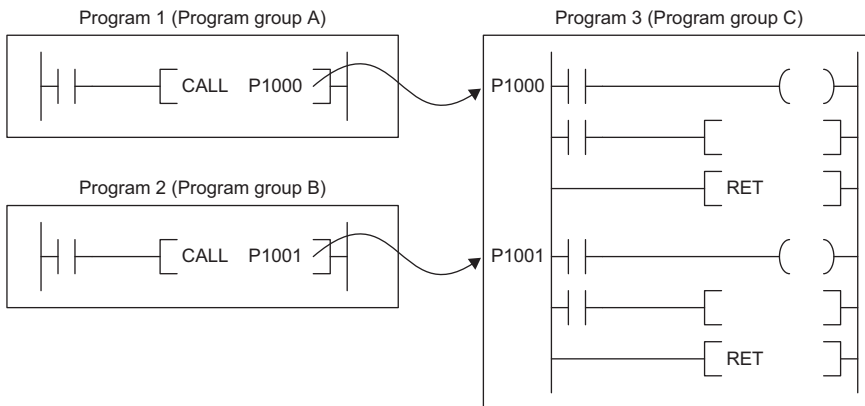
# 29.11 Pointer (P)

This device is used in the jump instructions (the CJ/SCJ/JMP instruction) and/or subroutine program call instructions (e.g. the CALL instruction). The type of pointer type includes the Global pointer and Local pointer. This pointer is used for the following objectives:

- Specifies the jumping target specification and label of the jump instructions (the CJ/SCJ/JMP instruction).
- Specifies the call target and label (the head of subroutine program) of the subroutine call instructions (e.g. the CALL instruction).

## Global pointer

This is the pointer which enables calling by the subroutine call instruction from all the program being executed.



### Effective use of number of points

The number of points for global pointers can be obtained with the following formula: "pointer number of points specified by the parameter" - "the total point number of local pointer being used for each program."

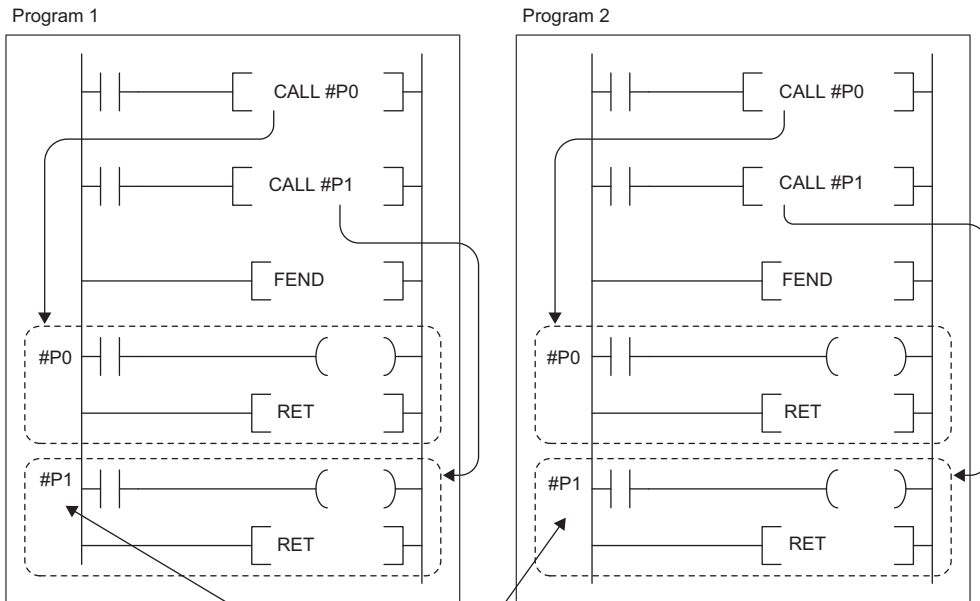
### Precautions

This section describes the precautions when the global pointer is used.

- A global pointer with the same pointer number cannot be set as a label in multiple points.

# Local pointer

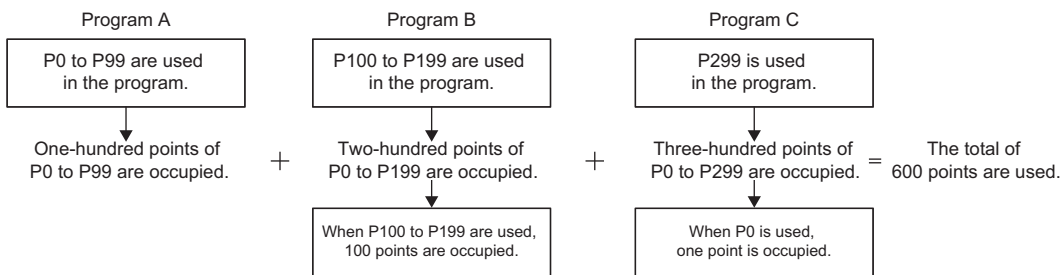
This is the pointer to be independently used in each program where the same pointer number can be used. This pointer is specified in the following format: # (pointer number) (e.g. #P0) (Page 265 Specification method for the local devices).



The local pointers that have 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

Item	Setting
Pointer Setting	
Global Pointer Start	0
Total Points	16384 Points
Global Pointer	4096 Points
Local Pointer	4096 Points
Pointer Type Label	8192 Points

## Displayed items

Item	Description	Setting range	Default
Global pointer head	Set the head of the global pointer.	P0 and over <sup>*1</sup>	0
Total Points	Show the total number of points for the pointer.	—	16384 points <sup>*2</sup>
Global pointer	Set the number of points of the global pointer.	<ul style="list-style-type: none"> <li>• R04CPU, R08CPU, R08PCPU, R16CPU, R16PCPU, R32CPU, R32PCPU: 0 to 16384 points (in increments of 1 point)</li> <li>• R120CPU, R120PCPU: 0 to 32768 points (in increments of 1 point)</li> </ul>	4096 points <sup>*3</sup>
Local pointer	Set the number of points of the local pointer.		4096 points <sup>*3</sup>
Pointer-type label	Set the number of points for the Pointer-type label assignment area.		8192 points <sup>*4</sup>

\*1 Up to the number of "Total points of pointer device area" - "Number of points of global pointer".

\*2 The default number of points is 32768 for the R120CPU and R120PCPU.

\*3 The default number of points is 8192 for the R120CPU and R120PCPU.

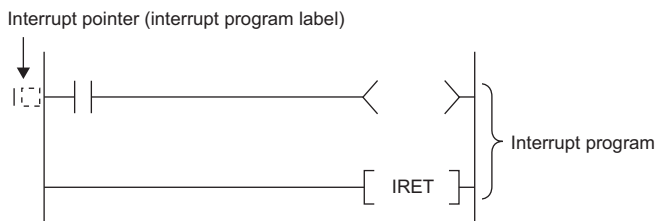
\*4 The default number of points is 16384 for the R120CPU and R120PCPU.

## Point

Specify a pointer number which is equal to or lower than "(end number of the pointer range specified in the parameter) - (number of points of the global pointers)."

# 29.12 Interrupt Pointer (I)

This device is used as a label located at the head of the interrupt program. This pointer can be used in all the programs being executed.



**Point** 🔍

Setting the execution type of program to the event execution type eliminates the need to write (I□) the interrupt pointer. (📖 Page 37 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
Interruption from module	I0 to I15	This is a pointer used for modules which have the interrupt function.
Interruption by the internal timer	I28 to I31	This interrupt pointer is used in fixed scan interrupts by the internal timer.
Inter-module synchronous interrupt	I44	This fixed scan interrupt pointer is used in the inter-module synchronization function.
Multiple CPU synchronous interrupt	I45	This fixed scan interrupt pointer is used in the multiple CPU synchronization function.
High-speed internal timer interrupt 2	I48	This interrupt pointer is used in fixed scan interrupts by the internal timer and can be specified in a shorter interval than interrupt pointer numbers I28 to I31.
High-speed internal timer interrupt 1	I49	
Interruption 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	Interruption from module	5 to 8	9	
I1			1st point	10
I2			2nd point	11
I3			3rd point	12
I4			4th point	13
I5			5th point	14
I6			6th point	15
I7			7th point	16
I8			8th point	17
I9			9th point	18
I10			10th point	19
I11			11th point	20
I12			12th point	21
I13			13th point	22
I14			14th point	23
I15			15th point	24
I28	Interruption by the internal timer	4	8	
I29			7	
I30			6	
I31			5	
I44	Inter-module synchronous interrupt	3	4	
I45	Multiple CPU synchronous interrupt		3	
I48	High-speed internal timer interrupt 2	2	2	
I49	High-speed internal timer interrupt 1	1	1	
I50 to I1023	Interruption 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 84 Interrupt priority)
- The interrupt priority order is the order which is executed when the interrupt factor with the same interrupt priority is generated. (👉 Page 87 Multiple interrupt execution sequence)

## 29.13 Other Devices

---

This section describes other devices.


### Network No. specification device (J)

---

This device is used when specifying a network number with the Link dedicated instruction.

**Point** 

For details, refer to the following.

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

---


### I/O No. specification device (U)

---

This device is used when specifying an I/O number with the intelligent function module dedicated instruction.

**Point** 

For details, refer to the following.

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

---

# 30 CONSTANTS

This section describes the constant.

## 30.1 Decimal Constant (K)

Used when decimal data is specified in the program. Specify this device using K□ character (e.g. K1234). The specification range depends on the argument data type of the instruction using the decimal constant as shown in the following table:

Argument data type of the instruction		Specification range for decimal constant
Data size	Data type name	
16 bit	Word (signed)	K-32768 to K32767
	Word (unsigned)/bit string (16 bit)	K0 to K65535
	16-bit data	K-32768 to K65535
32 bit	Double word (signed)	K-2147483648 to K2147483647
	Double word (unsigned)/bit string (32 bit)	K0 to K4294967295
	32-bit data	K-2147483648 to K4294967295

## 30.2 Hexadecimal Constant (H)

Used when hexadecimal data is specified in the program. Specify this device using H□ character (e.g. H1234). To specify data in BCD, specify each digit of hexadecimal value with numerical characters 0 to 9. The specification range depends on the argument data type of the instruction using the hexadecimal constant. The range is H0 to HFFFF for 16-bit data size, and H0 to HFFFFFFFF for 32-bit data size.

## 30.3 Real Constant (E)

Used when real number is specified in the program. There are two types of real numbers: single-precision real number and double-precision real number. Specify it using E□ character. (e.g. E1.234).

### Setting range for real numbers

The setting ranges are different between the single-precision real number and double-precision real number.

#### Specification range for single-precision real numbers

Indicates the specification range for single-precision real numbers.

$$-2^{128} < \text{device} \leq -2^{-126}, 0, 2^{-126} \leq \text{device} < 2^{128}$$

(E-3.40282347+38 to E-1.17549435-38, 0, E1.17549435-38 to E3.40282347+38)


#### Specification range for double-precision real number

Indicates the specification range for single-precision real numbers.

$$-2^{1024} < \text{device} \leq -2^{-1022}, 0, 2^{-1022} \leq \text{device} < 2^{1024}$$

(E-1.7976931348623157+308 to E-2.2250738585072014-308, 0, E2.2250738585072014-308 to E1.7976931348623157+308)

#### Point

For the number of significant digits of real number data input by the engineering tool, refer to the following.  
 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

### Activity when operation is in progress

Indicates the activity when operation is in progress.

#### When overflow and/or underflow occurs:

If an overflow and/or underflow occurs when the operation is in progress, the following process is performed:

- Activity when overflow occurs: an error is returned.
- Activity when underflow occurs: 0, rather than an error, is returned.

#### When a special value is input:

When a special value<sup>\*1</sup> 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  $\pm\infty$ .

### Expression on program

The following expressions are used:

- Normal expression: specifies the numerical value to be set with no modification (e.g.: 10.2345 is expressed as "E10.2345").
- Exponent expression: specifies the numerical value to be set as (numerical value)  $\times 10^n$  (e.g. 1234 is expressed as E1.234+3<sup>\*1</sup>).

\*1 +3 means 103.

## 30.4 Character String Constant

The character string can be specified by enclosing it with double-quotation marks (" "). (e.g.: "ABCDE") Note that the NULL character (00H) becomes the termination character.


### Rules for special character

The special characters shown below cannot be used after they are inputted into the character string as they are. They become available by adding the Escape Sequence "\$". (e.g.: ABC\$'D\$\$E)

Special character	Character string to be entered when editing the circuit diagram
\$	\$\$
'	'\$'
"	"\$"

#### Point

For details on character string data, refer to the following.

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

# 31 HOW TO USE DEVICES CONVENIENTLY

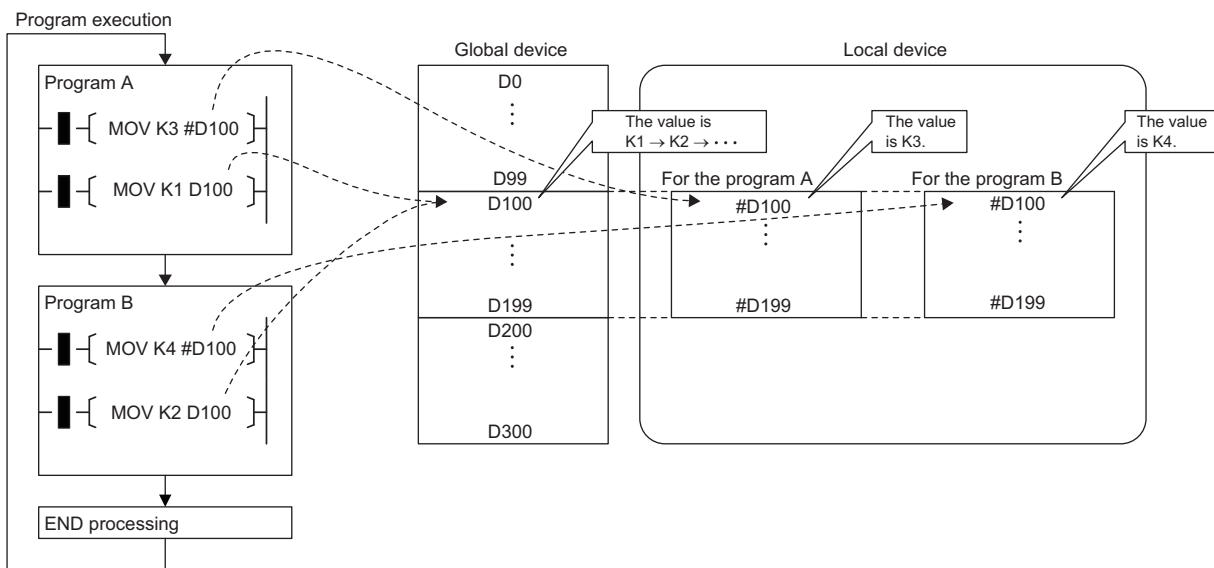
The device includes the global device and local device. As the convenient usage, there are index modification and indirect specification.

## 31.1 Global Device

This device can be shared by all the programs. All the devices that do not set as local device are handled as global device.

## 31.2 Local Device

This device can be used independently in each program. When creating multiple programs, programming can be completed without being aware of devices used in other programs.



### Point

The local device area must be reserved independently from the global device area. Therefore a local device and global device with the same device number can exist.

## Devices available as local device

This section lists devices available as local device.

- Internal relay (M)
- Edge relay (V)
- Timer (T, LT, ST, LST)
- Counter (C, LC)
- Data register (D)
- Pointer (P)

### Point

Because the index register (Z, LZ) saves/returns during program execution, it should be regarded separately as the local index register unlike other local devices. (Page 247 Index Register (Z/LZ))



## Local device area

The CPU module reserves the local device area on the device/label memory based on the number of points setting of the local device when:

- CPU module is powered off and on or reset.
- CPU module status is changed from STOP to RUN.

## When local device is used in subroutine program:

Local devices to be used vary depending on whether SM776 (Local device setting at CALL) is turned on or off. Local index register to be used is also determined according to the SM776 setting.

SM776	Local device to be used
Off	Uses local devices of the program file from which subroutine program is called.
On	Uses local devices of the program file into which the subroutine program is stored.

### Restriction

- In terms of on/off setting for SM776, the value (on or off) used when the subroutine call occurs is considered to be effective. Therefore when on/off setting for SM776 is switched in the subroutine program, the modified value (on or off) is not effective until the next subroutine call occurs.
- On/off setting for SM776 cannot be specified for each program file because it is enabled for each CPU module.

## When local device is used in interrupt and other programs:

Local device used for the interrupt program/fixed scan execution type program/event execution type program which uses occurrence of interruption as a trigger varies depending on whether SM777 (Local device setting in interrupt programs) is turned on or off. For local index register, the register of the program file which has been executed before these programs is used regardless of the SM777 setting.

SM777	Local device to be used
Off	Uses the local device of the program file which has been executed before the interrupt program/fixed scan execution type program/event execution type program which uses occurrence of interruption as a trigger.
On	Uses the local device of the program file of storage location.

### Restriction

- In terms of on/off setting for SM777, the value (on/off) used when executing the event execution type program which uses the interrupt program/fixed scan execution type program/occurrence of interruption as a trigger is considered to be effective. Therefore when on/off setting for SM777 is switched in the program to be executed, the modified value (on or off) is not effective until the event execution type program which uses the next interrupt program/fixed scan execution type program/occurrence of interruption as a trigger is executed.
- On/off setting for SM777 cannot be specified for each program file because it is enabled for each CPU module.

## Clearing local device

Local device can be cleared to 0 by the following operations:

- CPU module is powered off and on or reset.
- CPU module status is changed from STOP to RUN.

# Setting method for the local devices

Set the range where each device will be used as a local device and also set whether or not it should be used.

## Range setting


The range setting for local devices is common to all the programs. Therefore the range for local devices cannot be set for each program.

### Operating procedure

"Device/Label Memory Area Detailed Setting" window

Item	Setting
Device/Label Memory Area Detailed Setting	
Device Setting	<Detailed Setting>
Latch Type Setting of Latch Type Label	Latch (1)

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

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

2. Set the range where each device will be used as a local device.

"Device Setting" window

Item	Symbol	Local Device		
		Start	End	Points
Input	X			
Output	Y			
Internal Relay	M			
Link Relay	B			
Special Link Relay	SB			
Annunciator	F			
Edge Relay	V			
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 Register	SW			
Latch Relay	L			
Total Device				0.0K Word
Total Word Device				0.0K Word
Total Bit Device				0.0K Bit


### Point

Configure the setting range of the local device within the range which has been set for the number of device points. The number of local devices used is calculated by the following calculation formula. Set the number of local devices used so that the number is equal to or less than the capacity of the local device area.

$$\text{Total number of local devices used} = ((A \div 16) + B + (C \times 2) + (D \times 4) + ((E \times 2) \div 16)) \times F$$


- A: Number of points of the local devices M and V
- B: Number of points of the local devices D, T (current value), ST (current value), and C (current value)
- C: Number of points of the local device LC (current value)
- D: Number of points of the local devices LT and LST
- E: Number of points of the local devices T (contact/coil), ST (contact/coil), C (contact/coil), and LC (contact/coil)
- F: Number of programs using the local device

## Increment of setting

The increment of setting of a local device is the same as that of a global device. ( Page 109 Range of use of device points)

## Setting whether or not it should be used

Whether or not local devices should be used can be set for each program. Since the local device area of program for which "Do not use" has been set is not assured, it can suppress unnecessary consumption of device/label memory.

 [CPU Parameter] ⇒ "Program Setting"

### Operating procedure

"Program Setting" window

Item	Setting
Program Setting	<Detailed Setting>

"Detailed Setting" window

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

"Setting of Device/File Use Or Not" window

Execute Order	Program Name	Execution Type	Local Device Local Index Register Use or not	File Use Or Not	
				File Registers	Device Initial Value
1	MAIN	Scan	Use	Use File Setting	Use File Setting
2			Use	Use File Setting	Use File Setting
3			Use	Use File Setting	Use File Setting

1. Click "Detailed Setting" on the "Program Setting" window.
2. Click "Detailed Setting" of "Device/File Use or not".
3. Set whether or not it should be used for each program in the setting of "Local Device Local Index Register Use or not".

### Restriction

Do not use local devices in a program which is configured not to use local devices.

## Specification method for the local devices

To specify the local device in the program, add "#".

**Ex.**

For example, #D100, K4#M0, and @#D0 can be used.

### Point

Local device is listed with a preceding # symbol in the program. This symbol is helpful to differentiate local devices from global devices.

## Precautions

This section describes the precautions when using local devices.

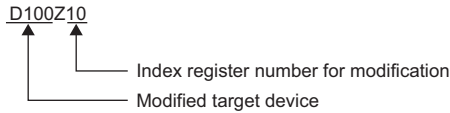
- Like global devices, the timer (T, LT, ST, LST) and counter (C, LC) specified as a local device cannot be checked for the device range. Therefore when operates the address in the index modification or indirect specification, be careful not to exceed the specified device range.
- Accessing the range including both global and local devices by the index modification is not allowed.
- When the range of the 32-bit index modification is across the setting ranges of local devices of the index register, proper index modification is impossible.
- Local devices are not latched.

# 31.3 Index Modification

Specify the device number using the index register. The device number to be used is "Device number of device targeted for modification" + "Contents of index register".

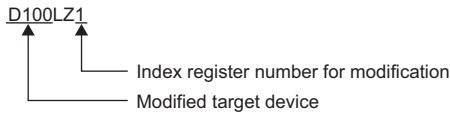
## 16-bit index modification

The device number is modified using the index register (Z). The modification range for the device in the case of the 16-bit index modification is -32768 to 32767.



## 32-bit index modification

The device number is modified using the long index register (LZ). The modification range for the device in the case of the 32-bit index modification is -2147483648 to 2147483647.



## Device for which Index modification can be performed

Indicates the devices that can be targeted for index modification.

Item	Description
16-bit index modification	X, Y, M, L, B, F, SB, V, T <sup>*1</sup> , LT <sup>*1</sup> , ST <sup>*1</sup> , LST <sup>*1</sup> , C <sup>*1</sup> , LC <sup>*1</sup> , D, W, SW, SM, SD, JnX, JnY, JnB, JnSB, JnW, JnSW, UnIG, U3EnIG, U3EnHG, R, ZR, RD, P <sup>*3</sup> , I <sup>*3</sup> , J, U, K, H
32-bit index modification	M, B, SB, T <sup>*1</sup> , LT <sup>*1</sup> , ST <sup>*1</sup> , LST <sup>*1</sup> , C <sup>*1</sup> , LC <sup>*1</sup> , D, W, SW, JnB <sup>*2</sup> , JnW <sup>*2</sup> , UnG <sup>*2</sup> , U3EnG <sup>*2</sup> , U3EnHG <sup>*2</sup> , R, ZR, RD, K, H

\*1 Can be used for the Contact, Coil and Current value.

\*2 For network numbers and the specification source of I/O numbers, 32-bit-based index modification cannot be used.

\*3 When it is used as an interrupt pointer, index modification cannot be applied.

## 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 229 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 229 Edge relay (V))

### Device Range Check for Index Modification

For details on the device range check performed when index modification is used, refer to the following.

☞ MELSEC iQ-R Programming Manual (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:

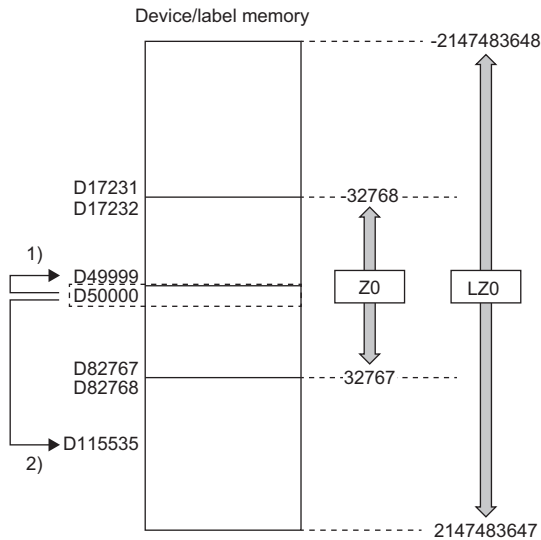
- Review the index modification block(s) within the program.
- For 32-bit-based index modification with ZZ expression, because the specified index register (Zn) and the immediately following index register (Zn+1) are used, caution must be taken to prevent duplicated index registers from being used.
- 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 248 Index register setting)

## When values are stored in the index registers

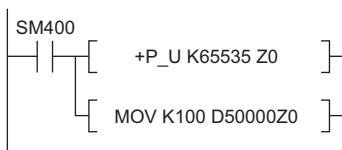
For 16-bit-based index modification using the index register (Z), the range is -32768 to 32767. Therefore, when values within the range from 32768 to 65535 are stored in the index register (Z) for an instruction which processes unsigned data, the instruction doesn't work in design because the range of the index modification will be -32768 to 32767. For the range of values larger than or equal to 32768, the long index register (LZ) must be used so that 32-bit-based index modification can be applied.

**Ex.**

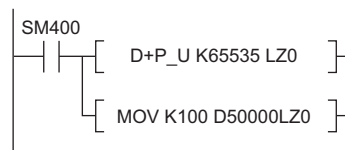
Operation for Index modification



1) When unexpected operation is executed



2) When proper operation is executed

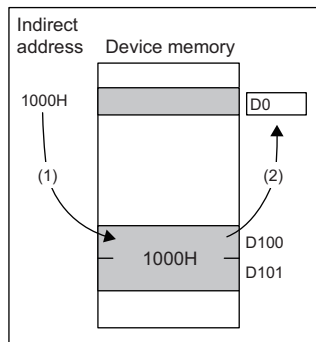
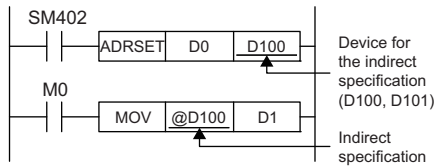


(1) When the value 65535 is stored in the index register (Z), D50000(-1) to D49999 are accessed because the value is turned into -1 when index modification is applied.

(2) When a value larger than or equal to 32768 is used for index modification, the value must be stored in the long index register (LZ). In doing so, the value 65535 is used as such for index modification using the long index register (LZ) and D50000 (65535) to D115535 become accessible.

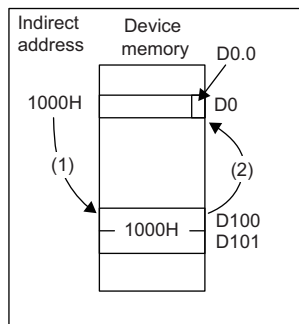
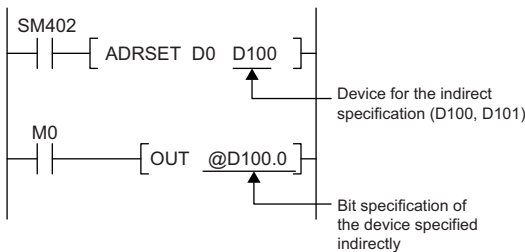
# 31.4 Indirect Specification

Specify the device using the indirect address of device. Store the indirect address of device to be specified into the device for indirect specification, and write as "@ + Device for indirect specification".



- (1) The indirect address of D0 is read into D100, D101.
- (2) The indirect address is used to indirectly specify D0.

Besides, specifying a bit of a word device allows the indirect specification for the instruction that specifies bits.



- (1) The indirect address of D0 is read into D100, D101.
- (2) The indirect address is used to output at the 0th bit of D0 which was indirectly specified using the indirect address.

The indirect specification can be used in the device/label memory or refresh memory.

## Indirect address of device

To specify, use the 32-bit data, and to hold the value, use the device of two words. The indirect address of the device can be obtained with the ADRSET instruction. The ADRSET instruction specifies the indirect address of the device using instructions that handle 32-bit data. For ADRSET instruction, refer to the following.

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

### Point

When the block or the file of the file register is switched through the RSET or QDRSET instruction, the indirect address refers to the one of the block or the file before they are switched. To allow the indirect address in the device for indirect specification to specify the block or file after the file register is switched, specify the ADRSET instruction to obtain the indirect address again after block or file are switched.

## Devices that can allow indirect specification

This section lists devices that can allow indirect specification.

Type	Device
Devices that can allow indirect specification where @ is added*1	T, ST, C, D, W, SW, FD, SD, Un\G, Jn\W, Jn\SW, U3En\G, U3En\HG, R, ZR, RD
Device that can acquire the indirect address through the ADRSET instruction*2	X, Y, M, L, B, F, SB, T, ST, C, D, W, SW, FX, FY, FD, SM, SD, R, ZR, RD

\*1 Also can be used for the local device. (e.g.: @#D0)

\*2 The indirect address of device can be obtained for the local device as well. (e.g.: ADRSET #D0 D100)

# APPENDICES

## Appendix 1 Error Codes

The CPU module stores the corresponding error code in the special register (SD) upon detection of an error using the self-diagnostics 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 engineering tool (📖 MELSEC iQ-R CPU Module User's Manual (Startup))
- Special register (SD0 (latest self-diagnostics error code), SD10 to SD25 (self-diagnostics error code)) (📖 Page 335 List of Special Register Areas)

This section describes errors that may occur in the CPU module and actions to be taken for the errors.


### Error code system

All error codes are given in hexadecimal format (4 digits) (16-bit unsigned integer). The type of error includes the error, which is detected through the self-diagnostics function of each module, and the common error, which is detected during communication between modules. The following table lists the error detection type and the error code ranges.

Error detection type	Range	Description
Detection by the self-diagnostics function of each module	0001H to 3FFFH	Error code specific to each module, such as self-diagnostics errors
Detected during communication between CPU modules	4000H to 4FFFH	Error in the CPU module
	7000H to 7FFFH	Error in the serial communication module
	B000H to BFFFH	Error in the CC-Link module
	C000H to CFFFH	Error in module provided with Ethernet
	D000H to DFFFH	Error in the CC-Link IE Field Network module
	E000H to EFFFH	Error in the CC-Link IE Controller Network module
F000H to FFFFH	Error in the MELSECNET/H network module or MELSECNET/10 network module	



## Detailed information

Upon detection of error through self-diagnostics function, the detailed information of the error cause is stored all together. The detailed information can be checked using the engineering tool. The following detailed information is added to each error code (up to three types of information are stored for each error code. The types differ depending on error code.) Detailed information 1 to 3 of the latest error code(s) can be checked with the special register (SD) as well (  Page 335 List of Special Register Areas)

Detailed information	Item	Description
Detailed information 1	Error location information <sup>*1</sup>	Information on the location in a program
	Drive/File information	Information on drive names and file names
	Parameter information	The information for the parameter, such as parameter storage location and parameter type, is indicated.
	System configuration information	The information for the system configuration, such as I/O No and Power supply No, is indicated.
	Frequency information	This section describes the information for the Frequency such as the write frequency into memory.
	Time information	The information for the time is indicated.
Detailed information 2	Drive/File information	Information on drive names and file names
	Annunciator information	Information about annunciators
	Parameter information	The information for the parameter, such as parameter storage location and parameter type, is indicated.
	System configuration information	The information for the system configuration, such as I/O No and Power supply No, is indicated.
	Process control instruction processing information	Information on processing blocks of process control instructions
Detailed information 3	Failure information	Information on failures

\*1 The step No, which is displayed in the program position information, is the step No that is counted from the head of the file. It might be sometimes different from the step No of the program which is displayed in error jump of engineering tool.

## 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 state will be in 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 179 CPU Module Operation Upon Error Detection Setting)

Besides, if the stop error occurs when the multiple CPU system is configured, the stop error target CPU module (all CPU modules or only the corresponding CPU module) can be set in parameter. (☞ Page 193 Stop setting)

### Continuation error

If a continuation error occurs, the CPU module continues its operation. (The operating state will remain the same.)

## How to clear errors

Continuation errors can be cleared. (☞ Page 182 Error Clear)

## List of error codes

### Self-diagnostics error codes of the CPU module (1000H to 3FFFH)

The following table lists the error codes detected by the self-diagnostics function of the CPU module.

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1000H	Power shutoff	<ul style="list-style-type: none"> <li>A momentary power failure has occurred.</li> <li>The power supply has been shut off.</li> </ul>	<ul style="list-style-type: none"> <li>Check the power supply status.</li> </ul>	—	Always
1080H	ROM write count error	<ul style="list-style-type: none"> <li>The number of writes to the flash ROM (data memory and system memory) exceeded 100000 times. (Number of writes &gt; 100000)</li> </ul>	<ul style="list-style-type: none"> <li>Replace the CPU module.</li> </ul>	Frequency information	At power-on, at RESET, at write
1090H	Battery error	<ul style="list-style-type: none"> <li>The voltage of the battery built in the CPU module has dropped below the specified value.</li> <li>The connector of the battery built in the CPU module is disconnected.</li> <li>The connector of the battery built in the CPU module is not securely connected.</li> </ul>	<ul style="list-style-type: none"> <li>Replace the battery.</li> <li>Connect the battery connector to use the file storage area in the device/label memory or the latch function.</li> <li>Check the connection status of the battery connector. If it is loose, securely connect the connector.</li> </ul>	—	Always
1100H	Memory card access error	<ul style="list-style-type: none"> <li>Data cannot be written to the memory card because the write protect switch is in the locked position.</li> </ul>	<ul style="list-style-type: none"> <li>Set the write protect switch of the memory card in the unlocked position.</li> </ul>	—	Always
1120H	SNTP clock setting error	<ul style="list-style-type: none"> <li>Time setting has failed when the CPU module is powered on or reset.</li> </ul>	<ul style="list-style-type: none"> <li>Check if the time is correctly set in parameter.</li> <li>Check if the specified SNTP server is operating normally and there is no failure on the network accessing the SNTP server computer.</li> </ul>	—	At power-on, at RESET
1124H	Default gateway/gateway IP address error	<ul style="list-style-type: none"> <li>The default gateway is not set correctly.</li> <li>The gateway IP address is not set correctly.</li> <li>The default gateway/gateway IP address (network address after the subnet mask) is different from that of the IP address of the own node.</li> </ul>	<ul style="list-style-type: none"> <li>Correct the default gateway IP address.</li> <li>Set the same network address as that of the IP address.</li> </ul>	Parameter information	Always
1128H	Own node port number error	<ul style="list-style-type: none"> <li>The port number is incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>Correct the port number.</li> </ul>	—	Always
1129H	Open specification port number error	<ul style="list-style-type: none"> <li>The port number setting value of the external device is incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>Correct the port number of the external device.</li> </ul>	—	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
112DH	Specified IP address error	<ul style="list-style-type: none"> <li>The IP address setting value of the external device for the open processing is incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>Correct the IP addresses.</li> <li>Check if the class of the IP address is set to A/B/C.</li> </ul>	—	Always
112EH	Connection establishment failed	<ul style="list-style-type: none"> <li>A connection could not be established in the open processing.</li> </ul>	<ul style="list-style-type: none"> <li>Check the operation of the external device.</li> <li>Check if the open processing has been performed in the external device.</li> <li>Correct the port number of the module, IP address/port number of the external device, and opening method.</li> <li>When the firewall is set in the external device, check if the access is permitted.</li> <li>Check if the Ethernet cable is disconnected.</li> </ul>	—	Always
1133H	Socket communications response send error	<ul style="list-style-type: none"> <li>The response send failed during socket communications.</li> </ul>	<ul style="list-style-type: none"> <li>Check the operation of the external device or switching hub.</li> <li>Since there may be congestion of packets on the line, send data after a certain period of time.</li> <li>Check if the connection cable is disconnected.</li> <li>Check that there is no connection failure with the switching hub.</li> </ul>	—	Always
1134H	TCP connection timeout	<ul style="list-style-type: none"> <li>A TCP ULP timeout error has occurred in the TCP/IP communication. (The external device does not send an ACK response.)</li> </ul>	<ul style="list-style-type: none"> <li>Check the operation of the external device.</li> <li>Correct the TCP ULP timeout value.</li> <li>Since there may be congestion of packets on the line, send data after a certain period of time.</li> <li>Check if the connection cable is disconnected.</li> </ul>	—	Always
1152H	IP address error	<ul style="list-style-type: none"> <li>The IP address is not set correctly.</li> </ul>	<ul style="list-style-type: none"> <li>Correct the IP addresses.</li> </ul>	Parameter information	Always
1155H	Connection number acquisition error	<ul style="list-style-type: none"> <li>The specified connection was already closed in TCP/IP communications.</li> <li>Open processing is not performed.</li> </ul>	<ul style="list-style-type: none"> <li>Perform the open processing for the specified connection.</li> <li>Check if the open processing has been performed in the external device.</li> </ul>	—	Always
1157H	Receive buffer securement error	<ul style="list-style-type: none"> <li>The specified connection was already closed in UDP/IP communications.</li> <li>Open processing is not performed.</li> </ul>	<ul style="list-style-type: none"> <li>Perform the open processing for the specified connection.</li> <li>Check if the open processing has been performed in the external device.</li> </ul>	—	Always
1165H	UDP/IP send failed	<ul style="list-style-type: none"> <li>Data was not sent correctly with UDP/IP.</li> </ul>	<ul style="list-style-type: none"> <li>Check the settings for connection with the external device.</li> <li>Check the operation of the external device or switching hub.</li> <li>Since there may be congestion of packets on the line, send data after a certain period of time.</li> <li>Check if the connection cable is disconnected.</li> <li>Check that there is no connection failure with the switching hub.</li> <li>Execute the PING test, and if the test was completed with an error, take the corrective action.</li> </ul>	—	Always
1166H	TCP/IP send failed	<ul style="list-style-type: none"> <li>Data was not sent correctly with TCP/IP.</li> </ul>	<ul style="list-style-type: none"> <li>Check the settings for connection with the external device.</li> <li>Check the operation of the external device or switching hub.</li> <li>Since there may be congestion of packets on the line, send data after a certain period of time.</li> <li>Check if the connection cable is disconnected.</li> <li>Check that there is no connection failure with the switching hub.</li> <li>Execute the PING test, and if the test was completed with an error, take the corrective action.</li> </ul>	—	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1167H	Unsend data send error	<ul style="list-style-type: none"> <li>Unsent data found, but could not be sent.</li> </ul>	<ul style="list-style-type: none"> <li>Check the settings for connection with the external device.</li> <li>Check the operation of the external device or switching hub.</li> <li>Since there may be congestion of packets on the line, send data after a certain period of time.</li> <li>Check if the connection cable is disconnected.</li> <li>Check that there is no connection failure with the switching hub.</li> <li>Execute the PING test, and if the test was completed with an error, take the corrective action.</li> </ul>	—	Always
1200H	Module moderate error	<ul style="list-style-type: none"> <li>A moderate error has been notified from the intelligent function module connected.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	System configuration information	Always
1210H	Module moderate error	<ul style="list-style-type: none"> <li>An inter-module synchronous signal error has been notified from the intelligent function module connected.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	System configuration information	Always
1220H	Another CPU module moderate error	<ul style="list-style-type: none"> <li>A moderate error has been notified from another CPU module.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Check the mounting status and reset status of other CPU modules.</li> </ul>	System configuration information	Always
1240H	Inter-module synchronization processing error	<ul style="list-style-type: none"> <li>The execution interval of a synchronous interrupt program has exceeded the set value.</li> <li>The run of an inter-module synchronous interrupt program (I44) has not finished within the inter-module synchronization period.</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (time information) in the module diagnostics to verify its value (time), and take the following actions:               <ol style="list-style-type: none"> <li>Review the processing details on the inter-module synchronous interrupt program so that its processing is to be completed within the period specified in the fixed scan interval setting.</li> <li>Set the period specified in the fixed scan interval setting to an appropriate value.</li> </ol> </li> </ul>	Time information	At interrupt occurrence
1241H	Inter-module synchronization processing error	<ul style="list-style-type: none"> <li>The execution interval of a synchronous interrupt program has exceeded the set value.</li> <li>The inter-module synchronous interrupt program (I44) has detected a period that has not been executed.</li> </ul>	<ul style="list-style-type: none"> <li>Review and correct the interrupt-prohibited sections and the interrupt programs with a high priority of interrupt so that the inter-module synchronous interrupt program can be executed.</li> </ul>	—	At interrupt occurrence
1260H	Multiple CPU synchronization processing error	<ul style="list-style-type: none"> <li>The execution interval of a synchronous interrupt program has exceeded the set value.</li> <li>The run of a multiple CPU synchronous interrupt program (I45) has not finished within the multiple CPU fixed scan communication period.</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (time information) in the module diagnostics to verify its value (time), and take the following actions:               <ol style="list-style-type: none"> <li>Review the processing details on the multiple CPU synchronous interrupt program so that its processing is to be completed within the period specified in the fixed scan interval setting.</li> <li>Set the period specified in the fixed scan interval setting to an appropriate value.</li> </ol> </li> </ul>	Time information	At interrupt occurrence

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
1262H	Multiple CPU synchronization processing error	<ul style="list-style-type: none"> <li>The program run section of a synchronous interrupt program has been exceeded.</li> <li>The multiple CPU synchronous interrupt program (I45) has not been completed within the program run section.</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (time information) in the module diagnostics to verify its value (time), and take the following actions:               <ol style="list-style-type: none"> <li>Review the processing details on the multiple CPU synchronous interrupt program so that its processing is to be completed within the program run section.</li> <li>Set the period specified in the fixed scan interval setting to an appropriate value.</li> </ol> </li> </ul>	Time information	At interrupt occurrence
1800H	Annunciator ON	<ul style="list-style-type: none"> <li>The annunciator (F) on status has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, and correct the program corresponding the displayed annunciator number.</li> </ul>	Error location information and annunciator information	At instruction execution
1830H	Receive queue full	<ul style="list-style-type: none"> <li>Number of reception requests of transient transmission exceeded upper limit of simultaneously processable requests.</li> </ul>	<ul style="list-style-type: none"> <li>Lower the transient transmission usage frequency, and then perform again.</li> </ul>	—	Always
1831H	Receive processing error	<ul style="list-style-type: none"> <li>Transient reception failed.</li> </ul>	<ul style="list-style-type: none"> <li>Lower the transient transmission usage frequency, and then perform again.</li> </ul>	System configuration information	Always
1832H	Transient data error	<ul style="list-style-type: none"> <li>Too many processings of transient transmission and cannot perform transient transmission.</li> </ul>	<ul style="list-style-type: none"> <li>Correct the transient transmission execution count.</li> </ul>	—	Always
1900H	Constant scan time error	<ul style="list-style-type: none"> <li>The scan time exceeded the constant scan time set in the CPU parameters.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the constant scan time setting.</li> </ul>	Time information	At END instruction execution
2000H	Module configuration error	<ul style="list-style-type: none"> <li>The module type set in the system parameters ("I/O Assignment Setting") differs from that of the module actually mounted.</li> </ul>	<ul style="list-style-type: none"> <li>Re-set the module type in the system parameters in accordance with the CPU module or intelligent function module actually mounted.</li> </ul>	System configuration information	At power-on, at RESET
2001H	Module configuration error	<ul style="list-style-type: none"> <li>The I/O numbers set in the system parameters ("I/O Assignment Setting") are overlapping between modules.</li> </ul>	<ul style="list-style-type: none"> <li>Re-set the I/O numbers in the system parameters in accordance with the intelligent function module or I/O module actually mounted.</li> </ul>	System configuration information	At power-on, at RESET
2002H	Module configuration error	<ul style="list-style-type: none"> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Re-set the number of points in the system parameters in accordance with the intelligent function module actually mounted.</li> </ul>	System configuration information	At power-on, at RESET
2004H	Module configuration error	<ul style="list-style-type: none"> <li>Nine or more CC-Link IE Controller Network modules* are mounted in the entire system.</li> <li>* The CC-Link IE built-in Ethernet interface module is included if the module is used as a CC-Link IE Controller Network module.</li> </ul>	<ul style="list-style-type: none"> <li>Reduce the number of CC-Link IE Controller Network modules* to eight or less in the entire system.</li> <li>* The CC-Link IE built-in Ethernet interface module is included if the module is used as a CC-Link IE Controller Network module.</li> </ul>	System configuration information	At power-on, at RESET
2005H	Module configuration error	<ul style="list-style-type: none"> <li>Two or more interrupt modules (QI60) with no interrupt pointer setting are mounted.</li> <li>The interrupt pointer numbers are overlapping in the interrupt module (QI60) with no interrupt pointer setting and a module with an interrupt pointer setting.</li> </ul>	<ul style="list-style-type: none"> <li>Mount only one QI60 in the entire system.</li> <li>Configure the interrupt pointer setting for the QI60.</li> <li>Correct the interrupt pointer setting.</li> </ul>	System configuration information	At power-on, at RESET, at STOP → RUN state
2006H	Module configuration error	<ul style="list-style-type: none"> <li>A module is mounted on the 65th slot or later.</li> </ul>	<ul style="list-style-type: none"> <li>Remove the module mounted on the 65th slot or later.</li> </ul>	System configuration information	At power-on, at RESET
2007H	Module configuration error	<ul style="list-style-type: none"> <li>A module is mounted on the slot whose number is later than that specified in the system parameters ("I/O Assignment Setting").</li> </ul>	<ul style="list-style-type: none"> <li>Remove the module mounted on the slot whose number is later than that specified in the system parameters.</li> </ul>	System configuration information	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2008H	Module configuration error	<ul style="list-style-type: none"> <li>A module is mounted over or across the maximum number of I/O points (4096).</li> </ul>	<ul style="list-style-type: none"> <li>Remove the module mounted over or across the maximum number of I/O points (4096).</li> <li>Replace the module mounted on the last slot to the one that does not exceed the maximum number of I/O points (4096).</li> </ul>	System configuration information	At power-on, at RESET
2009H	Module configuration error	<ul style="list-style-type: none"> <li>There is no response from the I/O module or intelligent function module accessed.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the I/O assignment setting in the system parameters.</li> <li>The possible cause is a hardware failure of the I/O module or intelligent function module accessed. Please consult your local Mitsubishi representative.</li> </ul>	System configuration information	Always
2020H	Module configuration error	<ul style="list-style-type: none"> <li>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").</li> </ul>	<ul style="list-style-type: none"> <li>Remove the unsupported module if any.</li> <li>Check whether modules are supported by the network type (module model name) set in the system parameters.</li> <li>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.</li> </ul>	System configuration information	At power-on, at RESET
2021H	Module configuration error	<ul style="list-style-type: none"> <li>In the 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.</li> </ul>	<ul style="list-style-type: none"> <li>Replace the Q series intelligent function module with the one (function version B) compatible with the multiple CPU system.</li> <li>Set the control CPU of the Q series intelligent function module incompatible with the multiple CPU system to CPU No.1.</li> </ul>	System configuration information	At power-on, at RESET
2040H	CPU module configuration error	<ul style="list-style-type: none"> <li>The number of CPU modules set in the system parameters ("I/O Assignment Setting") differs from the number of CPU modules actually mounted.</li> <li>The CPU module is mounted on the slot different from the one specified in the system parameters (I/O assignment setting).</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Correctly set the system parameters so that the setting and actual CPU module mounting status will be the same.</li> </ul>	System configuration information	At power-on, at RESET
2041H	CPU module configuration error	<ul style="list-style-type: none"> <li>The CPU module is not mounted on the slot that is set for the CPU module in the system parameters ("I/O Assignment Setting").</li> <li>The CPU module is mounted on the slot that is set for empty in the system parameters ("I/O Assignment Setting").</li> <li>An I/O module or intelligent function module is mounted between the CPU modules.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Remove the I/O module or intelligent function module mounted between the CPU modules.</li> </ul>	System configuration information	At power-on, at RESET
2043H	CPU module configuration error	<ul style="list-style-type: none"> <li>The CPU module is mounted on the inapplicable slot.</li> </ul>	<ul style="list-style-type: none"> <li>Mount the CPU module on the applicable slot (CPU slot or I/O slot 0 to 6).</li> <li>Remove the CPU module from the inapplicable slot.</li> </ul>	System configuration information	At power-on, at RESET
2044H	CPU module configuration error	<ul style="list-style-type: none"> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Re-set the host CPU No. in the system parameters in accordance with the mounting position of the CPU module.</li> </ul>	System configuration information	At power-on, at RESET
2050H	CPU module configuration error	<ul style="list-style-type: none"> <li>An unsupported CPU module is mounted.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	System configuration information	At power-on, at RESET

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2051H	CPU module configuration error	<ul style="list-style-type: none"> <li>A CPU module that does not support the online module change (direct change) function is used in the multiple CPU system.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the system configuration referring to the online module change manual.</li> <li>To disable the online module change (direct change) function, set "Disable" to the "Direct change setting" in the CPU parameters. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative. Please consult your local Mitsubishi representative.</li> </ul>	System configuration information	At power-on, at RESET
2060H	Base unit configuration error	<ul style="list-style-type: none"> <li>Eight or more extension base units are connected.</li> </ul>	<ul style="list-style-type: none"> <li>Reduce the number of extension base units to seven or less.</li> </ul>	System configuration information	At power-on, at RESET
2061H	Base unit configuration error	<ul style="list-style-type: none"> <li>Any of the following base units is connected: QA1S3□B, QA1S5□B/QA1S6□B, QA6□B, QA6ADP+A5□B/A6□B, or QA1S6ADP+A1S5□B/A1S6□B.</li> </ul>	<ul style="list-style-type: none"> <li>Remove the inapplicable base unit: QA1S3□B, QA1S5□B/QA1S6□B, QA6□B, QA6ADP+A5□B/A6□B□B, and QA1S6ADP+A1S5□B/A1S6□B.</li> </ul>	System configuration information	At power-on, at RESET
2063H	Base unit configuration error	<ul style="list-style-type: none"> <li>Extension base unit levels are overlapping.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the level setting of the extension base units.</li> </ul>	System configuration information	At power-on, at RESET
2070H	Base unit configuration error	<ul style="list-style-type: none"> <li>An unsupported base unit is connected.</li> <li>A GOT is bus-connected to the Q series extension base unit.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Disconnect the GOT bus-connected to the Q series extension base unit.</li> </ul>	System configuration information	At power-on, at RESET
2080H	Inter-module synchronization configuration error	<ul style="list-style-type: none"> <li>An inter-module synchronization signal error has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	System configuration information	At power-on, at RESET
20E0H	Module unrecognized	<ul style="list-style-type: none"> <li>A module that the CPU module cannot recognize is mounted.</li> <li>The module cannot be recognized because the control CPU setting of the system parameter setting differs from that of other CPU modules in the multiple CPU system.</li> </ul>	<ul style="list-style-type: none"> <li>Mount only applicable modules.</li> <li>Correct the system parameter settings for the CPU No.2 and later. The settings need to be the same between all the CPU modules.</li> <li>The possible cause is a hardware failure of the I/O module or intelligent function module accessed. Please consult your local Mitsubishi representative.</li> </ul>	System configuration information	Always
2100H	Memory error (extended SRAM cassette)	<ul style="list-style-type: none"> <li>The extended SRAM cassette set in the CPU parameters differs from the one actually inserted.</li> </ul>	<ul style="list-style-type: none"> <li>Re-set the extended SRAM cassette in the CPU parameters in accordance with the one actually inserted. If the same error code is displayed again, the possible cause is a hardware failure of the extended SRAM cassette. Please consult your local Mitsubishi representative.</li> </ul>	Drive/file information Parameter information	At power-on, at RESET
2101H	Memory error (extended SRAM cassette)	<ul style="list-style-type: none"> <li>The extended SRAM cassette is inserted or removed while the system is powered on.</li> </ul>	<ul style="list-style-type: none"> <li>Do not insert or remove the extended SRAM cassette during operation.</li> <li>Check if the extended SRAM cassette is securely inserted to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the extended SRAM cassette. Please consult your local Mitsubishi representative.</li> </ul>	Drive/file information	Always
2102H	Memory error (extended SRAM cassette)	<ul style="list-style-type: none"> <li>An error has been detected in the extended SRAM cassette.</li> </ul>	<ul style="list-style-type: none"> <li>Check if the extended SRAM cassette is securely inserted to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the extended SRAM cassette. Please consult your local Mitsubishi representative.</li> </ul>	Drive/file information	Always
2103H	Memory error (extended SRAM cassette)	<ul style="list-style-type: none"> <li>An unsupported extended SRAM cassette is inserted.</li> </ul>	<ul style="list-style-type: none"> <li>Replace the extended SRAM cassette with the one supported by the CPU module.</li> </ul>	Drive/file information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2120H	Memory card error	<ul style="list-style-type: none"> <li>The memory card was removed without the card being disabled.</li> </ul>	<ul style="list-style-type: none"> <li>Disable the memory card, and then remove it.</li> </ul>	Drive/file information	Always
2121H	Memory card error	<ul style="list-style-type: none"> <li>An error has been detected in the memory card.</li> </ul>	<ul style="list-style-type: none"> <li>Format the memory card, or re-insert the memory card. If the same error code is displayed again, the possible cause is a hardware failure of the memory card. Replace the memory card.</li> </ul>	Drive/file information	Always
2122H	Memory card error	<ul style="list-style-type: none"> <li>The CPU module did not start because the memory card was not restored during the startup processing.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Drive/file information	At power-on, at RESET
2160H	IP address duplication error	<ul style="list-style-type: none"> <li>Overlapping IP addresses have been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the IP addresses.</li> </ul>	—	Always
2180H	Invalid file	<ul style="list-style-type: none"> <li>An invalid file has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, select the correct file name, and write the specified file to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.</li> </ul>	Drive/file information	At power-on, at RESET, at STOP → RUN state
21A0H	File specification error	<ul style="list-style-type: none"> <li>The file specified in the CPU parameters does not exist.</li> <li>The memory card is disabled by SM606 (SD memory card forced disable instruction).</li> <li>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).</li> <li>The file specified in the boot file setting of the memory card parameters does not exist in the memory card.</li> </ul>	<ul style="list-style-type: none"> <li>Turn off SM606. (Cancel the disabled state.)</li> <li>Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, select the correct file name, and write the specified file to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the device/label memory in the CPU module or the memory card. Please consult your local Mitsubishi representative.</li> </ul>	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	<ul style="list-style-type: none"> <li>The file specified in parameter cannot be created.</li> </ul>	<ul style="list-style-type: none"> <li>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 number.</li> <li>Check the detailed information (drive/file information) of the error by executing module diagnostics using the engineering tool, and take either of the following action. <ol style="list-style-type: none"> <li>Format the corresponding drive.</li> <li>Delete unnecessary files on the corresponding drive to increase free space.</li> <li>Unlock the corresponding drive if it is locked.</li> </ol> </li> </ul>	Drive/file information Parameter information	At write, at power-on, at RESET, at STOP → RUN state
21A2H	File specification error	<ul style="list-style-type: none"> <li>The CPU module model set to the file using the engineering tool differs from that of the CPU module actually mounted.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Drive/file information	At write, at power-on, at RESET, at STOP → RUN state
2200H	Parameter error	<ul style="list-style-type: none"> <li>The system parameter file and CPU parameter file do not exist.</li> <li>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 SM606 (SD memory card forced disable instruction).</li> </ul>	<ul style="list-style-type: none"> <li>Write the system parameter file and CPU parameter file to the CPU module.</li> <li>Turn off SM606. (Cancel the disabled state.)</li> </ul>	Parameter information	At power-on, at RESET, at STOP → RUN state



Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2220H	Parameter error	<ul style="list-style-type: none"> <li>The parameter setting is corrupted.</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and write the displayed parameter setting to the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, the memory card, or the module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.</li> </ul>	Parameter information	At power-on, at RESET, at STOP → RUN state
2221H	Parameter error	<ul style="list-style-type: none"> <li>The set value is out of range.</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, the memory card, or the module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.</li> </ul>	Parameter information	At power-on, at RESET, at STOP → RUN state, at END instruction execution, at instruction execution, at module access
2222H	Parameter error	<ul style="list-style-type: none"> <li>Use of the function that is not supported by the module is enabled in parameter.</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (parameter information) of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, the memory card, or the module (I/O module or intelligent function module) connected. Please consult your local Mitsubishi representative.</li> </ul>	Parameter information	At power-on, at RESET, at STOP → RUN state
2223H	Parameter error	<ul style="list-style-type: none"> <li>The parameter that needs a reset of the CPU module was modified and overwritten.</li> </ul>	<ul style="list-style-type: none"> <li>Reset the CPU module, and run it again.</li> </ul>	Parameter information	At write, at power-on, at RESET, at STOP → RUN state
2224H	Parameter error	<ul style="list-style-type: none"> <li>A memory area cannot be ensured.</li> </ul>	<ul style="list-style-type: none"> <li>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.)</li> <li>Reduce the number of labels or local devices used.</li> </ul>	Parameter information	At write, at power-on, at RESET, at STOP → RUN state
2225H	Parameter error	<ul style="list-style-type: none"> <li>The CPU module model set to the project using the engineering tool differs from that of the CPU module actually mounted.</li> <li>The operation set in the memory card parameters cannot be performed. (The boot function cannot be executed.)</li> </ul>	<ul style="list-style-type: none"> <li>Correct the CPU module model set to the project in accordance with the CPU module actually mounted.</li> <li>Delete the memory card parameter settings.</li> <li>Remove the memory card so that the operation set in the memory card parameters will not be performed. (Do not execute the boot operation.)</li> </ul>	Parameter information	At write, at power-on, at RESET, at STOP → RUN state
2240H	Parameter error (module)	<ul style="list-style-type: none"> <li>In the multiple CPU system, the I/O module or intelligent function module controlled by another CPU module is specified in the module parameters.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Parameter information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2241H	Parameter error (module)	<ul style="list-style-type: none"> <li>The I/O numbers set in the system parameters differ from those of the module actually mounted.</li> <li>The target module is not mounted on the slot where the system parameters and module parameters are set.</li> <li>The module type set in parameter differs from that of the module actually mounted.</li> </ul>	<ul style="list-style-type: none"> <li>Check if the system configuration displayed on the system monitor window match the actual system configuration.</li> <li>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.</li> </ul>	Parameter information	At power-on, at RESET, at STOP → RUN state, at END instruction execution, at instruction execution, at module access
2242H	Parameter error (module)	<ul style="list-style-type: none"> <li>The intelligent function module has detected a module parameter error.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	System configuration information	At power-on, at RESET, at STOP → RUN state
2260H	Parameter error (network)	<ul style="list-style-type: none"> <li>Network numbers are overlapping.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Parameter information	At power-on, at RESET
2261H	Parameter error (network)	<ul style="list-style-type: none"> <li>Different network types (CC IE Control extended mode/normal mode) are set between the control station and the normal station.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Parameter information	At power-on, at RESET, at STOP → RUN state
2262H	Parameter error (network)	<ul style="list-style-type: none"> <li>The station type set in the module parameters differs that of the module actually mounted.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Parameter information	At power-on, at RESET, at STOP → RUN state
2263H	Parameter error (network)	<ul style="list-style-type: none"> <li>Even though the CC-Link IE module is mounted, a different module is set in the system parameters ("I/O Assignment Setting"), or CC-Link IE module parameters are not set.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Parameter information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2280H	Parameter error (refresh)	<ul style="list-style-type: none"> <li>The refresh setting is not set correctly. (Data were refreshed exceeding the file register capacity.)</li> <li>The refresh settings (number of points) are different from those of other numbered CPU modules.</li> </ul>	<ul style="list-style-type: none"> <li>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 any of 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.)</li> <li>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.)</li> </ul>	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)	<ul style="list-style-type: none"> <li>A device that cannot be used as a refresh device is specified.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Parameter information	At power-on, at RESET, at STOP → RUN state
2282H	Parameter error (refresh)	<ul style="list-style-type: none"> <li>The number of specified refresh points is invalid.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Parameter information	At power-on, at RESET, at STOP → RUN state
2283H	Parameter error (refresh)	<ul style="list-style-type: none"> <li>The total number of refresh points exceeded the maximum limit.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Parameter information	At power-on, at RESET, at STOP → RUN state
22E0H	Parameter verification error	<ul style="list-style-type: none"> <li>In the multiple CPU system, the system parameter settings of the host CPU module differ from those of other CPU modules.</li> <li>In the 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.</li> </ul>	<ul style="list-style-type: none"> <li>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 between 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.)</li> <li>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.)</li> </ul>	Parameter information System configuration information	At write, at power-on, at RESET, at STOP → RUN state
2300H	Security key authentication error	<ul style="list-style-type: none"> <li>The security key set to the program does not match the one registered to the CPU module (or extended SRAM cassette).</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the security key setting.</li> </ul>	Drive/file information	At power-on, at RESET, at STOP → RUN state
2301H	Security key authentication error	<ul style="list-style-type: none"> <li>The security key is set to the program, but it is not registered to the CPU module (or extended SRAM cassette).</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the security key setting.</li> </ul>	Drive/file information	At power-on, at RESET, at STOP → RUN state
2302H	Security key authentication error	<ul style="list-style-type: none"> <li>The security key set to the file is corrupted and does not match the one registered to the CPU module.</li> <li>The security key registered to the CPU module is corrupted and does not match the one set to the file.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	—	At power-on, at RESET, at STOP → RUN state
2303H	Security key authentication error	<ul style="list-style-type: none"> <li>The security key is registered to the CPU module and extended SRAM cassette.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the security key setting.</li> </ul>	—	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2320H	Remote password setting error	<ul style="list-style-type: none"> <li>The start I/O number of the remote password target module is set to other than 0H to 0FF0H.</li> <li>There is a problem on the slot specified by the start I/O number of the remote password setting.               <ol style="list-style-type: none"> <li>No module is mounted.</li> <li>The mounted intelligent function module does not support the remote password setting.</li> </ol> </li> </ul>	<ul style="list-style-type: none"> <li>Set the start I/O number of the remote password target module within the range 0H to 0FF0H.</li> <li>On the specified slot, mount an intelligent function module that supports the remote password setting.</li> </ul>	System configuration information	At power-on, at RESET, at STOP → RUN state
2321H	Remote password setting error	<ul style="list-style-type: none"> <li>In the multiple CPU system, the module controlled by another CPU module is specified by the start I/O number of the remote password setting.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the remote password setting.</li> </ul>	System configuration information	At power-on, at RESET, at STOP → RUN state
2400H	Module verification error	<ul style="list-style-type: none"> <li>The module information at power-on differs from the information of modules actually mounted.</li> <li>The I/O module or intelligent function module is not mounted properly or was removed during operation.</li> </ul>	<ul style="list-style-type: none"> <li>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 number.</li> </ul>	System configuration information	Always
2401H	Module verification error	<ul style="list-style-type: none"> <li>A CPU module, I/O module, or intelligent function module was mounted on the base unit during operation.</li> </ul>	<ul style="list-style-type: none"> <li>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 number.</li> <li>Do not mount a CPU module, I/O module, nor intelligent function module during operation.</li> </ul>	System configuration information	Always
2420H	Fuse blown error	<ul style="list-style-type: none"> <li>The output module with a blown fuse has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Check the FUSE LED of each output module, and replace the one with the FUSE LED on.</li> <li>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 number.</li> </ul>	System configuration information	Always
2440H	Module major error	<ul style="list-style-type: none"> <li>In the multiple CPU system, the control CPU setting in the system parameters is different from that of other numbered CPU modules.</li> <li>An error has been detected in the I/O module or intelligent function module during the initial processing.</li> </ul>	<ul style="list-style-type: none"> <li>Correct the system parameter settings for the CPU No.2 and later. The settings need to be the same between all the CPU modules.</li> <li>The possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.</li> </ul>	System configuration information	At power-on, at RESET
2441H	Module major error	<ul style="list-style-type: none"> <li>An error has been detected in the I/O module or intelligent function module when the instruction was executed.</li> </ul>	<ul style="list-style-type: none"> <li>The possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.</li> </ul>	Error location information and system configuration information	At instruction execution
2442H	Module major error	<ul style="list-style-type: none"> <li>An error has been detected in the I/O module or intelligent function module during the END processing.</li> </ul>	<ul style="list-style-type: none"> <li>The possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.</li> </ul>	System configuration information	At module access
2443H	Module major error	<ul style="list-style-type: none"> <li>An error has been detected in the I/O module or intelligent function module.</li> </ul>	<ul style="list-style-type: none"> <li>The possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.</li> </ul>	System configuration information	At module access
2450H	Module major error	<ul style="list-style-type: none"> <li>A major error has been notified from the intelligent function module connected.</li> <li>The I/O module or intelligent function module is not mounted properly or was removed during operation.</li> </ul>	<ul style="list-style-type: none"> <li>Check the connection status of the extension cable.</li> <li>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 number.</li> <li>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.</li> </ul>	System configuration information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2460H	Another CPU module major error	<ul style="list-style-type: none"> <li>An error has been detected in another CPU module during the initial processing.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	System configuration information	At power-on, at RESET
2461H	Another CPU module major error	<ul style="list-style-type: none"> <li>An error has been detected in another CPU module when the instruction was executed.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information and system configuration information	At instruction execution
2462H	Another CPU module major error	<ul style="list-style-type: none"> <li>An error has been detected in another CPU module during the END processing.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	System configuration information	At END instruction execution
2463H	Another CPU module major error	<ul style="list-style-type: none"> <li>An error has been detected in another CPU module.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	System configuration information	At power-on, at RESET
2470H	Another CPU module major error	<ul style="list-style-type: none"> <li>A major error has been notified from another CPU module.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	System configuration information	Always
2480H	Multiple CPU error	<ul style="list-style-type: none"> <li>In the multiple CPU system, an error has been detected in the CPU module where "Stop" is set in the operation mode setting parameter.</li> <li>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.)</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (system configuration information) of the error by executing module diagnostics using the engineering tool, identify the error CPU module, and eliminate the error cause.</li> <li>Remove the CPU module from the inapplicable slot.</li> </ul>	System configuration information	Always
2481H	Multiple CPU error	<ul style="list-style-type: none"> <li>In the 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.</li> </ul>	<ul style="list-style-type: none"> <li>Check the mounting status and reset status of the CPU modules other than CPU No.1.</li> </ul>	System configuration information	Always
24C0H	System bus error	<ul style="list-style-type: none"> <li>An error was detected on the system bus.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	System configuration information	At module access
24C1H	System bus error	<ul style="list-style-type: none"> <li>An error was detected on the system bus.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	System configuration information	At module access

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
24C2H	System bus error	<ul style="list-style-type: none"> <li>An error was detected on the system bus.</li> </ul>	<ul style="list-style-type: none"> <li>Check the connection status of the extension cable.</li> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	System configuration information	Always
24C3H	System bus error	<ul style="list-style-type: none"> <li>An error was detected on the system bus.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	System configuration information	At module access
24C4H	System bus error	<ul style="list-style-type: none"> <li>An error was detected on the system bus.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	System configuration information	At module access
24C5H	System bus error	<ul style="list-style-type: none"> <li>An error was detected on the system bus.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	—	At module access
24C6H	System bus error	<ul style="list-style-type: none"> <li>An error was detected on the system bus.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	—	At module access
24C8H	System bus error	<ul style="list-style-type: none"> <li>An error was detected on the system bus.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	—	At power-on, at RESET
24D0H	System bus error	<ul style="list-style-type: none"> <li>The extension level setting of the Q series extension base unit is overlapping with that of any other extension base units.</li> <li>An error was detected on the system bus.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the level setting of the Q series extension base unit.</li> <li>Check the connection status of the extension cable.</li> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	System configuration information	Always
24E0H	System bus error	<ul style="list-style-type: none"> <li>An error was detected on the system bus.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	System configuration information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2500H	WDT error	<ul style="list-style-type: none"> <li>The scan time exceeded the execution monitoring time set in parameter.</li> <li>The initial (1st) scan time exceeded the execution monitoring time set in the CPU parameters.</li> <li>The execution time of the fixed scan interrupt program exceeded the interrupt execution interval.</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take either of the following actions.</li> <li>(1) Check and correct the program so that it can be executed within the monitoring time set in parameter.</li> <li>(2) Change the monitoring time setting.</li> <li>Check and correct the fixed scan interrupt program so that the processing completes within the interrupt execution interval.</li> </ul> <p>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.</p>	Time information	Always
2501H	WDT error	<ul style="list-style-type: none"> <li>The scan time exceeded the execution monitoring time set in parameter.</li> <li>The 2nd or later scan time exceeded the execution monitoring time set in the CPU parameters.</li> <li>The execution time of the fixed scan interrupt program exceeded the interrupt execution interval.</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (time information) of the error by executing module diagnostics using the engineering tool, check the time setting, and take either of the following actions.</li> <li>(1) Check and correct the program so that it can be executed within the monitoring time set in parameter.</li> <li>(2) Change the monitoring time setting.</li> <li>Check and correct the fixed scan interrupt program so that the processing completes within the interrupt execution interval.</li> </ul> <p>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.</p>	Time information	Always
2520H	Invalid interrupt	<ul style="list-style-type: none"> <li>Even though an interrupt was requested, there is no interrupt factor.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	System configuration information	At interrupt occurrence
2521H	Invalid interrupt	<ul style="list-style-type: none"> <li>Even though an interrupt was requested, there is no interrupt factor.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	—	At interrupt occurrence
2522H	Invalid interrupt	<ul style="list-style-type: none"> <li>An interrupt was requested from the module with no interrupt pointer setting.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the interrupt pointer setting in the module parameters.</li> <li>Take measures so that no interrupt is requested from the module with no interrupt pointer setting.</li> <li>Check and correct the interrupt setting in the buffer memory of the intelligent function module.</li> <li>Correct the BASIC program executed in the QD51.</li> </ul>	System configuration information	At interrupt occurrence
2610H	Inter-module synchronization signal error	<ul style="list-style-type: none"> <li>An execution interval error of the synchronization interrupt program has been detected.</li> <li>An inter-module synchronization error has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	—	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2611H	Inter-module synchronization signal error	<ul style="list-style-type: none"> <li>An inter-module synchronization error has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	System configuration information	At power-on, at RESET, at END instruction execution
2630H	Multiple CPU synchronization signal error	<ul style="list-style-type: none"> <li>An execution interval error of the synchronization interrupt program has been detected.</li> <li>A multiple CPU synchronization error has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	—	Always
2631H	Multiple CPU synchronization signal error	<ul style="list-style-type: none"> <li>A multiple CPU synchronization error has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	System configuration information	At power-on, at RESET, at END instruction execution
2800H	I/O number or network number specification error	<ul style="list-style-type: none"> <li>The specified I/O number is out of range (other than 000H to FFFH, 3E0H to 3E3H).</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
2801H	I/O number or network number specification error	<ul style="list-style-type: none"> <li>The I/O number of the module that does not exist was specified.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information System configuration information	At instruction execution
2802H	I/O number or network number specification error	<ul style="list-style-type: none"> <li>The I/O number of the module that does not support the instruction was specified.</li> <li>The dedicated instruction specified in the program cannot be executed in the specified module or mode.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Check the execution propriety (including support status and execution mode) of the dedicated instruction, referring to the manual for the target module.</li> </ul>	Error location information System configuration information	At instruction execution
2803H	I/O number or network number specification error	<ul style="list-style-type: none"> <li>The I/O number of the module that cannot be specified in the instruction was specified.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information System configuration information	At instruction execution
2804H	I/O number or network number specification error	<ul style="list-style-type: none"> <li>The specified network number is out of range (other than 1 to 239).</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
2805H	I/O number or network number specification error	<ul style="list-style-type: none"> <li>The network number that does not exist was specified.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution



Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2806H	I/O number or network number specification error	<ul style="list-style-type: none"> <li>An I/O module or intelligent function module controlled by another CPU module was specified.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Delete the link direct device that specifies a network module controlled by another CPU module from the program.</li> <li>Specify a network module controlled by host CPU module by using the link direct device.</li> </ul>	Error location information System configuration information	At instruction execution
2807H	I/O number or network number specification error	<ul style="list-style-type: none"> <li>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.)</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
2810H	I/O number or network number specification error	<ul style="list-style-type: none"> <li>The I/O module or intelligent function module specified in the instruction cannot execute the instruction.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information System configuration information	At instruction execution
2820H	Device, label, or buffer memory specification error	<ul style="list-style-type: none"> <li>The device or label area used in the instruction exceeded the specified range.</li> <li>The file register file is not set or was accessed without setting it in the CPU parameters (file setting).</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Set the file register file in parameter, and access the file.</li> </ul>	Error location information Process control instruction processing information	At instruction execution, at END instruction execution
2821H	Device, label, or buffer memory specification error	<ul style="list-style-type: none"> <li>The device or label areas used in the instruction to store data are overlapping.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information Process control instruction processing information	At instruction execution
2822H	Device, label, or buffer memory specification error	<ul style="list-style-type: none"> <li>The device or label that cannot be used in the instruction was specified.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
2823H	Device, label, or buffer memory specification error	<ul style="list-style-type: none"> <li>The buffer memory area of the module specified in the instruction exceeded the specified range.</li> <li>The module specified in the instruction does not have buffer memory.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
2824H	Device, label, or buffer memory specification error	<ul style="list-style-type: none"> <li>The access prohibited area in the buffer memory was accessed.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
2840H	File name specification error	<ul style="list-style-type: none"> <li>The file specified in the instruction does not exist.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>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).</li> </ul>	Error location information Drive/file information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
2841H	File name specification error	<ul style="list-style-type: none"> <li>The program file specified in the instruction is not set in the CPU parameters (program setting).</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>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 CPU parameter.</li> </ul>	Error location information Drive/file information	At instruction execution
3000H	Boot function execution error	<ul style="list-style-type: none"> <li>The boot setting in the memory card parameters is incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the boot setting in the memory card parameters.</li> </ul>	Drive/file information	At power-on, at RESET
3001H	Boot function execution error	<ul style="list-style-type: none"> <li>When the boot function was executed, the file format processing failed.</li> </ul>	<ul style="list-style-type: none"> <li>Reset the CPU module, and execute the boot function again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.</li> </ul>	Drive/file information	At power-on, at RESET
3003H	Boot function execution error	<ul style="list-style-type: none"> <li>When the boot function was executed, the file passwords did not match.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the file password settings of the transfer source and transfer destination files.</li> <li>Delete the boot setting.</li> </ul>	Drive/file information	At power-on, at RESET
3004H	Boot function execution error	<ul style="list-style-type: none"> <li>When the boot function was executed, the CPU built-in memory capacity was exceeded.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the boot setting.</li> <li>Delete unnecessary files in the CPU built-in memory.</li> <li>Clear the CPU built-in memory by selecting "Clear" to "Operation Setting at CPU Built-in Memory Boot" in the memory card parameters, and execute the boot function.</li> </ul>	Drive/file information	At power-on, at RESET
3005H	Boot function execution error	<ul style="list-style-type: none"> <li>When the boot function is executed, the security key registered in the CPU module (or extended SRAM cassette) does not match the one that locks the boot source program.</li> <li>When the boot function is executed, the security key is not registered in the CPU module (or extended SRAM cassette) even though the boot source program is locked with the security key.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the security key setting.</li> <li>Delete the boot setting.</li> </ul>	Drive/file information	At power-on, at RESET
3100H	Program error	<ul style="list-style-type: none"> <li>The program includes any instruction that cannot be used or decoded in the CPU module.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Take measures to reduce noise.</li> <li>Write the program to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.</li> </ul>	Error location information	At instruction execution
3121H	Program error	<ul style="list-style-type: none"> <li>The number of devices used in the dedicated instruction specified in the program is incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3122H	Program error	<ul style="list-style-type: none"> <li>The function block or function specified in the program does not exist.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Take measures to reduce noise.</li> <li>Write the program(s) and FB program(s) to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.</li> </ul>	Error location information	At instruction execution
3140H	END instruction error	<ul style="list-style-type: none"> <li>The END (FEND) instruction does not exist in the program.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Take measures to reduce noise.</li> <li>Write the program(s) and FB program(s) to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.</li> </ul>	Error location information	At power-on, at RESET, at STOP → RUN state
3141H	FB/FUN program error	<ul style="list-style-type: none"> <li>The structure of FB/FUN program is incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>Write the program(s) and FB program(s) to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.</li> </ul>	Error location information	At instruction execution
3142H	Temporary area error	<ul style="list-style-type: none"> <li>The temporary area was used incorrectly.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Take measures to reduce noise.</li> <li>Write the program(s) and FB program(s) to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.</li> </ul>	Error location information	At instruction execution
3200H	Program execution error	<ul style="list-style-type: none"> <li>Memory/Device Setting set in the CPU parameters differ from those assigned to the programs, FB programs, and global label setting file. (After Memory/Device Setting were modified, only the CPU parameters were written to the CPU module.)</li> </ul>	<ul style="list-style-type: none"> <li>After Memory/Device Setting are modified, write the program file(s), FB file(s), and global label setting file together with the CPU parameter file to the CPU module.</li> <li>If no global label is used, delete the global label setting file.</li> </ul>	Drive/file information	At power-on, at RESET, at STOP → RUN state
3201H	Program execution error	<ul style="list-style-type: none"> <li>Even though no program is set in the CPU parameters, multiple program files exist.</li> </ul>	<ul style="list-style-type: none"> <li>If no global label is used, delete the global label setting file.</li> </ul>	Drive/file information	At power-on, at RESET, at STOP → RUN state
3202H	Program execution error	<ul style="list-style-type: none"> <li>The program file is incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>Write the program file to the CPU built-in memory again.</li> </ul>	Drive/file information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3203H	Program execution error	<ul style="list-style-type: none"> <li>No program file exists.</li> </ul>	<ul style="list-style-type: none"> <li>Check if the system parameter file, CPU parameter file, and program file exist.</li> <li>Write the system parameter file, CPU parameter file, and program file to the CPU built-in memory.</li> </ul>	Drive/file information	At power-on, at RESET, at STOP → RUN state
3205H	Program execution error	<ul style="list-style-type: none"> <li>After the global label setting file was modified, only the modified file was written to the programmable controller. Or, without writing the global label setting file, only the program file(s) and FB file(s) were written. (The global label setting file was not written to the programmable controller.)</li> <li>After an FB file was modified, only the modified file was written to the programmable controller. Or, without writing the modified FB file, only the program file(s) and global label setting file were written. (The modified FB file was not written to the programmable controller.)</li> <li>After the external device access setting of the global label setting was modified, only the global label assignment information was written to the programmable controller. Or, after the external device access setting of the global label setting was modified, only the program file(s) and FB file(s) were written to the programmable controller. (The global label setting file was not written.)</li> </ul>	<ul style="list-style-type: none"> <li>Write all the program file(s), FB file(s), global label setting file, and global label assignment information file to the programmable controller.</li> <li>If no global label is used, delete the global label setting file.</li> <li>If the external device access setting is not used, initialize the memory where the global label assignment information file is stored, and write the program file(s), FB file(s), and global label setting file to the programmable controller.</li> </ul>	Drive/file information	At power-on, at RESET, at STOP → RUN state
3206H	Program execution error	<ul style="list-style-type: none"> <li>After the program was modified, only the modified program file was written to the programmable controller. Or, without writing the modified program file, only the initial local label value file was written.</li> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Write both the program file(s) and initial local label value file to the programmable controller.</li> <li>Write both the global label setting file and initial global label value file to the programmable controller.</li> <li>If no initial global label value is used, delete the initial global label value file.</li> <li>If no initial local label value is used, delete the initial local label value file.</li> </ul>	Drive/file information	At power-on, at RESET, at STOP → RUN state
3207H	Program execution error	<ul style="list-style-type: none"> <li>The signal flow area to be used in the FB program cannot be secured.</li> </ul>	<ul style="list-style-type: none"> <li>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 program file.</li> </ul>	Drive/file information	At power-on, at RESET, at STOP → RUN state
3300H	Pointer setting error	<ul style="list-style-type: none"> <li>The total number of points of local and global pointers used in the program exceeded the points set in the CPU parameters (pointer device area).</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Check and correct the pointer device area point setting in the CPU parameters.</li> </ul>	Error location information	At power-on, at RESET, at STOP → RUN state
3301H	Pointer setting error	<ul style="list-style-type: none"> <li>The total number of points of pointer-type labels used in the program exceeded the points set in the CPU parameters (label assignment area).</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Check and correct the pointer-type label assignment area point setting in the CPU parameters.</li> </ul>	Error location information	At power-on, at RESET, at STOP → RUN state

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3302H	Pointer setting error	<ul style="list-style-type: none"> <li>Multiple global pointers with the same number are used in the program. (The pointer numbers are overlapping.)</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At power-on, at RESET, at STOP → RUN state
3303H	Pointer setting error	<ul style="list-style-type: none"> <li>Multiple local pointers with the same number are used in the program. (The pointer numbers are overlapping.)</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At power-on, at RESET, at STOP → RUN state
3320H	Interrupt pointer setting error	<ul style="list-style-type: none"> <li>The interrupt pointer numbers used in the files are overlapping.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At power-on, at RESET, at STOP → RUN state
3340H	FOR-NEXT instruction error	<ul style="list-style-type: none"> <li>The NEXT instruction was not executed even though the FOR instruction was executed. Or, there are more FOR instructions than NEXT instructions.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At END instruction execution
3341H	FOR-NEXT instruction error	<ul style="list-style-type: none"> <li>The NEXT instruction was executed even though the FOR instruction was not executed. Or, there are more NEXT instructions than FOR instructions.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
3342H	FOR-NEXT instruction error	<ul style="list-style-type: none"> <li>The BREAK instruction was executed even though the FOR instruction was not executed.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
3360H	Nesting depth error	<ul style="list-style-type: none"> <li>The number of nesting levels in the subroutine program exceeded its limit (16).</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) using the error jump function, and change the number of nesting levels (16 or less).</li> </ul>	Error location information	At instruction execution
3361H	Nesting depth error	<ul style="list-style-type: none"> <li>The number of nesting levels in the FOR instruction exceeded its limit (16).</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) using the error jump function, and change the number of nesting levels (16 or less).</li> </ul>	Error location information	At instruction execution
3362H	Nesting depth error	<ul style="list-style-type: none"> <li>The number of nesting levels in the DI instruction exceeded its limit (16).</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) using the error jump function, and change the number of nesting levels (16 or less).</li> </ul>	Error location information	At instruction execution
3363H	Nesting depth error	<ul style="list-style-type: none"> <li>The number of nesting levels in the function block or function exceeded its limit (32).</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) using the error jump function, and change the number of nesting levels (32 or less).</li> </ul>	Error location information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3380H	Pointer execution error	<ul style="list-style-type: none"> <li>The pointer specified in the instruction does not exist.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
3381H	Pointer execution error	<ul style="list-style-type: none"> <li>The RET instruction does not exist in the executed subroutine program.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At END instruction execution
3382H	Pointer execution error	<ul style="list-style-type: none"> <li>The RET instruction exists before the FEND instruction in the main program.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
33A0H	Interrupt pointer execution error	<ul style="list-style-type: none"> <li>The interrupt pointer corresponding to the interrupt input does not exist.</li> </ul>	<ul style="list-style-type: none"> <li>Check if the program corresponding to the interrupt pointer number set in the module parameters exists.</li> </ul>	—	At instruction execution
33A1H	Interrupt pointer execution error	<ul style="list-style-type: none"> <li>The IRET instruction does not exist in the executed interrupt program.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
33A2H	Interrupt pointer execution error	<ul style="list-style-type: none"> <li>The IRET instruction exists before the FEND instruction in the main program.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
33A3H	Interrupt pointer execution error	<ul style="list-style-type: none"> <li>The IRET instruction or STOP instruction was executed in the fixed scan execution type program.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
33C0H	FB/FUN execution error	<ul style="list-style-type: none"> <li>Before the FB/FUN program ends, the call source program ended.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Take measures to reduce noise.</li> <li>Write the program to the CPU module again. Then, reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.</li> </ul>	Error location information	At instruction execution
33D0H	Temporary area exceeded	<ul style="list-style-type: none"> <li>The secured temporary area size exceeded its limit.</li> </ul>	<ul style="list-style-type: none"> <li>Check the detailed information (error location information) of the error by executing module diagnostics using the engineering tool, display the error program (step) using the error jump function, and correct the number of nesting levels in the function.</li> </ul>	Error location information	At instruction execution
3400H	Operation error	<ul style="list-style-type: none"> <li>Division where the divisor is zero was performed.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information Process control instruction processing information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3401H	Operation error	<ul style="list-style-type: none"> <li>Data that cannot be converted by using the data conversion instruction was input.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
3402H	Operation error	<ul style="list-style-type: none"> <li>The operation was performed with the invalid data (-0, denormalized number, NaN (not a number), or <math>\pm\infty</math>).</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information Process control instruction processing information	At instruction execution
3403H	Operation error	<ul style="list-style-type: none"> <li>An overflow has occurred during the operation.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information Process control instruction processing information	At instruction execution
3404H	Operation error	<ul style="list-style-type: none"> <li>A string that is not supported in the instruction was specified.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
3405H	Operation error	<ul style="list-style-type: none"> <li>The input data was out of range.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information Process control instruction processing information	At instruction execution
3406H	Operation error	<ul style="list-style-type: none"> <li>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.)</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information Process control instruction processing information	At instruction execution
3420H	Operation error	<ul style="list-style-type: none"> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
3421H	Operation error	<ul style="list-style-type: none"> <li>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.</li> <li>The value set in SD771 is out of range.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Execute the SP.DEVST instruction again on another day, or change the value in SD771.</li> <li>Set the value in SD771 within the settable range.</li> </ul>	Error location information	At instruction execution
3422H	Operation error	<ul style="list-style-type: none"> <li>The structure of the PID control instruction is incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3423H	Operation error	<ul style="list-style-type: none"> <li>The size of data to be sent/received by the socket communications instruction exceeds the allowable range.</li> </ul>	<ul style="list-style-type: none"> <li>Check and change the send data size of the CPU module or the external device.</li> <li>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.</li> </ul>	Error location information	At instruction execution
3440H	Operation error	<ul style="list-style-type: none"> <li>In the 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).</li> </ul>	<ul style="list-style-type: none"> <li>Change "Fixed scan communication function" to "Use".</li> <li>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 (the one whose symbol starts with M(P)).</li> </ul>	Error location information	At instruction execution
3441H	Operation error	<ul style="list-style-type: none"> <li>In the multiple CPU system, the number of data points was specified exceeding the multiple CPU dedicated instruction areas applicable for each CPU module.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	Error location information	At instruction execution
34A0H	Operation error	<ul style="list-style-type: none"> <li>Response data of the socket communications instruction cannot be created.</li> </ul>	<ul style="list-style-type: none"> <li>Increase the request interval.</li> <li>Decrease the number of request nodes.</li> <li>Wait for a response to the previous request before sending the next request.</li> <li>Correct the timeout value.</li> </ul>	Error location information	At END instruction execution
3C00H	Hardware failure	<ul style="list-style-type: none"> <li>A hardware failure has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	Always
3C01H	Hardware failure	<ul style="list-style-type: none"> <li>A hardware failure has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	Always
3C02H	Hardware failure	<ul style="list-style-type: none"> <li>A hardware failure has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	At power-on, at RESET, at END instruction execution, at interrupt occurrence
3C03H	Hardware failure	<ul style="list-style-type: none"> <li>A hardware failure has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	Always
3C0FH	Hardware failure	<ul style="list-style-type: none"> <li>A hardware failure has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	Always
3C10H	Hardware failure	<ul style="list-style-type: none"> <li>A hardware failure has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	At power-on, at RESET



Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3C11H	Hardware failure	<ul style="list-style-type: none"> <li>A hardware failure has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	At END instruction execution, at instruction execution
3C12H	Hardware failure	<ul style="list-style-type: none"> <li>The waveform of the voltage out of the specified range has been detected in the power supply module.</li> <li>A hardware failure has been detected in the power supply module, CPU module, base unit, or extension cable.</li> </ul>	<ul style="list-style-type: none"> <li>Check the waveform of the voltage applied to the power supply module.</li> <li>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.</li> </ul>	Failure information	Always
3C13H	Hardware failure	<ul style="list-style-type: none"> <li>A hardware failure has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	Always
3C14H	Hardware failure	<ul style="list-style-type: none"> <li>A hardware failure has been detected.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	Always
3C20H	Memory error	<ul style="list-style-type: none"> <li>An error has been detected in the memory.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	At power-on, at RESET
3C21H	Memory error	<ul style="list-style-type: none"> <li>An error has been detected in the memory.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	At END instruction execution, at power-on, at RESET
3C22H	Memory error	<ul style="list-style-type: none"> <li>An error has been detected in the memory.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	At END instruction execution, at power-on, at RESET
3C2FH	Memory error	<ul style="list-style-type: none"> <li>An error has been detected in the memory.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	Always
3C30H	Memory error	<ul style="list-style-type: none"> <li>An error has been detected in the memory.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	At instruction execution
3C31H	Memory error	<ul style="list-style-type: none"> <li>An error has been detected in the memory.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	Always

Error code	Error name	Error details and cause	Action	Detailed information	Diagnostic timing
3C32H	Memory error	<ul style="list-style-type: none"> <li>An error has been detected in the memory.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	Always
3E00H	Operation circuit error	<ul style="list-style-type: none"> <li>An error has been detected in the CPU module.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	At power-on, at RESET
3E20H	Program execution error	<ul style="list-style-type: none"> <li>The entire program was executed without executing the END instruction.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	At END instruction execution
3E22H	Program execution error	<ul style="list-style-type: none"> <li>The FB/FUN program did not complete successfully.</li> </ul>	<ul style="list-style-type: none"> <li>Take measures to reduce noise.</li> <li>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.</li> </ul>	Failure information	At instruction execution

## Error codes detected by other causes than the self-diagnostics (4000H to 4FFFH)

The following table lists the error codes detected by other causes than the self-diagnostics function of the CPU module.

### ■ Error codes returned to request source at communication with CPU module

- Errors generated when the data communications are requested from the engineering tool, intelligent function module, or network system connected
- Errors generated with the data logging function

Because the following errors are not detected by the self-diagnostics function, the error codes are not stored in SD0.

Error code	Error name	Error details and cause	Action
4000H	Common error	• Serial communication sum check error	<ul style="list-style-type: none"> <li>• Connect the serial communication cable correctly.</li> <li>• Take measures to reduce noise.</li> </ul>
4001H	Common error	• An unsupported request was executed. (The request was executed to the CPU module that does not support the request.)	<ul style="list-style-type: none"> <li>• Check the command data of the SLMP/MC protocol.</li> <li>• Check the CPU module model name selected in the engineering tool.</li> <li>• Check the target CPU module model name.</li> </ul>
4002H	Common error	• An unsupported request was executed.	<ul style="list-style-type: none"> <li>• Check the command data of the SLMP/MC protocol.</li> <li>• Check the CPU module model name selected in the engineering tool.</li> <li>• Execute the request again.</li> <li>• 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.</li> </ul>
4003H	Common error	• Command for which a global request cannot be performed was executed.	<ul style="list-style-type: none"> <li>• Check the command data of the SLMP/MC protocol.</li> </ul>
4004H	Common error	<ul style="list-style-type: none"> <li>• Any operation for the CPU module is prohibited by the system protect function provided against the following events.</li> <li>• The CPU module is starting up.</li> </ul>	<ul style="list-style-type: none"> <li>• Perform operation again after the CPU module has completed the startup processing.</li> </ul>
4005H	Common error	• The volume of data handled according to the specified request is too large.	<ul style="list-style-type: none"> <li>• Check the command data of the SLMP/MC protocol.</li> </ul>
4006H	Common error	• Initial communication has failed.	<ul style="list-style-type: none"> <li>• When using serial communication, inquire of the external device manufacturer for support conditions.</li> <li>• When using serial communication, check the CPU module model name selected in the engineering tool.</li> <li>• When using Ethernet communication, shift the communication start timing.</li> </ul>
4008H	Common error	• The CPU module is BUSY. (The buffer is not vacant.)	<ul style="list-style-type: none"> <li>• Execute the request again after the specified period of time has elapsed.</li> </ul>
4010H	CPU module operation error	• Since the CPU module is running, the request contents cannot be executed.	<ul style="list-style-type: none"> <li>• Execute after setting the CPU module to STOP status.</li> </ul>
4013H	CPU module operation error	• Since the CPU module is not in a STOP status, the request contents cannot be executed.	<ul style="list-style-type: none"> <li>• Execute after setting the CPU module to STOP status.</li> </ul>
4021H	File related error	• The specified drive (memory) does not exist or there is an error.	<ul style="list-style-type: none"> <li>• Check the specified drive (memory) status.</li> <li>• Back up data in the CPU module, and then initialize the memory.</li> </ul>
4022H	File related error	<ul style="list-style-type: none"> <li>• The file with the specified file name or file No. does not exist.</li> <li>■When CPU Module Logging Configuration Tool is used</li> <li>• The data logging specifying a program name that does not exist was started.</li> </ul>	<ul style="list-style-type: none"> <li>• Check the specified file name and file No.</li> </ul>
4023H	File related error	• The file name and file No. of the specified file do not match.	<ul style="list-style-type: none"> <li>• Delete the file and then recreate the file.</li> </ul>
4024H	File related error	• The specified file cannot be handled by a user.	<ul style="list-style-type: none"> <li>• Do not access the specified file.</li> </ul>
4025H	File related error	• The specified file is processing the request from another engineering tool.	<ul style="list-style-type: none"> <li>• Forcibly execute the request. Or, execute the request again after the processing being performed ends.</li> </ul>
4026H	File related error	• The file password set in advance to the target drive (memory) must be specified.	<ul style="list-style-type: none"> <li>• Specify the file password set in advance, and then access to the drive (memory).</li> </ul>
4027H	File related error	• The specified range is larger than the file size.	<ul style="list-style-type: none"> <li>• Check the specified range and access within that range.</li> </ul>
4028H	File related error	• The same file already exists.	<ul style="list-style-type: none"> <li>• Forcibly execute the request. Or, change the file name and execute the request again.</li> </ul>

Error code	Error name	Error details and cause	Action
4029H	File related error	<ul style="list-style-type: none"> <li>The specified file capacity cannot be obtained.</li> </ul>	<ul style="list-style-type: none"> <li>Review the specified file capacity, and execute the request again.</li> </ul>
402AH	File related error	<ul style="list-style-type: none"> <li>The specified file is abnormal.</li> </ul>	<ul style="list-style-type: none"> <li>Back up data in the CPU module, and then initialize the memory.</li> </ul>
402BH	File related error	<ul style="list-style-type: none"> <li>The request contents cannot be executed in the specified drive (memory).</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after setting the CPU module to STOP status.</li> </ul>
402CH	File related error	<ul style="list-style-type: none"> <li>The requested operation cannot be executed currently.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after a while.</li> </ul>
4030H	Device specification error	<ul style="list-style-type: none"> <li>The specified device name cannot be handled.</li> </ul>	<ul style="list-style-type: none"> <li>Check the specified device name.</li> </ul>
		<ul style="list-style-type: none"> <li>■When CPU Module Logging Configuration Tool is used</li> <li>The data logging specifying a device that is not supported was started.</li> </ul>	
4031H	Device specification error	<ul style="list-style-type: none"> <li>The specified device No. is outside the range.</li> <li>The CPU module cannot handle the specified device.</li> </ul>	<ul style="list-style-type: none"> <li>Check the specified device No.</li> <li>Check the device assignment of the CPU module.</li> <li>Check the specified device name.</li> </ul>
		<ul style="list-style-type: none"> <li>■When CPU Module Logging Configuration Tool is used</li> <li>The data logging specifying a device number that does not exist was started.</li> </ul>	
4032H	Device specification error	<ul style="list-style-type: none"> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Check the device modification method.</li> <li>Check the specified device name.</li> </ul>
		<ul style="list-style-type: none"> <li>■When CPU Module Logging Configuration Tool is used</li> <li>The data logging specifying a device modification that is not supported was started.</li> </ul>	
4033H	Device specification error	<ul style="list-style-type: none"> <li>Writing cannot be done because the specified device is for system use.</li> </ul>	<ul style="list-style-type: none"> <li>Do not write the data in the specified device, and do not turn on or off.</li> </ul>
4034H	Device specification error	<ul style="list-style-type: none"> <li>The dedicated instruction cannot be executed since the completion device for the dedicated instruction does not turn on.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>
4040H	Intelligent function module specification error	<ul style="list-style-type: none"> <li>The request contents cannot be executed in the specified intelligent function module.</li> </ul>	<ul style="list-style-type: none"> <li>Check whether the specified module is the intelligent function module having the buffer memory.</li> </ul>
4041H	Intelligent function module specification error	<ul style="list-style-type: none"> <li>The access range exceeds the buffer memory range of the specified intelligent function module.</li> </ul>	<ul style="list-style-type: none"> <li>Check the start address and access number of points and access using a range that exists in the intelligent function module.</li> </ul>
4042H	Intelligent function module specification error	<ul style="list-style-type: none"> <li>The specified intelligent function module cannot be accessed.</li> </ul>	<ul style="list-style-type: none"> <li>Check that the specified intelligent function module is operating normally.</li> <li>Check the specified module for a hardware fault.</li> </ul>
4043H	Intelligent function module specification error	<ul style="list-style-type: none"> <li>The intelligent function module does not exist in the specified position.</li> </ul>	<ul style="list-style-type: none"> <li>Check the I/O number of the specified intelligent function module.</li> </ul>
		<ul style="list-style-type: none"> <li>■When CPU Module Logging Configuration Tool is used</li> <li>The data logging specifying a device that does not exist or cannot be accessed was started.</li> </ul>	
4044H	Intelligent function module specification error	<ul style="list-style-type: none"> <li>A control bus error occurred during access to the intelligent function module.</li> </ul>	<ul style="list-style-type: none"> <li>Check the specified intelligent function module and other modules and base units for a hardware fault.</li> </ul>
4049H	Intelligent function module specification error	<ul style="list-style-type: none"> <li>A request cannot be processed because the module extension parameter of the positioning module is being used for the positioning control.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>

Error code	Error name	Error details and cause	Action
4050H	Protect error	<ul style="list-style-type: none"> <li>The request contents cannot be executed because the write protect switch of the SD memory card is on.</li> </ul>	<ul style="list-style-type: none"> <li>Turn off the write protect switch.</li> </ul>
4051H	Protect error	<ul style="list-style-type: none"> <li>The specified drive (memory) cannot be accessed.</li> </ul>	Check the following and take action. <ul style="list-style-type: none"> <li>Is the drive (memory) the one that can be used?</li> <li>Is the specified drive (memory) correctly installed?</li> </ul>
4052H	Protect error	<ul style="list-style-type: none"> <li>The specified file attribute is read only, so the data cannot be written.</li> </ul>	<ul style="list-style-type: none"> <li>Do not write data in the specified file. Or, change the file attribute.</li> </ul>
4053H	Protect error	<ul style="list-style-type: none"> <li>An error occurred when writing data to the specified drive (memory).</li> </ul>	<ul style="list-style-type: none"> <li>Check the specified drive (memory). Or, write data again after changing the corresponding drive (memory).</li> </ul>
4054H	Protect error	<ul style="list-style-type: none"> <li>An error occurred when deleting the data in the specified drive (memory).</li> </ul>	<ul style="list-style-type: none"> <li>Check the specified drive (memory). Or, delete data again after replacing the specified drive (memory).</li> </ul>
4060H	Online registration error	<ul style="list-style-type: none"> <li>The online debug function (such as online change) and the data logging function are being executed with another engineering tool.</li> <li>■When CPU Module Logging Configuration Tool is used               <ul style="list-style-type: none"> <li>An attempt was made to write or delete data logging settings or to execute data logging to the setting registered by another request source.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Finish the operation of another engineering tool and then execute the function again.</li> <li>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.</li> </ul>
4061H	Online registration error	<ul style="list-style-type: none"> <li>Settings for the online debug function (such as online change) are incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>Register an online debug function (such as online change), and then execute the function.</li> <li>Execute again after checking the communication route such as the communication cable.</li> </ul>
4063H	Online registration error	<ul style="list-style-type: none"> <li>The registered number of locked files exceeded the maximum value.</li> </ul>	<ul style="list-style-type: none"> <li>Finish the file access from another engineering tool, and then execute again.</li> </ul>
4064H	Online registration error	<ul style="list-style-type: none"> <li>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.</li> <li>■When CPU Module Logging Configuration Tool is used               <ul style="list-style-type: none"> <li>The trigger logging was started in a state that the trigger condition has already been satisfied.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>Execute again after checking the communication route such as the communication cable.</li> <li>■When CPU Module Logging Configuration Tool is used               <ul style="list-style-type: none"> <li>Clear the satisfied trigger condition, and execute the trigger logging again.</li> </ul> </li> </ul>
4065H	Online registration error	<ul style="list-style-type: none"> <li>The device assignment information differs from the parameter.</li> </ul>	<ul style="list-style-type: none"> <li>Check the device assignment of the CPU module or the device assignment of the request data.</li> </ul>
4066H	Online registration error	<ul style="list-style-type: none"> <li>The specified file password is incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>Check and specify the correct file password.</li> </ul>
4067H	Online registration error	<ul style="list-style-type: none"> <li>Monitor communication has failed.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the communication route such as the communication cable.</li> </ul>
4068H	Online registration error	<ul style="list-style-type: none"> <li>Operation is disabled because it is being performed with another engineering tool.</li> </ul>	<ul style="list-style-type: none"> <li>Finish the operation of another engineering tool and then execute again.</li> </ul>
406AH	Online registration error	<ul style="list-style-type: none"> <li>The drive (memory) number other than 0 to 4 was specified.</li> </ul>	<ul style="list-style-type: none"> <li>Check the drive and specify the drive number correctly.</li> </ul>
406BH	Online registration error	<ul style="list-style-type: none"> <li>Online operation was interrupted due to a CPU module error.</li> </ul>	<ul style="list-style-type: none"> <li>Check the status of the CPU module by executing the module diagnostics.</li> <li>Identify the error, and take a corrective action referring to the troubleshooting section.</li> </ul>
4070H	Circuit verification error	<ul style="list-style-type: none"> <li>The program not yet corrected and the one corrected by online program change are different.</li> </ul>	<ul style="list-style-type: none"> <li>Read the program from the CPU module to match it with that of the engineering tool, and then execute online change again.</li> </ul>
4080H	Other errors	<ul style="list-style-type: none"> <li>Request data error</li> <li>■When CPU Module Logging Configuration Tool is used               <ul style="list-style-type: none"> <li>Request or setting data error</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Check the request data that has been specified.</li> <li>■When CPU Module Logging Configuration Tool is used               <ul style="list-style-type: none"> <li>Check the specified data, and write it to the CPU module again.</li> </ul> </li> </ul>
4081H	Other errors	<ul style="list-style-type: none"> <li>The search target data cannot be detected.</li> </ul>	<ul style="list-style-type: none"> <li>Check the data to be searched.</li> </ul>
4082H	Other errors	<ul style="list-style-type: none"> <li>The specified command cannot be executed because it is being executed.</li> </ul>	<ul style="list-style-type: none"> <li>Complete the processing for a request from another engineering tool and then execute the command again.</li> </ul>

Error code	Error name	Error details and cause	Action
4083H	Other errors	<ul style="list-style-type: none"> <li>An attempt was made to perform operation to the program not registered in parameter.</li> <li>■When CPU Module Logging Configuration Tool is used <ul style="list-style-type: none"> <li>The data logging specifying the program that is stored in the CPU module but not registered in the CPU parameters was started.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Register the program in parameter.</li> </ul>
4084H	Other errors	<ul style="list-style-type: none"> <li>The specified pointer (P or I) does not exist.</li> </ul>	<ul style="list-style-type: none"> <li>Check if the pointer (P or I) exist in the data.</li> </ul>
4085H	Other errors	<ul style="list-style-type: none"> <li>The pointer (P or I) cannot be specified because the program is not specified in parameter.</li> </ul>	<ul style="list-style-type: none"> <li>Register the program to be executed in parameter first, and specify the pointer (P or I).</li> </ul>
4086H	Other errors	<ul style="list-style-type: none"> <li>The specified pointer (P or I) has already been added.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the pointer number to be added.</li> </ul>
4087H	Other errors	<ul style="list-style-type: none"> <li>The number of pointers (P or I) exceeds its limit.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the specified pointer (P or I).</li> </ul>
4088H	Other errors	<ul style="list-style-type: none"> <li>The specified step number is not at the head of the instruction.</li> <li>The program differs from that stored in the CPU module.</li> <li>■When CPU Module Logging Configuration Tool is used <ul style="list-style-type: none"> <li>The data logging specifying the step number that does not exist or is not specified as the start number of the instruction was started.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the specified step number.</li> <li>Read the program from the CPU module to match it with that of the engineering tool, and then execute online change again.</li> </ul>
4089H	Other errors	<ul style="list-style-type: none"> <li>An attempt was made to insert/delete the END instruction by online program change.</li> </ul>	<ul style="list-style-type: none"> <li>Check the specified program file contents.</li> <li>Write the program after setting the CPU module to the STOP status.</li> </ul>
408AH	Other errors	<ul style="list-style-type: none"> <li>The file capacity exceeded after the online change was executed.</li> </ul>	<ul style="list-style-type: none"> <li>Check the capacity of the specified program file.</li> <li>Write the program after setting the CPU module to the STOP status.</li> </ul>
408BH	Other errors	<ul style="list-style-type: none"> <li>The remote request cannot be executed.</li> </ul>	<ul style="list-style-type: none"> <li>Reexecute after the CPU module is in a status where the mode request can be executed.</li> <li>For remote operation, set the parameter to "Enable remote reset".</li> </ul>
408DH	Other errors	<ul style="list-style-type: none"> <li>The instruction code that cannot be handled exists.</li> </ul>	<ul style="list-style-type: none"> <li>Check whether the model of the used CPU module is correct or not.</li> <li>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.</li> </ul>
408EH	Other errors	<ul style="list-style-type: none"> <li>The write step is illegal.</li> <li>The program differs from that stored in the CPU module.</li> </ul>	<ul style="list-style-type: none"> <li>Write the program after setting the CPU module to the STOP status.</li> <li>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.</li> <li>Read the program from the CPU module to match it with that of the engineering tool, and then execute online program change again.</li> </ul>
40C0H	Label communication error	<ul style="list-style-type: none"> <li>The specified label name does not exist.</li> </ul>	<ul style="list-style-type: none"> <li>Check the label setting. If the "Access from External Device" checkbox is not selected, check the checkbox.</li> </ul>
40C1H	Label communication error	<ul style="list-style-type: none"> <li>Label access with a label name has failed because the specified array element number is out of range.</li> </ul>	<ul style="list-style-type: none"> <li>Specify the array element number within the set range.</li> </ul>
40C2H	Label communication error	<ul style="list-style-type: none"> <li>Label access with a label name has failed because the bit array type label is not specified by bit specification.</li> </ul>	<ul style="list-style-type: none"> <li>Specify the label by bit specification, and access again.</li> </ul>
40C3H	Label communication error	<ul style="list-style-type: none"> <li>Label access with a label name has failed because the word array type label is not specified by word specification.</li> </ul>	<ul style="list-style-type: none"> <li>Specify the label by word specification, and access again.</li> </ul>
40C4H	Label communication error	<ul style="list-style-type: none"> <li>Label access with a label name has failed because the number of labels used in the program exceeded its limit.</li> </ul>	<ul style="list-style-type: none"> <li>Reduce the number of labels in the program, and access the labels multiple times.</li> </ul>
40C8H	Label communication error	<ul style="list-style-type: none"> <li>Registration of a label definition has failed because the number of registered labels reached its limit.</li> </ul>	<ul style="list-style-type: none"> <li>Reduce the number of registered labels with the "Access from External Device" checkbox checked.</li> </ul>

Error code	Error name	Error details and cause	Action
40C9H	Label communication error	<ul style="list-style-type: none"> <li>Registration of a label definition has failed because the registration target memory capacity reached its limit.</li> </ul>	<ul style="list-style-type: none"> <li>Reduce the number of registered labels with the "Access from External Device" checkbox checked.</li> <li>If the data memory is specified as the registration target memory, change the memory to the SD memory card.</li> <li>Change the settings of the functions that use the registration target memory.</li> </ul>
40CAH	Label communication error	<ul style="list-style-type: none"> <li>Labels could not be changed, added, nor deleted because the label communication data do not exist in the CPU module.</li> <li>■When CPU Module Logging Configuration Tool is used</li> <li>An attempt was made to change, add, or delete labels without creating the label communication data.</li> </ul>	<ul style="list-style-type: none"> <li>Write the label communication data to the programmable controller.</li> </ul>
40CBH	Label communication error	<ul style="list-style-type: none"> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>
40CCH	Label communication error	<ul style="list-style-type: none"> <li>Online program change has failed because the global setting file before modification and the global label assignment information do not match.</li> </ul>	<ul style="list-style-type: none"> <li>Write the global setting file and the global label assignment information to the programmable controller together after modification.</li> </ul>
40CEH	Label communication error	<ul style="list-style-type: none"> <li>An attempt was made to access a label that cannot be accessed with a label name.</li> </ul>	<ul style="list-style-type: none"> <li>Change the data type of the specified label to the one other than "Function Block" or "Pointer".</li> <li>Change the class of the specified label to the one other than "VAR_GLOBAL_CONSTANT".</li> <li>Specify a device that is not being modified. (Bit-specified word devices and digit-specified bit devices can be specified.)</li> </ul>
4100H	Other errors	<ul style="list-style-type: none"> <li>Hardware failure of the CPU module.</li> </ul>	<ul style="list-style-type: none"> <li>Replace the CPU module.</li> </ul>
4101H	Other errors	<ul style="list-style-type: none"> <li>Serial communication connection was executed for a different CPU module series.</li> </ul>	<ul style="list-style-type: none"> <li>Check the CPU module series.</li> </ul>
4103H	Other errors	<ul style="list-style-type: none"> <li>The instruction written by online program change is incorrect or invalid.</li> </ul>	<ul style="list-style-type: none"> <li>Execute online program change again, or write the program after setting the CPU module to the STOP status.</li> </ul>
4108H	Other errors	<ul style="list-style-type: none"> <li>The device monitor/test cannot be performed normally.</li> </ul>	<ul style="list-style-type: none"> <li>Execute the function again. Check that the access prohibited area is not accessed, and execute the function again.</li> </ul>
410AH	Other errors	<ul style="list-style-type: none"> <li>The specified command cannot be executed because the online program change is being executed.</li> <li>■When CPU Module Logging Configuration Tool is used</li> <li>The data logging specifying the step number as a sampling or trigger condition was started during online program change.</li> </ul>	<ul style="list-style-type: none"> <li>Execute the request again after the online program change.</li> </ul>
410BH	Other errors	<ul style="list-style-type: none"> <li>The monitor condition registration was cleared after the online program change was executed.</li> </ul>	<ul style="list-style-type: none"> <li>Execute the registration of monitoring condition again after the online program change.</li> </ul>
410CH	Other errors	<ul style="list-style-type: none"> <li>Writing to the specified data is not supported.</li> </ul>	<ul style="list-style-type: none"> <li>Check that the version of the engineering tool used is correct.</li> <li>Check the settings and make a correction.</li> </ul>
410EH	Other errors	<ul style="list-style-type: none"> <li>When the execution status of the online program change is in error, the online program change command was issued.</li> </ul>	<ul style="list-style-type: none"> <li>Issue the online program change cancel command, and execute the function again.</li> </ul>
410FH	Other errors	<ul style="list-style-type: none"> <li>During the execution of the online program change function, the cancel command was issued by the same request source.</li> </ul>	<ul style="list-style-type: none"> <li>Issue the command again after the currently-performed processing ends.</li> </ul>
4110H	CPU module error	<ul style="list-style-type: none"> <li>Since the CPU module is in a stop error status, it cannot execute the request.</li> </ul>	<ul style="list-style-type: none"> <li>Execute the request again after resetting the CPU module.</li> </ul>
4111H	CPU module error	<ul style="list-style-type: none"> <li>The requested operation cannot be performed since the other CPU modules have not yet started in the multiple CPU system.</li> </ul>	<ul style="list-style-type: none"> <li>Execute the request again after all the other CPU modules have started.</li> </ul>
4121H	File related error	<ul style="list-style-type: none"> <li>The specified drive (memory) or file does not exist.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the specified drive (memory) or file.</li> </ul>
4122H	File related error	<ul style="list-style-type: none"> <li>The specified drive (memory) or file does not exist.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the specified drive (memory) or file.</li> </ul>
4123H	File related error	<ul style="list-style-type: none"> <li>The specified drive (memory) is abnormal.</li> <li>■When CPU Module Logging Configuration Tool is used</li> <li>The data logging was started to the memory having an error.</li> </ul>	<ul style="list-style-type: none"> <li>Initialize the memory, and restore the drive (memory) to its normal state.</li> </ul>
4124H	File related error	<ul style="list-style-type: none"> <li>The specified drive (memory) is abnormal.</li> </ul>	<ul style="list-style-type: none"> <li>Initialize the memory, and restore the drive (memory) to its normal state.</li> </ul>

Error code	Error name	Error details and cause	Action
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.	• Execute again after a while.
		■When CPU Module Logging Configuration Tool is used • An operation was performed to a file being accessed.	
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).



Error code	Error name	Error details and cause	Action
414AH	Intelligent function module specification error	<ul style="list-style-type: none"> <li>Operation was executed from the CPU module other than the control CPU to the intelligent function module in the multiple CPU system.</li> </ul>	<ul style="list-style-type: none"> <li>Execute the operation again from the control CPU of the target module.</li> </ul>
414CH	Intelligent function module specification error	<ul style="list-style-type: none"> <li>The specified buffer memory address cannot be accessed.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the buffer address.</li> </ul>
4150H	File related error	<ul style="list-style-type: none"> <li>An attempt was made to initialize the drive (memory) protected by the system.</li> </ul>	<ul style="list-style-type: none"> <li>Do not initialize the target drive (memory) as it cannot be initialized.</li> </ul>
4151H	File related error	<ul style="list-style-type: none"> <li>An attempt was made to delete the file/folder protected by the system.</li> </ul>	<ul style="list-style-type: none"> <li>Do not delete the target file as it cannot be deleted.</li> </ul>
41C1H	File related error	<ul style="list-style-type: none"> <li>The format information data of the specified drive (memory) is abnormal.</li> </ul>	<ul style="list-style-type: none"> <li>The file information data may be corrupted.</li> <li>Back up data in the CPU module, and then initialize the memory.</li> </ul>
41C2H	File related error	<ul style="list-style-type: none"> <li>File open specification data for file access is wrong.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the specification data.</li> </ul>
41C3H	File related error	<ul style="list-style-type: none"> <li>Simultaneously accessible files exceeded the maximum.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after decreasing file operations.</li> </ul>
41C4H	File related error	<ul style="list-style-type: none"> <li>Simultaneously accessible files exceeded the maximum.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after decreasing file operations.</li> </ul>
41C5H	File related error	<ul style="list-style-type: none"> <li>The specified file does not exist.</li> <li>■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.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the file.</li> </ul>
41C7H	File related error	<ul style="list-style-type: none"> <li>The specified file or drive (memory) does not exist.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the file or drive (memory).</li> </ul>
41C8H	File related error	<ul style="list-style-type: none"> <li>The size of the specified file has exceeded that of the existing file.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the size of the specified file.</li> <li>If the error recurs after re-execution, the file information data may be corrupted.</li> <li>Back up data in the CPU module, and then initialize the memory.</li> </ul>
41C9H	File related error	<ul style="list-style-type: none"> <li>Access to the file sector has failed.</li> <li>The format information data of the target drive (memory) is abnormal.</li> </ul>	<ul style="list-style-type: none"> <li>Back up data in the CPU module, and then initialize the memory.</li> </ul>
41CAH	File related error	<ul style="list-style-type: none"> <li>Access to the file sector has failed.</li> <li>The format information data of the target drive (memory) is abnormal.</li> </ul>	<ul style="list-style-type: none"> <li>Back up data in the CPU module, and then initialize the memory.</li> </ul>
41CBH	File related error	<ul style="list-style-type: none"> <li>The file name is specified in a wrong method.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the file name.</li> </ul>
41CCH	File related error	<ul style="list-style-type: none"> <li>The specified file does not exist. Or, the specified subdirectory does not exist.</li> <li>■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.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the name of the file and subdirectory.</li> </ul>
41CDH	File related error	<ul style="list-style-type: none"> <li>An access to the file is prohibited in the system.</li> <li>■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.</li> </ul>	<ul style="list-style-type: none"> <li>Do not access the specified file or subdirectory.</li> <li>Execute again after checking the file and subdirectory.</li> </ul>
41CEH	File related error	<ul style="list-style-type: none"> <li>The file cannot be written because the specified file is read-only.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the attribute of the specified file.</li> </ul>
41CFH	File related error	<ul style="list-style-type: none"> <li>The specified drive (memory) has been used exceeding the capacity.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the drive (memory) capacity.</li> </ul>

Error code	Error name	Error details and cause	Action
41D0H	File related error	<ul style="list-style-type: none"> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after increasing the free space of the drive (memory).</li> <li>Delete files in the drive (memory), and execute the function again.</li> </ul>
41D1H	File related error	<ul style="list-style-type: none"> <li>The file name is specified in a wrong method.</li> <li>The SD memory card is being disabled by SM606 (SD memory card forced disable instruction).</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the file name.</li> <li>If the error recurs after re-execution, the file information data may be corrupted.</li> <li>Back up data in the CPU module, and then initialize the memory.</li> <li>Cancel the SD memory card forced disable instruction.</li> </ul>
41D5H	File related error	<ul style="list-style-type: none"> <li>The file of the same name exists.</li> </ul>	<ul style="list-style-type: none"> <li>Forcibly execute the request, or execute after changing the file name.</li> </ul>
41D6H	File related error	<ul style="list-style-type: none"> <li>The format information data of the specified drive (memory) is abnormal.</li> </ul>	<ul style="list-style-type: none"> <li>The file information data may be corrupted.</li> <li>Back up data in the CPU module, and then initialize the memory.</li> </ul>
41D7H	File related error	<ul style="list-style-type: none"> <li>The format information data of the specified drive (memory) is abnormal.</li> </ul>	<ul style="list-style-type: none"> <li>The file information data may be corrupted.</li> <li>Back up data in the CPU module, and then initialize the memory.</li> </ul>
41D8H	File related error	<ul style="list-style-type: none"> <li>The specified file is being accessed.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after a while.</li> </ul>
41DFH	File related error	<ul style="list-style-type: none"> <li>The specified drive (memory) is write-protected.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after canceling the write protect of the specified drive (memory).</li> </ul>
41E0H	File related error	<ul style="list-style-type: none"> <li>The specified drive (memory) is abnormal or does not exist.</li> </ul>	<ul style="list-style-type: none"> <li>Execute the operation again after checking that the SD memory card has been inserted.</li> <li>Back up data, and then initialize the memory.</li> </ul>
41E1H	File related error	<ul style="list-style-type: none"> <li>Access to the SD memory card has failed.</li> </ul>	<ul style="list-style-type: none"> <li>Back up data, and then write the data to the data memory.</li> </ul>
41E4H	File related error	<ul style="list-style-type: none"> <li>Access to the SD memory card has failed.</li> </ul>	<ul style="list-style-type: none"> <li>Execute the operation again after checking that the SD memory card has been inserted.</li> <li>Execute the operation again after replacing the SD memory card.</li> <li>Back up data, and then initialize the memory.</li> </ul>
41E7H	File related error	<ul style="list-style-type: none"> <li>The format information data of the specified drive (memory) is abnormal.</li> </ul>	<ul style="list-style-type: none"> <li>The file information data may be corrupted.</li> <li>Back up data in the CPU module, and then initialize the memory.</li> </ul>
41E8H	File related error	<ul style="list-style-type: none"> <li>The format information data of the specified drive (memory) is abnormal.</li> </ul>	<ul style="list-style-type: none"> <li>The file information data may be corrupted.</li> <li>Back up data in the CPU module, and then initialize the memory.</li> </ul>
41E9H	File related error	<ul style="list-style-type: none"> <li>The specified file is being accessed.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after a while.</li> </ul>
41EBH	File related error	<ul style="list-style-type: none"> <li>The file name is specified in a wrong method.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after checking the file name.</li> </ul>
41ECH	File related error	<ul style="list-style-type: none"> <li>The file system of the specified drive (memory) is logically corrupted.</li> </ul>	<ul style="list-style-type: none"> <li>The file information data may be corrupted.</li> <li>Back up data in the CPU module, and then initialize the memory.</li> </ul>
41EDH	File related error	<ul style="list-style-type: none"> <li>The specified drive (memory) does not have continuous free space. (The free space for file is sufficient but the continuous free space is insufficient.)</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after deleting unnecessary files.</li> </ul>
41EFH	File related error	<ul style="list-style-type: none"> <li>Creation of the power failure backup data in the specified drive (memory) has failed.</li> </ul>	<ul style="list-style-type: none"> <li>Execute the operation again after checking that the SD memory card has been inserted.</li> </ul>
41F0H	File related error	<ul style="list-style-type: none"> <li>The power failure backup data of the specified drive (memory) are corrupted.</li> </ul>	<ul style="list-style-type: none"> <li>Execute the operation again after checking that the SD memory card has been inserted.</li> </ul>
41F3H	File related error	<ul style="list-style-type: none"> <li>The file size is larger than the following:</li> <li>the value to be acquired when 2 bytes are subtracted from 4G bytes.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>
41F4H	File related error	<ul style="list-style-type: none"> <li>Since the operation prohibited by the system is performed, the requested processing cannot be performed.</li> </ul>	<ul style="list-style-type: none"> <li>Do not request the file operation because it is prohibited by the system.</li> </ul>
41F5H	File related error	<ul style="list-style-type: none"> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Check the command, and request the command applicable to the target file.</li> </ul>

Error code	Error name	Error details and cause	Action
41F8H	File related error	<ul style="list-style-type: none"> <li>The data is being accessed from another engineering tool.</li> </ul>	<ul style="list-style-type: none"> <li>Data are being written to the program memory or being transferred to the backup memory.</li> </ul> Access the file after the currently-performed processing ends.
41FAH	File related error	<ul style="list-style-type: none"> <li>Program was written beyond the area where the program can be executed.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after reducing either the already written program or newly written program.</li> </ul>
41FBH	File related error	<ul style="list-style-type: none"> <li>The specified file is already being manipulated from the engineering tool.</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after the currently performed operation is completed.</li> </ul>
41FCH	File related error	<ul style="list-style-type: none"> <li>An attempt was made to initialize the drive (memory) being used.</li> </ul>	<ul style="list-style-type: none"> <li>Stop all accesses to the specified drive (memory), and execute the request again.</li> </ul>
41FDH	File related error	<ul style="list-style-type: none"> <li>There are no data written to the data memory.</li> </ul>	<ul style="list-style-type: none"> <li>Write all the required files to the programmable controller.</li> </ul>
41FEH	File related error	<ul style="list-style-type: none"> <li>The SD memory card has not been inserted.</li> <li>The SD memory card is being disabled.</li> <li>The SD memory card is being disabled by SM606 (SD memory card forced disable instruction).</li> </ul> <p>■When CPU Module Logging Configuration Tool is used</p> <ul style="list-style-type: none"> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Insert the SD memory card.</li> <li>Remove the SD memory card, and insert it again.</li> <li>Cancel the SD memory card forced disable instruction.</li> </ul>
41FFH	File related error	<ul style="list-style-type: none"> <li>The type of the SD memory card differs.</li> </ul>	<ul style="list-style-type: none"> <li>Check the type of the SD memory card.</li> </ul>
4269H	Other errors	<ul style="list-style-type: none"> <li>The remote RUN (function) cannot be executed.</li> </ul>	<ul style="list-style-type: none"> <li>Execute the function again after a while.</li> </ul>
4270H	Debug function error	<ul style="list-style-type: none"> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Register data logging to the memory where the data logging is being performed. Or, stop the data logging being performed and register again.</li> </ul> <p>■When CPU Module Logging Configuration Tool is used</p> <ul style="list-style-type: none"> <li>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.</li> </ul>
4271H	Debug function error	<ul style="list-style-type: none"> <li>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).</li> </ul>	<ul style="list-style-type: none"> <li>Stop the data logging. Or, write, delete, or register data logging to the setting number where no data logging is being performed.</li> </ul>
4272H	Debug function error	<ul style="list-style-type: none"> <li>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).</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul> <p>■When CPU Module Logging Configuration Tool is used</p> <ul style="list-style-type: none"> <li>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.</li> </ul>
4275H	Debug function error	<ul style="list-style-type: none"> <li>Auto logging is being performed.</li> </ul>	<ul style="list-style-type: none"> <li>After the auto logging, replace the SD memory card and execute again.</li> </ul>
4276H	Debug function error	<ul style="list-style-type: none"> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Stop the data logging, and then execute the function.</li> </ul>

Error code	Error name	Error details and cause	Action
4277H	Debug function error	<ul style="list-style-type: none"> <li>The number of stored files has exceeded the value set in advance.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>
		<p>■When CPU Module Logging Configuration Tool is used</p> <ul style="list-style-type: none"> <li>The data logging was started in a state that the number of saved files has exceeded the specified number. (The operation in the case of exceeding the number of saved files is set to "Stop".) The data logging was started in a state that the number of saved files has exceeded the specified number. (The operation in the case of exceeding the number of saved files is set to "Overwrite".)</li> </ul>	<p>■When CPU Module Logging Configuration Tool is used</p> <ul style="list-style-type: none"> <li>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.</li> </ul>
4278H	Debug function error	<ul style="list-style-type: none"> <li>An attempt was made to register data logging in a state that the stored file number has reached to the maximum (FFFFFFFF). Or, the number reached to the maximum during the execution.</li> </ul>	<ul style="list-style-type: none"> <li>The stored file number in the storage target memory has reached to the maximum of FFFFFFFF. Delete files, or change the storage destination and then register.</li> </ul>
		<p>■When CPU Module Logging Configuration Tool is used</p> <ul style="list-style-type: none"> <li>The data logging was started in a state that the stored file number has reached to the maximum (FFFFFFFF). Or, the number reached to the maximum during the execution.</li> </ul>	<p>■When CPU Module Logging Configuration Tool is used</p> <ul style="list-style-type: none"> <li>The stored file number in the storage target memory has reached to the maximum of FFFFFFFF. Delete files or change the storage target memory, and then perform the data logging.</li> </ul>
427AH	Debug function error	<ul style="list-style-type: none"> <li>Common setting file does not exist.</li> </ul>	<ul style="list-style-type: none"> <li>Write the common settings to the target memory.</li> <li>Register data logging to the memory where the common settings are stored.</li> </ul>
		<p>■When CPU Module Logging Configuration Tool is used</p> <ul style="list-style-type: none"> <li>The data logging was started to the memory where the common settings are not stored.</li> </ul>	<p>■When CPU Module Logging Configuration Tool is used</p> <ul style="list-style-type: none"> <li>Write the common settings to the target memory.</li> <li>Start the data logging to the memory where the common settings are stored.</li> </ul>
427BH	Debug function error	<ul style="list-style-type: none"> <li>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).</li> </ul>	<ul style="list-style-type: none"> <li>Stop the data logging destined for the same storage, and then register. Or, change the storage destination of the file, and then register.</li> </ul>
		<p>■When CPU Module Logging Configuration Tool is used</p> <ul style="list-style-type: none"> <li>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).</li> </ul>	<p>■When CPU Module Logging Configuration Tool is used</p> <ul style="list-style-type: none"> <li>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.</li> </ul>
4282H	Debug function error	<ul style="list-style-type: none"> <li>The registration was performed with the internal buffer capacity set to 0.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the internal buffer capacity setting.</li> </ul>
4283H	Debug function error	<ul style="list-style-type: none"> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the internal buffer capacity setting.</li> <li>Reduce the number of records before trigger.</li> </ul>
4284H	Debug function error	<ul style="list-style-type: none"> <li>The error codes registered for memory dump are incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the error codes.</li> </ul>
4285H	Debug function error	<ul style="list-style-type: none"> <li>A non-executable function has been executed during collection or save in memory dump.</li> </ul>	<ul style="list-style-type: none"> <li>Execute the function again after the completion of save in memory dump.</li> </ul>
4291H	Debug function error	<ul style="list-style-type: none"> <li>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".</li> </ul>	<ul style="list-style-type: none"> <li>After the completion of the online program change, check the step number specified and execute real-time monitor again.</li> </ul>
4292H	Debug function error	<ul style="list-style-type: none"> <li>Real-time monitor does not begin during the execution of real-time monitor.</li> </ul>	<ul style="list-style-type: none"> <li>Start real-time monitor after the stop of the real-time monitor being executed.</li> </ul>
4293H	Debug function error	<ul style="list-style-type: none"> <li>Execution fails because the internal buffer exceeds its maximum capacity.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the settings of internal buffer capacity, and then try again.</li> </ul>
4294H	Debug function error	<ul style="list-style-type: none"> <li>Execution fails because the settings of internal buffer capacity has been changed during the execution of functions that consume the internal buffer.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>
433CH	Maintenance and inspection error	<ul style="list-style-type: none"> <li>The error was not cleared. (Error clear was performed during execution of error clear.)</li> </ul>	<ul style="list-style-type: none"> <li>Execute again after a while.</li> </ul> <p>If the same error code is displayed again after re-execution, the possible cause is a hardware failure of the target module. Please consult your local Mitsubishi representative.</p>

Error code	Error name	Error details and cause	Action
433DH	Maintenance and inspection error	<ul style="list-style-type: none"> <li>The target module cannot handle the error clear.</li> </ul>	<ul style="list-style-type: none"> <li>Check the target module. (Check the module in which the error occurred.)</li> </ul>
4400H	Security function error	<ul style="list-style-type: none"> <li>A file protected by a password has been opened without unlocking the password.</li> </ul>	<ul style="list-style-type: none"> <li>Enter a correct password and perform password authentication.</li> </ul>
4401H	Security function error	<ul style="list-style-type: none"> <li>Read password authentication has failed when required.</li> <li>The file password format is incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>Set a correct read password and perform password authentication.</li> <li>Access the file with the correct method.</li> </ul>
4402H	Security function error	<ul style="list-style-type: none"> <li>Write password authentication has failed when required.</li> <li>The file password format is incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>Set a correct write password and perform password authentication.</li> <li>Access the file with the correct method.</li> </ul>
4403H	Security function error	<ul style="list-style-type: none"> <li>Both passwords for reading and for writing that are set upon Create, Change, Delete, or Disable do not match the previous ones.</li> </ul>	<ul style="list-style-type: none"> <li>Set correct passwords for both reading and writing, and perform password authentication.</li> </ul>
4404H	Security function error	<ul style="list-style-type: none"> <li>A file error was detected before or after performing Create, Change, or Delete.</li> </ul>	<ul style="list-style-type: none"> <li>Initialize the drive including the target file by initializing the memory.</li> <li>Write the target file to the programmable controller again, and then register or cancel the file password.</li> </ul>
4408H	Security function error	<ul style="list-style-type: none"> <li>File password authentication has failed when required.</li> </ul>	<ul style="list-style-type: none"> <li>Set a correct password and perform password authentication again.</li> </ul>
4409H	Security function error	<ul style="list-style-type: none"> <li>File password authentication has failed when required.</li> </ul>	<ul style="list-style-type: none"> <li>Set a correct password and perform password authentication again 1 minutes later.</li> </ul>
440AH	Security function error	<ul style="list-style-type: none"> <li>File password authentication has failed when required.</li> </ul>	<ul style="list-style-type: none"> <li>Set a correct password and perform password authentication again 5 minutes later.</li> </ul>
440BH	Security function error	<ul style="list-style-type: none"> <li>File password authentication has failed when required.</li> </ul>	<ul style="list-style-type: none"> <li>Set a correct password and perform password authentication again 15 minutes later.</li> </ul>
440CH	Security function error	<ul style="list-style-type: none"> <li>File password authentication has failed when required.</li> </ul>	<ul style="list-style-type: none"> <li>Set a correct password and perform password authentication again 60 minutes later.</li> </ul>
440DH	Security function error	<ul style="list-style-type: none"> <li>File password authentication has failed when required.</li> </ul>	<ul style="list-style-type: none"> <li>Set a correct password and perform password authentication again 60 minutes later.</li> </ul>
440EH	Security function error	<ul style="list-style-type: none"> <li>The security function was activated and password authentication cannot be performed.</li> </ul>	<ul style="list-style-type: none"> <li>Set a correct password and perform password authentication again after a certain period of time.</li> </ul>
4410H	Security function error	<ul style="list-style-type: none"> <li>The file of the locked CPU module is accessed without the security key authentication.</li> </ul>	<ul style="list-style-type: none"> <li>Register the security key which locks the CPU module to the engineering tool.</li> <li>When the project is opened, lock the project with the security key which locks the CPU module.</li> <li>When the CPU module is locked, the access control target file cannot be accessed using the following functions or external devices. <ul style="list-style-type: none"> <li>FTP server function</li> <li>SLMP/MC protocol</li> <li>GOT</li> <li>EZSocket</li> </ul> </li> </ul>
4412H	Security function error	<ul style="list-style-type: none"> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Hardware failure of the CPU module. Replace the CPU module.</li> </ul>
4413H	Security function error	<ul style="list-style-type: none"> <li>Since the CPU module is locked and 32 engineering tools are reading and writing programs simultaneously, another engineering tool cannot read or write programs.</li> </ul>	<ul style="list-style-type: none"> <li>Wait until the number of engineering tools which are reading and writing programs decreases to 31 or less, and read or write programs.</li> </ul>
4414H	Security function error	<ul style="list-style-type: none"> <li>Since the CPU module is locked, the requested processing cannot be performed.</li> </ul>	<ul style="list-style-type: none"> <li>Do not request any processing since the requested processing cannot be performed in this state.</li> </ul>
4415H	Security function error	<ul style="list-style-type: none"> <li>Since the CPU module is not locked, the requested processing cannot be performed.</li> </ul>	<ul style="list-style-type: none"> <li>Do not request any processing since the requested processing cannot be performed in this state.</li> </ul>
4416H	Security function error	<ul style="list-style-type: none"> <li>Since the CPU module is in locking or unlocking operation, the requested processing cannot be performed.</li> </ul>	<ul style="list-style-type: none"> <li>Request the processing after the locking or unlocking operation ends.</li> </ul>
4417H	Security function error	<ul style="list-style-type: none"> <li>When an attempt was made to write or delete the security key, no extended SRAM cassette was inserted to the CPU module.</li> </ul>	<ul style="list-style-type: none"> <li>Check that the extended SRAM cassette is correctly inserted, and then perform the operation again.</li> </ul>
4418H	Security function error	<ul style="list-style-type: none"> <li>The security key cannot be changed or deleted because any locked program exists in the CPU module.</li> </ul>	<ul style="list-style-type: none"> <li>Unlock all the programs in the CPU module.</li> </ul>

Error code	Error name	Error details and cause	Action
4422H	Security function error	<ul style="list-style-type: none"> <li>The access target CPU module does not support the security key information stored in the engineering tool.</li> </ul>	<ul style="list-style-type: none"> <li>Change the security key information version of the engineering tool in accordance with the version supported by the target CPU module.</li> </ul>
4424H	Security function error	<ul style="list-style-type: none"> <li>The security key set to the file batch-written by executing the online change function does not match the one registered to the CPU module (or extended SRAM cassette).</li> </ul>	<ul style="list-style-type: none"> <li>Set the same security key registered to the CPU module to the file, and then execute the online change function.</li> </ul>
4425H	Security function error	<ul style="list-style-type: none"> <li>The file with a security key was batch-written by executing the online change function, but no security key is registered to the CPU module (or extended SRAM cassette).</li> </ul>	<ul style="list-style-type: none"> <li>Clear the security key set to the file, and then execute the online change function.</li> </ul>
4905H	Other errors	<ul style="list-style-type: none"> <li>The capacity of label area used exceeded its limit.</li> </ul>	<ul style="list-style-type: none"> <li>Check the program, delete the unused local and global label definitions, compile the program, and then write the program to the programmable controller.</li> <li>Change the label area size in parameter, compile the program, and write the program to the programmable controller.</li> </ul>
4906H	Other errors	<p>The file register data cannot be cleared by the following reasons.</p> <ul style="list-style-type: none"> <li>The QDRSET instruction was executed more than once after the CPU module was powered on or reset.</li> <li>The file register setting parameter is set to the one other than "Use Common File Register in All Programs".</li> </ul>	<ul style="list-style-type: none"> <li>If the QDRSET instruction is being executed, power on or reset the CPU module.</li> <li>The operation cannot be performed if the file register setting parameter is not set to "Use Common File Register in All Programs".</li> </ul>
4A00H	Network error	<ul style="list-style-type: none"> <li>Access to the specified station cannot be made since the routing parameters are not set to the start source CPU module and/or relay CPU module.</li> <li>For routing via a multiple CPU system, the control CPU of the network module for data routing has not started.</li> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Set to the related stations the routing parameters for access to the specified station.</li> <li>Retry after a while. Or, start communication after checking that the system for data routing has started.</li> <li>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.</li> </ul>
4A01H	Network error	<ul style="list-style-type: none"> <li>The network of the number set to the routing parameters does not exist.</li> <li>The specified CPU module cannot be communicated through the network that is not supported by the CPU module.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the routing parameters set to the related stations.</li> <li>Set communication through the network that is supported by the specified CPU module.</li> </ul>
4A02H	Network error	<ul style="list-style-type: none"> <li>Access to the specified station cannot be made.</li> </ul>	<ul style="list-style-type: none"> <li>Check the network module for error, or check that the modules are not in offline.</li> <li>Check if the network numbers/PC numbers are correctly set.</li> </ul>
4A03H	Network error	<ul style="list-style-type: none"> <li>A request for network test was issued.</li> </ul>	<ul style="list-style-type: none"> <li>Check the request of the SLMP/MC protocol.</li> </ul>
4A20H	IP communication test error	<ul style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the IP address settings of the CPU module.</li> <li>Check and correct the IP address of the request destination device.</li> <li>Check and correct the IP address of the CC-Link IE module.</li> <li>Check and correct the IP address of the request source device.</li> </ul>
4A21H	IP communication test error	<ul style="list-style-type: none"> <li>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.</li> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the IP address settings of the CPU module.</li> <li>Check and correct the IP address of the request destination device.</li> </ul>
4A22H	IP communication test error	<ul style="list-style-type: none"> <li>The IP address is not set to the CC-Link IE module on the path that IP packets travel.</li> </ul>	<ul style="list-style-type: none"> <li>Set the IP address to the CC-Link IE module used as a master station.</li> <li>Check the communication status with the master station when the CC-Link IE module is used as a local station.</li> <li>Replace the CC-Link IE module (master station) with the one that supports the IP packet transfer function.</li> <li>Conduct the IP communication test again after the CC-Link IE module is started up.</li> <li>Check and correct the IP address of the request destination device.</li> </ul>

Error code	Error name	Error details and cause	Action
4A23H	IP communication test error	<ul style="list-style-type: none"> <li>The CPU module on the path that IP packets travel does not support the IP packet transfer function.</li> <li>The CPU module does not support the IP packet transfer function.</li> <li>Routing parameters are set so that IP packets are routed to the CPU module.</li> </ul>	<ul style="list-style-type: none"> <li>Replace the CPU module with the one supporting the IP packet transfer function.</li> <li>Correct routing parameter so that IP packets are routed to the CPU module that supports the IP packet transfer function.</li> <li>Check and correct the IP address of the request destination device.</li> </ul>
4A24H	IP communication test error	<ul style="list-style-type: none"> <li>The CC-Link IE module on the path that IP packets travel does not support the IP packet transfer function.</li> <li>Routing parameters are set so that IP packets are routed to the network module that does not support the IP packet transfer function.</li> <li>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.</li> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>Replace the CC-Link IE module with the one supporting the IP packet transfer function.</li> <li>Correct routing parameter so that IP packets are routed to the CC-Link IE module that supports the IP packet transfer function.</li> <li>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.</li> <li>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.</li> <li>Check and correct the IP address of the request destination device.</li> </ul>
4A25H	IP communication test error	<ul style="list-style-type: none"> <li>The IP packet transfer setting is not set.</li> <li>Routing parameters are set so that IP packets are routed to the CPU module where IP packet transfer setting is not set.</li> </ul>	<ul style="list-style-type: none"> <li>Select "Use" for the IP packet transfer function setting in the CPU parameters.</li> <li>Correct routing parameter so that IP packets are routed to the CPU module where IP packet transfer setting is set.</li> <li>Check and correct the IP address of the request destination device.</li> </ul>
4A27H	IP communication test error	<ul style="list-style-type: none"> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>
4A28H	IP communication test error	<ul style="list-style-type: none"> <li>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.</li> <li>When multiple CC-Link IE modules with the same network number are connected in the 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.</li> <li>When multiple CC-Link IE modules with the same network number are connected in the 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.</li> </ul>	<ul style="list-style-type: none"> <li>Correct the routing parameter setting so that IP packets travel the same path for both request and response transmission.</li> <li>When multiple CC-Link IE modules with the same network number are connected in the multiple CPU system, set the module mounted on the lowest number slot as a controlled module that transfers IP packets.</li> <li>When multiple CC-Link IE modules with the same network number are connected in the 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.</li> </ul>
4A29H	IP communication test error	<ul style="list-style-type: none"> <li>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 by Ether net.</li> </ul>	<ul style="list-style-type: none"> <li>Check and correct the IP address settings of the CPU module.</li> <li>Check and correct the IP address of the request destination device.</li> </ul>
4A2AH	IP communication test error	<ul style="list-style-type: none"> <li>The IP address of a device on CC-Link IE or the CPU module is not specified.</li> </ul>	<ul style="list-style-type: none"> <li>Specify the IP address of a device on CC-Link IE or the CPU module.</li> </ul>
4B00H	Target module error	<ul style="list-style-type: none"> <li>An error occurred in the access destination or relay station.</li> <li>The specified transfer setup (request destination module I/O number) is invalid.</li> </ul>	<ul style="list-style-type: none"> <li>Take corrective action after checking the error that occurred at the specified access destination or the relay station to the accessed station.</li> <li>Check the transfer setup (request destination module I/O number or PC number) in the request data of SLMP/MC protocol.</li> <li>Check the stop error, and take action.</li> </ul>
4B02H	Target module error	<ul style="list-style-type: none"> <li>The request is not addressed to the CPU module.</li> </ul>	<ul style="list-style-type: none"> <li>Perform operation for the module that can execute the specified function.</li> </ul>
4B03H	Target module error	<ul style="list-style-type: none"> <li>The specified route is not supported by the specified CPU module version.</li> <li>The communication target CPU module is not mounted.</li> </ul>	<ul style="list-style-type: none"> <li>Check whether the specified route is supported or not.</li> <li>Check the mounting status of the CPU module.</li> <li>Check the stop error, and take action.</li> </ul>

Error code	Error name	Error details and cause	Action
4B04H	Target module error	<ul style="list-style-type: none"> <li>The specified transfer setup (request destination module I/O number) is not supported.</li> </ul>	<ul style="list-style-type: none"> <li>In the target setup, an invalid value is set as the start I/O number of the target module. Set the start I/O number of the target module correctly, and communicate again.</li> </ul>
4C00H	Data logging function error	<ul style="list-style-type: none"> <li>There is not enough free space for storing the result file in the target memory.</li> </ul>	<ul style="list-style-type: none"> <li>Increase the free space, and create the result file again.</li> </ul>
4C01H	Data logging function error	<ul style="list-style-type: none"> <li>The result file cannot be written to the target memory because the SD memory card is write-protected or the folder/file structure is incorrect.</li> </ul>	<ul style="list-style-type: none"> <li>Unlock the write protect switch of the SD memory card, and write the result file again.</li> <li>Check that the SD memory card is not damaged.</li> <li>Check that files/folders used in the SD memory card are not deleted.</li> </ul>
4C02H	Data logging function error	<ul style="list-style-type: none"> <li>The SD memory card was removed while the data logging function was being executed (data logging status: Collecting, Trigger waiting (collecting before trigger), Collecting after trigger, or Saving the logging data).</li> </ul>	<ul style="list-style-type: none"> <li>Insert the SD memory card, and execute the function again.</li> </ul>
4C03H	Data logging function error	<ul style="list-style-type: none"> <li>The number of files in the root directory and subdirectory in the target memory exceeded the limit.</li> </ul>	<ul style="list-style-type: none"> <li>Increase the free space of the drive (memory), and execute the function again.</li> <li>Delete files in the drive (memory), and execute the function again.</li> </ul>
4C04H	Data logging function error	<ul style="list-style-type: none"> <li>During auto logging, a data logging was not registered due to a registration failure of the data logging with another setting number.</li> </ul>	<ul style="list-style-type: none"> <li>Clear the error, and start auto logging.</li> </ul>
4C05H	Data logging function error	<ul style="list-style-type: none"> <li>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).</li> </ul>	<ul style="list-style-type: none"> <li>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).</li> <li>Stop the data logging function specifying the step number.</li> </ul>
4C06H	Data logging function error	<ul style="list-style-type: none"> <li>System error</li> </ul>	<ul style="list-style-type: none"> <li>Check the specified data, and write it to the CPU module again.</li> </ul>



## ■ Error codes related to the online module change function

The following table lists the codes of errors detected by 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 is executed while the CPU module is in the error state (stop error).	The module cannot be changed online. Power off the programmable controller, and replace the module.
4111H		The online module change function is executed in the multiple CPU system even though other CPU modules have not been started up.	Wait until all the CPU modules start up, and execute the function again.
4202H		<ul style="list-style-type: none"> <li>An attempt was made to change two modules directly at the same time.</li> <li>A module that is not ready to be removed was removed.</li> </ul>	<ul style="list-style-type: none"> <li>For the module where online module change is in progress, continue the change. The other module where change was attempted later cannot be changed online. Power off the programmable controller, and replace the module.</li> <li>Turns on SM1602 (Module removal request flag) and check that SD1617 (Online module change progress status) turns to Module removal completion (5). Then, continue the change from the procedure of module mounting.</li> </ul>
4205H		The online module change function is executed in the 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 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. <ul style="list-style-type: none"> <li>Mount a module of the same model, and perform the module recognition processing again.</li> </ul>
4215H		<ul style="list-style-type: none"> <li>The module recognition processing is performed without the module being mounted.</li> <li>The module selection processing is performed to the slot that is set as "Empty".</li> <li>The module newly-mounted to the system is not mounted correctly.</li> </ul>	<ul style="list-style-type: none"> <li>Mount a module, and perform the module recognition processing again.</li> <li>Check and correct the target base unit number or slot number, and turn on SM1600 (Module selection request flag) again.</li> <li>Mount the module correctly.</li> </ul>
4216H		The module newly-mounted to the system online has failed. (The CPU module cannot access to the module.) <ul style="list-style-type: none"> <li>The module has failed.</li> <li>The module is not mounted correctly.</li> </ul>	After the module is changed online, execute the request again. Or, power off the system, and replace the module with the one that supports the online module change function. <ul style="list-style-type: none"> <li>Replace the module with the normal one.</li> <li>Mount the module correctly.</li> </ul>
4218H		<ul style="list-style-type: none"> <li>The online change target module (before change) does not support the online module change function.</li> <li>The module newly-mounted to the system online (after change) does not support the online module change function.</li> <li>The online module change function is executed to change the MELSEC-Q series module.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>The module cannot be changed online. Power off the programmable controller, and replace the module.</li> </ul>
421AH		The online module change function is executed to change the module controlled by another CPU module.	Execute the function from the control CPU module.

# Appendix 2 Event List

The CPU module collects information from each module such as errors detected by the module, operations done for the module, and network errors, and stores it in the data memory or on the SD memory card. (Page 184 Event History Function) When an event occurs, its event code and description can be read using an engineering tool.



For details on events occurred on each CPU module, refer to the manual for the module used.

## Guide for reference of event list

The event list contains the following information.

Item	Description
Event code	The ID number of each event
Event type	The type of each event
Event category	The category of each event
Detected event	The description of each detected event
Detailed information 1 to 3	Details of each detected event

### Detailed information

The contents of Detailed information 1 to 3

Detailed information	Item	Description
Detailed information 1	Operation source information	Information on the operation source <ul style="list-style-type: none"> <li>• Connection port (connection information such as Ethernet or USB)</li> <li>• I/O No.</li> <li>• CPU number (the CPU number of each CPU module in a multi-CPU system)</li> <li>• Network Number</li> <li>• Station No.</li> <li>• IP Address</li> </ul>
	Event history file information	Information on the event history file
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 security keys
	Remote password information	Information on the remote password
	File password information	Information on the file password
	Disconnected IP address information	Information on the disconnected IP address
	Drive and file information	Information on drive names and file names
	Copy source drive and file information	Information on the copying drive name and file name
	Operation target information	Information about the operation target (I/O No.)
	Clock information (before change)	Information about the clock before change
	Remote operation type information	Information about the remote operation type
	Device and label information	Information about devices and labels
System configuration information	Information about the system configuration	
Detailed information 3	Clock information (after change)	Information about the clock after change
	Copy destination drive and file information	Information on drive names and file names

# Event list

The following table lists events related to the CPU module.

Event code	Event type	Event category	Detected event	Description	Detailed information				
					Detailed information 1	Detailed information 2	Detailed information 3		
0100	System	Info	Link-up	The CPU module has been entered into the link-up state as a result of operation such as connection of the network cable from an external device.	Operation source information	Communication speed and communication mode	—		
0110			TCP connection communication start/end	Communication with an external device through TCP connection has been started. Or, communication with an external device through TCP connection has been terminated.					
0120			FTP connection start/termination	FTP connection with an external device has been started. Or, FTP connection with an external device has been terminated.					
0130			Receive frame error	A receive frame error has been detected.	—				
0140			SNTP server time synchronization failure	Time setting by the time synchronization function time has been failed because there was no response from the SNTP server.					
0400			Power-on and reset	The CPU module has been power-on or reset.		—			
0410			Boot Operation	Boot operation has been performed.					
0420			Event history file generation	A event history file has been generated.	Event history file information				
0800			Warning	Warning	Link-down	The CPU module has been entered into the link-down state as a result of operation such as disconnection of the network cable from an external device.		Operation source information	Communication speed and communication mode
0904					Socket communications send error	Sending a message over socket communication failed.			
0906	Alive check error	Alive check of the target device could not be completed within the period specified by the response monitoring timer value.							
0907	Divided message reception timeout error	<ul style="list-style-type: none"> <li>All the data could not be received within the period specified by the response monitoring timer value.</li> <li>Data as long as the data length could not be received.</li> <li>The remaining part of the message divided at the TCP/IP level could not be received within the period specified by the response monitoring timer value.</li> </ul>							
0908	IP composition timeout error	An IP composition timeout error occurred (the remaining part of the divided data could not be received to cause the time-out).							
0909	TCP-specified port number error	A port number being used for opened connection was set (for TCP/IP).							
090A	UDP-specified port number error	A port number being used for opened connection was set (for UDP/IP).							
1000 -	Error	Error			When a self-diagnostics error occurs, information on the error is logged as an event.				

Event code	Event type	Event category	Detected event	Description	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
10100	Security	Info	Security key registration/deletion	A security key was registered or deleted.	Operation source information	Security key operation information	—
10200			Remote password lock	The lock processing for the remote password was performed.		Remote password information	
10201			Remote password unlock successful	The lock processing for the remote password was performed and completed successfully.			
10202			Remote password unlock failed	The lock processing for the remote password was performed and was not completed successfully.			
10300			Access from address prohibited by IP filter setting accepted	Access from an IP address for which access is prohibited by the IP filter setting was accepted.	Disconnected IP address information		

Event code	Event type	Event category	Detected event	Description	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
10400	Security	Info	File password registration/change/deletion successful	Registration, change, or deletion of a file password was performed and completed successfully.	Operation source information	File password information	—
10401			File password registration/change/deletion failed	Registration, change, or deletion of a file password was performed and was not completed successfully.			
10402			File password clear successful	Clear of a file password was performed and completed successfully.			
10403			File password clear failed	Clear of a file password was performed and was not completed successfully.			
10500			Forced disable applied	Forced disable was enabled.	—	—	—
10501			Forced disable clear	Forced disable was disabled.	—	—	—
20100			Operation		Error clear	Error clear was performed.	Operation source 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 forced disabled	The SD memory card forced disable function was performed and the SD memory card is ready for removal.					
24000	Clock setting	The clock setting was specified.			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.					
24100	Operating status change (RUN)	The operating status was changed to RUN.					
24101	Operating status change (STOP)	The operating status was changed to STOP.					
24102	Operating status change (PAUSE)	The operating status was changed to PAUSE.					
24200	Creation of new folders, writes to files/folders* <sup>1</sup>	<ul style="list-style-type: none"> <li>• A new folder was created.</li> <li>• A new file was created or data was written to a file.</li> </ul>			Operation source information	Drive and file information	—
24201	File copy* <sup>1</sup>	A file was copied.					
24202	Folder/file rename* <sup>1</sup>	A folder name or file name was changed.					
25000	Online module change	The online module change processing has completed.			—	System configuration information	—
2A200	Warning				Memory initialization* <sup>1</sup>	The memory was initialized.	—
2A201			Device/label zero clear	A device or label was cleared to zero.	—	Device and label information	—
2A202			Folder/file deletion* <sup>1</sup>	A folder or file was deleted.	—	Drive and file information	—

\*1 For file-related events such as write to and deletion of files, operations for the following are logged into the event history:

- Program file
- FB Program file
- Parameter file
- Data logging setting file (Common setting file, Individual setting file)
- Memory dump setting file

# Appendix 3 Troubleshooting by Symptom

If any function of the CPU module does not operate as designed, perform troubleshooting by checking the following items. If the ERROR LED or USER LED is on or flashing, eliminate the error cause using the engineering tool.

## When the POWER LED of the power supply module turns off

When the POWER LED of the power supply module turns off, check the following items.

Check item	Action
The power supply module is not mounted on the base unit properly.	Remove the power supply module from the base unit, and mount it back on the base unit. Then, restore power to the system.
The READY LED of the CPU module is on.	The power supply module has failed. Replace the power supply module.
Power supply voltage is not appropriate.	Supply power voltage within the specified range. (MELSEC iQ-R Module Configuration Manual)
The internal current consumption within the entire system exceeded the rated output current of the power supply module.	Review the system configuration so that the internal current consumption does not exceed the rated output current. (MELSEC iQ-R Module Configuration Manual)
The POWER LED turns on when power is restored to the system after all modules, except the power supply module, have been removed.	One of the modules except the power supply module has failed. Repeatedly supply power to the system, returning the modules to the system one by one. The last module mounted immediately before the POWER LED turns off has failed. Replace the corresponding module.

If the POWER LED of the power supply module does not turn on even after the items above are checked and the actions are taken, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.

## When the READY LED of the CPU module turns off

When the READY LED of the CPU module turns off, check the following items.

Check item	Action
The CPU module is not 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.
The READY LED of another module is on.	A major error has occurred in the CPU module. Replace the CPU module.
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.)	The power supply module before the replacement has failed. Replace the power supply module.
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.)	One of the modules except the power supply module has failed. Repeatedly supply power to the system, returning the modules to the system one by one. The last module mounted immediately before the READY LED turns off has failed. Replace the corresponding module.

If the READY LED of the CPU module does not turn on even after the items above are checked and the actions are taken, the possible cause is a hardware failure of the power supply module. Please consult your local Mitsubishi representative.

## When the specific extension base unit cannot be recognized

When the specific extension base unit cannot be recognized, check the following items.

Check item	Action
The extension base unit that cannot be recognized is the Q series extension base unit.	Refer to the troubleshooting for the case where the specific Q series extension base unit cannot be recognized. (☞ Page 317 When the specific Q series extension base unit cannot be recognized)
The POWER LED of the power supply module of the extension base unit that cannot be recognized is turned off.	Refer to the troubleshooting for the power supply module. (☞ Page 316 When the POWER LED of the power supply module turns off)
The extension cable is not correctly mounted.	Correctly mount the extension cable.
The specific extension base unit cannot be recognized even after the above items were checked.	Execute in the order shown below. <ul style="list-style-type: none"> <li>• Replace the extension cable.</li> <li>• Replace the corresponding extension base unit.</li> <li>• Replace the base unit at one step before the corresponding extension base unit.</li> </ul>

If the specific extension base unit cannot be recognized even after the items above are checked and the actions are taken, the possible cause is a hardware failure. Please consult your local Mitsubishi representative.

## When the specific Q series extension base unit cannot be recognized

When the specific Q series extension base unit cannot be recognized, check the following items.

Check item	Action
The POWER LED of the power supply module of the Q series extension base unit that cannot be recognized is turned off.	Refer to the troubleshooting for the power supply module. (☞ User's Manual (Hardware Design, Maintenance, and Inspection)) for the Q series CPU module used
The connector pin for setting the number of extension base units is not correctly mounted.	Correctly mount the connector pin for setting the number of extension base units.
The extension cable is not correctly mounted.	Correctly mount the extension cable.
The specific Q series extension base unit cannot be recognized even after the above items were checked.	Execute in the order shown below. <ul style="list-style-type: none"> <li>• Replace the extension cable.</li> <li>• Replace the corresponding Q series extension base unit.</li> <li>• Replace the base unit at one step before the corresponding Q series extension base unit.</li> </ul>

If the specific Q series extension base unit cannot be recognized even after the items above are checked and the actions are taken, the possible cause is a hardware failure. Please consult your local Mitsubishi representative.

## When data cannot be written to the programmable controller

When data cannot be written to the programmable controller, check the following items.

Check item	Action
Check if password is not registered.	Authenticate a password using the engineering tool.
Check if the SD memory card is write-protected while the SD memory card is being targeted for writing.	Clear the write-protect for the SD memory card.
Check if the SD memory card is formatted while the SD memory card is being targeted for writing.	Carry out formatting for the SD memory card.
Check if the data to be written exceeds the capacity of CPU built-in memory or SD memory card.	Check the CPU built-in memory or SD memory card for their free space.

If data cannot be written to the programmable controller even after the items above are checked and the actions are taken, the possible cause is a hardware failure. Please consult your local Mitsubishi representative.

## When data cannot be read from the programmable controller

---

When data cannot be read from the programmable controller, check the following items.


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 target memory (CPU built-in memory, SD memory card, Intelligent function unit) to be read.

If data cannot be read from the programmable controller even after the items above are checked and the actions are taken, the possible cause is a hardware failure. Please consult your local Mitsubishi representative.

## When Ethernet function cannot be used

---

For troubleshooting when using Ethernet function, refer to the following.

 MELSEC iQ-R Ethernet User's Manual (Application)



# 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> <ul style="list-style-type: none"><li>• S: System</li><li>• U: sets at the user side (program, engineering tool, GOT, other testing operations from external device).</li><li>• U/S: User and system</li></ul> <Set timing> <ul style="list-style-type: none"><li>• Every END: Data is set every time END processing is performed.</li><li>• Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN).</li><li>• Status change: Data is set when the status is changed.</li><li>• Error: Data is set when an error occurs.</li><li>• Instruction execution: Data is set when an instruction is executed.</li><li>• Request: Data is set when requested by a user (using the special relay).</li><li>• Writing: Data is set when a user performs a writing operation.</li><li>• During END: Data is set when END processing is performed.</li><li>• 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.</li></ul>



Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

## Diagnostic information

The following is the list of special relay areas relating to the diagnostic information.

No.	Name	Data stored	Details	Set by (setting timing)
SM0	Latest self diagnostics error (including annunciator ON)	Off: No error On: Error	<ul style="list-style-type: none"> <li>This relay switches to on when the self diagnostics returns an error (including the case when an error is detected by turning the annunciator On).</li> <li>The On state is maintained even after the error has been later cleared.</li> </ul>	S (Error)
SM1	Latest self diagnostics error (not including annunciator On)	Off: No error On: Error	<ul style="list-style-type: none"> <li>This relay switches to on when the self diagnostics returns an error (not including the case when an error is detected by turning the annunciator On).</li> <li>The On state is maintained even after the error has been later cleared.</li> </ul>	S (Error)
SM50	Error reset	Off→On: Error reset request On→Off: Error reset complete	<ul style="list-style-type: none"> <li>This relay clears the error state when the mode transfers from off to on.</li> <li>This relay switches from on to off when the error reset has been completed.</li> </ul>	U/S (Status change)
SM51	Battery low latch	Off: Normal On: Battery low	<ul style="list-style-type: none"> <li>This relay switches to on when the battery voltage of the CPU module drops below the specified value.</li> <li>The On state is maintained even after the battery voltage has been later recovered to the normal value.</li> <li>This relay synchronizes with BAT LED.</li> </ul>	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	<ul style="list-style-type: none"> <li>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.</li> <li>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.</li> </ul>	S (Error)
SM56	Instruction execution fault	Off: Normal On: Instruction execution fault state	<ul style="list-style-type: none"> <li>This relay switches to on when an error which can be classified as instruction execution fault is returned.</li> <li>The On state is maintained even after the error has been later cleared.</li> </ul>	S (Error)
SM60	Fuse Blown	Off: Normal On: Fuse blown is detected	<ul style="list-style-type: none"> <li>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.</li> <li>The fuse blown state check is also done for output modules on the remote I/O station.</li> </ul>	S (Error)
SM61	I/O module verify error	Off: Normal On: Error	<ul style="list-style-type: none"> <li>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.</li> <li>I/O module verification is also done for modules on the remote I/O station.</li> </ul>	S (Error)
SM62	Annunciator	Off: Not detected On: Detected	<ul style="list-style-type: none"> <li>This relay switches to on when at least one annunciator is turned on.</li> <li>This relay returns to off when all the annunciators are turned off.</li> </ul>	S (Instruction execution)
SM80	Detailed information 1: Flag in use	Off: Not used On: In use	This relay switches to on if the detailed information n exists when SM0 switched to on.	S (Status change)
SM112	Detailed information 2: Flag in use			


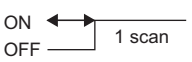
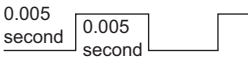
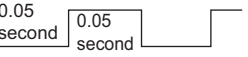

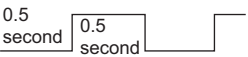


## System information

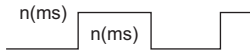
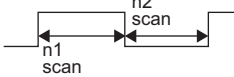


The following is the list of special relay areas relating to the system information.

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	<ul style="list-style-type: none"> <li>Clock data stored in SD210 to SD216 is written into the CPU module when this relay is switched from off to on.</li> <li>This relay switches from on to off when writing of clock data stored in SD210 to SD216 into the clock element is completed.</li> </ul>	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 of the clock data (SD210 to SD216), and to off when no error is generated.	S (Request)
SM213	Clock data read request	Off: Non-processing On: Reading request	Clock data is loaded into SD210 to SD216 when this relay is in On state.	U
SM220	CPU No.1 preparation completed	Off: CPU No.n preparation not completed On: CPU No.n preparation completed	<ul style="list-style-type: none"> <li>This relay switches to on at the time when access from the CPU module on other CPUs to the CPU module for CPU No. n is enabled during power-on or resetting.</li> <li>This relay is used as an interlock to access the CPU module for the CPU No. n when the multiple CPU synchronization setting is configured to asynchronous mode.</li> </ul>	S (Status change)
SM221	CPU No.2 preparation completed			
SM222	CPU No.3 preparation completed			
SM223	CPU No.4 preparation completed			
SM230	CPU No.1 error flag	Off: CPU No.n normal On: CPU No.n stop error state	<ul style="list-style-type: none"> <li>This relay is off when the CPU module for the CPU No.n is normal (including a continuation error period).</li> <li>This relay is on when the CPU module for the CPU No.n is in stop error state.</li> </ul>	S (Status change)
SM231	CPU No.2 error flag			
SM232	CPU No.3 error flag			
SM233	CPU No.4 error flag			
SM240	CPU No.1 reset flag	Off: CPU No.n not being reset On: CPU No.n in reset mode	<ul style="list-style-type: none"> <li>This relay switches to off when the CPU module of the CPU No.1 is not being reset.</li> <li>This relay is on while the CPU module of the CPU No.1 is being reset (including the case when the CPU module is removed from the base unit). Other CPUs also enter into reset mode.</li> </ul>	S (Status change)
SM241	CPU No.2 reset flag			
SM242	CPU No.3 reset flag			
SM243	CPU No.4 reset flag			

## System clock

The following is the list of special relay areas relating to the system clock.

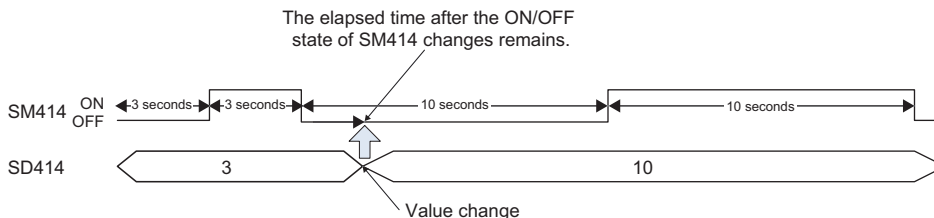
No.	Name	Data stored	Details	Set by (setting timing)
SM400	Always On	ON _____ OFF	Always On	Power-on to RUN or STOP to RUN
SM401	Always Off	ON OFF _____	Always Off	S (Status change)
SM402	After RUN, On for one scan only	ON  OFF	<ul style="list-style-type: none"> <li>This relay is on during only one scan after RUN mode starts.</li> <li>This relay is enabled only for the scan execution type program.</li> </ul>	S (Status change)
SM403	After RUN, Off for one scan only	ON  OFF	<ul style="list-style-type: none"> <li>This relay is off during only one scan after RUN mode starts.</li> <li>This relay is enabled only for the scan execution type program.</li> </ul>	S (Status change)
SM409	0.01 second clock		<ul style="list-style-type: none"> <li>This relay repeats on/off at 5ms intervals.</li> <li>The initial state when CPU module is powered on or reset is off.</li> <li>This relay switches to on or off after a specified time is elapsed even when the program execution is in progress.</li> </ul>	S (Status change)
SM410	0.1 second clock		<ul style="list-style-type: none"> <li>This relay repeats on/off at certain intervals.</li> <li>The initial state when CPU module is powered on or reset is off.</li> <li>This relay switches to on or off after a specified time is elapsed even when the program execution is in progress.</li> </ul>	S (Status change)
SM411	0.2 second clock		<ul style="list-style-type: none"> <li>This relay repeats on/off at certain intervals.</li> <li>The initial state when CPU module is powered on or reset is off.</li> <li>This relay switches to on or off after a specified time is elapsed even when the program execution is in progress.</li> </ul>	S (Status change)
SM412	1 second clock		<ul style="list-style-type: none"> <li>This relay repeats on/off at certain intervals.</li> <li>The initial state when CPU module is powered on or reset is off.</li> <li>This relay switches to on or off after a specified time is elapsed even when the program execution is in progress.</li> </ul>	S (Status change)
SM413	2 second clock		<ul style="list-style-type: none"> <li>This relay repeats on/off at certain intervals.</li> <li>The initial state when CPU module is powered on or reset is off.</li> <li>This relay switches to on or off after a specified time is elapsed even when the program execution is in progress.</li> </ul>	S (Status change)
SM414	2n second clock		<ul style="list-style-type: none"> <li>This relay repeats on/off at regular intervals specified in units of seconds on the SD414. (When the value on SD414 is changed, the elapsed time count starting when on/off state on SM414 was last changed continues, and on/off state will be changed when the new specified time is elapsed.*1)</li> <li>The initial state when CPU module is powered on or reset is off.</li> <li>This relay switches to on or off after a specified time is elapsed even when the program execution is in progress.</li> </ul>	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SM415	2n (ms) clock		<ul style="list-style-type: none"> <li>This relay repeats on/off at regular intervals specified in units of milliseconds on the SD415 (When the value on SD414 is changed, the elapsed time count starting when on/off state on SM414 was last changed continues, and on/off state will be changed when the new specified time is elapsed.*1)</li> <li>The initial state when CPU module is powered on or reset is off.</li> <li>This relay switches to on or off after a specified time is elapsed even when the program execution is in progress.</li> </ul>	S (Status change)
SM420	User timing clock No.0		<ul style="list-style-type: none"> <li>This relay repeats on/off at specified scan intervals.</li> <li>The initial state when CPU module is powered on or reset is off.</li> <li>The on/off scan interval is set using the DUTY instruction. (n1: ON scan interval, n2: OFF scan interval)</li> </ul>	S (Every END)
SM421	User timing clock No.1			
SM422	User timing clock No.2			
SM423	User timing clock No.3			
SM424	User timing clock No.4			
SM440	On only initial I44 execution after RUN		<ul style="list-style-type: none"> <li>This relay is on during the first execution of the inter-module synchronous interrupt program (I44) after RUN mode starts and off during the second execution onwards.</li> <li>Note that the relay status while the DI instruction is executed is as follows: On during the first execution of the inter-module synchronous interrupt program (I44) after the DI instruction is cleared and Off during the second execution onwards. (no change after the second DI clear).</li> <li>This contact is enabled only for the inter-module synchronous interrupt program.</li> </ul>	S (Power-on to RUN or STOP to RUN)
SM441	On only initial I45 execution after RUN		<ul style="list-style-type: none"> <li>This relay is on during the first execution of the multiple CPU synchronous interrupt program (I45) after RUN mode starts and off during the second execution onwards.</li> <li>Note that the relay status while the DI instruction is executed is as follows: On during the first execution of the multiple CPU synchronous interrupt program (I45) after the DI instruction is cleared and Off during the second execution onwards. (no change after the second DI clear).</li> <li>This contact is enabled only for the multiple CPU synchronous interrupt program.</li> </ul>	S (Power-on to RUN or STOP to RUN)

\*1 The following figure shows an example of the process expected when the value for SD414 or SD415 is changed.

**Ex.**

If the value of SD414 is switched from 3 to 10:



## Fixed scan function information

The following is the list of special relay areas relating to the fixed scan function information.

No.	Name	Data stored	Details	Set by (setting timing)
SM480	Cycle overrun flag for inter-module synchronization program (I44)	Off: No error for the inter-module synchronization program (Normal) On: Error state for the inter-module synchronization program	<ul style="list-style-type: none"> <li>This relay switches to on when the inter-module synchronous interrupt program (I44) has not been completed within the specified inter-module synchronization cycle or the program cannot be executed due to various reasons, such as execution of a higher-priority interrupt program and interrupt disabling by the instruction execution.</li> <li>On state is maintained even after the program is later completed within the specified inter-module synchronization cycle (Clear by turning power off and on or resetting).</li> </ul>	S (Status change)
SM481	Cycle overrun flag for multiple CPU synchronization program (I45)	Off: No error for the multiple CPU synchronization program (Normal) On: Error state for the multiple CPU synchronization program	<ul style="list-style-type: none"> <li>This relay switches to on when the multiple CPU synchronization program (I45) has not been completed within the specified fixed scan communication cycle or the program cannot be executed due to various reasons, such as execution of a higher-priority interrupt program and interrupt disabling by the instruction execution.</li> <li>On state is maintained even after the program is later completed within the specified fixed scan communication cycle (Clear by turning power off and on or resetting).</li> </ul>	S (Status change)
SM484	Execution section excess error flag for multiple CPU synchronization interrupt program	Off: No execution section excess error for the multiple CPU synchronization interrupt program (Normal) On: Execution section excess error state for the multiple CPU synchronization interrupt program	<ul style="list-style-type: none"> <li>This relay switches to on when the program is executed exceeding the program execution section within the specified multiple CPU synchronization cycle.</li> <li>On state is maintained even after the program is later completed within the multiple CPU synchronous interrupt program execution section (cleared by turning power off and on or resetting).</li> </ul>	S (Status change)
SM488	Inter-module synchronization error (out of synchronization was detected on the CPU module)	Off: No error for the inter-module synchronization signal (Normal) On: Error state for the inter-module synchronization signal	<ul style="list-style-type: none"> <li>This relay switches to on when the inter-module synchronization signal cannot be identified within the inter-module synchronization cycle specified in the parameter or more than one signal has been identified within the same inter-module synchronization cycle.</li> <li>On state is maintained even after the inter-module synchronization signal can be later identified within the specified inter-module synchronization cycle (cleared by turning power off and on or resetting).</li> </ul>	S (Status change)

## Drive information

The following is the list of special relay areas relating to the drive information.

No.	Name	Data stored	Details	Set by (setting timing)
SM600	Memory card enabled/ disabled flag	Off: Disabled On: Enabled	This relay is on when an SD memory card is enabled (This relay switches to on when a valid SD memory card is attached and prepared for use).	S (Status change)
SM601	Memory card protect flag	Off: Not protected On: Protected	This relay is on when the write protect switch of the SD memory card is set to on.	S (Status change)
SM603	Memory card (drive 2) flag	Off: No SD memory card inserted On: SD memory card inserted	This relay is on when an SD memory card is attached (This relay switches to on when an SD memory card is attached regardless of its state (enabled/disabled) and type).	S (Status change)
SM604	Memory card in-use flag	Off: Not used On: In use	This relay is on when the SD memory card is being used.	S (Status change)
SM605	Memory card remove/attach prohibit flag	Off: Allowed to remove/attach On: Not allowed to remove/ attach	This relay can be switched to on when prohibiting the SD memory card from being removed/attached. When this relay is on, if SM607 is set to on, the system switches it to off.	U/S
SM606	SD memory card forced disable instruction	Off: Cancel instruction On: Forced disable instruction	<ul style="list-style-type: none"> <li>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.</li> <li>This relay can be switched to off to cancel the instruction that forces the SD memory card to be disabled.</li> </ul>	U
SM607	SD memory card forced disable state flag	Off: Not disabled by the SD memory card forced disable instruction On: Disabled by the SD memory card forced disable instruction	<ul style="list-style-type: none"> <li>This relay switches to on when the SD memory card is disabled by switching SM606 to on.</li> <li>This relay switches to off when the forcibly-disabled SD memory card is enabled by switching SM606 to off.</li> </ul>	S (Status change)
SM626	Extended SRAM cassette insertion flag	Off: Extended SRAM cassette is not attached On: Extended SRAM cassette is attached	This relay is on when the extended SRAM cassette is attached.	S (Status change)
SM628	Program memory write error	Off: Write error On: No write operation/ normal	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: Writing is in progress On: No 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	Number of rewriting operations error to program memory flag	Off: The number of rewriting operations reaches 100000 On: The number of rewriting operations is lower than 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: Write error On: No write operation/ normal	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: Writing is in progress On: No 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)

No.	Name	Data stored	Details	Set by (setting timing)
SM634	Number of rewriting operations error to data memory flag	Off: The number of rewriting operations reaches 100000 On: The number of rewriting operations is lower than 100000	This relay switches to on when the number of data memory rewriting operations reaches 100000 (CPU module must be replaced).	S (Writing)



## Instruction related

The following is the list of special relay areas relating to the instruction-related items.

No.	Name	Data stored	Details	Set by (setting timing)
SM699	Dedicated instruction skip flag	Off: Instruction is executing or completed On: Instruction has not been executed	<ul style="list-style-type: none"> <li>This relay indicates whether the built-in Ethernet function instruction (the SP.SOCOPEN/SP.SOCCLOSE/SP.SOCRCV/S.SOCRCVS/SP.SOCSND/SP.ECPRTCL 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.)</li> <li>This relay saves/returns when the interrupt program is being executed.</li> </ul>	S (Status change)
SM700	Carry flag	Off: Carry off On: Carry on	<ul style="list-style-type: none"> <li>This relay is a carry flag used while the application instruction is executing.</li> <li>This relay saves/returns when the interrupt program is being executed.</li> </ul>	S (Instruction execution)
SM701	Number of output characters selection	Off: Outputs until reaching NULL code On: Outputs 16 characters	<ul style="list-style-type: none"> <li>ASCII codes are output until reaching NULL (00H) code, when SM701 is off.</li> <li>16 characters of ASCII codes are output, when SM701 is on.</li> <li>This relay saves/returns when the interrupt program is being executed.</li> </ul>	U
SM702	Search method	Off: Sequential search On: Dichotomizing search	<ul style="list-style-type: none"> <li>The search method in the search instruction can be specified by using this relay.</li> <li>To use the dichotomizing search, data must be sorted.</li> <li>This relay saves/returns while the interrupt program is being executed.</li> </ul>	U
SM703	Sort order	Off: Ascending On: Descending	<ul style="list-style-type: none"> <li>This relay can be used to specify how to arrange data in the sort instruction: ascending or descending.</li> <li>This relay saves/returns while the interrupt program is being executed.</li> </ul>	U
SM704	Block comparison	Off: Mismatch is detected On: Completely match	<ul style="list-style-type: none"> <li>This relay switches to on when all the data conditions are met in the block data comparison instruction.</li> <li>This relay saves/returns while the interrupt program is being executed.</li> </ul>	S (Instruction execution)
SM709	DT/TM instruction improper data detection flag	Off: No improper data On: Improper data is detected	<ul style="list-style-type: none"> <li>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.</li> <li>This relay saves/returns while the interrupt program is being executed.</li> </ul>	S (Instruction execution)/U
SM752	Dedicated instruction End bit control flag	Off: End bit automatically controlled On: End bit 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 access in progress	Off: File access is not in progress On: File access is in progress	<ul style="list-style-type: none"> <li>This relay is on while the file access is in progress in the SP.FWRITE/SP.FREAD/SP.DEVST instructions.</li> <li>This relay is on while accessing the SD memory card and data memory.</li> </ul>	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SM754	BIN/DBIN instruction error control flag	Off: Executes error detection On: No execute error detection	This relay can be switched to on when the error detection is not desirable in the BIN/DBIN instruction.	U
SM755	Scaling data check settings	Off: Performs data check On: Not perform data check	This relay can be used to enable/disable the check whether the scaling data is sorted in ascending order when the SCL/DSCL/SCL2/DSCL2 instruction is being executed.	U
SM756	Module access completion wait control flag	Off: Not wait the completion On: Waits the completion	This relay can be used to determine whether the system must wait until the access is completed before starting the next instruction when the write access instruction to the buffer memory of another module is being executed.	U
SM775	Selection of refresh processing during the COM instruction execution	Off: Executes all the refresh processings On: Executes the refresh processing specified in SD775	This relay can be used to select the refresh processing target when the COM instruction is executed from two options: all refresh processing or only refresh processing specified in SD775.	U
SM776	Local device setting at CALL	Off: Disables local devices On: Enables local devices	This relay can be used to determine whether the local devices on the subroutine program called when the CALL instruction is being executed are enabled or not.	U
SM777	Local device setting in interrupt programs	Off: Disables local devices On: Enables local devices	This relay can be used to determine whether the local devices are enabled or not when the interrupt program is being executed.	U
SM792	PID bumpless processing (for the complete differentiation PIDCONT instruction)	Off: Matching On: Not matching	This relay can be used to specify whether to match SV to PV in manual mode.	U
SM794	PID bumpless processing (for the inexact differential S.PIDCONT instruction)	Off: Matching On: Not matching	This relay can be used to specify whether to match SV to PV in manual mode.	U
SM796	Number of used blocks information for the multiple CPU dedicated instruction (for CPU No.1)	Off: The specified number of blocks is reserved On: The number of blocks specified in SD796 is not reserved	<ul style="list-style-type: none"> <li>This relay switches to on when the number of remaining blocks in the dedicated instruction transfer area to be used in the multiple CPU dedicated instruction (target machine: CPU No.1) drops below the number of blocks specified in SD796. This relay is also on when the instruction is being executed.</li> <li>This relay switches to off when there exist free blocks during the END processing.</li> </ul>	S (Instruction execution/ During END)
SM797	Number of used blocks information for the multiple CPU dedicated instruction (for CPU No.2)	Off: The specified number of blocks is reserved On: The number of blocks specified in SD797 is not reserved	<ul style="list-style-type: none"> <li>This relay switches to on when the number of remaining blocks in the dedicated instruction transfer area to be used in the multiple CPU dedicated instruction (target machine: CPU No.2) drops below the number of blocks specified in SD797. This relay is also on when the instruction is being executed.</li> <li>This relay switches to off when there exist free blocks during the END processing.</li> </ul>	S (Instruction execution/ During END)
SM798	Number of used blocks information for the multiple CPU dedicated instruction (for CPU No.3)	Off: The specified number of blocks is reserved On: The number of blocks specified in SD798 is not reserved	<ul style="list-style-type: none"> <li>This relay switches to on when the number of remaining blocks in the dedicated instruction transfer area to be used in the multiple CPU dedicated instruction (target machine: CPU No.3) drops below the number of blocks specified in SD798. This relay is also on when the instruction is being executed.</li> <li>This relay switches to off when there exist free blocks during the END processing.</li> </ul>	S (Instruction execution/ During END)
SM799	Number of used blocks information for the multiple CPU dedicated instruction (for CPU No.4)	Off: The specified number of blocks is reserved On: The number of blocks specified in SD799 is not reserved	<ul style="list-style-type: none"> <li>This relay switches to on when the number of remaining blocks in the dedicated instruction transfer area to be used in the multiple CPU dedicated instruction (target machine: CPU No.4) drops below the number of blocks specified in SD799. This relay is also on when the instruction is being executed.</li> <li>This relay switches to off when there exist free blocks during the END processing.</li> </ul>	S (Instruction execution/ During END)

No.	Name	Data stored	Details	Set by (setting timing)
SM816*1	Hold mode	Off: Value not held On: Value held	Whether to hold the output value or not is specified when the input value is found to be exceeding the valid range during the range check processing of the S.IN instruction.	U
SM817*1	Hold mode	Off: Value not held On: Value held	This relay is used to specify whether or not to hold the output values of the S.OUT1, S.OUT2, and S.DUTY instructions if a sensor error occurs.	U

\*1 Only the Process CPU supports these special relay areas.

## Data logging function

The following is the list of special relay areas relating to the data logging function.

No.	Name	Data stored	Details	Set by (setting timing)
SM1200	Auto logging setting file and registration status	Off: Mismatch On: Matching	<ul style="list-style-type: none"> <li>This relay is on when the auto logging setting executed (registered) matches with the content of the configuration file stored in the target memory. This relay is off when the setting does not match with the content.</li> </ul>	S (Status change)
SM1201	SD memory card setting file in use flag	Off: Not used On: In use	<ul style="list-style-type: none"> <li>This relay switches to on when the data logging configuration file stored in an SD memory card is being used. This relay switches to on when one or more data logging of the settings No.1 to 10 is registered.</li> <li>On state is maintained even when the data logging later enters into suspend/waiting for start without collection/waiting for RUN without collection mode. However, the relay switches to off when all the data logging stops.</li> </ul>	S (Status change)
SM1202	Data memory setting file in use flag	Off: Not used On: In use	<ul style="list-style-type: none"> <li>This relay switches to on when the data logging configuration file stored in the data memory is being used. This relay switches to on when one or more data logging of the settings No.1 to 10 is registered.</li> <li>On state is maintained even when the data logging later enters into suspend/waiting for start without collection/waiting for RUN without collection mode. However, the relay switches to off when all the data logging stops.</li> </ul>	S (Status change)
SM1210	Data logging setting No.1 Data logging preparation	Off: Not prepared On: Prepared	<ul style="list-style-type: none"> <li>This relay switches to on when the data logging preparation is completed.</li> <li>On state is maintained even when the data logging later enters into suspend/waiting for start without collection/waiting for RUN without collection mode. This relay switches to off when the data logging stops.</li> </ul>	S (Initial)
SM1211	Data logging setting No.1 data logging start	Off: Suspended/waiting for start On: Start	<ul style="list-style-type: none"> <li>This relay switches to on when the data logging starts. The relay turns off when the data logging is in suspend/waiting for start without collection mode. Associated special relay, such as the data logging data collection in progress, data logging completion, data logging triggering, and post data logging triggering, switches to off simultaneously.</li> <li>This relay also switches to off when the CPU module is moved from RUN to STOP mode and therefore the data collection is halted.</li> </ul>	S (Status change)
SM1212	Data logging setting No.1 data logging data collection in progress	Off: No data logging collection operation On: Data logging data collection in progress	This relay switches to on when the data logging starts data collection.	S (Status change)
SM1213	Data logging setting No.1 data logging completion	Off: Data logging is not completed On: Data logging is completed	This relay switches to on when the data logging is completed. For continuous logging, the corresponding bit switches to on when writing reaches the maximum number of storage files and data logging is completed (with "Stop" set for the operation at the time when the number of save files exceeds the limit). For trigger logging, the corresponding bit switches to on when data collection for the specified number of records has been completed followed by writing into the SD memory card after the trigger condition was satisfied. The bit also switches to on when an error is generated (except for data logging error caused by online program change) during the data logging execution.	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SM1214	Data logging setting No.1 Data logging triggering	Off→On: triggering occurs	The system switches this relay to on when the specified trigger condition is satisfied.	S (Status change)
SM1215	Data logging setting No.1 Post data logging triggering	Off: Not post triggering On: Post triggering	<ul style="list-style-type: none"> <li>This relay switches to on once data logging triggering occurs. On state is maintained even when the data logging is completed. However, this relay is off when the data logging is in suspend/waiting for start without collection/stop mode</li> <li>This relay also switches to off when the CPU module is moved from RUN to STOP mode and therefore the data collection is halted.</li> </ul>	S (Status change)
SM1216	Data logging setting No.1 Data logging error	Off: No error On: Error	<ul style="list-style-type: none"> <li>This relay switches to on when a data logging function error is generated.</li> <li>This relay switches to off when the setting is registered or by the stop command from CPU Module Logging Configuration Tool.</li> </ul>	S (Status change)
SM1217	Data logging setting No.1 Data logging data saving into memory card in progress	Off: Data saving is not in progress On: Data saving is in progress	This relay switches to on when data in the internal buffer are being saved into an SD memory card as a part of the data logging process.	S (Status change)
SM1218	Data logging setting No.1 Logging data storage file switching in progress	Off: Storage file switching is not in progress On: Storage file switching is in progress	This relay switches to on when storage file switching is in progress.	S (Status change)
SM1220 to SM1228	Data logging setting No.2	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1220 to SM1228).	Same configuration as the setting No.1
SM1230 to SM1238	Data logging setting No.3	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1230 to SM1238).	Same configuration as the setting No.1
SM1240 to SM1248	Data logging setting No.4	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1240 to SM1248).	Same configuration as the setting No.1
SM1250 to SM1258	Data logging setting No.5	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1250 to SM1258).	Same configuration as the setting No.1
SM1260 to SM1268	Data logging setting No.6	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1260 to SM1268).	Same configuration as the setting No.1
SM1270 to SM1278	Data logging setting No.7	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1270 to SM1278).	Same configuration as the setting No.1
SM1280 to SM1288	Data logging setting No.8	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1280 to SM1288).	Same configuration as the setting No.1
SM1290 to SM1298	Data logging setting No.9	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1290 to SM1298).	Same configuration as the setting No.1
SM1300 to SM1308	Data logging setting No.10	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SM1300 to SM1308).	Same configuration as the setting No.1
SM1312 to SM1321	Data logging setting No.1 to 10 Data logging suspend/resume flag	Off→On: Suspend On→Off: Resume	<ul style="list-style-type: none"> <li>When this relay changes from off to on, the data logging function is suspended. In an off state of the data logging start SM, no processing is performed.</li> <li>When this relay changes from on to off, the data logging function resumes. In an on state of the data logging start SM, no processing is performed.</li> </ul>	U

## Memory dump function

The following is the list of special relay areas relating to the memory dump function. (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SM1472	Memory dump in progress	Off: Memory dump not executed On: Memory dump in progress	Turns on if memory dump is in progress, and turns off if memory dump is not executed.	S (Status change)
SM1473	Memory dump completed	Off: Memory dump not completed On: Memory dump completed	Turns on at the time when collection of the data and save to the SD memory card are completed after the establishment of the trigger condition. Thereafter, the on state remains even if memory dump is executed again. If the trigger condition is established again, On changes to Off.	S (Status change)

## 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	<ul style="list-style-type: none"> <li>When this relay is changed from off to on, IP address settings stored in SD1520 to SD1525 are written to the IP address storage area (system memory) of the CPU module.</li> <li>Once write to the IP address storage area (system memory) is completed (regardless of whether it was successful or not), this relay is turned off.</li> </ul>	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) failed, and off when it was successful.	S (Status change)
SM1522	IP address storage area clear request	Off→On: Clear request exists On→Off: Clear completed	When this relay is changed from off to on, the IP address storage area (system memory) is cleared. Once clear of the IP address storage area (system memory) 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) failed, and off when it was successful.	S (Status change)
SM1524	Initial processing successful completion state	Off: The initial processing was completed successfully. On: The initial processing was not completed successfully.	This relay is turned on when the initial processing for the Ethernet function was completed successfully. 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: The initial processing was completed with an error. On: The initial processing was not completed with an error.	This relay is turned on when the initial processing for the Ethernet function was completed with an error. When the initial processing was completed successfully, SM1524 is turned on and SM1525 stays off.	S (Status change)

## Online module change function

The following is the list of special relay areas relating to the online module change function. (Process CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SM1600	Module selection request flag	Off→On: Module selection 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: Module being 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: Module removal 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 ready 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: Module recognition 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: Module control resumption 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: Processing not completed On: Processing 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)

No.	Name	Data stored	Details	Set by (setting timing)
SM1616	Online module change availability flag	Off: Disabled On: Enabled	This relay turns on when only the Process CPU is used in a single CPU system. In a multiple CPU system, if any of the CPU modules other than the Process CPU does not support the online module change function, the relay turns off. If all the CPU modules support the function, the relay turns on. Note that the setting details in the direct change setting cannot be checked with SM1616. To check the direct change setting, check the CPU parameters.	S (Initial)
SM1617	Online module change status flag	Off: Function not executed On: Function being executed	This relay turns on when SM1600 is turned on to start the online module change processing. The relay turns off upon completion of the online module change processing.	S (Status change)
SM1618	Online module change error flag	Off: No error On: Error	This relay turns on when an error is detected. This relay turns off when the error cause is eliminated and the online module change related request is executed. An error occurs in selecting a module. Thus, turn off the relay before module selection.	S (Status change)/U (Request)
SM1619	Disable request flag during online module change	Off: No disable request On: Disable request detected	This relay turns on when a disable request is issued during the online module change processing. The relay turns off upon completion of the online module change processing.	S (Status change)



# 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> <ul style="list-style-type: none"><li>• S: System</li><li>• U: sets at the user side (program, engineering tool, GOT, other testing operations from external device).</li><li>• U/S: User and system</li></ul> <Set timing> <ul style="list-style-type: none"><li>• Every END: Data is set every time END processing is performed.</li><li>• Initial: Data is set when initial processing is performed (e.g. powering on the system, changing the operating status from STOP to RUN).</li><li>• Status change: Data is set when the status is changed.</li><li>• Error: Data is set when an error occurs.</li><li>• Instruction execution: Data is set when an instruction is executed.</li><li>• Request: Data is set when requested by a user (using the special relay).</li><li>• Switch change: Data is set when the switch of the CPU module is changed.</li><li>• Card insertion/removal: Data is set when an SD memory card is inserted or removed.</li><li>• Writing: Data is set when a user performs a writing operation.</li><li>• During END: Data is set when END processing is performed.</li></ul>

**Point** 

Do not change the data set by the system in a program or by a device test. Doing so may result in system down or communication failure.

## Diagnostic information

The following is the list of special register areas relating to the diagnostic information.

No.	Name	Data stored	Details	Set by (setting timing)
SD0	Latest self diagnostics error code	Latest self diagnostics error code	<ul style="list-style-type: none"> <li>• Error codes are stored in a hexadecimal value when the diagnostics detects an error.</li> <li>• This information is the same as the latest item in the event history.</li> </ul>	S (Error)
SD1	Clock time for self diagnosis error occurrence	Clock time for self diagnosis error occurrence	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)	

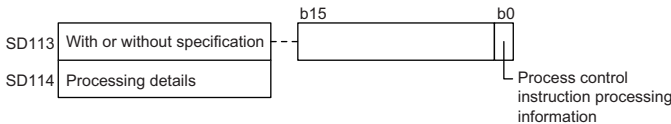
No.	Name	Data stored	Details	Set by (setting timing)
SD10	Self diagnostics error code	Self diagnostics error code 1	The maximum of 16 types of error codes are stored into SD10 onwards when the diagnostics detects an error (The same error code as one already stored in SD10 onwards is not stored). The 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 diagnostics error code 2		
SD12		Self diagnostics error code 3		
SD13		Self diagnostics error code 4		
SD14		Self diagnostics error code 5		
SD15		Self diagnostics error code 6		
SD16		Self diagnostics error code 7		
SD17		Self diagnostics error code 8		
SD18		Self diagnostics error code 9		
SD19		Self diagnostics error code 10		
SD20		Self diagnostics error code 11		
SD21		Self diagnostics error code 12		
SD22		Self diagnostics error code 13		
SD23		Self diagnostics error code 14		
SD24		Self diagnostics error code 15		
SD25		Self diagnostics error code 16		
SD53	AC/DC DOWN	The number of AC/DC DOWN detections	The value of this register increments by one and stored as a BIN code, each time input voltage drops to 85% (AC power)/65% (DC power) or less of the nominal value while the CPU module is carrying out an operation. A counting cycle from 0 to 65535 to 0 is repeated.	S (Error)
SD60	Number of modules with fuse blown	Number of modules with fuse blown	The lowest number of module in which a fuse blew is stored.	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.	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)

No.	Name	Data stored	Details	Set by (setting timing)
SD64 to SD79	Table of detected annunciator numbers	Detected annunciator number	<ul style="list-style-type: none"> <li>Numbers of activated annunciators are sequentially registered into SD64 to SD79, each time an annunciator (F) is turned on by the SET F instruction.</li> <li>Number of the annunciator deactivated by the RST F instruction is removed from registers SD64 to SD79, and the numbers of annunciators which were lined up behind the removed one move forward one by one. When the number of detected annunciators has already reached 16, the number of a newly detected annunciator is not stored into SD64 to SD79.</li> </ul> <p>For details on the annunciator, refer to the annunciator (F) (Page 227 Annunciator (F))</p>	S (Instruction execution)
SD80	Detailed information 1 information category	Detailed information 1 information category code	<ul style="list-style-type: none"> <li>Detailed information 1 information category code is stored.  b15 to b8 b7 to b0  <span style="border: 1px solid black; padding: 2px;">Not used (fixed to 0)</span> <span style="border: 1px solid black; padding: 2px;">Information block code</span> </li> <li>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</li> </ul>	S (Error)

No.	Name	Data stored	Details	Set by (setting timing)
SD81 to SD111	Detailed information 1	Detailed information 1	<ul style="list-style-type: none"> <li>Detailed information 1 corresponding to the error code (SD0) is stored.</li> <li>There are seven types of information to be stored as shown in the following figures:</li> <li>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), and (24)).</li> </ul> <p>(1) Error location information</p> <p>(2) Drive number and file name</p> <p>(4) Parameter information</p>	S (Error)

No.	Name	Data stored	Details	Set by (setting timing)		
SD81 to SD111	Detailed information 1	Detailed information 1	<p>(5) System configuration information</p> <p>(6) Number of times information When there is no specification, 0 is set to each SD.</p> <p>(7) Time information When there is no specification, 0 is set to each SD.</p> <p>(24) Failure information Failure information is system information.</p>	S (Error)		
SD112	Detailed information 2 information category	Detailed information 2 information category code	<ul style="list-style-type: none"> <li>Detailed information 2 information category code is stored.</li> </ul> <p>b15 to b8 b7 to b0</p> <table border="1"> <tr> <td>Not used (fixed to 0)</td> <td>Information block code</td> </tr> </table> <ul style="list-style-type: none"> <li>The following codes are stored into the information category code.</li> </ul> <p>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</p>	Not used (fixed to 0)	Information block code	S (Error)
Not used (fixed to 0)	Information block code					

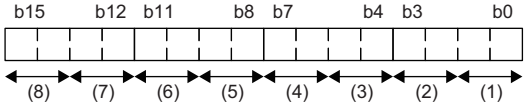
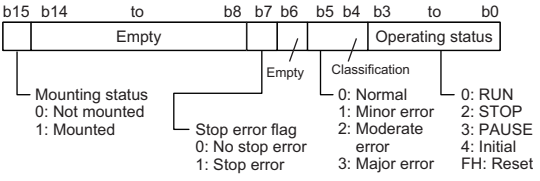
No.	Name	Data stored	Details	Set by (setting timing)
SD113 to SD143	Detailed information 2	Detailed information 2	<ul style="list-style-type: none"> <li>Detailed information 2 corresponding to the error code (SD0) is stored.</li> <li>There are five types of information to be stored.</li> <li>The type of information can be checked in SD112. (The code stored in SD112 corresponds to the following (2) to (5), and (25).)</li> </ul> <p>(2) Drive number and file name</p> <p>(3) Annunciator number</p> <p>(4) Parameter information</p> <p>(5) System configuration information</p>	S (Error)

No.	Name	Data stored	Details	Set by (setting timing)
SD113 to SD143	Detailed information 2	Detailed information 2	<p>(25) Process control instruction processing information</p>  <p>The following are processing blocks stored in SD114 (Processing details).</p> <ol style="list-style-type: none"> <li>1: — (Instruction with no processing block)</li> <li>2: Range check</li> <li>3: Input limiter</li> <li>4: Engineering value inverse transformation</li> <li>5: Digital filter</li> <li>6: Input addition</li> <li>7: Variation rate &amp; upper/lower limiter</li> <li>8: Reset windup</li> <li>9: Output conversion</li> <li>10: Output ON time conversion</li> <li>11: Variation rate check</li> <li>12: Integrated value calculation</li> <li>13: Control cycle determination</li> <li>14: SV setting</li> <li>15: Tracking</li> <li>16: Gain (Kp) operation</li> <li>17: PID operation</li> <li>18: Deviation check</li> <li>19: PID operation 1 (operation processing of Bn or Cn)</li> <li>20: PID operation 2 (operation processing of Dn)</li> <li>21: PID operation 3 (operation processing of ΔMV)</li> <li>22: PIDP operation</li> <li>23: Operating time monitor</li> <li>24: SPI operation</li> <li>25: IPD operation</li> <li>26: BPI operation</li> <li>27: Engineering value transformation</li> <li>28: Variation rate limiter</li> <li>29: Ratio calculation</li> <li>30: Upper/lower limit check</li> <li>31: Loop stop</li> <li>32: MV correction</li> <li>33: Two-position (on/off) control</li> <li>34: Three-position (on/off) control</li> <li>35: Operation constant check</li> <li>36: SV count-up</li> <li>37: MVLGS operation</li> <li>38: Output</li> <li>39: Input check</li> <li>40: Timeout determination</li> <li>41: Step manipulated value set</li> <li>42: Sampling cycle determination</li> <li>43: Response waveform observation</li> <li>44: Identification processing</li> <li>45: PID constants calculation</li> </ol>	S (Error)



## 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	Switch status	CPU switch status	The switch status of the CPU module is stored as follows: 0: RUN, 1: STOP	S (Switch change)
SD201	LED status	CPU-LED state	This register stores the information that indicates LED status (0: off, 1: on, 2: flashing) of the CPU module in the following bit patterns.   (1)READY (2)ERROR (3)PROGRAM RUN (4)USER (5)BATTERY (6)CARD READY (7)CARD ACCESS (8)FUNCTION	S (Status change)
SD203	CPU module operating status	CPU module operating status	The operating status of the CPU module is stored as follows: 0: RUN, 2: STOP, 3: PAUSE	S (Every END)
SD210	Clock data	Clock data (year)	The year value (four digits) of the clock data is stored as a BIN code.	S/U (Request)
SD211		Clock data (month)	The month value of the clock data is stored as a BIN code.	S/U (Request)
SD212		Clock data (day)	The day value of the clock data is stored as a BIN code.	S/U (Request)
SD213		Clock data (hour)	The hour value of the clock data is stored as a BIN code.	S/U (Request)
SD214		Clock data (minute)	The minute value of the clock data is stored as a BIN code.	S/U (Request)
SD215		Clock data (second)	The second value of the clock data is stored as a BIN code.	S/U (Request)
SD216		Clock data (day of the week)	The day of the week value of the clock data is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)	S/U (Request)
SD218	Time zone setting value	Time zone (in minutes)	The time zone setting value specified in the parameter is stored in increments of minutes. Example: when the setting value of the time zone is "UTC+9," the result of the following equation is stored into SD218: $9 \times 60$ (minutes) = 540.	S (Initial)
SD228	Multiple CPU system information	Number of CPU modules	The number of CPU modules which constitute a multiple CPU system is stored (one to four, including those reserved).	S (Initial)
SD229		CPU module number in multiple CPU system	The number of this CPU No. is stored when a multiple CPU system is configured.	S (Initial)
SD230		CPU No.1 operating status	The operation information for each CPU No. is stored (the amount of stored information depending on the number of CPU modules indicated in SD228 is stored).  	S (During END/Error)
SD231		CPU No.2 operating status		
SD232	CPU No.3 operating status			
SD233	CPU No.4 operating status			
SD241	Number of extension base units	0: base unit only 1 to 7: number of extension base units	The maximum number of implemented extension base units is stored.	S (Initial)

No.	Name	Data stored	Details	Set by (setting timing)
SD242	Identification for whether or not Q series module can be mounted	Identification of the base type 0: Q series module cannot be mounted (There is no base unit that can mount the Q series unit) 1: Q series module can be mounted (There is a base unit that can mount the Q series unit)	Identify whether or not Q series module can be mounted. When no SD memory card is inserted, the value is fixed to 0.  	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.  	S (Initial)
SD250	Latest I/O for implemented unit	Latest I/O number for implemented unit	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 SD261	Number of points assigned to bit devices	X number of points assigned (L)	The number of points of the device X currently set is stored as 32-bit data.	S (Initial)
		X number of points assigned (H)		
SD262 SD263		Y number of points assigned (L)	The number of points of the device Y currently set is stored as 32-bit data.	S (Initial)
		Y number of points assigned (H)		
SD264 SD265		M number of points assigned (L)	• The number of points of the device M currently set is stored as 32-bit data. • The number of points assigned is stored even when the number of points assigned to M is 32K points or less.	S (Initial)
		M number of points assigned (H)		
SD266 SD267		B number of points assigned (L)	• The number of points of the device B currently set is stored as 32-bit data. • The number of points assigned is stored even when the number of points assigned to B is 32K points or less.	S (Initial)
		B number of points assigned (H)		
SD268 SD269		SB number of points assigned (L)	• The number of points of the device SB currently set is stored as 32-bit data. • The number of points assigned is stored even when the number of points assigned to SB is 32K points or less.	S (Initial)
		SB number of points assigned (H)		
SD270 SD271		F number of points assigned (L)	The number of points of the device F currently set is stored as 32-bit data.	S (Initial)
		F number of points assigned (H)		
SD272 SD273		V number of points assigned (L)	The number of points of the device V currently set is stored as 32-bit data.	S (Initial)
		V number of points assigned (H)		
SD274 SD275		L number of points assigned (L)	The number of points of the device L currently set is stored as 32-bit data.	S (Initial)
		L number of points assigned (H)		

No.	Name	Data stored	Details	Set by (setting timing)
SD280	Word device number of points assigned	D number of points assigned (L)	• The number of points of the device D currently set is stored as 32-bit data.	S (Initial)
SD281		D number of points assigned (H)	• The number of points assigned is stored even when the number of points assigned to D is 32K points or less.	
SD282		W number of points assigned (L)	• The number of points of the device W currently set is stored as 32-bit data.	S (Initial)
SD283		W number of points assigned (H)	• The number of points assigned is stored even when the number of points assigned to W is 32K points or less.	
SD284		SW number of points assigned (L)	• The number of points of the device SW currently set is stored as 32-bit data.	S (Initial)
SD285		SW number of points assigned (H)	• The number of points assigned is stored even when the number of points assigned to SW is 32K points or less.	
SD288	Timer-type device number of points assigned	T number of points assigned (L)	• The number of points of the device T currently set is stored as 32-bit data.	S (Initial)
SD289		T number of points assigned (H)	• The number of points assigned is stored even when the number of points assigned to T is 32K points or less.	
SD290		ST number of points assigned (L)	• The number of points of the device ST currently set is stored as 32-bit data.	S (Initial)
SD291		ST number of points assigned (H)	• The number of points assigned is stored even when the number of points assigned to ST is 32K points or less.	
SD292		C number of points assigned (L)	• The number of points of the device C currently set is stored as 32-bit data.	S (Initial)
SD293		C number of points assigned (H)	• The number of points assigned is stored even when the number of points assigned to C is 32K points or less.	
SD294		LT number of points assigned (L)	• The number of points of the device LT currently set is stored as 32-bit data.	S (Initial)
SD295		LT number of points assigned (H)	• The number of points assigned is stored even when the number of points assigned to LT is 32K points or less.	
SD296		LST number of points assigned (L)	• The number of points of the device LST currently set is stored as 32-bit data.	S (Initial)
SD297		LST number of points assigned (H)	• The number of points assigned is stored even when the number of points assigned to LST is 32K points or less.	
SD298		LC number of points assigned (L)	• The number of points of the device LC currently set is stored as 32-bit data.	S (Initial)
SD299		LC number of points assigned (H)	• The number of points assigned is stored even when the number of points assigned to LC is 32K points or less.	
SD300	Number of points assigned to index registers	Z number of points assigned	The number of points of the device Z currently set is stored.	S (Initial)
SD302	Number of points assigned to long index registers	LZ number of points assigned	The number of points of the device LZ currently set is stored.	S (Initial)
SD306	Number of points assigned to file registers	ZR number of points assigned (L)	• The number of points of the device ZR currently set is stored as 32-bit data.	S (Initial)
SD307		ZR number of points assigned (H)	• The number of points assigned is stored even when the number of points assigned to ZR is 32K points or less.	
SD308	Number of points assigned to refresh devices	RD number of points assigned (L)	• The number of points of the device RD currently set is stored as 32-bit data.	S (Initial)
SD309		RD number of points assigned (H)	• The number of points assigned is stored even when the number of points assigned to RD is 32K points or less.	
SD312	File register block number	File register block number	The block number for 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.	<ul style="list-style-type: none"> <li>The value in this register increments by one for each second after the CPU module enters in RUN mode.</li> <li>A counting cycle from 0 to 65535 to 0 is repeated.</li> </ul>	S (Status change)
SD414	2n second clock setting	Unit setting for 2n second clock	<ul style="list-style-type: none"> <li>n value of the 2n second clock is stored (Default: 30).</li> <li>Configurable range is -32768 to 32767 (0 to FFFFH).</li> </ul>	U
SD415	2nms second clock setting	Unit setting for 2nms second clock	<ul style="list-style-type: none"> <li>n value of the 2nms second clock is stored (Default: 30).</li> <li>Configurable range is -32768 to 32767 (0 to FFFFH).</li> </ul>	U
SD420	Scan counter	The number of counts that is counted once for each scan.	<ul style="list-style-type: none"> <li>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).</li> <li>A counting cycle from 0 to 65535 to 0 is repeated.</li> </ul>	S (Every END)

## Fixed scan function information

The following is the list of special register areas relating to the fixed scan function information.

No.	Name	Data stored	Details	Set by (setting timing)
SD480	Number of cycle overrun events for inter-module synchronization cycle program (I44)	0: No cycle overrun event 1~65535: the accumulated number of cycle overrun events	The number of events in which the inter-module synchronous interrupt program (I44) has not been completed within the inter-module synchronization cycle or the program cannot be executed due to various reasons, such as execution of a higher-priority interrupt program and interrupt disabling by the instruction execution is stored. When the count exceeds 65535, it returns to 0 and starts a new cycle. The number of cycle overrun events is counted regardless of the setting content for the error check setting of the RAS setting (execution check of the inter-module synchronous interrupt (I44)).	S (Status change)
SD481	Number of cycle overrun events for multiple CPU synchronization program (I45)	0: No cycle overrun event 1~65535: the accumulated number of cycle overrun events	The number of events in which the multiple CPU synchronization program (I45) has not been completed within the fixed scan communication cycle or the program cannot be executed due to various reasons, such as execution of a higher-priority interrupt program and interrupt disabling by the instruction execution is stored. When the count exceeds 65535, it returns to 0 and starts a new cycle. The number of cycle overrun events is counted regardless of the setting content for the error check setting of the RAS setting (execution check of the multiple CPU synchronization program (I45)).	S (Status change)
SD484	Number of execution section excess errors for multiple CPU synchronization interrupt program	0: no execution section excess error for the multiple CPU synchronization interrupt program (Normal) 1 to 65535: the accumulated number of execution section excess errors for the multiple CPU synchronization interrupt program	The number of events in which the program is executed exceeding the program execution section within the specified multiple CPU synchronization cycle is stored. When the count exceeds 65535, it returns to 0 and starts a new cycle. Note that the number of error occurrences is counted regardless of the CPU module operation setting for error detections within the RAS setting of the CPU parameter.	S (Status change)
SD500	Execution program number	The execution type of the program being executed	The program number which is currently executed is stored as a BIN value.	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)
SD518	Initial scan time	Initial scan time (unit: ms)	<ul style="list-style-type: none"> <li>The initial scan time is stored into SD518 and SD519 (it is measured in increments of <math>\mu</math>s).</li> <li>SD518: stores a value in the ms place (storage range: 0 to 65535)</li> <li>SD519: stores a value in the <math>\mu</math>s place (storage range: 0 to 999)</li> <li>This register is cleared to 0 when the mode transfers from STOP to RUN mode.</li> </ul>	S (Every END)
SD519		Initial scan time (unit: $\mu$ s)		
SD520	Current scan time	Current scan time (unit: ms)	<ul style="list-style-type: none"> <li>The current scan time is stored into SD520 and SD521 (it is measured in increments of <math>\mu</math>s).</li> <li>SD520: stores a value in the ms place (storage range: 0 to 65535)</li> <li>SD521: stores a value in the <math>\mu</math>s place (storage range: 0 to 999)</li> <li>Example: If the current scan time is 23.6ms, the following values are stored: SD520 = 23 SD521 = 600</li> <li>This register is cleared to 0 when the mode transfers from STOP to RUN mode.</li> </ul>	S (Every END)
SD521		Current scan time (unit: $\mu$ s)		
SD522	Minimum scan time	Minimum scan time (unit: ms)	<ul style="list-style-type: none"> <li>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 <math>\mu</math>s).</li> <li>SD522: stores a value in the ms place (storage range: 0 to 65535)</li> <li>SD523: stores a value in the <math>\mu</math>s place (storage range: 0 to 999)</li> <li>This register is cleared to 0 when the mode transfers from STOP to RUN mode.</li> </ul>	S (Every END)
SD523		Minimum scan time (unit: $\mu$ s)		
SD524	Maximum scan time	Maximum scan time (unit: ms)	<ul style="list-style-type: none"> <li>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 <math>\mu</math>s).</li> <li>SD524: stores a value in the ms place (storage range: 0 to 65535)</li> <li>SD525: stores a value in the <math>\mu</math>s place (storage range: 0 to 999)</li> <li>This register is cleared to 0 when the mode transfers from STOP to RUN mode.</li> </ul>	S (Every END)
SD525		Maximum scan time (unit: $\mu$ s)		
SD526	END processing time	END processing time (unit: ms)	<ul style="list-style-type: none"> <li>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 <math>\mu</math>s).</li> <li>SD526: stores a value in the ms place (storage range: 0 to 65535)</li> <li>SD527: stores a value in the <math>\mu</math>s place (storage range: 0 to 999)</li> <li>This register is cleared to 0 when the mode transfers from STOP to RUN mode.</li> </ul>	S (Every END)
SD527		END processing time (unit: $\mu$ s)		
SD528	Constant scan waiting time	Constant scan waiting time (unit: ms)	<ul style="list-style-type: none"> <li>The waiting time specified in the constant scan setting process is stored into SD528 and SD529 (it is measured in increments of <math>\mu</math>s).</li> <li>SD528: stores a value in the ms place (storage range: 0 to 65535)</li> <li>SD529: stores a value in the <math>\mu</math>s place (storage range: 0 to 999)</li> <li>This register is cleared to 0 when the mode transfers from STOP to RUN mode.</li> </ul>	S (Every END)
SD529		Constant scan waiting time (unit: $\mu$ s)		
SD530	Scan program execution time	Scan program execution time (unit: ms)	<ul style="list-style-type: none"> <li>The execution time of the scan program for one scan is stored into SD530 and SD531 (it is measured in increments of <math>\mu</math>s).</li> <li>SD530: stores a value in the ms place (storage range: 0 to 65535)</li> <li>SD531: stores a value in the <math>\mu</math>s place (storage range: 0 to 999)</li> <li>This register is cleared to 0 when the mode transfers from STOP to RUN mode.</li> </ul>	S (Every END)
SD531		Scan program execution time (unit: $\mu$ s)		

## 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	<p>This register indicates the type of mounted SD memory cards.</p>	S (Initial, card insertion/removal)																
SD604	SD memory card (drive 2) usage status	SD memory card (drive 2) usage status	<p>Usage status of the SD memory card is stored using the following bit pattern. (On indicates being used.)</p> <table border="1"> <tr> <td>b0: Event history</td> <td>b8: Not used</td> </tr> <tr> <td>b1: Module extension parameter<sup>*1</sup></td> <td>b9: Not used</td> </tr> <tr> <td>b2: Label communication data</td> <td>b10: Not used</td> </tr> <tr> <td>b3: Not used</td> <td>b11: Not used</td> </tr> <tr> <td>b4: Not used</td> <td>b12: Not used</td> </tr> <tr> <td>b5: Not used</td> <td>b13: Not used</td> </tr> <tr> <td>b6: Not used</td> <td>b14: Not used</td> </tr> <tr> <td>b7: Not used</td> <td>b15: Not used</td> </tr> </table>	b0: Event history	b8: Not used	b1: Module extension parameter <sup>*1</sup>	b9: Not used	b2: Label communication data	b10: Not used	b3: Not used	b11: Not used	b4: Not used	b12: Not used	b5: Not used	b13: Not used	b6: Not used	b14: Not used	b7: Not used	b15: Not used	S (Status change)
b0: Event history	b8: Not used																			
b1: Module extension parameter <sup>*1</sup>	b9: Not used																			
b2: Label communication data	b10: Not used																			
b3: Not used	b11: Not used																			
b4: Not used	b12: Not used																			
b5: Not used	b13: Not used																			
b6: Not used	b14: Not used																			
b7: Not used	b15: Not used																			
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	<p>Usage status of the device/label memory is stored using the following bit pattern. (On indicates being used.)</p> <table border="1"> <tr> <td>b0: File register</td> <td>b8: Not used</td> </tr> <tr> <td>b1: Not used</td> <td>b9: Not used</td> </tr> <tr> <td>b2: Not used</td> <td>b10: Not used</td> </tr> <tr> <td>b3: Not used</td> <td>b11: Not used</td> </tr> <tr> <td>b4: Not used</td> <td>b12: Not used</td> </tr> <tr> <td>b5: Not used</td> <td>b13: Not used</td> </tr> <tr> <td>b6: Not used</td> <td>b14: Not used</td> </tr> <tr> <td>b7: Not used</td> <td>b15: Not used</td> </tr> </table>	b0: File register	b8: Not used	b1: Not used	b9: Not used	b2: Not used	b10: Not used	b3: Not used	b11: Not used	b4: Not used	b12: Not used	b5: Not used	b13: Not used	b6: Not used	b14: Not used	b7: Not used	b15: Not used	S (Status change)
b0: File register	b8: Not used																			
b1: Not used	b9: Not used																			
b2: Not used	b10: Not used																			
b3: Not used	b11: Not used																			
b4: Not used	b12: Not used																			
b5: Not used	b13: Not used																			
b6: Not used	b14: Not used																			
b7: Not used	b15: Not used																			
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)																

No.	Name	Data stored	Details	Set by (setting timing)																
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.)  <table border="1"> <tr> <td>b0: Event history</td> <td>b8: Not used</td> </tr> <tr> <td>b1: Module extension parameter*1</td> <td>b9: Not used</td> </tr> <tr> <td>b2: Label communication data</td> <td>b10: Not used</td> </tr> <tr> <td>b3: Not used</td> <td>b11: Not used</td> </tr> <tr> <td>b4: Not used</td> <td>b12: Not used</td> </tr> <tr> <td>b5: Not used</td> <td>b13: Not used</td> </tr> <tr> <td>b6: Not used</td> <td>b14: Not used</td> </tr> <tr> <td>b7: Not used</td> <td>b15: Not used</td> </tr> </table>	b0: Event history	b8: Not used	b1: Module extension parameter*1	b9: Not used	b2: Label communication data	b10: Not used	b3: Not used	b11: Not used	b4: Not used	b12: Not used	b5: Not used	b13: Not used	b6: Not used	b14: Not used	b7: Not used	b15: Not used	S (Status change)
b0: Event history	b8: Not used																			
b1: Module extension parameter*1	b9: Not used																			
b2: Label communication data	b10: Not used																			
b3: Not used	b11: Not used																			
b4: Not used	b12: Not used																			
b5: Not used	b13: Not used																			
b6: Not used	b14: Not used																			
b7: Not used	b15: Not used																			
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	S (Initial)																
SD629	Program memory write (transfer) status	Write (transfer) status display (percent)	This register displays write (transfer) status to the program memory in percentage (0 to 100%). The initial value is "0". Upon completion of writing, this register is set to "100". It is set to "0" at the time when the write command is issued.	S (Writing)																
SD630	Index for the number of program memory write operations	Index of the number of write operations up to now	<ul style="list-style-type: none"> <li>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.</li> <li>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.</li> </ul>	S (Writing)																
SD631																				
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	Index for the number of data memory write operations	Index of the number of write operations up to now	<ul style="list-style-type: none"> <li>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.</li> <li>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.</li> </ul>	S (Writing)																
SD635																				
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.)  <table border="1"> <tr> <td>b0: Used for real-time monitor</td> <td>b8: Not used</td> </tr> <tr> <td>b1: Not used</td> <td>b9: Not used</td> </tr> <tr> <td>b2: Not used</td> <td>b10: Not used</td> </tr> <tr> <td>b3: Not used</td> <td>b11: Not used</td> </tr> <tr> <td>b4: Not used</td> <td>b12: Not used</td> </tr> <tr> <td>b5: Not used</td> <td>b13: Not used</td> </tr> <tr> <td>b6: Not used</td> <td>b14: Not used</td> </tr> <tr> <td>b7: Not used</td> <td>b15: Not used</td> </tr> </table>	b0: Used for real-time monitor	b8: Not used	b1: Not used	b9: Not used	b2: Not used	b10: Not used	b3: Not used	b11: Not used	b4: Not used	b12: Not used	b5: Not used	b13: Not used	b6: Not used	b14: Not used	b7: Not used	b15: Not used	S (Status change)
b0: Used for real-time monitor	b8: Not used																			
b1: Not used	b9: Not used																			
b2: Not used	b10: Not used																			
b3: Not used	b11: Not used																			
b4: Not used	b12: Not used																			
b5: Not used	b13: Not used																			
b6: Not used	b14: Not used																			
b7: Not used	b15: Not used																			

No.	Name	Data stored	Details	Set by (setting timing)
SD642	Internal buffer capacity	Internal buffer capacity Low-order (in K bytes)	The capacity of the internal buffer is stored in K bytes.	S (Initial)
SD643		Internal buffer capacity High-order (in K bytes)	The capacity of the internal buffer is stored in K bytes.	S (Initial)
SD644	Internal buffer free area space	Internal buffer free area space Low-order (in K bytes)	This register stores the following in K bytes: internal buffer free space that is not set in the internal buffer capacity setting of the CPU parameters.	S (Status change)
SD655		Internal buffer free area space High-order (in K bytes)	This register stores the following in K bytes: internal buffer free space that is not set in the internal buffer capacity setting of the CPU parameters.	S (Status change)

\*1 It is stored when used for the Ethernet function of CPU module.

\*2 When the save destination for the event history is the data memory: the event history will be stored at the timing of powering off and on and resetting the CPU module. Accordingly, since the data will be written into the data memory, "100" is stored. (When the save destination for the event history is the SD memory, the initial value remains "0".)

## 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. 1~8: 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	In this register, interrupt disabling level for each priority setting value is stored. Interrupt by items with the same priority level specified by instructions, such as the interrupt disabling instruction (the DI instruction), the interrupt disabling instruction (the DI instruction) by items with the specified priority level or lower, and the interrupt enabling instructions (the EI instruction) is disabled in a lump. 1: disabling interrupt by items with the priority level 1 or lower (disabling interrupt by items with any priority levels) (default) 2: disabling interrupt by items with the priority level 2 or lower 3: disabling interrupt by items with the priority level 3 or lower 4: disabling interrupt by items with the priority level 4 or lower 5: disabling interrupt by items with the priority level 5 or lower 6: disabling interrupt by items with the priority level 6 or lower 7: disabling interrupt by items with the priority level 7 or lower 8: disabling interrupt by items with the priority level 8 or lower 0: No priority specification (enabling interrupt by items with any priority level)	S (Status change)
SD760	Unicode text file faulty area in the DBIMPORT instruction	Row number of the Unicode text file (lower)	If an error in the Unicode text file is detected when the DBIMPORT(P) instruction is executed, the row number of the Unicode text file is stored. The value is stored when the error completion signal of the DBIMPORT(P) instruction is turned on and is cleared to 0 when the DBIMPORT(P) instruction is executed.	S (Status change)
SD761		Row number of the Unicode text file (upper)		S (Status change)
SD771	Specification of the number of write instruction executions to data memory	Specification of the number of write instruction executions to data memory	In this register, the maximum number of the data memory writing instruction (SP.DEVST) executions per day is specified. When the number of data memory writing instruction executions exceeds the value specified in this register, an error is generated. Setting range is from 1 to 32767. If a value is set outside the range, an error is generated during the data memory writing instruction execution.	U



No.	Name	Data stored	Details	Set by (setting timing)												
SD775	Selection of refresh processing during the COM instruction execution	b0 to b5 and b6 0: no refresh execution 1: refresh execution b15 0: Device/Label access service processing execution 1: No device/Label access service processing execution	In this register, whether or not each processing is executed during the COM instruction execution is selected (Default: 0). The specification on the SD775 is effective when SM775 is turned on. 	U												
SD792 and SD793	PID limit restriction settings (for complete differentiation)	0: Limit restriction applied 1: No limit restriction	The limit restriction for each PID loop is specified as follows: (for the PIDCONT instruction) <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">b15</td> <td></td> <td style="text-align: center;">b1</td> <td style="text-align: center;">b0</td> </tr> <tr> <td>SD792</td> <td style="text-align: center;">Loop 16</td> <td style="text-align: center;">to</td> <td style="text-align: center;">Loop 2    Loop 1</td> </tr> <tr> <td>SD793</td> <td style="text-align: center;">Loop 32</td> <td style="text-align: center;">to</td> <td style="text-align: center;">Loop 18   Loop 17</td> </tr> </table>	b15		b1	b0	SD792	Loop 16	to	Loop 2    Loop 1	SD793	Loop 32	to	Loop 18   Loop 17	U
b15		b1	b0													
SD792	Loop 16	to	Loop 2    Loop 1													
SD793	Loop 32	to	Loop 18   Loop 17													
SD794 and SD795	PID limit restriction settings (for incomplete differentiation)	0: Limit restriction applied 1: No limit restriction	The limit restriction for each PID loop is specified as follows: (for the S.PIDCONT instruction) <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">b15</td> <td></td> <td style="text-align: center;">b1</td> <td style="text-align: center;">b0</td> </tr> <tr> <td>SD794</td> <td style="text-align: center;">Loop 16</td> <td style="text-align: center;">to</td> <td style="text-align: center;">Loop 2    Loop 1</td> </tr> <tr> <td>SD795</td> <td style="text-align: center;">Loop 32</td> <td style="text-align: center;">to</td> <td style="text-align: center;">Loop 18   Loop 17</td> </tr> </table>	b15		b1	b0	SD794	Loop 16	to	Loop 2    Loop 1	SD795	Loop 32	to	Loop 18   Loop 17	U
b15		b1	b0													
SD794	Loop 16	to	Loop 2    Loop 1													
SD795	Loop 32	to	Loop 18   Loop 17													
SD796	Maximum number of blocks used for the multiple CPU dedicated instruction (for CPU No.1)	The maximum number of blocks to be used for the dedicated instruction Depending on the number of CPU modules which constitute a multiple CPU system, the range is as follows.*1 When constituting two modules: 2 to 599 When constituting three modules: 2 to 299 When constituting four modules: 2 to 199 (Default: 2).	<ul style="list-style-type: none"> <li>The maximum number of blocks used for the multiple CPU dedicated instruction is specified (for CPU No.1).</li> <li>When executing the multiple CPU dedicated instruction on CPU No. 1, if the number of free blocks in the dedicated instruction transfer area is less than the setting value on this register, SM796 is turned on.</li> <li>This value is used as interlock signal for the continuous executions of the multiple CPU dedicated instruction.</li> </ul>	U												
SD797	Maximum number of blocks setting used for the multiple CPU dedicated instruction (for CPU No.2)	The maximum number of blocks to be used for the dedicated instruction Depending on the number of CPU modules which constitute a multiple CPU system, the range is as follows.*1 When constituting two modules: 2 to 599 When constituting three modules: 2 to 299 When constituting four modules: 2 to 199 (Default: 2).	<ul style="list-style-type: none"> <li>The maximum number of blocks used for the multiple CPU dedicated instruction is specified (for CPU No.2).</li> <li>When executing the multiple CPU dedicated instruction on CPU No. 2, if the number of free blocks in the dedicated instruction transfer area is less than the setting value on this register, SM797 is turned on.</li> <li>This value is used as interlock signal for the continuous executions of the multiple CPU dedicated instruction.</li> </ul>	U												
SD798	Maximum number of blocks setting used for the multiple CPU dedicated instruction (for CPU No.3)	The maximum number of blocks to be used for the dedicated instruction Depending on the number of CPU modules which constitute a multiple CPU system, the range is as follows.*1 When constituting two modules: 2 to 599 When constituting three modules: 2 to 299 When constituting four modules: 2 to 199 (Default: 2).	<ul style="list-style-type: none"> <li>The maximum number of blocks used for the multiple CPU dedicated instruction is specified (for CPU No.3).</li> <li>When executing the multiple CPU dedicated instruction on CPU No. 3, if the number of free blocks in the dedicated instruction transfer area is less than the setting value on this register, SM798 is turned on.</li> <li>This value is used as interlock signal for the continuous executions of the multiple CPU dedicated instruction.</li> </ul>	U												
SD799	Maximum number of blocks setting used for the multiple CPU dedicated instruction (for CPU No.4)	The maximum number of blocks to be used for the dedicated instruction Depending on the number of CPU modules which constitute a multiple CPU system, the range is as follows.*1 When constituting two modules: 2 to 599 When constituting three modules: 2 to 299 When constituting four modules: 2 to 199 (Default: 2).	<ul style="list-style-type: none"> <li>The maximum number of blocks used for the multiple CPU dedicated instruction is specified (for CPU No.4).</li> <li>When executing the multiple CPU dedicated instruction on CPU No. 4, if the number of free blocks in the dedicated instruction transfer area is less than the setting value on this register, SM799 is turned on.</li> <li>This value is used as interlock signal for the continuous executions of the multiple CPU dedicated instruction.</li> </ul>	U												
SD816 <sup>*2</sup> SD817 <sup>*2</sup>	Execution cycle	Execution cycle	An execution cycle (unit: second) of process control instructions is set in real number.	U												
SD818 <sup>*2</sup>	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												


No.	Name	Data stored	Details	Set by (setting timing)
SD820 <sup>*2</sup>	Dummy device	Dummy device	A dummy device used in process control instructions is set.	U
SD821 <sup>*2</sup>				

\*1 When the value out of the range is specified, operation runs while its value is being regarded as max value of each range of multiple CPU system configuration.

\*2 Only the Process CPU supports these special register areas.

## Data logging function

The following is the list of special register areas relating to the data logging function.

No.	Name	Data stored	Details	Set by (setting timing)
SD1210 SD1211	Data logging setting No.1 Latest storage file number	Latest storage file number	<ul style="list-style-type: none"> <li>The latest storage file number</li> <li>This register is cleared to 0 by the stop command from CPU Module Logging Configuration Tool.</li> </ul>	S (Status change)
SD1212 SD1213	Data logging setting No.1 Oldest storage file number	Oldest storage file number	<ul style="list-style-type: none"> <li>The oldest storage file number</li> <li>This register is cleared to 0 by the stop command from CPU Module Logging Configuration Tool.</li> </ul>	S (Status change)
SD1214	Data logging setting No.1 Internal buffer free space	Free space size of the internal buffer (K bytes)	<ul style="list-style-type: none"> <li>The amount of free space of the internal buffer is stored in K bytes. The smaller the value, the higher the generating ratio of processing overflow.</li> <li>For the trigger logging, it refers to the total capacity of internal buffer until the data for the number of records is collected after a trigger occurred.</li> <li>This register is cleared to 0 by the stop command from CPU Module Logging Configuration Tool.</li> </ul>	S (Error)
SD1215	Data logging setting No.1 Number of processing overflow occurrences	Number of processing overflow occurrences	<ul style="list-style-type: none"> <li>The number of data logging processing overflow occurrences</li> <li>When overflow occurs, data is lost.</li> <li>When the count exceeds 65535, it returns to 0 and starts a new cycle.</li> <li>With "Stop" set for the operation at the time when the number of save files exceeds the limit, a processing overflow may occur until the stop operation is completed after the collection of data corresponding to the specified number of storage files was completed.</li> <li>This register is cleared to 0 when the setting is registered or by the stop command from CPU Module Logging Configuration Tool.</li> </ul>	S (Error)
SD1216	Data logging setting No.1 data logging error cause	data logging error cause	<p>Cause of the error generated during data logging operations is stored.</p> <p>0: no error Other than 0: for details on values stored when an error occurs, refer to the list of error codes found in the following manual: (  Page 272 List of error codes)</p>	S (Error)
SD1220 to SD1226	Data logging setting No.2	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1230 to SD1236	Data logging setting No.3	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1245 to SD1246	Data logging setting No.4	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1250 to SD1256	Data logging setting No.5	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1260 to SD1266	Data logging setting No.6	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1270 to SD1276	Data logging setting No.7	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1280 to SD1286	Data logging setting No.8	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1290 to SD1296	Data logging setting No.9	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1
SD1300 to SD1306	Data logging setting No.10	Same configuration as the setting No.1	Data configuration is the same as the setting No.1 (SD1210 to SD1216).	Same configuration as the setting No.1

## 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	<p>The mask pattern for interrupt pointers is stored as shown in the following figure:</p> <table border="1"> <tr> <td></td> <td style="text-align: center;">b15</td> <td></td> <td></td> <td style="text-align: center;">b1</td> <td style="text-align: center;">b0</td> </tr> <tr> <td>SD1400</td> <td>I15</td> <td style="text-align: center;">to</td> <td></td> <td>I1</td> <td>I0</td> </tr> <tr> <td>SD1401</td> <td>I31</td> <td style="text-align: center;">to</td> <td></td> <td>I17</td> <td>I16</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="text-align: center;">-</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td style="text-align: center;">-</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td style="text-align: center;">-</td> <td></td> <td></td> </tr> <tr> <td>SD1463</td> <td>I1023</td> <td style="text-align: center;">to</td> <td></td> <td>I1009</td> <td>I1008</td> </tr> </table>		b15			b1	b0	SD1400	I15	to		I1	I0	SD1401	I31	to		I17	I16				-						-						-			SD1463	I1023	to		I1009	I1008	S (at execution)
	b15			b1	b0																																									
SD1400	I15	to		I1	I0																																									
SD1401	I31	to		I17	I16																																									
			-																																											
			-																																											
			-																																											
SD1463	I1023	to		I1009	I1008																																									

## Memory dump function

The following is the special register area relating to the memory dump function. (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SD1472	Memory dump error cause	Memory dump error cause	<p>The cause of the error occurred during the memory dump function is stored. 0: No error            Other than 0: For details on the values stored when an error occurs, refer to the list of error codes. (Page 272 List of error codes)</p>	S (upon error)

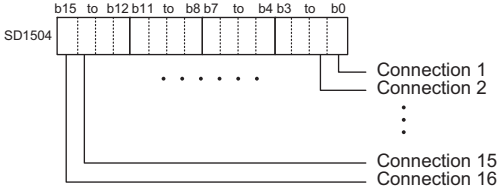

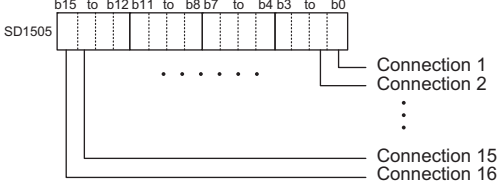

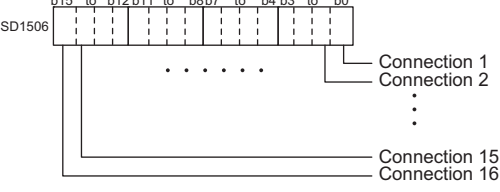

## Real-time monitor function

The following is the special register area relating to the real-time monitor function. (Programmable controller CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SD1484	Real-time monitor internal buffer free space	Free space size of the internal buffer (K bytes)	<p>The amount of free space of the internal buffer is stored in K bytes. The smaller the value, the higher the probability of occurrence of processing overflow.</p>	S (Status change)

## Ethernet function

The following is the list of special register areas relating to the Ethernet function.

No.	Name	Data stored	Details	Set by (setting timing)
SD1504	Open completion signal	In this register, open completion status is stored.	<p>Open statuses of connection No.1 to 16 are stored. (0: Close/Open not completed, 1: Open completed).</p>  <p>For details on the On/Off timing, refer to the following:   MELSEC iQ-R Ethernet User's Manual (Application)</p>	S (Status change)
SD1505	Open request signal	In this register, open request status is stored.	<p>Open processing statuses of connection No.1 to 16 are stored. (0: No open request, 1: Open request exists).</p>  <p>For details on the On/Off timing, refer to the following:   MELSEC iQ-R Ethernet User's Manual (Application)</p>	S (Status change)
SD1506	Socket communications receive status signal	In this register, receive status is stored.	<p>Receiving statuses of connection No.1 to 16 are stored. (0: No data received, 1: Data receiving completed)</p>  <p>For details on the On/Off timing, refer to the following:   MELSEC iQ-R Ethernet User's Manual (Application)</p>	S (Status change)

No.	Name	Data stored	Details	Set by (setting timing)									
SD1520	IP address setting	IP Address (lower)	<ul style="list-style-type: none"> <li>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)</li> <li>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.</li> </ul> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td>b15 to b8</td> <td>b7 to b0</td> </tr> <tr> <td>SD1520</td> <td>3rd byte</td> <td>4th byte</td> </tr> <tr> <td>SD1521</td> <td>1st byte</td> <td>2nd byte</td> </tr> </table>		b15 to b8	b7 to b0	SD1520	3rd byte	4th byte	SD1521	1st byte	2nd byte	S (Status change)/U
		b15 to b8		b7 to b0									
SD1520	3rd byte	4th byte											
SD1521	1st byte	2nd byte											
SD1521	IP Address (upper)												
SD1522		Subnet mask pattern (lower)	<ul style="list-style-type: none"> <li>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)</li> <li>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.</li> </ul> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td>b15 to b8</td> <td>b7 to b0</td> </tr> <tr> <td>SD1522</td> <td>3rd byte</td> <td>4th byte</td> </tr> <tr> <td>SD1523</td> <td>1st byte</td> <td>2nd byte</td> </tr> </table>		b15 to b8	b7 to b0	SD1522	3rd byte	4th byte	SD1523	1st byte	2nd byte	S (Status change)/U
		b15 to b8		b7 to b0									
SD1522	3rd byte	4th byte											
SD1523	1st byte	2nd byte											
SD1523	Subnet mask pattern (upper)												
SD1524		Default Router IP Address (lower)	<ul style="list-style-type: none"> <li>Specify the Default Router 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)</li> <li>Upon completion of writing or clearing the IP address to/from the IP address storage area (system memory), the value of default router IP address stored in the IP address storage area (system memory) is stored.</li> </ul> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td>b15 to b8</td> <td>b7 to b0</td> </tr> <tr> <td>SD1524</td> <td>3rd byte</td> <td>4th byte</td> </tr> <tr> <td>SD1525</td> <td>1st byte</td> <td>2nd byte</td> </tr> </table>		b15 to b8	b7 to b0	SD1524	3rd byte	4th byte	SD1525	1st byte	2nd byte	S (Status change)/U
		b15 to b8		b7 to b0									
SD1524	3rd byte	4th byte											
SD1525	1st byte	2nd byte											
SD1525	Default Router IP Address (upper)												
SD1526	IP address storage area writing error cause	The error cause for the case when failing in writing into IP address storage area is stored.	<p>The error cause generated when writing into IP address storage area (system memory) will be stored. (Linked with SM1521)</p> <p>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.</p>	S (Status change)									
SD1527	IP address storage area clearing error cause	The cause of the error for the case when failing in clearing IP address storage area is stored.	<p>The cause of the error generated during writing into IP address storage area (system memory) is stored. (Linked with SM1523)</p> <p>0H: no error  200H: clear error  400H: Impossible to execute the clear processing because write processing is in progress.</p>	S (Status change)									

## Online module change function

The following is the list of special register areas relating to the online module change function. (Process CPU only)

No.	Name	Data stored	Details	Set by (setting timing)
SD1600	Module selection (base unit No.)	Base unit No. where the online change target module is mounted	<p>The base unit number where the online change target module is mounted is specified.</p> <ul style="list-style-type: none"> <li>0: Main base unit</li> <li>1: Extension base unit 1</li> <li>⋮</li> <li>7: Extension base unit 7</li> <li>FFFFH: Not specified (Default)</li> </ul> <p>The specified value returns to its default (FFFFH) upon completion of the online module change processing. When changing a module directly, the system stores the corresponding base unit number upon removal of the target module.</p>	S (Status change)/U (Request)
SD1601	Module selection (slot No.)	Slot No. where the online change target module is mounted	<p>The slot number where the online change target module is mounted is specified.</p> <ul style="list-style-type: none"> <li>0: Slot No.0</li> <li>1: Slot No.1</li> <li>⋮</li> <li>11: Slot No.11</li> <li>FFFFH: Not specified (Default)</li> </ul> <p>The specified value returns to its default (FFFFH) upon completion of the online module change processing. When changing a module directly, the system stores the corresponding slot number upon removal of the target module.</p>	S (Status change)/U (Request)
SD1602	I/O No. of the module being changed online	I/O No. of the module being changed online	<p>The value of the I/O number of the module being changed online divided by 16 is stored.</p> <p>Other than FFFFH: I/O No. ÷ 16FFFFH: Not set (Default)</p> <p>The stored value returns to its default (FFFFH) upon completion of the online module change processing.</p>	S (Status change)
SD1617	Online module change progress status	Online module change progress status	<p>The online module change progress status is stored.</p> <ul style="list-style-type: none"> <li>0: Normal operation</li> <li>1: Module being selected</li> <li>2: Module selected</li> <li>3: Module removal requested</li> <li>4: Module removal ready</li> <li>5: Module removed</li> <li>6: Module mounted</li> <li>7: Module being recognized</li> <li>8: Module recognized</li> <li>9: Module control resumed</li> </ul> <p>The value becomes 0 upon completion of the online module change processing.</p>	S (Status change)
SD1618	Online module change error code	0: Normal operation Other than 0: Error code	<p>The corresponding error code is stored when an error is detected 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.</p> <p>For the value stored, refer to the list of error codes. (☞ Page 311 Error codes related to the online module change function)</p> <p>An error occurs in selecting a module. Thus, the error code needs to be cleared to 0 before module selection.</p>	S (Status change)/U (Request)
SD1619	Disable request error code during online module change	0: Normal operation Other than 0: Error code	<p>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.</p> <p>For the value stored, refer to the list of error codes. (☞ Page 311 Error codes related to the online module change function)</p>	S (Status change)

# Appendix 6 Processing Time


---

The scan time of the CPU module is the sum of the instruction execution time, the program execution time, and the END processing time. Each of the processing time that constitutes the scan time is as follows.

## Instruction execution time

---

Total of each instruction processing time used for the program to be executed by CPU module For the processing time of each instruction, refer to the following.

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

## Program execution time

---

Shows the processing time when multiple programs are executed. When the interrupt program is executed, the overhead time during the interrupt program being executed is also included.

### Processing time when multiple programs are executed (program switching time)

---

When multiple programs are executed, the scan time becomes longer.

#### ■When multiple programs run

The processing time [ $\mu$ s] with multiple programs run is given by: the number of program files  $\times$  1.2.

### Overhead time at executing interrupt program and fixed scan execution type program

---

Shows the overhead time when executing the interrupt program and fixed scan execution type program.

#### Point

The processing time for the interrupt program and fixed scan execution type program is as shown below.  
Instruction processing time for each instruction + Overhead time + Each refresh processing time (when setting)

---



## ■ Overhead time when executing the interrupt program

The overhead time when executing the interrupt program includes the one before executing the interrupt program and the other when the interrupt program finished.

- Overhead time before executing the interrupt program

Interrupt factor	Condition	Overhead time
Internal timer interrupt (I28 to I31)	Saves/stores the file register (R) block number.	19.5 $\mu$ s
	Not save/store the file register (R) block number.	12.9 $\mu$ s
Inter-module synchronous interrupt (I44)	Saves/stores the file register (R) block number.	25.4 $\mu$ s
	Not save/store the file register (R) block number.	17.0 $\mu$ s
Multiple CPU synchronous interrupt (I45)	Saves/stores the file register (R) block number.	24.2 $\mu$ s
	Not save/store the file register (R) block number.	16.1 $\mu$ s
High-speed internal timer interrupt 2 (I48), high-speed internal timer interrupt 1 (I49)	Saves/stores the file register (R) block number.	25.8 $\mu$ s
	Not save/store the file register (R) block number.	18.1 $\mu$ s
Interruption from module (I0 to I15, I50 to I1023)	Saves/stores the file register (R) block number.	19.1 $\mu$ s
	Not save/store the file register (R) block number.	12.8 $\mu$ s

- Overhead time when finishing the interrupt program

Interrupt factor	Condition	Overhead time
Internal timer interrupt (I28 to I31)	Saves/stores the file register (R) block number.	16.0 $\mu$ s
	Not save/store the file register (R) block number.	8.5 $\mu$ s
Inter-module synchronous interrupt (I44)	Saves/stores the file register (R) block number.	19.0 $\mu$ s
	Not save/store the file register (R) block number.	11.4 $\mu$ s
Multiple CPU synchronous interrupt (I45)	Saves/stores the file register (R) block number.	18.7 $\mu$ s
	Not save/store the file register (R) block number.	10.5 $\mu$ s
High-speed internal timer interrupt 2 (I48), high-speed internal timer interrupt 1 (I49)	Saves/stores the file register (R) block number.	19.4 $\mu$ s
	Not save/store the file register (R) block number.	10.3 $\mu$ s
Interruption from module (I0 to I15, I50 to I1023)	Saves/stores the file register (R) block number.	15.7 $\mu$ s
	Not save/store the file register (R) block number.	8.5 $\mu$ s

## ■ Overhead time when executing the fixed scan execution type program

Shows the Overhead time when executing the fixed scan execution type program.

Condition	Overhead time
Saves/stores the file register (R) block number.	37.9 $\mu$ s
Not save/store the file register (R) block number.	20.6 $\mu$ s

## END processing time

The END processing time is as shown below.

- Common processing time
- I/O refresh processing time
- Network refresh processing time
- Intelligent function module refresh processing time
- Multiple CPU refresh processing time
- Prolonged time of END processing when executing each function
- Device/label access service processing time

### Common processing time

Shows the common processing time of CPU module to be treated by the system.

Condition	Common processing time
One unit of CPU module (module is not mounted)	65μs (when the Process CPU is used: 75μs)

### I/O refresh processing time

The I/O refresh processing time for module mounted on main base unit and extension base unit can be calculated by the following formula.

I/O refresh processing time [μs] = ((Number of input refresh points<sup>\*1</sup> × KM1) + (Number of unit having number of input points × KM2) + KM3) + ((Number of output refresh points<sup>\*2</sup> × KM4) + (Number of unit having number of output points × KM5) + KM6)

\*1 Shows the numeric value that is obtained through the division of number of input points by 16.

\*2 Shows the numeric value that is obtained through dividing the number of output points by 16.

Condition			Constant value	
MELSEC iQ-R series module	Main base unit	Input	KM1	0.04
			KM2	0.56
			KM3	11.2
		Output	KM4	0.02
			KM5	0.94
			KM6	6.30
	Extension base unit	Input	KM1	0.04
			KM2	0.56
			KM3	11.2
		Output	KM4	0.02
			KM5	0.94
			KM6	6.30
Q series module	RQ extension base unit	Input	KM1	1.15
			KM2	1.33
			KM3	24.0
		Output	KM4	0.03
			KM5	1.20
			KM6	6.21
	Q series extension base unit	Input	KM1	1.85
			KM2	1.98
			KM3	39.4
		Output	KM4	0.02
			KM5	1.29
			KM6	6.01

## Network refresh processing time

Shows the Network refresh processing time.

### ■CC-Link IE Controller Network refresh processing time

The refresh processing time for the module mounted on the main base unit and the extension base unit can be calculated by the following formula.

$$\alpha T, \alpha R [\text{ms}] = \text{KM1} + \text{KM2} \times ((\text{LB} + \text{LX} + \text{LY} + \text{SB}) \div 16 + \text{LW} + \text{SW}) + \alpha U$$

$$\alpha U [\text{ms}] = \text{KM3} \times (\text{SBU} \div 16 + \text{SWU})$$

- $\alpha T$ : Link refresh time (sending side)
- $\alpha R$ : Link refresh time (receiving side)
- $\alpha U$ : Unit label (SB/SW) refresh time
- LB: Total number of points of Link relay (LB) that is refreshed by the station<sup>\*1</sup>
- LW: Total number of points of Link register (LW) that is refreshed by the station<sup>\*1</sup>
- LX: Total number of points of Link input (LX) that is refreshed by the station<sup>\*1</sup>
- LY: Total number of points of Link output (LY) that is refreshed by the station<sup>\*1</sup>
- SB: Number of points of link special relay (SB)<sup>\*2</sup>
- SW: Number of points of link special register (SW)<sup>\*2</sup>
- SBU: Number of points of link special relay (SB)<sup>\*3</sup>
- SWU: Number of points of link special register (SW)<sup>\*3</sup>

Condition		Constant value
KM1 ( $\times 10^{-3}$ )		27
KM2 ( $\times 10^{-3}$ )	Main base unit <sup>*4</sup>	0.01
	Extension base unit <sup>*4</sup>	0.12
KM3 ( $\times 10^{-3}$ )	Main base unit <sup>*4</sup>	0.03
	Extension base unit <sup>*4</sup>	0.15

\*1 Shows the total number of points of link device that has been set by the refresh parameter setting and the network range assignment setting. Note that the number of points that has been assigned in the reserved station is excluded.

\*2 Shows the number of points when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.

\*3 Shows the number of points when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.

\*4 Shows the base unit type provided with network module targeted for refresh.

## ■CC-Link IE Field Network refresh processing time

The refresh processing time for the module mounted on the main base unit and the extension base unit can be calculated by the following formula.

$$\alpha T, \alpha R [\text{ms}] = \text{KM1} + \text{KM2} \times ((\text{RX} + \text{RY} + \text{SB}) \div 16 + \text{RW}_r + \text{RW}_w + \text{SW}) + \alpha U$$

$$\alpha U [\text{ms}] = \text{KM3} \times (\text{SBU} \div 16 + \text{SWU})$$

- $\alpha T$ : Link refresh time (sending side)
- $\alpha R$ : Link refresh time (receiving side)
- $\alpha U$ : Unit label (SB/SW) refresh time
- RX: Total number of points of remote input (RX) refreshed by the master station/local station<sup>\*1</sup>
- RY: Total number of points of remote input (RY) refreshed by the master station/local station<sup>\*1</sup>
- RW<sub>w</sub>: Total number of points of remote register (RW<sub>w</sub>) refreshed by the master station/local station<sup>\*1</sup>
- RW<sub>r</sub>: Total number of points of remote register (RW<sub>r</sub>) refreshed by the master station/local station<sup>\*1</sup>
- SB: Number of points of link special relay (SB)<sup>\*2</sup>
- SW: Number of points of link special register (SW)<sup>\*2</sup>
- SBU: Number of points of link special relay (SB)<sup>\*3</sup>
- SWU: Number of points of link special register (SW)<sup>\*3</sup>

Condition		Constant value
KM1 ( $\times 10^{-3}$ )		27
KM2 ( $\times 10^{-3}$ )	Main base unit <sup>*4</sup>	0.01
	Extension base unit <sup>*4</sup>	0.12
KM3 ( $\times 10^{-3}$ )	Main base unit <sup>*4</sup>	0.03
	Extension base unit <sup>*4</sup>	0.15

\*1 Shows the range specified in the refresh parameter and the total number of points of link device that has been set by the network configuration setting. Note that the number of points that has been assigned in the reserved station is excluded.

\*2 Shows the number of points when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.

\*3 Shows the number of points when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.

\*4 Shows the base unit type provided with network module targeted for refresh.

## ■CC-Link refresh processing time

The refresh processing time for the module mounted on the main base unit and the extension base unit can be calculated by the following formula. (Remote net Ver.1 mode, Remote net Ver.2 mode)

$$\alpha T, \alpha R [\text{ms}] = \text{KM1} + \text{KM2} \times ((\text{RX} + \text{RY} + \text{SB}) \div 16 + \text{RW}_r + \text{RW}_w + \text{SW}) + \alpha U$$

$$\alpha U [\text{ms}] = \text{KM3} \times (\text{SBU} \div 16 + \text{SWU})$$

- $\alpha T$ : Link refresh time (sending side)
- $\alpha R$ : Link refresh time (receiving side)
- $\alpha U$ : Unit label (SB/SW) refresh time
- RX: Total number of points of remote input (RX) refreshed by the master station/local station<sup>\*1</sup>
- RY: Total number of points of remote input (RY) refreshed by the master station/local station<sup>\*1</sup>
- RW<sub>w</sub>: Total number of points of remote register (RW<sub>w</sub>) refreshed by the master station/local station<sup>\*1</sup>
- RW<sub>r</sub>: Total number of points of remote register (RW<sub>r</sub>) refreshed by the master station/local station<sup>\*1</sup>
- SB: Number of points of link special relay (SB)<sup>\*2</sup>
- SW: Number of points of link special register (SW)<sup>\*2</sup>
- SBU: Number of points of link special relay (SB)<sup>\*3</sup>
- SWU: Number of points of link special register (SW)<sup>\*3</sup>

Condition		Constant value
KM1 ( $\times 10^{-3}$ )		27
KM2 ( $\times 10^{-3}$ )	Main base unit <sup>*4</sup>	0.01
	Extension base unit <sup>*4</sup>	0.12
KM3 ( $\times 10^{-3}$ )	Main base unit <sup>*4</sup>	0.03
	Extension base unit <sup>*4</sup>	0.15

\*1 Shows the range specified in the refresh parameter and the total number of points of link device that has been set by the network configuration setting. Note that the number of points that has been assigned in the reserved station is excluded.

\*2 Shows the number of points when module label is not used. Calculation is executed while it is regarded as "0" when module label is used.

\*3 Shows the number of points when module label is used. Calculation is executed while it is regarded as "0" when module label is not used.

\*4 Shows the base unit type provided with network module targeted for refresh.

## Intelligent function module refresh processing time

The following calculation formula shows the refresh processing time for the intelligent function module mounted on the main base unit and the extension base unit.

Refresh processing time of one module [ $\mu\text{s}$ ] = Read refresh time<sup>\*1</sup> + Write refresh time<sup>\*1</sup>

\*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 [ $\mu\text{s}$ ]

Read refresh time × KM1 + Refresh time for 1st piece setting item (A) + Refresh time for 2nd piece setting item (A) + ... + Refresh time for n-th piece setting item (A) + KM2

### ■Write refresh time [ $\mu\text{s}$ ]

Write refresh time × KM4 + Refresh time for 1st piece setting item (B) + Refresh time for 2nd piece setting item (B) + ... + Refresh time for n-th piece setting item (B) + KM5

- A:  $\text{KM3} \times \text{Number of refresh transfer (word)} [\mu\text{s}]$
- B:  $\text{KM6} \times \text{Number of refresh transfer (word)} [\mu\text{s}]$
- n: Number of blocks for refresh parameter
- KM1 to KM6: Constant value as shown below

Condition		Constant value	
MELSEC iQ-R series module	KM1	Module on the main base unit → CPU module	0.98
		Module on the extension base unit → CPU module	0.98
	KM2	Module on the main base unit → CPU module	11.6
		Module on the extension base unit → CPU module	11.6
	KM3	Module on the main base unit → CPU module	0.05
		Module on the extension base unit → CPU module	0.05
	KM4	CPU module → Main base unit	0.58
		CPU module → Extension base unit	0.58
	KM5	CPU module → Main base unit	9.10
		CPU module → Extension base unit	9.10
	KM6	CPU module → Main base unit	0.01
		CPU module → Extension base unit	0.01
Q series module	KM1	Module on the RQ extension base unit → CPU module	1.47
		Module on the Q series extension base unit → CPU module	2.92
	KM2	Module on the RQ extension base unit → CPU module	21.2
		Module on the Q series extension base unit → CPU module	20.2
	KM3	Module on the RQ extension base unit → CPU module	0.38
		Module on the Q series extension base unit → CPU module	0.91
	KM4	CPU module → Module on the RQ extension base unit	0.83
		CPU module → Module on the Q series extension base unit	1.20
	KM5	CPU module → Module on the RQ extension base unit	15.8
		CPU module → Module on the Q series extension base unit	15.0
	KM6	CPU module → Module on the RQ extension base unit	0.43
		CPU module → Module on the Q series extension base unit	0.97

## Multiple CPU refresh processing time

Shows the Multiple CPU refresh processing time.

Refresh processing time[ $\mu$ s] = Send refresh time + Receive refresh time

Send refresh time [ $\mu$ s] =  $KM1 + KM2 \times$  Number of points of send word

Receive refresh time [ $\mu$ s] =  $KM3 + KM4 \times$  Number of other CPU modules +  $KM4 \times$  Number of points of receive word

### ■Refresh (when END)

Shows the constant value when refresh is executed in END processing at CPU buffer memory area.

Constant	Constant value
KM1	6
KM2	0.03
KM3	43
KM4	21
KM5	0.04

### ■Refresh (when I45 is executed)

Shows the constant value when refresh is executed in Multiple CPU synchronous interrupt (I45) at fixed scan communication area.

Constant	Constant value
KM1	20
KM2	0.007
KM3	20
KM4	4
KM5	0.007

## Prolonged time of END processing when executing each function

This section describes the prolonged time of END processing when executing each function.

### ■Latch processing time

If the latch function is used and the range for the latch time setting is effective, the scan time becomes longer. Also when the time setting is configured for the latch time setting, the scan time may be prolonged in the next END processing after the specified time has passed. The increase in scan time when the latch range is set can be calculated by the following formula.  
 Increase in scan time[μs] = (KM1 × Number of settings of latch range setting<sup>\*1</sup>) + (KM2 × (Number of points of bit device with latch specified ÷ 16 + Number of points of word device with latch specified + Number of points of double word device with latch specified × 2)) + KM3

\*1 The latch range (1) and the latch range (2) are counted as different device types.

Condition		Constant value
When set to "each scan"	KM1	1.00
	KM2	0.09
	KM3	1.20
When set to "hour"	KM1	1.0
	KM2	0.004
	KM3	17.5

### ■Data logging function

When the data logging function is executed, the scan time becomes longer. The increase in scan time can be calculated by the following formula.

Increase in scan time [μs] = KM1 + (KM2 × Number of data logging settings) + (KM3 × Number of device points of internal device<sup>\*1</sup>)

\*1 Shows the total number of points of data logging setting No.1 to No.10.

Condition		Constant value (NZ1MEM-□GBSD)
When all the columns are outputted under the condition of file format of Unicode text file	KM1	34
	KM2	34
	KM3	0.08

### ■Processing time when file register is used

When "Use File Register of Each Program" has been set through setting the file register, the scan time becomes longer. The increase in the scan time [ms] is given by: the number of program files × 0.016.



When "Use Common File Register in All Programs" has been set, the scan time may not become longer.

### ■Memory dump function

The execution of the memory dump function results in an increase in the scan time. The increase in scan time can be calculated by the following formula.

Increase in scan time [μs] = (KM1 × Number of points<sup>\*1</sup>) + KM2

Constant	Setting of the internal buffer capacity	Constant value
KM1	32K bytes	0.70
	256K bytes	4.70
	3040K bytes	40.00
KM2	32K bytes	320.00
	256K bytes	2290.00
	3040K bytes	16400.00

\*1 Indicates the total number of points (total number of words) of devices that are set in the device area, file storage area, and refresh data register area (unit: K words).



## ■Real-time monitor function

The execution of the real-time monitor function results in an increase in the scan time. The increase in scan time can be calculated by the following formula.

Increase in scan time [ $\mu$ s] =  $KM1 + (KM2 \times \text{Number of points of word device}) + (KM3 \times \text{Number of points of bit device})$

- In monitoring internal user devices

Constant	Constant value
KM1	149.00
KM2	3.20
KM3	0.80

- In monitoring file registers

Constant	Constant value
KM1	149.00
KM2	3.50
KM3	1.10

- In monitoring module access devices/link direct devices

Constant	Constant value
KM1	156.00
KM2	11.60
KM3	8.90

## Device/label access service processing time

Indicates the Device/Label access service processing time when "Processing time = One time" in the setting of Device/Label access service processing.

Condition			Device/label access service processing time
Connected via USB	Ladder block change during RUN (online program change)	100 steps are inserted into the head of program of 40K steps.	1.0ms maximum <sup>*1</sup>
	Monitor data registration	Data register (D) (Number of device points = 32 points) is registered to monitor.	0.03ms maximum
When connecting the Ethernet port (TCP)	Ladder block change during RUN (online program change)	100 steps are inserted into the head of program of 40K steps.	1.0ms maximum <sup>*1</sup>
	Monitor data registration	Data register (D) (Number of device points = 32 points) is registered to monitor.	0.07ms maximum
When connecting the Ethernet port (UDP)	Ladder block change during RUN (Online program change)	100 steps are inserted into the head of program of 40K steps.	1.0ms maximum <sup>*1</sup>
	Monitor data registration	Data register (D) (Number of device points = 32 points) is registered to monitor.	0.07ms maximum

\*1 When data are written to the program using a pointer (P) during RUN, the processing time is extended depending on the pointer number used. For example, when the data are written to the program using P8191 during RUN, the processing time is extended up to 3.0ms.

## Data logging function processing time

Shows the processing time taken to store the data when executing the data logging function. Shows the minimum specified time value that can collect without losing the data when executing the continuous logging.

### When the file format is Unicode text file

Shows the collection interval at which data can be collected under the following conditions.

- Scan time = 1.5ms (up to 3ms)
- Internal buffer capacity setting = 128K bytes as per one setting (default setting)
- Collection setting = Time specification (data collection at time interval)
- Data setting = Data register (D) (Data-type: word-signed (decimal number type))
- Output setting = Day/time column (output format is default), Index column are outputted.
- Operation when save setting = file switch timing: 10000 records, save file number are exceeded: overwriting
- SD memory card: NZ1MEM-2/4/8/16GBSD used

Number of points		Collection interval where data can be collected	
		When the NZ1MEM-2GBSD is used	When the NZ1MEM-4/8/16GBSD is used
8 points	(8 points × 1 setting)	1.0ms	1.0ms
16 points	(16 points × 1 setting)	1.0ms	2.0ms
64 points	(64 points × 1 setting)	2.0ms	5.0ms
128 points	(128 points × 1 setting)	4.0ms	7.0ms
256 points	(128 points × 2 setting)	7.0ms	13.0ms
1280 points	(128 points × 10 setting)	24.0ms	26.0ms

### When the file format is binary file

Shows the Collection interval at which data can be collected under the following conditions.

- Scan time = 1.5ms (up to 3ms)
- Internal buffer capacity setting = 128K bytes as per one setting (default setting)
- Collection setting = Time specification (data collection at time interval)
- Data setting = Data register (D) (Data-type: word-signed (decimal number type))
- Output setting = Day/time column (output format is default), Index column are outputted.
- Operation when save setting = File switch timing: 10000 records, save file number are exceeded: overwriting
- SD memory card: NZ1MEM-2/4/8/16GBSD used

Number of points		Collection interval where data can be collected	
		When the NZ1MEM-2GBSD is used	When the NZ1MEM-4/8/16GBSD is used
8 points	(8 points × 1 setting)	1.0ms	1.0ms
16 points	(16 points × 1 setting)	1.0ms	2.0ms
64 points	(64 points × 1 setting)	2.0ms	5.0ms
128 points	(128 points × 1 setting)	4.0ms	7.0ms
256 points	(128 points × 2 setting)	7.0ms	13.0ms
1280 points	(128 points × 10 setting)	24.0ms	26.0ms

## Memory dump function processing time

This section describes the time (unit: seconds) taken to save all of the data when the memory dump function is in execution. The following table shows the time taken for data save under the following conditions:

- Scan time = 1.5ms (up to 3ms)
- Internal buffer capacity setting = 256K bytes (by default)
- SD memory card: NZ1MEM-2/4/8/16GBSD used

Number of points <sup>*1</sup>	Time taken for data save when the memory dump function is in execution	
	When the NZ1MEM-2GBSD is used	When the NZ1MEM-4/8/16GBSD is used
550.4K points <sup>*2</sup>	17.00 seconds	25.00 seconds
1224K points <sup>*3</sup>	35.00 seconds	56.00 seconds
5320K points <sup>*4</sup>	145.00 seconds	240.00 seconds

- \*1 Indicates the total number of points (total number of words) of devices that are set in the device area, file storage area, and refresh data register area.
- \*2 Indicates the number of device points (user device: 38.4K points (words), refresh data register: 512K points (words)) with the CPU parameters set to default values.
- \*3 Assumes the following conditions without an extended SRAM cassette:
- User device: 40K points (words), file register: 160K points (words), and refresh data register: 1024K points (words)
  - With " Use Common File Register in All Programs" selected in the file register setting.
- \*4 Assumes the following conditions with an 8M extended SRAM cassette:
- Total number of points (total number of words) of user devices, local devices, and file registers: 4296K points (words), and refresh data register: 1024K points (words)
  - With " Use File Register of Each Program" selected in the file register setting.
  - The number of program files is 10.

## Real-time monitor function processing time

This section shows the minimum value of monitor interval that allows monitoring without data loss during the execution of the real-time monitor function. The following table shows the time taken for data save under the following conditions:

- Scan time = 1.5ms (up to 3ms)<sup>\*1</sup>
- Data = agrees with the "Number of points of word device" column in the table below
- Timing = Condition specification (step No. specification)<sup>\*2</sup>
- Internal buffer capacity = 32K bytes (default)
- Fixed scan interrupt timing = generating a fixed scan interrupt on the head step number of the program where step numbers are specified
- Performance of the system of personal computer = CPU: Intel® Core™ i3-2100 CPU (3.10GHz), Memory: 2GB, OS: Windows® 7 Professional SP1 32 bits

\*1 The scan time during non-execution of the real-time monitor function shall be an average of 1.0ms.

\*2 Program name: Arbitrary, Step No.: Head step of the program, Execution condition: Set to Always

Condition	Number of points of word device	Value of monitor interval
Connected via USB	1 point	0.15ms
	8 points	0.25ms
	16 points	0.45ms
When connecting the Ethernet port (TCP)	1 point	0.55ms
	8 points	1.25ms
	16 points	1.35ms
When connecting the Ethernet port (UDP)	1 point	0.15ms
	8 points	0.30ms
	16 points	0.50ms

## Database function processing time

This section shows the time taken for the completion signal of the completion device to turn on from the execution of each instruction of the database access instructions. The table below shows the processing time under the following conditions:

- Number of tables: 1
- Number of records: 10000
- Number of fields: 4 (Assume that the data type of field is specified as WORD/DWORD/WSTRING (16 characters) and WSTRING (32 characters).)
- Index: specified
- Transaction: specified<sup>\*1</sup>

Instruction		Time taken to complete the execution of instruction
Database import	DBIMPORT	320 seconds
Database export	DBEXPORT	220 seconds
Database connection	DBOPEN	First time <sup>*2</sup>
		Second time or later
Database disconnection	DBCLOSE	3ms
Database record addition	DBINSERT	210ms
Database record update	DBUPDATE	260ms
Database record search	DBSELECT	180ms
Database record deletion	DBDELETE	300ms

\*1 The DBINSERT/DBUPDATE/DBSELECT/DBDELETE instructions only are subject to this condition.

\*2 After power-on, initial database access involves the diagnostics to check whether the database is corrupted, and thus it takes time to execute the DBOPEN instruction.

## Process control function processing time

This is the time required to process the control loop when the process control function is executed. The following table lists the processing time of some control loop examples.

Loop type	Configuration	Processing time
Two-degree-of-freedom PID control (S2PID)	S.IN, S.PHPL, S.2PID, and S.OUT1 instructions	160 $\mu$ s
PID control (SPID)	S.IN, S.PHPL, S.PID, and S.OUT1 instructions	151 $\mu$ s
Monitor (SMON)	S.IN and S.PHPL instructions	73 $\mu$ s

For the processing time of each process control instruction, refer to the following.

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

# Appendix 7 Parameter List

This section lists parameters.

## System parameters

The following is the list of system parameters.

Item		Parameter No.
I/O Assignment	Base/Power/Extension Cable Setting	Base/Power Supply Module/Extension Cable model name setting 0203H
	I/O Assignment Setting	Slot 0201H
		Module Name 0203H
		Module Type/Points/Start XY/Module Status Setting 0200H
		Control PLC Setting 0202H
Setting of Points Occupied by Empty Slot 0100H		
Multiple CPU Setting	Number of CPU modules 0301H	
	Communication Setting between CPU	Refresh Area Setting 0303H
		CPU Buffer Memory Setting (Refresh (At the END)) 0304H
		CPU Buffer Memory Setting (Refresh (At I45 Exe.)) 0308H
		PLC Unit Data 0309H
		Fixed Scan Communication Function —
		Fixed Scan Communication Area Setting 0307H
	Fixed Scan Communication Setting	Fixed Scan Interval Setting of Fixed Scan Communication 0306H
		Fixed Scan Communication Function and Inter-module Synchronization Function 0306H
	Operation Mode Setting	Stop Setting 0302H
Synchronous Startup Setting 030AH		
Other PLC Control Module Setting	I/O Setting Outside Group 0305H	
Synchronization Setting within the Modules	Use Inter-module Synchronization Function in System —	
	Select Synchronous Target Unit between Unit 0101H	
	Synchronous Fixed Scan Interval Setting within the Modules 0101H	
	Synchronous Master Setting within the Modules 0102H	

# 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	
	File Setting for Device Data Storage	3303H	
Memory/Device Setting	Device/Label Memory Area Setting	Extended SRAM Cassette	3404H
		Device/Label Memory Area Capacity Setting	3400H
		Device Points	3401H
		Local Device	3405H
		Latch Range Setting	3407H
		Latch Type Setting of Latch Type Label	3408H
	Index Register Setting	3402H	
	Refresh Memory Setting	3403H	
	Device Latch Interval Setting	3406H	
	Pointer Setting	340BH	
	Internal Buffer Capacity Setting	340AH	
	RAS Setting	Scan Time Monitoring Time (WDT) Setting	3500H
Constant Scan Setting		3503H	
Error Detection Setting		3501H	
CPU Module Operation Setting at Error Detected		3501H	
LED Display Setting		3502H	
Event History Setting		3504H	
Online module change function setting		Direct change setting	3505H
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
	FB/FUN File Setting	3702H	
Refresh Setting between Multiple CPU	Refresh Setting (At the END)	3901H	
	Refresh Setting (At I45 Exe.)	3902H	
Routing Setting	Routing Setting	3800H	

## Memory card parameters

The following is the list of memory card parameters.

Item	Parameter No.
Boot Setting	2000H
Setting of File/Data Use or Not in Memory Card	2010H

# Appendix 8 List of the Targets of Device/Label Access Service Processing Setting

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

Applicable function	Description
Functions to perform read/write to files accessed by programs	When read/write is performed to the relevant files during execution of a program, file inconsistency may occur. Therefore, such operation is performed during the END processing.
Functions to perform write to devices/labels	When write is performed to devices/labels during execution of a program, the result of operation processing may become inconsistent. Therefore, such operation is performed during the END processing.

## Communication functions via the MC protocol

Among communication functions using the MC protocol, those to which the device/label access service processing setting can be applied are as follows.

Function			Command
Device memory	Multiple blocks batch read		0406 (00□0)
	Multiple blocks batch write		1406 (00□0)
	Batch read	In units of bits	0401 (00□1)
		In units of words	0401 (00□0)
	Batch write	In units of bits	1401 (00□1)
		In units of words	1401 (00□0)
	Random read		0403 (00□0)
	Test (Random write)	In units of bits	1402 (00□1)
		In units of words	1402 (00□0)
	Monitor		0802 (00□0)
Files	New file creation <sup>*1</sup>		1820 (0000)
	File copy <sup>*1</sup>	File password function incompatible	1824 (0000)
		File password function compatible	1824 (0004)
	File open <sup>*1</sup>	File password function incompatible	1827 (0000)
		File password function compatible	1827 (0004)
	File read <sup>*1</sup>		1828 (0000)
	File write <sup>*1</sup>		1829 (0000)
	File close <sup>*1</sup>		182A (0000)
Programmable controller CPU monitoring	Registration <sup>*2</sup>		0630 (0000)
	Clear <sup>*2</sup>		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) <sup>*1</sup>
	File transfer (FTP client) <sup>*1</sup>

\*1 Only the file register file can support it. (Other file access is always executed asynchronously from the program.)

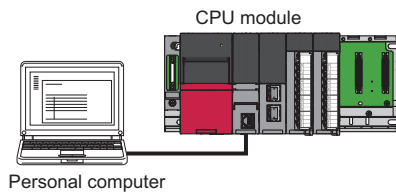
# Appendix 9 How to Use CPU Module Logging Configuration Tool

This section describes the system configuration and specifications, and operation/configuration procedures with CPU Module Logging Configuration Tool when using the data logging function.

## System configuration

This section describes the system configuration when using the data logging function.

### Overall system configuration



### Software

This section describes the software used for the data logging function.

Name	Content
CPU Module Logging Configuration Tool	This software is used for configuration and maintenance of the data logging.
GX LogViewer	This software graphically displays data collected by the data logging function.

#### Point

GX LogViewer is not an essential tool for using the data logging function. Use this tool for various objectives, such as graphically displaying data logged by this function. GX LogViewer is beyond the scope of this manual unless it specifically relates to the topic of this manual. For details on GX LogViewer, refer to the following:

GX LogViewer Version1 Operating Manual

### Supported versions of CPU Module Logging Configuration Tool

Each CPU module can be used with CPU Module Logging Configuration Tool of the supported version (listed in the table below) or any later version.

Version of CPU Module Logging Configuration Tool	Compatible CPU module
"1.35M" or later	Programmable controller CPU
"1.43V" or later	Process CPU

### Operating environment of CPU Module Logging Configuration Tool

For details on the operating environment for CPU Module Logging Configuration Tool, refer to following manual which is stored in the installer.

CPU Module Logging Configuration Tool/GX LogViewer Installation Instructions (BCN-P5999-0393)

## Communication route

To connect the CPU module to a personal computer, use the following methods:

### ■Connection through a USB port

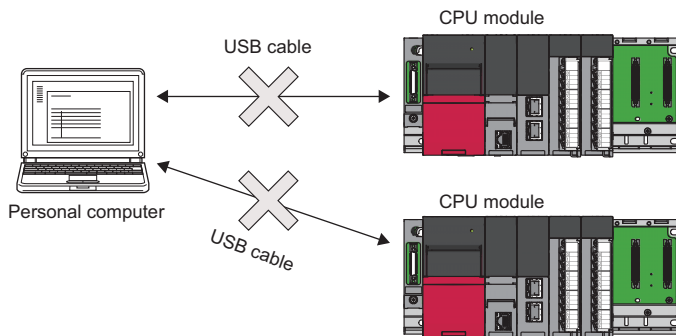
Use a USB cable. The following types of USB cables can be used:

- USB cable (USB A type—USB miniB type)
- USB cable (USB B type—USB miniB type)

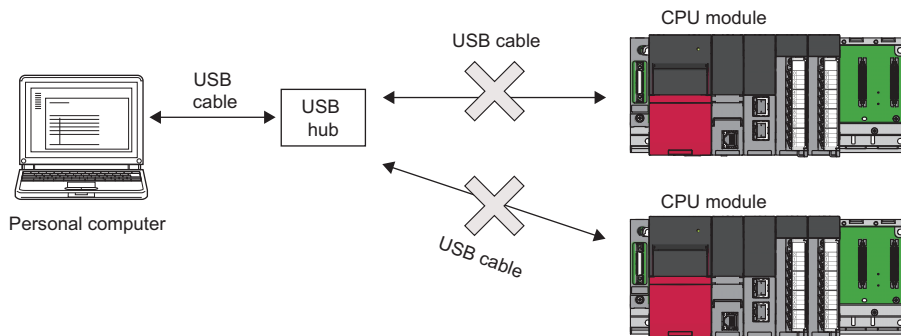
#### Restriction

Only one CPU module can be connected to the same personal computer simultaneously. The following connection configurations are prohibited.

- Connecting a personal computer with multiple USB ports to more than one CPU module



- Connecting to more than one CPU module via a USB hub



### ■Connection through an Ethernet port

- Connection via a hub

Connect the CPU module via a hub to a personal computer on the same local network. Note that IP address of the CPU module must be specified. Also the personal computer should have the same network address as the CPU module.

#### Restriction

Only local area network can be used for connections. Connections via the Internet are not allowed.

- Direct connection

One-to-one direct connection with an Ethernet crossing cable is possible. This method requires no hub. Note that IP address of the CPU module need not be specified in this method (broadcast address is used for this communication).

## ■Precautions about Ethernet connection

- Do not directly connect to a personal computer via LAN line. Load imposed on the LAN line adversely affects communications by other devices.
- Do not configure the direct connection setting when using one-to-one connection via a hub between the CPU module and a personal computer.
- If the following conditions are met, the direct connection communication may be disabled. If the communication is disabled, review the settings of the CPU module and personal computer.

### Ex.

When all the bits of the CPU module-side IP address that correspond to 0 part of the personal computer-side subnet mask are on or off:

CPU module-side IP address: 64.64.255.255

Personal computer-side IP address: 64.64.1.1

Personal computer-side subnet mask: 255.255.0.0

### Ex.

When all the bits of the CPU module-side IP address that correspond to the host address of each class of the personal computer-side IP address are on or off<sup>\*1\*2</sup>:

CPU module-side IP address: 64.64.255.255

Personal computer-side IP address: 192.168.0.1

Personal computer-side subnet mask: 255.255.0.0

\*1 The IP address for each class is as follows:

- Class A: 0.x.x.x to 127.x.x.x
- Class B: 128.x.x.x to 191.x.x.x
- Class C: 192.x.x.x to 223.x.x.x

\*2 The host address for each class is 0 part in the following addresses:

- Class A: 255.0.0.0
- Class B: 255.255.0.0
- Class C: 255.255.255.0

### Ex.

When the CPU module-side IP address is obtained through DHCP:

- Disable Windows firewall setting, if enabled.
- Do not set direct connection method in a configuration in which more than one IP address is enabled simultaneously as described in the following list.
  - An IP address is assigned to each Ethernet port (network device) of the personal computer with more than one port.
  - Wireless LAN setting is enabled as well as the Ethernet port on the personal computer.
  - More than one IP address is assigned to a single Ethernet port of the personal computer.

# Specifications

This section describes the specifications of the data logging function.

## Functional specifications

The following table lists functional specifications.

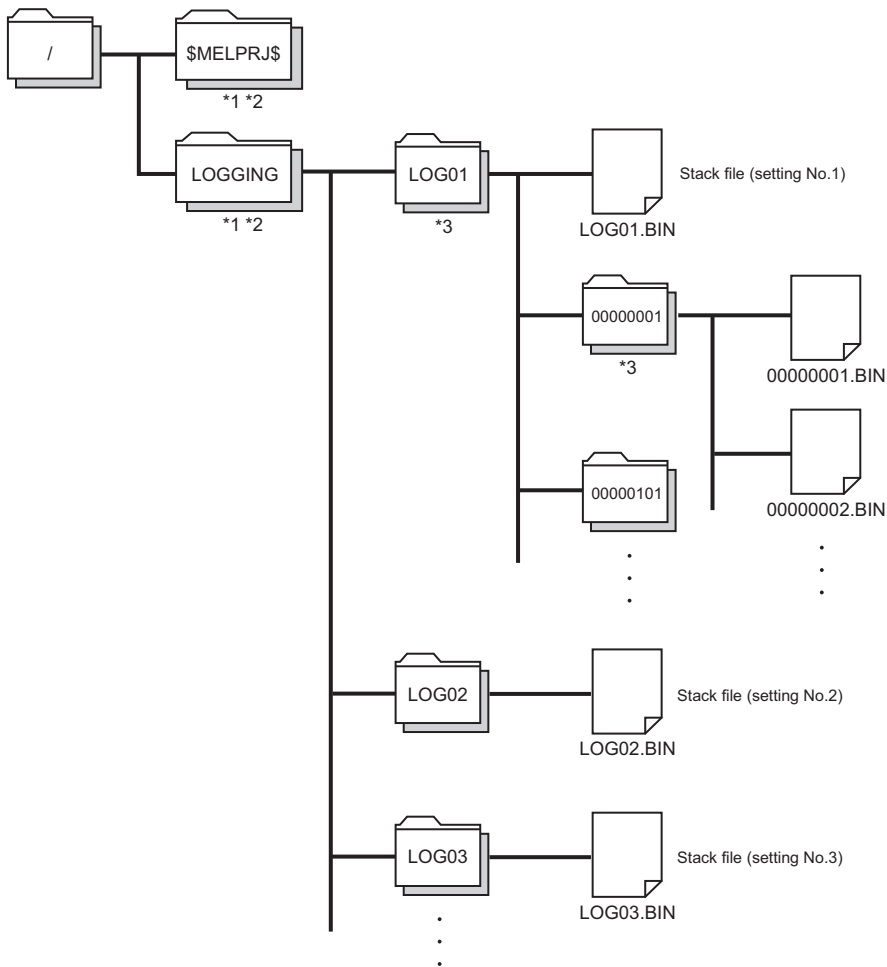
Item		Specifications	
Number of data logging settings		10	
Data storage location		<ul style="list-style-type: none"> <li>Data memory (only data logging configuration file)</li> <li>SD memory card</li> </ul>	
Logging type		<ul style="list-style-type: none"> <li>Continuous logging</li> <li>Trigger logging</li> </ul>	
Data collection	Collection interval	<ul style="list-style-type: none"> <li>every scan operation</li> <li>Time specification</li> <li>Interrupt occurrence</li> <li>Condition specification (device specification, step No. specification)</li> </ul>	
	Number of points for collection	Maximum of 1280 (128 per setting)	
	AND conjunction	AND conjunction of the device specification and step No. specification is enabled.	
Data processing	Trigger logging	Trigger condition	<ul style="list-style-type: none"> <li>Condition specification (device change specification, step No. specification)</li> <li>When trigger instruction executed</li> </ul>
		AND conjunction	AND conjunction of the device change specification and step No. specification is enabled.
		Trigger logging range	Number of records specified before and after the trigger establishment
		Number of trigger establishments (number of events that can be handled as trigger)	one
		Number of records	Maximum of 1000000
File output	File name		Additional information plus file number
	File storage format		<ul style="list-style-type: none"> <li>Unicode text file</li> <li>Binary file</li> </ul>
	Data type		<ul style="list-style-type: none"> <li>Bit</li> <li>Word (signed)</li> <li>Double word (signed)</li> <li>Word (unsigned)</li> <li>Double word (unsigned)</li> <li>Single-precision real number</li> <li>Double-precision real number</li> <li>Character string</li> <li>Numeric string</li> <li>Time</li> </ul>
	Data output format	Unicode text file	<ul style="list-style-type: none"> <li>Decimal format</li> <li>Hexadecimal format</li> <li>Decimal fraction format</li> <li>Exponent format</li> </ul>
		Binary file	<ul style="list-style-type: none"> <li>Word (signed)</li> <li>Double word (signed)</li> <li>Word (unsigned)</li> <li>Double word (unsigned)</li> <li>Single-precision real number</li> <li>Double-precision real number</li> </ul>
	Output file handling	Storage file switching	File switching timing
Maximum number of storage files			1 to 65535

Item		Specifications
Other functions	Operation settings when entering into RUN mode	This function sets data logging operations when entering into RUN mode after the data logging setting is registered.
	Auto logging	When inserting an SD memory card, which holds data logging setting, into the CPU module, the data logging automatically starts based on the data logging setting information on the SD memory card.
	SD memory card replacement	SD memory cards can be replaced using the SD memory card forced disable function even when the data logging is in progress.

## Folder configuration

The following figure shows the folder configuration of the SD memory card attaching to a CPU module.


A



\*1 Folder names cannot be modified.

\*2 Do not create folders/files under the \$MELPRJ\$ and LOGGING folders using a personal computer and other.

\*3 To remove unnecessary folders, use the following methods:

- Use a personal computer.
- Logging file operation (  Page 413 Logging file operation).

## Data output type

This section describes each of file output types.

### Unicode text file output type

This section describes the format specifications of the Unicode text file output type and output content of each data.

- Format specifications

Item	Description
Delimiter	Tab
Return code	CRLF (0x0D, 0x0A)
Character code	Unicode
Character encoding schema	UTF-16 (Little-Endian)
Field data	Not enclosed by double quotation marks (" ") Double quotation marks (" ") and commas (,) cannot be used in each data.*1

\*1 When "Output device comments for data" is selected in the output setting; if double quotation marks (" ") and/or commas (,) are included in a device comment, the following processing occurs:

- Whole data is enclosed by double quotation marks (" ").
- When the data includes double quotation marks (" "), another set of double quotation marks will be appended to each of them.

- File format example

Output items can be specified (  Page 401 Output)

	File type	Model information_ file version	No. for data type information line	No. for data name line	No. for device comment line	No. for data start line	No. for comment line			
File information line	[LOGGING]	RCP1_1	3	4	5	6	2			
Comment line	LOG01									
Data type information line	DATETIME[YYYY/MM/DD hh:mm:ss.sss]	INTERVAL	STEP NO.	PROGRAM NO.	PROGRAM NAME	INDEX	SHORT[DEC.0]	BIT[1:]	TRIGGER[*]	
Data name line	TIME(UTC+09:00)	INTERVAL[μs]	STEP NO.	PROGRAM NO.	PROGRAM NAME	INDEX	D1	M1	Trigger	
Device comment line								Start contact		
Data line	2014/7/15 13:29:50.858	100	4	1	MAIN	1	1234	1		
	2014/7/15 13:29:50.858	100	4	1		2	1234	0		
	2014/7/15 13:29:50.858	100	4	1		3	1234	0		
	2014/7/15 13:29:50.858	100	4	1		4	0	0		
	2014/7/15 13:29:50.858	100	4	1		5	0	0		
	2014/7/15 13:29:50.858	100	4	1		6	0	0	*	
	2014/7/15 13:29:50.858	100	4	1						

Date row
Data sampling interval row
Execution step No. row
Execution program No. row
Index row
Data row
Trigger occurrence information row

- Output content for each data

· File information row

File-related information is displayed.

Item	Description	Size
File type	[LOGGING] is output.	14 bytes
Model information_ file version	File version that indicates model information (fixed to "RCP1_1") is displayed.	12 bytes
No. for data type information line	Numerical value indicating the position of the data type information row from the top of the file is placed.	2 bytes
No. for data name line	Numerical value indicating the position of the data name row from the top of the file is placed.	2 bytes
No. for device comment line	Numerical value indicating the position of the device comment row from the top of the file is placed.	2 bytes
No. for data start line	Numerical value indicating the starting position of the data row from the top of the file is placed.	2 bytes
No. for comment line	Numerical value indicating the position of the comment row from the top of the file is placed. When the comment row is not output, this field is blank.	0 to 2 bytes



**Ex.**

The total size of the file information row can be obtained by the following equation: (when comment is output)  
 $14(\text{file type}) + 12(\text{model information\_file version}) + 2(\text{data type information row number}) + 2(\text{data name row number}) + 2(\text{device comment row number}) + 2(\text{data start row number}) + 2(\text{comment row number}) + 10(\text{number of tabs}) + 4(\text{CR} + \text{LF}) = 50 \text{ bytes}$

- Comment row file information row
- Comments are displayed.

Item	Description	Size
Comment	Comment specified in CPU Module Logging Configuration Tool is output (the comment can contain up to 256 characters. <sup>*2</sup> When no comment is set, a blank row is output).	0 to 512 bytes

\*2 Double quotation marks (" "), commas (,), and semicolons (;) cannot be used.

**Ex.**

The total size of the comment row can be obtained by the following equation:  
 Character size of the specified comment (depending on the specified character string) + 4(CR + LF)

- Data type information row

The data type for each column is displayed. This information is output in the following format: (data type)[(additional information)].

Item	"Data type" output content	Size	"Additional information" output content	Size
Date/time column	DATETIME	16 bytes	Format is output. [YYYY/MM/DD hh:mm:ss.s]	6 to 68 bytes
Data collection interval column	INTERVAL	16 bytes	No additional information	0 byte
Execution step No. column	STEP NO.	16 bytes		
Execution program number column	PROGRAM NO.	22 bytes		
Execution program name column	PROGRAM NAME	24 bytes		
Index column	INDEX	10 bytes		
Data column	Bit type: BIT	6 bytes	Bit type: [1;0]	10 bytes
	16-bit integer (unsigned): USHORT	12 bytes	For decimal format: [DEC.0]	14 bytes
	16-bit integer (signed): SHORT	10 bytes		
	32-bit integer (unsigned): ULONG	10 bytes	For hexadecimal format: [HEX]	10 bytes
	32-bit integer (signed): LONG	8 bytes		
	Single-precision floating point (32-bit): FLOAT	10 bytes	For exponent expression: [EXP. (number of digits of decimal part)]	14 to 16 bytes
	Double-precision floating point (64-bit): DOUBLE	12 bytes		
	Character string type: STRING	12 bytes	Character string type, numeric string type: the specified data length value (unit: bytes) is output.	6 to 10 bytes
	Numeric string type: RAW	6 bytes		
Time: TIME	8 bytes	No additional information	0 byte	
Trigger occurrence information column	TRIGGER	14 bytes	[(string occurred)] is output (semicolons (;), double quotation marks (" "), and commas (,) cannot be used).	6 to 512 bytes

**Ex.**

The size of the data type information row can be obtained by the following equation, when performing the data logging of 128 16-bit integer (signed) data items in decimal format (when selecting the option which outputs date/time column with the format YYYY/MM/DD hh:mm:ss.s, data collection interval column, execution step No. column, execution program number column, execution program name column, and index column):

$$(16 + 46)(\text{date/time column}) + 16(\text{data collection interval column}) + 16(\text{execution step No. column}) + 22(\text{execution program number column}) + 24(\text{execution program name column}) + 10(\text{index column}) + (10 + 14) \times 128(\text{data column}) + 264(\text{number of tabs}) + 4(\text{CR} + \text{LF}) = 3490 \text{ bytes}$$

· Data name row

The data name for each column is displayed.

Item	Description	Size
Date/time column	TIME (time zone) is output.	28 bytes
Data collection interval column	INTERVAL[us] is output.	24 bytes
Execution step No. column	STEP NO. is output.	16 bytes
Execution program number column	PROGRAM NO. is output.	22 bytes
Execution program name column	PROGRAM NAME is output.	24 bytes
Index column	INDEX is output.	10 bytes
Data column	The specified data name is output.	512 bytes
Trigger occurrence information column	Trigger is output.	14 bytes

**Ex.**

The size of the data name row can be obtained by the following equation, when performing the data logging of 128 data items from D100 to D227 (when selecting the option which outputs date/time column, data collection interval column, execution step No. column, execution program number column, execution program name column, and index column):

$$28(\text{date/time column}) + 24(\text{data collection interval column}) + 16(\text{execution step No. column}) + 22(\text{execution program number column}) + 24(\text{execution program name column}) + 10(\text{index column}) + (6 \times 128)(\text{data column}) + 264(\text{number of tabs}) + 4(\text{CR} + \text{LF}) = 1138 \text{ bytes}$$

· Device comment row

The device comment row for each column is displayed.

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

\*3 When "Output device comments for data" is selected in the output setting: if double quotation marks (" ") and/or commas (,) are included in a device comment, the size of the data to be used becomes larger than the size of character string of the device comment.

**Ex.**

The size of the device comment row can be obtained by the following equation, when performing the data logging of 128 data items within the range of D100 to D227 (when selecting the option which outputs date/time column with the format YYYY/MM/DD hh:mm:ss.s, data collection interval column, execution step No. column, execution program number column, execution program name column, and index column; only the range D200 to D227; specifying "Output device comments for data"):

$$0(\text{date/time column}) + 0(\text{data collection interval column}) + 0(\text{execution step number column}) + 0(\text{execution program number column}) + 0(\text{execution program name column}) + 0(\text{index column}) + (0 \times 100 + 14 \times 28)(\text{data column}) + 264(\text{number of tabs}) + 2(\text{CR} + \text{LF}) = 660 \text{ bytes}$$

· Data row

The collected data value is displayed. All the data items collected during a single collection is displayed in a single row.

Item	Description	Size
Date/time column	Information is output according to the data row output character string specified in the format.	2 to 64 bytes
Data collection interval column	The time interval from the previous collection time to the current collection time is output. If the maximum display range is exceeded, the count returns to 1 and starts again to output a new time interval (unit: $\mu$ s, display range: 1 to 100000000000).	2 to 24 bytes
Execution step No. column	Execution step No. on CPU Module Logging Configuration Tool at the time interval and timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the step No. at the time of execution of END instruction in the last executed program on the scan. With the collection condition "Interrupt occurrence" specified, the resulting output is the step No. at the time of execution of IRET instruction in the interrupt program. If the system operation (such as system interruption) is running during collection interval and collection timing or the FB program is in running, "0" is output.	2 to 12 bytes
Execution program number column	Execution program number (FB file number) on CPU Module Logging Configuration Tool at the time interval and timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the program number of the last executed program in the scan. If two or more system operations, such as collection timing and system interrupt, occur simultaneously, — is output.	2 to 6 bytes
Execution program name column	Execution program name (FB file name) on CPU Module Logging Configuration Tool at the time interval and timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the program name of the last executed program in the scan. If a program name with the same program number has already been in the file, a null value is output. If two or more system operations, such as collection timing and system interrupt, occur simultaneously, "** SYSTEM" is output.	0 to 200 bytes
Index column	A value which increments in ascending order from 1 is output. When it exceeds the upper limit, it returns to 1 and increments again (range: 1 to 4294967295).	2 to 20 bytes
Data column	When bits are specified: bit On = 1 and bit Off = 0 are output.	2 bytes
	When unsigned/signed word type is specified: data value is output according to the specified output type.	<ul style="list-style-type: none"> <li>• Decimal format: 2 to 22 bytes</li> <li>• Hexadecimal format: 2 to 16 bytes</li> <li>• Exponent format: 10 to 42 bytes</li> </ul>
	When unsigned/signed double word type, single-precision real number, or double-precision real number is specified: data value is output according to the specified output type.	<ul style="list-style-type: none"> <li>• Decimal format: 2 to 22 bytes<sup>*4</sup></li> <li>• Hexadecimal format: 2 to 16 bytes</li> <li>• Exponent format: 10 to 44 bytes</li> </ul>
	When character string is specified: the specified character string is output.	2 to 256 bytes
	When numeric string is specified: the character string which represented by the hexadecimal in increments of a byte is output without clearance.	4 to 1024 bytes
	When time is specified: T#-24d20h31m23s648ms to T#24d20h31m23s647ms is displayed.	28 to 40 bytes
Trigger occurrence information column	The specified character string is output when the trigger occurs. In other cases, no character string is output.	0 to 1024 bytes

\*4 When single-precision real number or double-precision real number is specified, if the numerical value to be output does not fall into the range -2147483648.0 to 4294967295.0, it is displayed in an equivalent format to "exponent format and the number of decimal part digits is nine."

**Ex.**

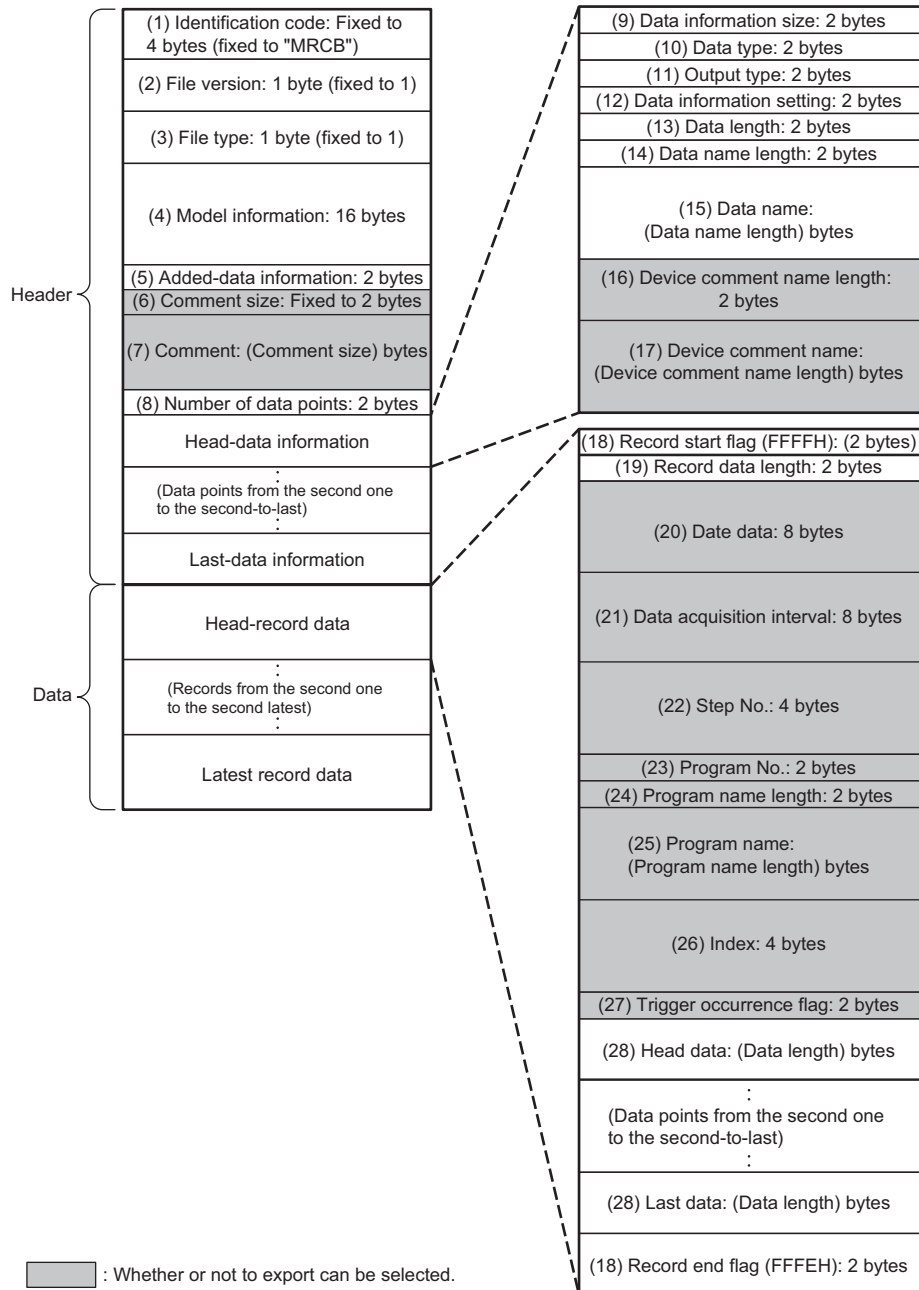
The size of the data row can be obtained by the following equation, when performing the data logging of 128 unsigned word type data items within the range of D100 to D227 in decimal format (when selecting the option which outputs date/time column with the format YYYY/MM/DD hh:mm:ss.s, data collection interval column, execution step No. column, execution program number column, execution program name column, and index column):

$$42(\text{date/time column}) + 24(\text{data collection interval column}) + 12(\text{execution step No. column}) + 6(\text{execution program number column}) + 200(\text{execution program name column}) + 20(\text{index column}) + (12 \times 128)(\text{data column}) + 264(\text{number of tabs}) + 4(\text{CR} + \text{LF}) = 2108 \text{ bytes}$$

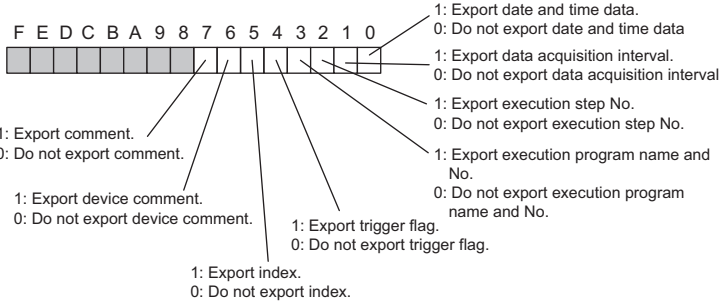
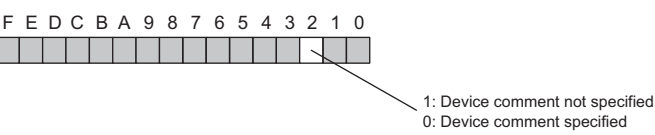
### Binary file output format

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

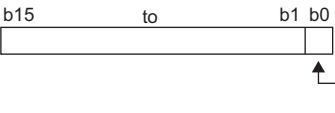
- Configuration figure of binary format



• Details of each data

No.	Item	Description	Size (byte)																													
(1)	Identification code	The MRCB is always output to identify the file.	4																													
(2)	File version	File version 1 is displayed.	1																													
(3)	File type	The file type is output. (fixed to 1: Continuous/trigger logging)	1																													
(4)	Model name information	The module model name that outputted binary file is output. R□CPU is output to the first eight bytes and 00H is output to the last eight bytes.*1	16																													
(5)	Added data information	<p>The output selection setting for the data that can be output is output.</p> 	2																													
(6)	Comment size	The comment length of (7) Comment is output.	2																													
(7)	Comment	The comment specified in the setting is output in Unicode.	2 to 512																													
(8)	Number of data	The number of data points of the data information ((9) to (17)) for data logging is output.	2																													
(9)	Data information size	The total size of the data information ((9) to (17)) for data logging is output.	2																													
(10)	Data type	<p>The numeric value shown below is output depending on the data type.</p> <p>0000H: Bit            0001H: Word (signed)            0002H: Double word (unsigned)            0003H: Word (signed)            0004H: Double word (unsigned)            0005H: Single-precision real number            0006H: Double-precision real number            0007H: Character string            0008H: Numeric string            0009H: Time</p>	2																													
(11)	Output format	The same numerical value as the value in (10) Data type is output. However, when the data type is bit, character string, numeric string, or time, FFFFH is output because the output type cannot be specified.	2																													
(12)	Data information setting	<p>The data-related information is output.</p> 	2																													
(13)	Data length	The data length of data is output. When the data type is the bit type, it will be output as two bytes.	2																													
(14)	Data name length	The length of the data name specified in the setting is output.	2																													
(15)	Data name	The data name specified in the setting is output in Unicode.	2 to 512																													
(16)	Device comment length	The length of the device comment name specified in the setting is output.	2																													
(17)	Device comment name	The device comment name specified in the setting is output in Unicode.	0 to 2048																													
(18)	Record Start Flag, Record End Flag	The flags for identifying the start and end of the record are output. The FFFFH is output for record start while the FFFEh is output for record end as the fixed flag.	2																													
(19)	Record data length	The total size of (20) Day and time data to (28) Last data is output.	2																													
(20)	Day and time data	<p>The Day and time data is output.</p> <table border="1" data-bbox="427 1803 925 1937"> <tr> <td colspan="2">b15</td> <td>to</td> <td colspan="2">b0</td> <td rowspan="2">Year: Last 2 digits of the year, Month: 1 to 12</td> </tr> <tr> <td>Year</td> <td>Month</td> <td></td> <td>Day</td> <td>Time</td> </tr> <tr> <td>Day</td> <td>Time</td> <td></td> <td>Minute</td> <td>Second</td> <td>Day: 1 to 31, Time: 0 to 23</td> </tr> <tr> <td>Minute</td> <td>Second</td> <td></td> <td colspan="2">Millisecond</td> <td>Minute: 0 to 59, Second: 0 to 59</td> </tr> <tr> <td colspan="5"></td> <td>Millisecond: 0 to 999</td> </tr> </table>	b15		to	b0		Year: Last 2 digits of the year, Month: 1 to 12	Year	Month		Day	Time	Day	Time		Minute	Second	Day: 1 to 31, Time: 0 to 23	Minute	Second		Millisecond		Minute: 0 to 59, Second: 0 to 59						Millisecond: 0 to 999	8
b15		to	b0		Year: Last 2 digits of the year, Month: 1 to 12																											
Year	Month		Day	Time																												
Day	Time		Minute	Second	Day: 1 to 31, Time: 0 to 23																											
Minute	Second		Millisecond		Minute: 0 to 59, Second: 0 to 59																											
					Millisecond: 0 to 999																											
(21)	Data sampling interval	The time interval from the previous collection time to the current collection time is output. (Unit: μs, Display range: 1 to 100000000000 (When it exceeds the max value, it returns to "1" and incrementing runs again.)) After logging collection started, 0 is stored at first during data collection.	8																													



No.	Item	Description	Size (byte)
(22)	Step No.	The step No. executed with the engineering tool at the timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the step No. at the time of execution of END instruction in the last executed program on the scan. With the collection condition "Interrupt occurrence" specified, the resulting output is the step No. at the time of execution of IRET instruction in the interrupt program. If the system operation (such as system interruption) is running during collection interval and collection timing or the FB program is in running, "0" is output.	4
(23)	Program No.	The program No. (FB File No) executed on the engineering tool at the timing in which data was collected is output. With the collection condition "Each scanning cycle" specified, the resulting output is the program number of the last executed program in the scan. If system operation (such as system interruption) is running at collection interval and collection timing, "0" is output.	2
(24)	Program name length	The name length of a program that is executed at the time interval and timing in which data was collected is output. If the same program number or program name has already been in the file, "0" is output.	2
(25)	Program name	The program name that is executed at the time interval and timing in which data was collected is output in Unicode. With the collection condition "Each scanning cycle" specified, the resulting output is the program name of the last executed program in the scan. If system operation (such as system interruption) is running at collection interval and collection timing, "* SYSTEM" is output.	0 to 200
(26)	Index	The index number ranging from 1 to 4294967295 of data, which was collected by the data logging function, is output. When it exceeds the max value, it returns to "1" and incrementing runs again. If missing occurs in processing data, index will be reassigned from 1 again.	4
(27)	Trigger occurrence flag	The trigger occurrence information is output. 	2
(28)	Data	Data collected by the data logging function is output corresponding to (13) Data length and (10) Data type. <ul style="list-style-type: none"> <li>• When bits are specified: bit On = 1 and bit Off = 0 are output.</li> <li>• When word type (signed/unsigned) or double-word type (signed/unsigned) is specified: the data values are output in the specified unit.</li> <li>• When single-precision real number or double-precision real number is specified: data value is output in the specified unit. (Page 389 Numerical value range for each output type)</li> <li>• When character string type is specified: the character string with the specified size is output. If the character string terminator "0" exists in the middle of data, NULL is generated on from said point onward until the terminator of the specified size.</li> <li>• When numeric string type is specified: the data value with the specified size is output.</li> <li>• When time is specified: T#-24d20h31m23s648ms to T#24d20h31m23s647ms is displayed in Unicode.</li> </ul>	<ul style="list-style-type: none"> <li>• Bit: 2</li> <li>• Word (signed/unsigned): 2</li> <li>• Double word (signed/unsigned): 4</li> <li>• Single-/double-precision real number: 4</li> <li>• Character/numeric string: 1 to 256</li> <li>• Time: 28 to 40</li> </ul>

\*1 Any of the following values is output to  depending on the model.

R04CPU: 04, R08CPU and R08PCPU: 08, R16CPU and R16PCPU: 16, R32CPU and R32PCPU: 32, R120CPU and R120PCPU: 120

## Numerical value range for each output type

This section describes the numerical value ranges that can be output for each output type.

### Integer type

The following table lists the numerical value ranges that can be expressed for each integer type.

Output format	Lower limit	Upper limit
Word (unsigned)	0	65535
Word (signed)	-32768	32767
Double word (unsigned)	0	4294967295
Double word (signed)	-2147483648	2147483647

### Real number type

The following table lists the numerical value ranges that can be expressed for each real number type.

Output format	Negative value		Positive value	
	Lower limit	Upper limit	Lower limit	Upper limit
Single-precision real number	-3.4028235E+38	-1.401298E-45	1.401298E-45	3.4028235E+38
Double-precision real number	-1.79769313486231570E+308	-4.94065645841246544E-324	4.94065645841246544E-324	1.79769313486231570E+308

When the value of the data exceeds the numerical value range, the following rule is applied to the output.









- When the value exceeds the upper limit of the positive value, +Inf is output.
- When the value falls below the lower limit of the negative value, -Inf is output.
- When the value is between the upper limit of the negative value and the lower limit of the positive value, 0 is output.

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

## Data logging procedure

---


This section describes the data logging procedure.

- 1.** Install and launch CPU Module Logging Configuration Tool. ( CPU Module Logging Configuration Tool/GX LogViewer Installation Instructions (BCN-P5999-0393))
- 2.** Configure the data logging setting. ( Page 395 Setting data logging)
- 3.** If auto logging is used, configure the common settings (auto logging common settings). ( Page 407 Auto logging common setting)
- 4.** Attach an SD memory card into the CPU module, and power up the module.
- 5.** Connect the CPU module to a personal computer. ( Page 376 System configuration,  Page 408 Transfer setup).
- 6.** Write the settings to the CPU module or the SD memory card. ( Page 410 Writing logging setting)
- 7.** Switch the CPU module to RUN state to start the data logging. ( Page 411 Logging status and operation)
- 8.** Stop the data logging and read the data logging file. ( Page 413 Logging file operation)
- 9.** Check the file into which data has been read.

---

### Point

For details on installation and uninstallation procedures of CPU Module Logging Configuration Tool, refer to the following.

 CPU Module Logging Configuration Tool/GX LogViewer Installation Instructions (BCN-P5999-0393)

---



# Menu items and setting methods

This section describes the operation method and/or setting windows of CPU Module Logging Configuration Tool.

## Menu structure

The following table describes the menu structure of CPU Module Logging Configuration Tool.

Menu item		Description
Project	New	Create a new project.
	Open	Open a stored project file.
	Save	Overwrite an edited project to the file and saves it.
	Save As	Save an edited project with a new file name.
	Read Logging Setting from Memory Card(SD)	Read the data logging setting written in the SD memory card attached to the personal computer.
	Write Logging Setting into Memory Card(SD)	Write the settings being edited in a format with which the CPU module can operate. Once the settings is directly written into an SD memory card attached to the personal computer and then the SD memory card is mounted onto the CPU module, the data logging starts.
	Recent Files	Open a recently used file.
	Exit	Exit CPU Module Logging Configuration Tool.
Edit	Delete Data Logging Setting	Remove the data logging setting selected in the Edit item tree.
	Copy and Add Data Logging Setting	Copy and add the data logging setting selected in the Edit item tree.
	Batch Data Insertion	Configure the multiple setting items at once.
	Cut Setting Item	Cut the setting with a table format.
	Copy Setting Item	Copy the setting with a table format.
	Paste Setting Item	Paste the setting with a table format.
	Insert Copied Setting Item	Paste the copied setting with a table format.
	Insert Cut Setting Item	Paste the cut setting with a table format.
	Delete Setting Item	Remove the setting with a table format.
	Move Setting Item Upward	Move up a setting item.
	Move Setting Item Downward	Move down a setting item.
	Device Batch Replacement	Replace devices for all the settings.
Online	Transfer Setup	Configure the communication setting used for connection to the CPU module.
	Read Logging Setting	Read the setting from the CPU module.
	Write Logging Setting	Write the setting to the CPU module.
	Delete Data Logging Setting	Remove the setting data from the CPU module.
	Logging Status and Operation	Check the data logging status.
	Logging File Operation	Connect to the CPU module and reads or removes the files on the attached SD memory card.
Tool	Start GX LogViewer	Launch GX LogViewer.
Help	Open Manual	E-Manual Viewer opens and its manual is displayed.
	Connect to MITSUBISHI ELECTRIC FA site	The window to connect to the Mitsubishi Electric Corporation FA site opens.
	About Configuration tool	The product information is displayed.

## Project management

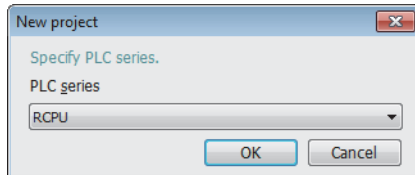
This function creates and saves the project, and reads/writes it from/to an SD memory card.

### ■Creating new project

Create a new project.

 [Project] ⇒ [New]

### Window



### Displayed items

Item	Description
PLC series	Select "RCPU".

### ■Opening project

Open a stored project file.

 [Project] ⇒ [Open]

### ■Saving project


- Overwriting

Overwrite an edited project to the file and saves it.

 [Project] ⇒ [Save]

- Saving with a new file name


Save an edited project with a new file name.

 [Project] ⇒ [Save As]

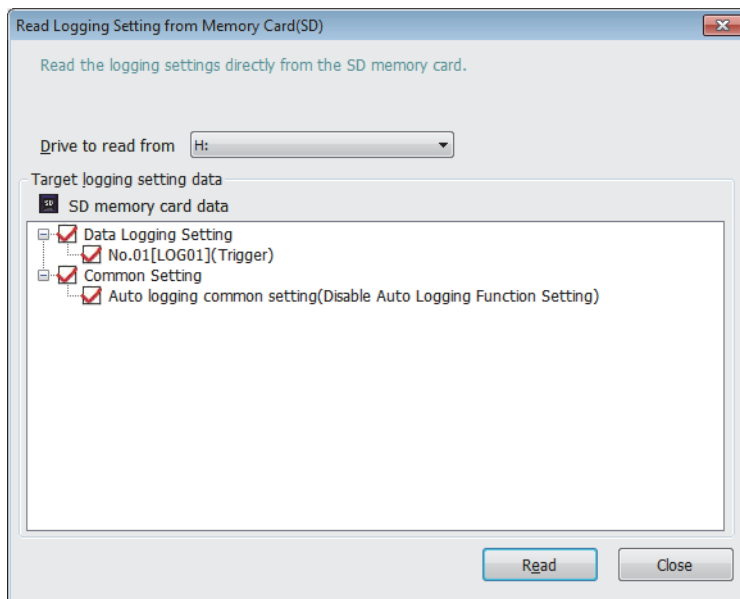
## ■ Reading logging setting from SD memory card

The following procedure is to read the data logging setting written in an SD memory card attached to the personal computer.

### Operating procedure

1. Attach an SD memory card to the personal computer.
2. Open the following window.  
 [Project] ⇒ [Read Logging Setting from Memory Card(SD)]
3. Select the drive from which data is read and data to be read.
4. Click the [Read] button.

### Window



### Displayed items

Item	Description
Drive to read from	Select the drive where the data to be read is stored.
Target logging setting data	Select the data item to be read.


### Point

Any existing data (data logging setting with the same setting number or common setting) on the target is overwritten.

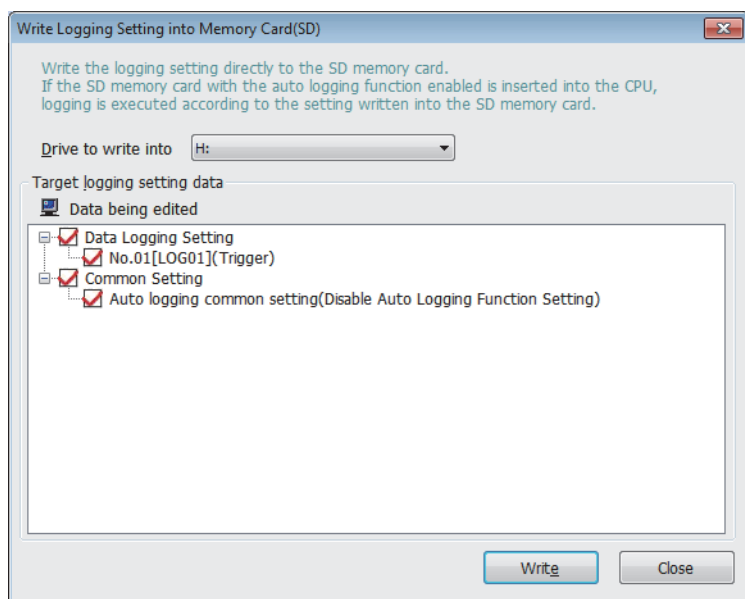
## ■ Writing logging setting to SD memory card

The following procedure is to write the settings being edited in a format with which the CPU module can operate. Once writing the settings directly into an SD memory card attached to the personal computer and attaching the card to the CPU module, the data logging starts.

### Operating procedure

1. Attach an SD memory card to the personal computer.
2. Open the following window.  
 [Project] ⇒ [Write Logging Setting into Memory Card(SD)]
3. Select the drive to which data is written and data to be written.
4. Click the [Write] button.

### Window



### Displayed items

Item	Description
Drive to write into	Select the drive where the data to be written is stored.
Target logging setting data	Select the data to be written.

#### Point



Any existing data (data logging setting with the same setting number or common setting) on the target is overwritten.

## Setting data logging

This menu item launches a wizard that helps users to configure the required settings for using the data logging function.

 Edit item tree ⇒ [RCPU] ⇒ [Data Logging Setting] ⇒ [Edit] button

### ■ Logging type

The following window configures the data logging type and file format ( Page 146 Logging Type,  Page 382 Data output type).

### Window

First off, select a logging type.

**Logging type**  
Select a logging type.

**Continuous logging**  
Logging is carried out continuously at the specified data sampling intervals.  
Interval at which or conditions under which to carry out logging can also be specified.

**Trigger logging**  
By monitoring data, data before and after a condition held true is logged.

**File format**  
Select the file format which outputs the logging.


**Unicode text file**  
The data can be checked not only by GX LogViewer but also by text editor or table calculation soft.

**Binary file**  
The data in the file can be checked by GX LogViewer.  
Compare with Unicode text file, the file volume can be decreased.

### Displayed items

Item	Description	Setting range	Default
Logging type	Select the logging type.	<ul style="list-style-type: none"><li>• Continuous logging</li><li>• Trigger logging</li></ul>	Continuous logging
File format	Select the output file format. (  Page 155 Storage format of data logging files)	<ul style="list-style-type: none"><li>• Unicode text file</li><li>• Binary file</li></ul>	Unicode text file

## ■Collection

The following window configures the collection interval and/or collection start conditions (  Page 141 Data Collection Conditions)

### Window

Specify the sampling interval and start conditions.

Sampling interval

**Each scanning cycle**  
Samples data at each sequence scanning cycle.

**Time specification**     
Samples data at the specified time interval.  
 Sample data at the first END processing after the specified time has elapsed(W)

**Interrupt occurrence(M)**  
Sample data by multiple interval of specified interrupt cycle (interrupt cycle [ms]).  
The specified interrupt cycle set in GX Works3 parameter will be operated.  
Interrupt pointer(Q)   (1-50) x Interrupt cycle [ms]

**Condition specification**  
Specifies data sampling timing by device data conditions or step No.  
If both "device specification" and "step No. specification" are selected, data will be sampled when both conditions are met.

**Device specification**



Device	Conditional formula	Radix	Value
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Data type

**Step No. specification**  
Samples data when the status immediately before execution of the specified step satisfies the specified execution conditions.  
If two or more programs are being executed, specify the program name to sample data.

Program name	Step No.	Execution condition
<input type="text"/>	<input type="text"/>	<input type="text"/>

### Displayed items

Item	Description	Setting range	Default
Each scanning cycle	Select this item to collect scan data obtained for each scan operation.	—	—
Time specification	Collection time value	Specify the collection interval value.	• ms: 1 to 32767 • second: 1 to 86400
	Sample data at the first END processing after the specified time has elapsed	Select this checkbox to collect data at a timing when the first END processing is done after the specified time interval is elapsed.	—
Interruption Occurrence	Interrupt pointer	Select the interrupt cycle.	• I28: interrupt by internal timer • I29: interrupt by internal timer • I30: interrupt by internal timer • I31: interrupt by internal timer • I44: inter-module synchronous interrupt • I45: multiple CPU synchronous interrupt • I48: high-speed internal timer interrupt 2 • I49: high-speed internal timer interrupt 1
	Multiple	Specify a multiple applied to the interval value.	1 to 50
Condition specification	Device specification	Specify the data collection timing according to the device data conditions.	 Page 143 Device specification
	Step No. specification	Specify the data collection timing based on the step No.	 Page 144 Step No. specification

## ■Data

The following window configures the various items such as data format of the target collection device.

A

### Window

Set the data for logging.

A total of up to 128 device points can be set.  
Bit digit specification is using points corresponding to data type. (1 point for word type and 2 points for double-word type)

No.	Device		Data type	Size	Output type
	Head	Last			
001					
002					
003					
004					
005					
006					
007					
008					
009					
010					
011					
012					
013					
014					
015					
016					
017					
018					
019					
020					
021					
022					
023					
024					
025					
026					
027					
028					
029					
030					
031					
032					
033					
034					
035					
036					
037					
038					
039					
040					
041					


### Displayed items

Item	Description	Setting range	Default
No.	In this column, the data setting numbers from 001 to 128 are displayed.	—	—
Device	Head	Specify the start device number.	
	Last		
Data type	Select the type of data to be collected.	☞ Page 140 Data type	
Size	Specify the data size when the data type is set to "String" or "Raw".	1 to 256 bytes	
Output Format	Clicking the [...] button at the rightmost part of each row displays the "Output Format (integer-float)" list. Select the format to be used when data is output to the file.	☞ Page 382 Data output type	

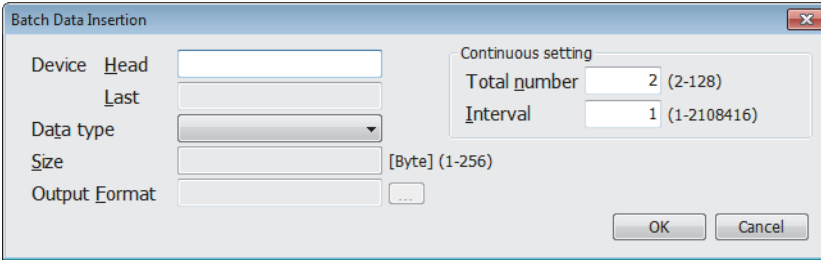
## ■ Batch insertion of data

The following window is to insert data items into the data list at once. Data is inserted into the blank rows of the list of the "Data" setting window sequentially from the top (when a setting already exists in the target insertion row, the row is skipped without overwriting it).


### Operating procedure

1. Open the following window.  
 [Edit] ⇨ [Batch Data Insertion]
2. Configure the setting items and continuous settings, and click the [OK] button.

### Window




### Displayed items

Item	Description	Setting range	Default	
Device	Same as the data setting (  Page 397 Data)	—	—	
Head				
Last				
Data type				
Size				
Output Format	Specify the total number of data items to be inserted at once.	2 to 128		
Continuous setting				Total number
Interval				Specify the device interval of data to be inserted at once.



## ■ Trigger

The following window specifies the trigger condition when the trigger logging is selected (  Page 148 Trigger Condition)

### Window

Make trigger setting.

Condition specification  
Sets trigger condition with device data values and step No. If both "Device change specification" and "Step No. specification" are selected, an AND condition of each setting is required to be met.

Device change specification

Device  Conditional formula  Radix  Value



Data type

Step No. specification  
Conditions met when the status immediately before execution of the specified step satisfies the specified execution conditions.  
 If two or more programs are being executed, specify the program name to satisfy the conditions.


Target program name  Step No.  Execution condition

When trigger instruction executed  
Trigger conditions met when LOGTRG instruction is executed.

### Displayed items

Item		Description	Setting range	Default
Condition specification	Device change specification	Configure the trigger condition based on the device data condition.	 Page 148 Device change specification	Checked
	Step No. specification	Configure the trigger condition based on the step No.	 Page 144 Step No. specification	—
When trigger instruction executed		Trigger condition is established when the LOGTRG instruction is executed.	—	—

## ■Number of records

The following window specifies the number of records to be output before and after trigger occurrences when the trigger logging is selected (  Page 147 Number of records)

### Window

Data before and after trigger condition rises will be logged.  
Specify the numbers of records before and after trigger.

No. of records (before trigger)  [Record] (0-999999)

No. of records (after trigger)  [Record] (1-1000000)

Total No. of records  [Record] (1-1000000)

### Displayed items

Item	Description	Setting range	Default
No. of records (before trigger)	Specify the number of records to be output as pre-trigger record.	0 to 999999	1
No. of records (after trigger)	Specify the number of records to be logged during and after a trigger occurrence.	1 to 1000000	1
Total No. of records	View the total number of pre-trigger and post-trigger records.	—	2

## ■Output

The following window specifies the items to be output into the file. (☞ Page 382 Data output type)

A

### Window

Setting items to be output to a file.

<p><b>Date</b> Carry out the logging with a time stamp attached to data.</p> <p><input checked="" type="checkbox"/> Output date <input type="button" value="Set Date Line Format..."/></p>	<p><b>Device comment output</b> Output device comments instead of devices in the data.</p> <p><input checked="" type="checkbox"/> Output device comments(B)</p> <p>Target memory <input type="text"/></p> <p>Target comment file name <input type="text"/></p> <p>Target comment number <input type="text"/> (1-16) <small>It can be checked by 'View-&gt; Multiple Comment Display Setting' of GX Works3.</small></p>
<p><b>Trigger information</b> Data on which a trigger occurred is logged after attached with a mark.</p> <p><input checked="" type="checkbox"/> Output trigger information <input type="button" value="Set String for Trigger..."/></p>	<p><b>Comment</b> Outputs comments on the settings. The specified comment will be output at the top of the logging file.</p> <p><input checked="" type="checkbox"/> Output comments</p> <p>Comment(K) <input type="text" value="LOG01"/></p>
<p><b>Index</b> Output index numbers for checking the continuity of logging.</p> <p><input checked="" type="checkbox"/> Output index</p>	
<p><b>Data sampling interval</b> Output data sampling interval time.</p> <p><input checked="" type="checkbox"/> Output data sampling interval</p>	
<p><b>Execution program name or execution program No.</b> Output execution program name and execution program No. which the data has been sampled.</p> <p><input checked="" type="checkbox"/> Output execution program name or program No.</p>	
<p><b>Execution step No.</b> Output step No. which the data has been sampled.</p> <p><input checked="" type="checkbox"/> Output execution step No.</p>	

## Displayed items


Item		Description	Setting range	Default
Date	Output date	Add a time stamp to data for the data logging. The date/time format to be output can be configured by clicking the [Date Line Output Format] button.	*1*2	Checked
Trigger information	Output trigger information	Add a mark to data items that are associated with a trigger occurrence for the data logging. A character string to be added to the trigger occurrence data can be configured by clicking the [String for Indicating Trigger Occurrence] button	256 characters or less	
Index	Output index	Output the index number used for checking the logging continuity.	—	
Data sampling interval	Output data sampling interval	Output the data collection interval.		
Execution program name or execution program No.	Output execution program name or execution program No.	Output the execution program name and execution program number used for the data collection.		
Execution step No.	Output execution step No.	Output the step No. used for the data collection.		
Device comment output	It is output as the device comment.	If the collection data was specified with the device, the device comment is output in addition to the device.		Not checked
	Target memory	Select the storage memory for the comment file used when the comment is displayed.	<ul style="list-style-type: none"> <li>• Memory card (SD)</li> <li>• Data memory</li> </ul>	—
	Target comment file name	Set the comment file name used when the comment is displayed.	60 characters or less	—
	Target comment No. <sup>*3</sup>	Specify the comment number to be displayed.	1 to 16	—
Comment	Output comments	Output the comment at the top of the file.	—	Checked
	Comment <sup>*4</sup>	Input the comment in this box.	256 characters or less (line feed not allowed)	LOG01

\*1 Data output format can be created by combining the following formats.


- Year: YYYY for four-digit expression; YY for two-digit expression
- Month: MM
- Day: DD
- Hour: hh
- Minute: mm
- Second: ss
- Millisecond: ms (three-digit expression), or s, ss, sss, ssss, sssss, ssssss, or sssssss (second unit after the decimal point, maximum of seven digits)

Example: YY/MM/DD hh:mm:ss.sss → 10/10/13 09:44:35.241

\*2 When either of "Year," "Month," "Day," "Hour," "Minute," or "Second" is omitted, if opening the data logging file by GX LogViewer, the index expression is used rather than the time expression.


 GX LogViewer Version1 Operating Manual

\*3 The target comment No refers to the numbers, which correspond to the comment numbers in the [View] ⇒ [Multiple Comment Display Setting] of the Engineering Tool.

\*4 When the following user action is detected, character entry will be disabled ( Page 415 Supported characters within the data logging setting)

- Entered a character which cannot be handled with the OS language character code.
- Entered a character whose language code is different from the one for characters already input in the same data logging setting.

## ■ Save

The following window configures the target storage for data logging file and switching timing of storage files (  Page 152 Switching to a storage file)

A

## Window

Specify the save destination and switching timing of data logging files.

Logging file save setting

Save destination in the SD memory card

Specify the logging file save directory (folder name). Data will be stored in the following folder:

/LOGGING/ LOG01

Folder to store file to be saved

Select the additional information to the folder name stored in file to be saved.

Date(J)  Time(K) Example 00000001

File name

Simple setting

The following information can be added to the file name. Select the information.

Folder name  Date  Time

Optional setting

The optional information set here can be added to the file name. Date and time can also be set. For details, refer to the manual.

Format

Example 00000001.TXT

Add date type

Date to establish file switching condition(U)  
Adding date when the file switching condition was satisfied.

File creation date(V)  
Adding date to the file when it was created.

File switching setting

Number of files to be saved

Specify the maximum number of files to be saved.

Number of files to be saved  (1 to 65535)

Operation when exceeds the number of files:

Overwrite  
Files with lower numbers are deleted and logging continues.

Stop  
Logging stops.

File switching timing

Specify the file switching timing.

Number of records  [Record] (1 to 65500)  
A logging file is switched to the new one when the number of records exceeds the specified value.

File size  [KB] (10 to 16384)  
A logging file is switched to the new one when the file size exceeds the specified value.

## Displayed items

Item		Description	Setting range	Default	
Logging file save setting	Save destination in the SD memory card	Specify the storage folder for the data logging file.	60 characters or less (double-byte character not allowed)	LOG01	
	Folder to store file to be saved	Select information to be added to the name of the folder which stores the storage file.	—	Not checked	
	File name	Simple setting	Select information to be added to the name of the storage file.	—	Checked
		Optional setting	Specify information format to be added to the name of the storage file.	*1*2	—
		Add date type	Add date/time when the file switching condition is satisfied and when the file is created if the optional setting is selected.	—	—
File switching setting	Number of files to be saved	Number of files to be saved	Specify the maximum number of storage files.	1 to 65535	1
		Operation when the number of files exceeds the set value	Select the action when the number of storage files is exceeded (☞ Page 154 Action to take when the maximum number of storage files is exceeded)	<ul style="list-style-type: none"> <li>• Overwrite</li> <li>• Stop</li> </ul>	Overwrite
	File switch timing*3	Select the timing at which the file is replaced with new one (☞ Page 153 File switching condition)	<ul style="list-style-type: none"> <li>• Number of records: 1 to 65500</li> <li>• File size: 10 to 16384K bytes</li> </ul>	—	

\*1 Date and/or time can be added in any format by using the following character strings.

- Year: YYYY for four-digit expression; YY for two-digit expression
- Month: MM
- Day: DD
- Day of the week: ddd (Sunday: Sun, Monday: Mon, Tuesday: Tue, Wednesday: Wed, Thursday: Thu, Friday: Fri, Saturday: Sat)
- Hour: hh
- Minute: mm
- Second: ss


Example: for June 18, 2014 (Wednesday), 09:30:15, YYYYMMDDdddhmmss → 20140618Wed093015\_00000001.bin

Also when using the additional information simply as a character string rather than the above format, any character string can be added by enclosing it with double-quotation marks (" ").

Example: when adding the character string "address" to the file name, "address" → address\_00000001.bin can be used.

- \*2 Maximum of 64 characters (including underscore (\_), serial number (eight digits), period, and extension) can be used. However, when specifying a character string that contains double quotation marks (" "), the maximum number reduces by the number of the double quotation marks.
- \*3 Reducing the setting value results in frequent file switching, so that it is possible that the scan time and/or the device/label access service processing time can be extended.

## ■ Logging operation

The following window specifies the data logging operation when the mode transfers to RUN mode (  Page 156 Setting the behavior at the time of a transition to RUN)

### Window

Specify logging operation.

Operation at transition to RUN  
 Specifies operation at transition to RUN in case that the PLC module power source is switched to RUN from ON or by reset operation, or the PLC module operation status is switched to RUN from STOP.

Auto Start  
 Start by User Operation

### Displayed items

Item	Description	Setting range	Default
Operation at transition to RUN	Select the operation when the mode transfers to RUN mode.	<ul style="list-style-type: none"> <li>• Auto Start</li> <li>• Start by User Operation</li> </ul>	Auto Start

## ■Finish

The following window is to give the data logging setting a name.

### Window

All data required for data logging have been collected.  
Press the "Complete" button to complete setting.

To reflect the settings to the PLC, select [Online]->[Write Logging Setting].

Name the data logging.

Data logging name

Free space volume below in SD memory card will be necessary to execute logging of the set content.  
Larger volume might be necessary depending on status of SD memory card.


Total Size of Output Logging Files  [MB]


To execute logging of the settings, the following internal buffer capacity is required.  
Please set internal buffer capacity as needed.

Required Internal Buffer Capacity in Logging  [KB]

The internal buffer capacity can be set in  
'Parameter->Control CPU->CPU Parameter->Memory/Device Setting->Internal Buffer Volume Setting' of GX Works3.  
Default value: 128[KB]

### Displayed items

Item	Description	Setting range	Default
Data logging name*1	Give the data logging setting being configured a name.	32 characters or less	LOG01
Total Size of Output Logging Files	View the total capacity of the data logging file which is output based on the specified settings. The total capacity can be increased/decreased by adding/removing the items to be output to the file.	—	1
Capacity of internal buffer necessary for logging	View the internal buffer capacity required to execute the data logging based on the specified settings. This value can be specified with the internal buffer capacity setting of engineering tool (  Page 154 Internal buffer capacity setting)	—	1

\*1 When the following user action is detected, character entry will be disabled (  Page 415 Supported characters within the data logging setting)

- Entered a character which cannot be handled with the OS language character code.
- Entered a character whose language code is different from the one for characters already input in the same data logging setting.



## Common setting

The common settings include the auto logging common setting.

### ■ Auto logging common setting

The following window configures the required settings for using the auto logging function. (☞ Page 157 Auto logging)

#### Window

Set for auto logging function.

Enable the auto logging function  
 Inserting an SD memory card into the CPU in RUN status starts operation.  
 Save the target data logging setting in the SD memory card before inserting the card.

The auto logging function does not operate according to the logging setting in the data memory.  
 Logging being operated when the SD memory card is inserted will stop.

**Auto logging terminate condition**  
 Specify conditions for terminating auto logging.  
 If two or more conditions are selected, logging will terminate when any of the conditions is met.

**Data logging stop**  
 Select when to terminate auto logging operation.

When all data loggings stop  
 When any of the data loggings stops

**Timer**

Complete with timer  
 Terminates logging after a specified period of time has elapsed after logging starts.

Elapsed time  [s] (1-86400)

#### Displayed items


Item	Description	Setting range	Default
Enable the auto logging function	Select this item to use auto logging.	—	Not checked
Auto logging terminate condition	Data logging stop	<ul style="list-style-type: none"> <li>• When all data loggings stop</li> <li>• When any of the data loggings stops</li> </ul>	When all data loggings stop
	Timer	—	—
	Complete with timer	Select this checkbox to stop the auto logging operation by a timer.	—
	Elapsed time	Specify the time interval from the start of the data logging until stopping it.	1 to 86400 seconds

## Online operation

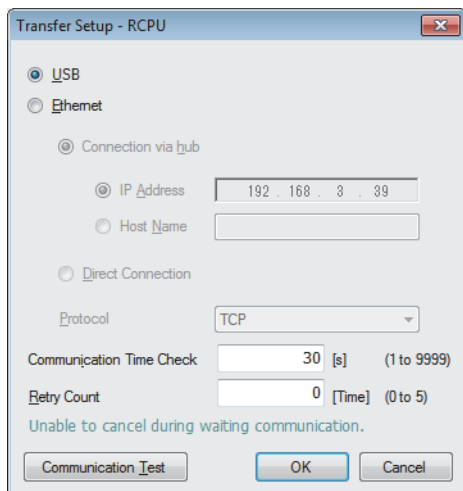
The online operation enables users to read/write/remove the data logging settings, view the data logging status, and operate the data logging file.

### Transfer setup

The following window specifies the communication route between the CPU module and a personal computer.

 [Online] ⇒ [Transfer Setup]

### Window



### Displayed items

Item		Description	
USB		Select this item when connecting with a USB cable.	
Ethernet	Connection via HUB	IP Address	
		Host Name	
	Direct Connection		Select this item for direct connection with an Ethernet cable.
	Protocol		Select the protocol used when connecting via a hub.
Communication Time Check		Specify the communication time.	
Retry Count		Specify the number of retries.	
[Communication Test] button		This button checks the communication status.	

## ■ Reading logging setting

The following procedure reads the data logging setting from the target memory.

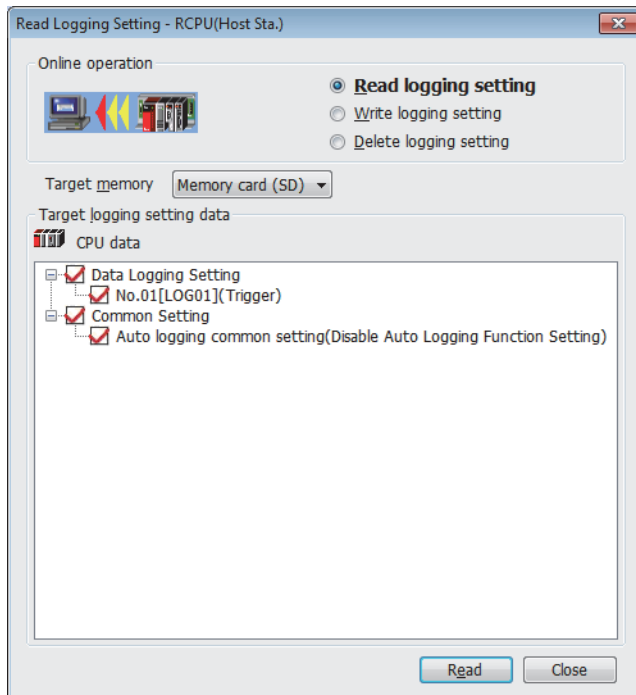
### Operating procedure

1. Open the "Read Logging Setting" window.

 [Online] ⇒ [Read Logging Setting]

2. Select the memory where the data to be read is stored from the "Target Memory" list.
3. Select the checkbox corresponding to the data item to be read in the "Target logging setting data" list, and click the [Read] button.

### Window



### Displayed items

Item	Description
Target Memory	Select the memory where the data to be read is stored.
Target logging setting data	Select the data item to be read.

#### Point

Any existing data (data logging setting with the same setting number or common setting) on the target is overwritten.

## ■ Writing logging setting

The following procedure is to write the data logging setting to the target memory.

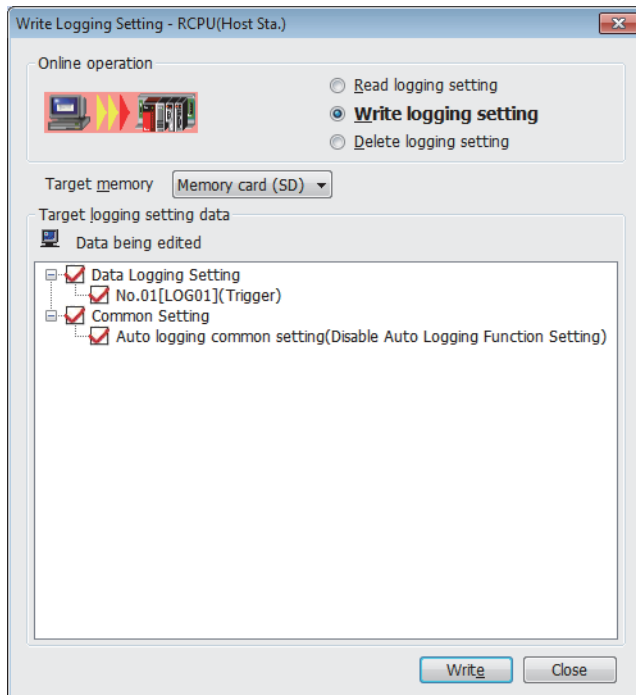
### Operating procedure

1. Open the "Write Logging Setting" window.

 [Online] ⇒ [Write Logging Setting]

2. Select the memory where the data to be written is stored from "Target Memory" list.
3. Select the checkbox in the "Target logging setting data" list corresponding to the data item to be written, and click the [Write] button.

### Window



### Displayed items

Item	Description
Target Memory	Select the memory where the data to be written is stored.
Target logging setting data	Select the data to be written.

#### Point


Any existing data (data logging setting with the same setting number or common setting) on the target is overwritten.

## ■Deleting logging setting

The following procedure removes the data logging setting on the target memory.

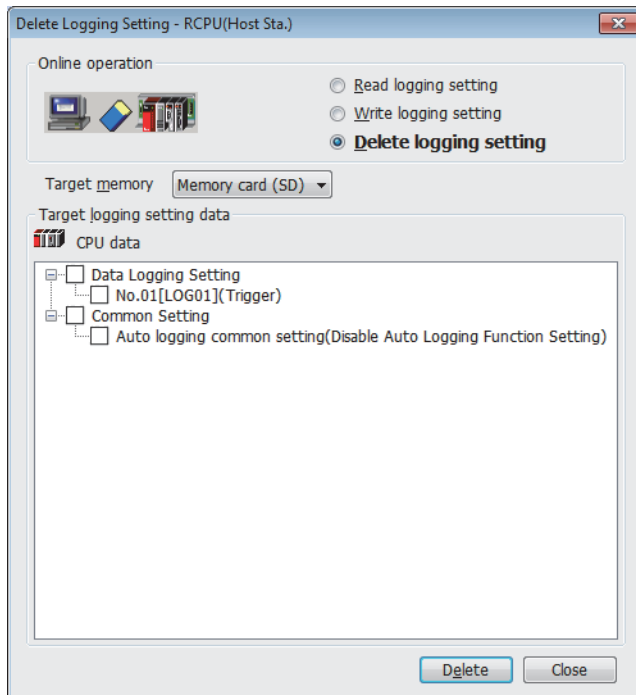
### Operating procedure

1. Open the "Delete Data Logging Setting" window.

 [Online] ⇒ [Delete Data Logging Setting]

2. Select the memory where the data to be removed is stored from the "Target Memory" list.
3. Select the checkbox corresponding to the data item to be removed in the "Target logging setting data" list, and click the [Delete] button.

### Window



### Displayed items

Item	Description
Target Memory	Select the memory where the data to be removed is stored.
Target logging setting data	Select the data to be removed.

## ■Logging status and operation

The following procedure is to execute or stop the data logging. Also the data logging status can be checked through this procedure.

### Operating procedure

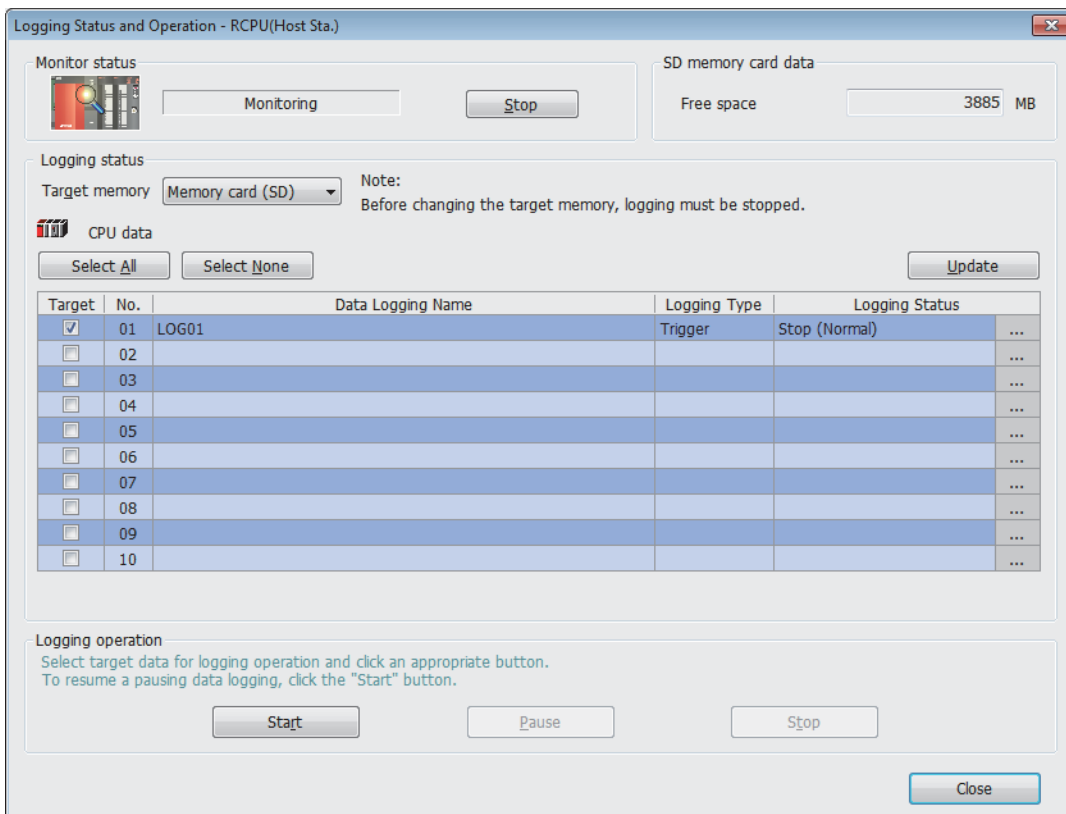
1. Open the "Logging Status and Operation" window.

 [Online] ⇒ [Logging Status and Operation]

2. Specify the target memory (either data memory or SD memory card) where the effective setting data is stored.
3. Select the checkbox corresponding to the setting number to be executed (One or more numbers can be selected).
4. Start the data logging by clicking the [Start] button (when multiple items are selected, they are executed simultaneously).
5. To suspend the data logging, click the [Pause] button, and to stop the data logging, click the [Stop] button (when multiple items are selected, they are executed simultaneously).

- The data logging cannot be started even when writing the setting and turning power off and on or resetting. Be sure to click the [Start] button to start the data logging (except for the auto logging).
- With regards to the trigger logging, the data logging setting registration attempt fails if the trigger condition is satisfied.
- It takes a certain time to stop or suspend the data logging after either of these commands is issued by CPU Module Logging Configuration Tool (because the data logging is not stopped or suspended unless the data stored in the internal buffer data has been transferred into the SD memory card in response to these commands).
- There may be a case where a time-out error occurs and the data logging is suspended after CPU Module Logging Configuration Tool starts the logging.

## Window



## Displayed items


Item	Description	
Monitor Status	[Start (Stop)] button	Start or stop monitoring.
SD memory card data	Free space	View the amount of free space of the SD memory card.
Logging status	Target Memory	Select the memory used for this operation.*1
	[Select All] button	Select all the checkboxes in the setting data list.
	[Deselect All] button	Clear all the checkboxes in the setting data list.
	[Update] button	Update monitoring status.
	Target	Select the target setting data for this operation (one or more numbers can be selected).
Logging operation	[...] button	Clicking this button when an error occurs displays the error details window.
	[Start] button	Execute the logging of the selected setting data.
	[Pause] button	Suspend the logging of the selected setting data.
	[Stop] button	Stop the logging of the selected setting data.

\*1 This menu item can be selected only when all the data logging statuses are "Stop."

## ■ Logging file operation

The following procedure is to save or remove data logging files on an SD memory card from/to the personal computer.

### Operating procedure

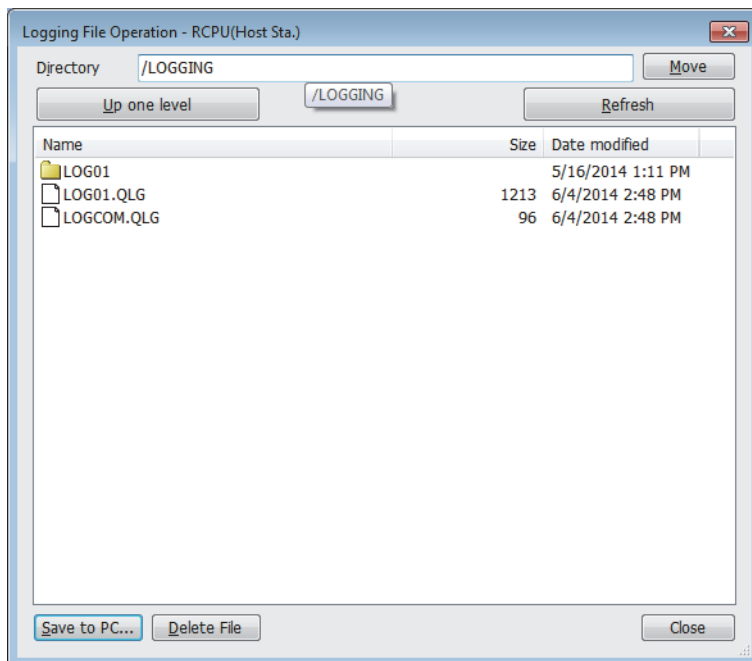
1. Open the "Logging File Operation" window.  
 [Online] ⇒ [Logging File Operation]
2. Specify the directory and select the targeted file.
3. To save click the [Save to PC] button, and to remove click the [Delete] button.

### Point

Attempting the following operations may result in delay of other monitor update because a certain time period is required for saving data logging files.

- When saving data logging files during the data logging execution.
- When saving a large data logging file.

### Window



### Displayed items

Item	Description
Directory	View the path to the displayed folder. To change the folder, specify the target folder path.
[Move] button	Move to the specified folder.
[Up one level] button	Move up one level in the folder hierarchy.
Refresh	Update the displayed content.
[Save to PC] button	Display the "Save As" window and save the selected file to the personal computer.
[Delete] button	Remove the selected file or folder.


## Help operation

The following procedures allow to view or use the help function of CPU Module Logging Configuration Tool.

### ■Opening user's manual

E-Manual Viewer opens and its manual is displayed.


#### Operating procedure

 [Help] ⇒ [Open Manual]

### ■Connection to MITSUBISHI ELECTRIC FA Global Website

Access Mitsubishi Electric Corporation FA site home page.

#### Operating procedure

 [Help] ⇒ [Connection to MITSUBISHI ELECTRIC FA Global Website]

### ■Checking version information

Check the version of CPU Module Logging Configuration Tool.

#### Operating procedure

 [Help] ⇒ [Version Information]



## Supported characters

This section describes the supported characters.

### Supported characters for CPU Module Logging Configuration Tool

Any characters that can be expressed by Unicode are supported. However, the supported characters vary for each position as shown in the following table. Note that if attempting to input an unsupported character, the entry is rejected or a message window appears in response to the improper entry.

Place where character is used		Support status of the target character																Japanese characters	
		(SP)*1	"	'	*	+	,	/	:	;	<	>	?	[	\	]			.
Data logging setting	• Data logging name • Trigger occurrence information column • Comment	○	×	○	○	○	×	○	○	×	○	○	○	○	○	○	○	○	○
	Date/time column of data row output format	○	×	○	○	○	×	○	○	×	○	○	○	○	○	○	○	○	×
	File save destination	×	×	○	×	○	×	×	×	×	×	×	○	×	○	×	×	×	×
Logging file operation	Directory	×	×	○	×	×	×	○	×	×	×	×	○	×	○	×	○	×	×
Connection to Mitsubishi Electric Corporation FA site	URL	×	×	○	○	○	×	○	○	×	○	○	○	○	○	○	○	○	○

\*1 (SP) means a space.

### Supported characters for file and/or folder (directory) name

Characters in the shaded area can be used.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NULL		(SP)	0	@	P	`	p				-	タ	ミ		
1			!	1	A	Q	a	q			。	ア	チ	ム		
2			"	2	B	R	b	r			「	イ	ツ	メ		
3			#	3	C	S	c	s			」	ウ	フ	モ		
4			\$	4	D	T	d	t				エ	ト	ヤ		
5			%	5	E	U	e	u			,	オ	ナ	ユ		
6			&	6	F	V	f	v			.	カ	ニ	ヨ		
7			'	7	G	W	g	w			ヲ	キ	ヌ	ウ		
8			(	8	H	X	h	x			ア	ク	ネ	リ		
9			)	9	I	Y	i	y			イ	ケ	ノ	ル		
A			*	:	J	Z	j	z			ウ	コ	ハ	レ		
B			+	;	K	[	k	{			エ	サ	ヒ	ロ		
C			,	<	L	¥	l				ヤ	シ	フ	ワ		
D			-	=	M	]	m	}			ユ	ス	ハ	ソ		
E			.	>	N	^	n	~			ヨ	セ	ホ	シ		
F			/	?	O	_	o				ツ	ソ	マ	。		

### Supported characters within the data logging setting

The following actions prevent characters from being input for the data logging setting.

- Entered a character which cannot be handled with the OS language character code. Example: A user attempted to input one-byte katakanas to the data logging setting created by English version CPU Module Logging Configuration Tool.
- Entered a character whose language code is different from the one for characters already input in the same data logging setting. Example: A user attempted to enter a text including both Japanese-specific characters (e.g. one-byte katakana) and Chinese characters.

## Installing a USB driver

To communicate with the CPU module via USB, a USB driver needs to be installed. This section describes the installation procedure of a USB driver. If multiple MELSOFT products are installed, refer to their installed location.

### ■Windows® XP

#### Operating procedure

1. Connect a personal computer to the CPU module with a USB cable, and power on the programmable controller.
2. Select "Install from a list or specific location (Advanced)" on the "Found New Hardware Wizard" window.
3. On the next window, select "Search for the best driver in these locations". Check the "Include this location in the search" checkbox, and specify the "Easysocket\USBDrivers" folder where the CPU Module Logging Configuration Tool has been installed.

#### Precautions

If the driver cannot be installed, check the following setting on Windows®.

Select [Control Panel] ⇒ [System] ⇒ [Hardware], and click the [Driver Signing] button. If "Block - Never install unsigned driver software" is selected, the USB driver may not be installed. Select "Ignore - Install the software anyway and don't ask for my approval" or "Warn - Prompt me each time to choose an action", and then install the USB driver.

### ■Windows Vista®

#### Operating procedure

1. Connect a personal computer to the CPU module with a USB cable, and power on the programmable controller.
2. Select "Locate and install driver software (recommended)" on the "Found New Hardware" window.
3. On the next window, select "Browse my computer for driver software (advanced)".
4. On the next window, select "Search for the best driver in these locations". Check the "Include subfolders" checkbox, and specify the "Easysocket\USBDrivers" folder where the CPU Module Logging Configuration Tool has been installed.

#### Precautions

If "Windows can't verify the publisher of this driver software" appears on the "Windows Security" window, select "Install this driver software anyway".

### ■Windows® 7 and later

#### Operating procedure

1. Connect a personal computer to the CPU module with a USB cable, and power on the programmable controller.
2. Select [Start] ⇒ [Control Panel] ⇒ [System and Security] ⇒ [Administrative Tools] ⇒ [Computer Management] ⇒ [Device Manager]. Right-click "Unknown device", and click "Update Driver Software".
3. On the window appeared, select "Browse my computer for driver software". Then, specify the "Easysocket\USBDrivers" folder where the CPU Module Logging Configuration Tool has been installed on the next window.

# Appendix 10 Added and Enhanced Functions

This section describes added and enhanced functions of the CPU module and the engineering tool, as well as the corresponding firmware versions of the CPU module and software versions of the engineering tool.

—: No restrictions on the firmware version of the CPU module and the software version of the engineering tool

## Programmable controller CPU

Added and enhanced function	Firmware version of the CPU module	Software version of the engineering tool	Reference
LED specifications (enhanced use of the FUNCTION LED)	"06" or later	"1.005F" or later	Page 181 LED display setting
Memory dump function			Page 166 MEMORY DUMP FUNCTION
Real-time monitor function	"06" or later	"1.40S" or later*1	Page 176 REAL-TIME MONITOR FUNCTION GX LogViewer Version1 Operating Manual
CC-Link IE Controller Network function of the RJ71EN71		"1.005F" or later	MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup), MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)
Mounting of MELSEC iQ-R series modules occupying two slots	"08" or later	—	For the applicable modules, refer to the following. MELSEC iQ-R Module Configuration Manual

\*1 Indicates the compatible software version of GX LogViewer.

## Process CPU

No function is added or enhanced.

# MEMO

---

# INDEX

## Symbols

"\$MELPRJ\$" folder . . . . . 64

## A

Accuracy of Constant Scan . . . . . 93  
Accuracy of timers . . . . . 233  
Annunciator (F) . . . . . 227  
Auto logging . . . . . 157

## B

Behavior at the time of a transition to RUN . . . . . 156  
Block number . . . . . 249  
Block switching method . . . . . 249  
Boot operation . . . . . 128

## C

Character string constant . . . . . 261  
Clearing file registers . . . . . 250  
Clock function . . . . . 75  
Constant scan . . . . . 93  
Constants . . . . . 259  
Continuous logging . . . . . 146  
Controlled module . . . . . 190  
Counter (C) . . . . . 238  
CPU buffer memory . . . . . 64  
CPU buffer memory access device . . . . . 246  
CPU Module Logging Configuration Tool . . . . . 376  
CPU parameters . . . . . 372

## D

Data inconsistency . . . . . 56  
Data logging . . . . . 139  
Data logging file . . . . . 155  
Data logging states . . . . . 150  
Data memory . . . . . 64  
Data register (D) . . . . . 241  
Database . . . . . 95  
Database access instructions . . . . . 96  
Database function . . . . . 95  
Decimal . . . . . 259  
Decimal constant (K) . . . . . 259  
Device initial value . . . . . 111  
Device/label access service processing . . . . . 132  
Device/label memory . . . . . 63  
Double-precision real number . . . . . 260

## E

Edge relay (V) . . . . . 229  
END processing . . . . . 23  
Error clear . . . . . 182  
Error codes . . . . . 270  
Event execution type program . . . . . 37  
Event history . . . . . 184  
Event history file . . . . . 185  
Exact differential . . . . . 89

## F

File password . . . . . 221  
File register (R/ZR) . . . . . 249  
File register setting . . . . . 250  
File registers common to all programs . . . . . 250  
Files . . . . . 66  
Fixed scan execution mode . . . . . 35  
Fixed scan execution type program . . . . . 32  
Fixed scan interval . . . . . 33  
Function device . . . . . 242  
Function input (FX) . . . . . 242  
Function output (FY) . . . . . 242  
Function register (FD) . . . . . 242

## G

Global device . . . . . 262  
Global pointer . . . . . 253  
GX LogViewer . . . . . 376

## H

Hexadecimal . . . . . 259  
Hexadecimal constant (H) . . . . . 259  
High-speed timer . . . . . 231  
High-speed timer (T/ST) . . . . . 231

## I

I/O No. specification device (U) . . . . . 258  
I/O refresh . . . . . 22  
Index modification . . . . . 247,266  
Index register (Z) . . . . . 247  
Index register setting . . . . . 248  
Indirect specification . . . . . 269  
Inexact differential . . . . . 89  
Initial execution type program . . . . . 31  
Initial processing (when powered on) . . . . . 22  
Initial processing (when switched to RUN) . . . . . 22  
Initial scan time . . . . . 23  
Initial scan time monitoring time . . . . . 23  
Input (X) . . . . . 226  
Internal buffer . . . . . 154  
Internal buffer capacity setting . . . . . 110  
Internal relay (M) . . . . . 227  
Interrupt factor . . . . . 256  
Interrupt pointer (I) . . . . . 256  
Interrupt pointer number . . . . . 256  
Interrupt priority . . . . . 84  
Interrupt program . . . . . 45  
IP filter . . . . . 221

## J

Jump instructions . . . . . 253

## L

Label initial value . . . . . 111  
Latch . . . . . 120  
Latch (1) . . . . . 120

Latch (2) . . . . .	120
Latch clear . . . . .	125
Latch relay (L) . . . . .	227
Librarization of programs . . . . .	41
Link direct device . . . . .	243
Link refresh . . . . .	244
Link register (W) . . . . .	241
Link relay (B) . . . . .	227
Link special register (SW) . . . . .	241
Link special relay (SB) . . . . .	229
Local pointer . . . . .	254
LOGTRG instruction . . . . .	149
LOGTRGR instruction . . . . .	163
Long counter (LC) . . . . .	238
Long index register (LZ) . . . . .	247
Long retentive timer (LST) . . . . .	231
Long timer (LT) . . . . .	230
Low-speed timer . . . . .	231
Low-speed timer (T/ST) . . . . .	231

## M

Master control instructions . . . . .	252
Maximum counting speed for counters . . . . .	241
Memory card parameters . . . . .	373
Memory configuration . . . . .	61
Memory dump file . . . . .	171
Memory dump function . . . . .	166
Memory dump status . . . . .	171
Missing data . . . . .	155
Module access device . . . . .	245
Module label . . . . .	20
Momentary power failure . . . . .	59
Multiple CPU synchronous interrupt . . . . .	218
Multiple CPU system . . . . .	189
Multiple interrupt . . . . .	84
Multiple interrupt function . . . . .	84

## N

Nesting (N) . . . . .	252
Nesting structure . . . . .	252
Network No. specification device (J) . . . . .	258
Non-controlled module . . . . .	190

## O

Online change . . . . .	78
Online ladder block change . . . . .	78
Operating environment . . . . .	376
Other devices . . . . .	258
Output (Y) . . . . .	226
Output mode at operating status change (STOP to RUN) . . . . .	114
Overflow . . . . .	260

## P

Parameter No. . . . .	371
PAUSE state . . . . .	57
PID control . . . . .	89
PID control instruction . . . . .	89
Pointer (P) . . . . .	253
Pointer setting . . . . .	255
Priority for interrupt factors . . . . .	257
Process control function . . . . .	91

Process control instruction . . . . .	91
Process CPU . . . . .	20
Processing time . . . . .	358
Program arrangement change . . . . .	41
Program cache memory . . . . .	62
Program execution type . . . . .	31
Program list monitor . . . . .	23
Program memory . . . . .	62
Programmable controller CPU . . . . .	20

## R

Real constant (E) . . . . .	260
Real number . . . . .	260
Real-time monitor function . . . . .	176
Refresh . . . . .	207
Refresh data register (RD) . . . . .	251
Refresh memory . . . . .	64
Refresh memory setting . . . . .	251
Remote operation . . . . .	103
Remote password . . . . .	221
Remote RESET . . . . .	105
Remote RUN . . . . .	103
Remote STOP . . . . .	103
Resetting counters . . . . .	239
Retentive timer (ST) . . . . .	231
RUN state . . . . .	57
RUN-PAUSE contacts . . . . .	105

## S

Saving/restoring of the file register (R) block number . . . . .	54
Saving/restoring of the index register (Z, LZ) . . . . .	55
Scan execution type program . . . . .	32
Scan monitoring function . . . . .	73
Scan time . . . . .	23
Scan time monitoring time . . . . .	73
Scan time monitoring time setting . . . . .	73
SD memory card . . . . .	64
SD memory card forced disable . . . . .	126
Security function . . . . .	221
Security key authentication . . . . .	221
Serial number method . . . . .	249
Setting RUN-PAUSE contacts . . . . .	105
Setting time zone . . . . .	76
Single-precision real number . . . . .	260
Special register (SD) . . . . .	242
Special relay (SM) . . . . .	242
Stack file . . . . .	152
Standby type program . . . . .	41
Steps reserved for online ladder block change . . . . .	79
STOP state . . . . .	57
Storage file . . . . .	153
Subroutine call instruction . . . . .	44
Subroutine program . . . . .	44
System clock . . . . .	77
System device . . . . .	242
System parameters . . . . .	371

## T

Time setting . . . . .	75
Time zone . . . . .	76
Timer (T) . . . . .	230
Timer current value . . . . .	232

Timer limit setting . . . . .	232
Timer time limit value. . . . .	232
Transaction . . . . .	101
Trigger condition . . . . .	148
Trigger instruction . . . . .	149
Trigger logging . . . . .	146
Troubleshooting by symptom . . . . .	316

## U

---

Underflow. . . . .	260
Unicode text file . . . . .	97
Unit label assigned area . . . . .	251
User device . . . . .	226

## W

---

Watchdog timer . . . . .	73
--------------------------	----



# REVISIONS

\*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
June 2014	SH(NA)-081264ENG-A	First edition
July 2014	SH(NA)-081264ENG-B	Error correction
October 2014	SH(NA)-081264ENG-C	<ul style="list-style-type: none"> <li>■Added functions Memory dump function, real-time monitor function, LED specifications (enhanced)</li> <li>■Added or modified parts SAFETY PRECAUTIONS, RELEVANT MANUALS, Section 1.4, 1.5, 4.4, 6.2, 8.1, 12.1, 12.2, 12.3, 12.4, Chapter 14, Section 17.1, 17.2, 20.2, Chapter 22, Section 22.1, 22.2, 22.4, 22.5, 22.12, 22.13, 23.1, 23.2, 23.3, 23.4, 23.5, 23.6, 23.7, 23.8, Chapter 24, Section 25.1, 25.3, 26.4, 30.2, Appendix 1, 2, 4, 5, 6, 10</li> </ul>
November 2014	SH(NA)-081264ENG-D	<ul style="list-style-type: none"> <li>■Added or modified parts Section 1.4, 23.2, 28.2, Appendix 6</li> </ul>
January 2015	SH(NA)-081264ENG-E	<ul style="list-style-type: none"> <li>■Added models R08PCPU, R16PCPU, R32PCPU, R120PCPU</li> <li>■Added or modified parts SAFETY PRECAUTIONS, INTRODUCTION, TERMS, Section 1.4, 4.2, 4.4, Chapter 9, 10, Section 14.1, 14.2, 14.4, Chapter 15, 18, Section 23.12, Chapter 24, 25, Section 27.1, 27.2, 29.1, 29.2, 29.11, 30.3, Appendix 1, 2, 4, 5, 6, 7, 10</li> </ul>

Japanese manual number: SH-081224-E

This manual confers no industrial property rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2014 MITSUBISHI ELECTRIC CORPORATION



# WARRANTY

---

Please confirm the following product warranty details before using this product.

## **1. Gratis Warranty Term and Gratis Warranty Range**

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

- (1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
- (2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
  1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
  2. Failure caused by unapproved modifications, etc., to the product by the user.
  3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
  4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
  5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
  6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
  7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

## **2. Onerous repair term after discontinuation of production**

- (1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.
- (2) Product supply (including repair parts) is not available after production is discontinued.

## **3. Overseas service**

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

## **4. Exclusion of loss in opportunity and secondary loss from warranty liability**

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation of damages caused by any cause found not to be the responsibility of Mitsubishi, loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products, special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products, replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

## **5. Changes in product specifications**

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

# TRADEMARKS

---

Microsoft, Windows, Windows Vista, Windows NT, Windows XP, Windows Server, Visio, Excel, PowerPoint, Visual Basic, Visual C++, and Access are either registered trademarks or trademarks of Microsoft Corporation in the United States, Japan, and other countries.

Intel, Pentium, and Celeron are either registered trademarks or trademarks of Intel Corporation in the United States and other countries.

Ethernet is a trademark of Xerox Corp.

The SD and SDHC logos are either registered trademarks or trademarks of SD-3C, LLC.

All other company names and product names used in this manual are either trademarks or registered trademarks of their respective companies.





SH(NA)-081264ENG-E(1501)MEE

MODEL: RCPU-U-OU-E

MODEL CODE: 13JX20

## **mitsubishi electric corporation**

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN  
NAGOYA WORKS : 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA, JAPAN

When exported from Japan, this manual does not require application to the  
Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.