MITSUBISHI ELECTRIC

Programmable Controller

MELSEC iQ-R series

MELSEC iQ-R C Controller Module User's Manual

-R12CCPU-V

SAFETY PRECAUTIONS

(Read these precautions before using this product.)

Before using this product, please read this manual and the relevant manuals carefully and pay full attention to safety to handle the product correctly. If products are used in a different way from that specified by manufacturers, the protection function of the products may not work properly.

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

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

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

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

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

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

[Considerations for using this manual]

- For fail-safe circuits for a C Controller system, refer to the following.
  ☞ Page 354 General Safety Requirements

[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 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.

[Design Precautions]

⚠WARNING

- Configure a circuit so that the external power supply is turned off first and then the programmable controller. If the programmable controller is turned off first, an accident may occur due to an incorrect output or malfunction.
- For the operating status of each station after a communication failure, refer to manuals for the network used. For the manuals, please consult your local Mitsubishi representative. Incorrect output or malfunction due to a communication failure may result in an accident.
- When connecting an external device with a CPU module or intelligent function module to modify data of a running programmable controller, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification, parameter change, forced output, or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Improper operation may damage machines or cause accidents. When a Safety CPU is used, data cannot be modified while the Safety CPU is in SAFETY MODE.
- Especially, when a remote programmable controller is controlled by an external device, immediate action cannot be taken if a problem occurs in the programmable controller due to a communication failure. To prevent this, configure an interlock circuit in the program, and determine corrective actions to be taken between the external device and CPU module in case of a communication failure.
- Do not write any data to the "system area" and "write-protect area" of the buffer memory in the module. Also, do not use any "use prohibited" signals as an output signal from the CPU module to each module. Doing so may cause malfunction of the programmable controller system. For the "system area", "write-protect area", and the "use prohibited" signals, refer to the user's manual for the module used. For areas used for safety communications, they are protected from being written by users, and thus safety communications failure caused by data writing does not occur.
- If a communication cable is disconnected, the network may be unstable, resulting in a communication failure of multiple stations. Configure an interlock circuit in the program to ensure that the entire system will always operate safely even if communications fail. Incorrect output or malfunction due to a communication failure may result in an accident. When safety communications are used, an interlock by the safety station interlock function protects the system from an incorrect output or malfunction.

[Precautions for using C Controller modules]

- In the settings of refresh parameters, link output (LY) refresh devices and remote output (RY) refresh devices do not allow the specification of Y. Thus, the CPU module holds the device status even after its status is changed to STOP.

[Design Precautions]

⚠CAUTION

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

[Precautions for using C Controller modules]

- After a C Controller module is powered on or is reset, the time taken to enter the RUN status varies depending on the system configuration, parameter settings, program size, and/or connection status with external devices. Design circuits so that the entire system will always operate safely, regardless of the time.

[Security Precautions]

⚠WARNING

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

[Installation Precautions]

⚠WARNING

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

[Installation Precautions]

⚠CAUTION

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

[Wiring Precautions]

⚠WARNING

- Shut off the external power supply (all phases) used in the system before installation and wiring. Failure to do so may result in electric shock or cause the module to fail or malfunction.
- After installation and wiring, attach a blank cover module (RG60) to each empty slot before powering on the system for operation. Also, attach an extension connector protective cover[*1] to each unused extension cable connector as necessary. Directly touching any conductive parts of the connectors while power is on may result in electric shock.
  *1 For details, please consult your local Mitsubishi Electric representative.

[Wiring Precautions]

⚠CAUTION

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

[Wiring Precautions]

⚠CAUTION

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

[Startup and Maintenance Precautions]

⚠WARNING

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

[Startup and Maintenance Precautions]

⚠CAUTION

- When connecting an external device with a CPU module or intelligent function module to modify data of a running programmable controller, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification, parameter change, forced output, or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Improper operation may damage machines or cause accidents.
- Especially, when a remote programmable controller is controlled by an external device, immediate action cannot be taken if a problem occurs in the programmable controller due to a communication failure. To prevent this, configure an interlock circuit in the program, and determine corrective actions to be taken between the external device and CPU module in case of a communication failure.
- Do not disassemble or modify the modules. Doing so may cause failure, malfunction, injury, or a fire.
- Use any radio communication device such as a cellular phone or PHS (Personal Handy-phone System) more than 25cm away in all directions from the programmable controller. Failure to do so may cause malfunction.
- Shut off the external power supply (all phases) used in the system before mounting or removing the module. Failure to do so may cause the module to fail or malfunction.
- Tighten the screws within the specified torque range. Undertightening can cause drop of the component or wire, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
- After the first use of the product, do not perform each of the following operations more than 50 times (IEC 61131-2/JIS B 3502 compliant).
  Exceeding the limit may cause malfunction.
  - Mounting/removing the module to/from the base unit
  - Inserting/removing the extended SRAM cassette or battery-less option cassette to/from the CPU module
  - Mounting/removing the terminal block to/from the module
  - Connecting/disconnecting the extension cable to/from the base unit
- After the first use of the product, do not insert/remove the SD memory card to/from the CPU module more than 500 times. Exceeding the limit may cause malfunction.
- Do not touch the metal terminals on the back side of the SD memory card. Doing so may cause malfunction or failure of the module.
- Do not touch the integrated circuits on the circuit board of an extended SRAM cassette or a battery-less option cassette. Doing so may cause malfunction or failure of the module.

[Startup and Maintenance Precautions]

⚠CAUTION

- 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 of a control panel must be performed by qualified maintenance personnel with knowledge of protection against electric shock. Lock the control panel so that only qualified maintenance personnel can operate it.
- Before handling the module, touch a conducting object such as a grounded metal to discharge the static electricity from the human body. Wearing a grounded antistatic wrist strap is recommended. Failure to discharge the static electricity may cause the module to fail or malfunction.
- After unpacking, eliminate static electricity from the module to prevent electrostatic discharge from affecting the module. If an electrostatically charged module comes in contact with a grounded metal object, a sudden electrostatic discharge of the module may cause failure. For details on how to eliminate static electricity from the module, refer to the following.
  Antistatic Precautions Before Using MELSEC iQ-R Series Products (FA-A-0368)
- Use a clean and dry cloth to wipe off dirt on the module.

[Operating Precautions]

⚠CAUTION

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

[Disposal Precautions]

⚠CAUTION

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

[Transportation Precautions]

⚠CAUTION

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

INTRODUCTION

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

Relevant product

R12CCPU-V

CONSIDERATIONS FOR USE

C Controller module has an embedded real-time operating system, VxWorks, manufactured by Wind River Systems, Inc. in the United States. We, Mitsubishi, make no warranty for the Wind River Systems product and will not be liable for any problems and damages caused by the Wind River Systems product during use of C Controller module.
For the problems or specifications of the Wind River Systems product, refer to the corresponding manual or consult Wind River Systems, Inc.
Contact information is available on the following website.

- Wind River Systems, Inc.: www.windriver.com

CONTENTS

SAFETY PRECAUTIONS 1
INTRODUCTION 10
CONSIDERATIONS FOR USE 10
RELEVANT MANUALS 18
TERMS 19
GENERIC TERMS AND ABBREVIATIONS 19

PART 1 PART NAMES

CHAPTER 1 PART NAMES 22

PART 2 PROCEDURE BEFORE OPERATION

CHAPTER 2 STARTUP PROCEDURE 26

2.1 Initialization 28
2.2 Hardware Diagnostics 30
Hardware diagnostics timing 30
Diagnostics types 30
Performing diagnostics 31
Operation at error detection 34
2.3 Inserting and Removing an SD Memory Card 35
Insertion procedure 35
Removal procedure 35
2.4 Connecting and Disconnecting a USB Mass Storage Class-compliant Device 36
Connection procedure 36
Disconnection procedure 36
2.5 Switch Operation 37
RESET/STOP/RUN switch 37
MODE/SELECT switch 38
2.6 Creating a Project 40
2.7 Connecting to a Personal Computer 40
2.8 Setting Parameters 41
2.9 Writing Parameters to a C Controller Module 43
2.10 Creating a User Program 44
Programming procedure 44
2.11 Checking VxWorks Image File 48
2.12 Creating a Script File 49
2.13 Registering a User Program 51

PART 3 SYSTEM CONFIGURATION

CHAPTER 3 SYSTEM CONFIGURATION 54

3.1 Overall Configuration 54
3.2 Peripheral Configuration 55
Available software 56

SD memory card 57

PART 4 SPECIFICATIONS

CHAPTER 4 PERFORMANCE SPECIFICATIONS 60

CHAPTER 5 DOT MATRIX LED DISPLAY SPECIFICATIONS 62

CHAPTER 6 WIRING SPECIFICATIONS 64

6.1 Ethernet Port 64
6.2 RS-232 Interface 66
6.3 USB Connector 67
6.4 HMI (GOT) 68
Connection route 68

PART 5 C CONTROLLER MODULE OPERATION

CHAPTER 7 EXECUTING PROGRAMS 70

7.1 Execution Order 70
7.2 Initial Processing 70
7.3 I/O Access Timing 71

CHAPTER 8 OPERATION PROCESSING IN C CONTROLLER MODULE 72

8.1 Operation Processing Depending on Operating Status 72
8.2 Operation Processing at Momentary Power Failure 73

CHAPTER 9 MEMORY CONFIGURATION OF C CONTROLLER MODULE 74

9.1 Memory Configuration 74
Program memory 74
System memory 74
CPU buffer memory 74
Device/label memory 75
Data memory 75
SD memory card 75
USB Mass Storage Class-compliant device 75
9.2 Memory Operation 76
9.3 Memory Life 76
9.4 Files 77
Drive names and file systems 77
File type and storage destination memory 77
File and folder configuration 78

PART 6 FUNCTIONS

CHAPTER 10 FUNCTION LIST 82

CHAPTER 11 BASIC FUNCTIONS 84

11.1 Device Access Function 84

11.2 **Label Communication Function** ... 85
11.3 **Data Analysis Function** ... 87
11.4 **Interrupt Function to a C Controller Module** ... 88
Interrupt pointer number and interrupt factor ... 88
Interrupt procedure ... 89
11.5 **MELSEC Data Link Function Setting Function** ... 91
11.6 **Output Mode Setting Function at STOP to RUN** ... 93
11.7 **Boot Function of Parameters** ... 94
11.8 **Remote Operation Function** ... 97
Remote RUN/STOP ... 97
Remote PAUSE ... 98
Remote RESET ... 98
Remote operation and operating status of a C Controller module ... 100
11.9 **Clock Function** ... 103
Clock data setting ... 103
Time zone setting ... 104
Daylight saving time function ... 105
11.10 **RAS Function** ... 108
Watchdog timer (WDT) function ... 108
Self-diagnostics function ... 110
Error clear function ... 114
Event history function ... 116
11.11 **Fixed Cycle Processing Function** ... 121
Setting a fixed cycle processing interval ... 121
Checking the interval ... 121
11.12 **Inter-module Synchronization Function** ... 122
Fixed cycle synchronization function ... 123
Interaction with cycle of the fixed cycle communication of the multiple CPU system function ... 126
Parameter setting ... 128
11.13 **System Timer Mode Switching Function** ... 130

CHAPTER 12 MOTION MODULE CONTROL FUNCTION 131

12.1 **Module Labels** ... 131
Usage ... 133
Header file ... 136
12.2 **I/O Signals** ... 137
12.3 **Buffer Memory** ... 137
12.4 **MCFB** ... 138
Correspondence table for MCFBs and instruction classes ... 139
Definition of an instruction class ... 140
Usage ... 142

CHAPTER 13 Ethernet COMMUNICATION FUNCTIONS 145

13.1 **Connection with MELSOFT Product or GOT** ... 145
Connection via a hub ... 145
Direct connection ... 147
13.2 **SLMP Communication Function** ... 151
13.3 **FTP Function** ... 155
13.4 **Time Setting Function** ... 156
13.5 **Telnet Function** ... 157

13.6 Parameter Settings . . . 159
Own node settings . . . 159
External device configuration . . . 160
FTP server settings . . . 162
Time setting . . . 163
Telnet server settings . . . 164

CHAPTER 14 CC-Link IE Field Network Basic FUNCTION 165

14.1 Cyclic Transmission Function . . . 166
Data flow and link device assignment . . . 166
Link refresh . . . 170
Operation of link scans . . . 171
Group number setting . . . 172
Input and output status when an error occurs . . . 174
Output status when a C Controller module is in STOP state . . . 175
Reserved station specification . . . 175
14.2 Programming . . . 176
Interlock program for cyclic transmission . . . 176
14.3 Parameter Settings . . . 178
CC-Link IEF Basic setting . . . 178
14.4 CC-Link IE Field Network Basic Diagnostics . . . 183
Acquiring diagnostic information of remote stations . . . 184
14.5 Processing Time . . . 185
Link scan time . . . 185
Transmission delay time . . . 186

CHAPTER 15 SECURITY FUNCTION 187

15.1 Individual Identification Information Read Function . . . 187
15.2 File Access Restriction Function . . . 188
15.3 IP Filter Setting Function . . . 190
15.4 Service Setting Function . . . 192
15.5 Lockout Function . . . 194

CHAPTER 16 ACCESS FUNCTION USING NETWORK MODULE 195

16.1 Data Communication via Network . . . 196
16.2 Cyclic Transmission Function . . . 196
Access by link refresh . . . 198
Direct access . . . 202
Buffer memory access . . . 205
16.3 Transient Transmission Function . . . 206
Message communication . . . 206
Access to devices on another station . . . 207
16.4 Access Methods via a Network Module . . . 209
CC-Link IE Controller Network module . . . 209
CC-Link IE Field Network module . . . 211
CC-Link IE TSN module . . . 214
MELSECNET/H network module . . . 217
CC-Link module . . . 219
16.5 Processing Time . . . 220
Link refresh time . . . 220

Refresh cycle . . . 223
Transmission delay time of cyclic transmission . . . 224
Reduction of link refresh time . . . 227

CHAPTER 17 MULTIPLE CPU SYSTEM FUNCTION 228

17.1 Out-of-Group I/O Fetch . . . 229
Accessing controlled module . . . 229
Accessing non-controlled module . . . 229
17.2 Operation Settings . . . 231
Stop setting . . . 231
Synchronous startup setting . . . 232
Clock data . . . 232
17.3 Multiple CPU Parameter Check . . . 233
17.4 Data Communication between CPU Modules . . . 234
Used memory . . . 234
Fixed scan communication setting . . . 239
Error detection setting . . . 239
CPU number-based data assurance . . . 240
Functions that can be used for communication . . . 246
Communication using CPU buffer memory . . . 246
Communication using fixed cycle communication area . . . 248
17.5 Interrupt from Another CPU . . . 253
Interrupt from a C Controller module . . . 253
Interrupt from a programmable controller CPU . . . 255
17.6 Issuing an Interrupt to Another CPU . . . 259
Interrupt to a C Controller module (another CPU) . . . 259
Interrupt to motion CPU . . . 259

CHAPTER 18 FIRMWARE UPDATE FUNCTION 261

PART 7 DEVICES

CHAPTER 19 DEVICES 264

19.1 Device List . . . 264
19.2 Device Descriptions . . . 265

PART 8 C CONTROLLER MODULE OPERATION EXAMPLE

CHAPTER 20 SYSTEM CONFIGURATION EXAMPLE 268

CHAPTER 21 SETTING THE MODULE 270

21.1 Initializing a C Controller Module . . . 270
21.2 Setting Parameters . . . 270

CHAPTER 22 PROGRAMMING 277

22.1 Program Example and Control Description . . . 277
22.2 Creating a Project . . . 280
22.3 Preparing a User Program . . . 289

CHAPTER 23 CHECKING OPERATIONS 304
23.1 Enabling Outputs (Y) from a User Program 304
23.2 Checking Operations with the Dot Matrix LED and Lamps 305

PART 9 MAINTENANCE AND INSPECTION

CHAPTER 24 MAINTENANCE AND INSPECTION 308
24.1 Daily Inspection 308
24.2 Periodic Inspection 309

PART 10 TROUBLESHOOTING

CHAPTER 25 TROUBLESHOOTING PROCEDURE 312
25.1 Checking with CW Configurator 313

CHAPTER 26 TROUBLESHOOTING BY SYMPTOM 316
26.1 POWER LED on a Power Supply Module Turns OFF 316
26.2 READY LED on a C Controller Module Does not Turn ON (Green) 317
26.3 READY LED on a C Controller Module Keeps Flashing 318
26.4 Ethernet Communication between a C Controller Module and a Personal Computer cannot be Established 318
26.5 File Access Fails 320
26.6 Connection with Peripherals Fails 320
26.7 File Read from a C Controller Module Fails 320
26.8 An Error Occurs during User Program Execution 321
26.9 The Serial Communication cannot be Established 321
26.10 Communication with an Ethernet Device cannot be Established 321
26.11 Drive Name of an SD Memory Card is not Displayed Properly 322
26.12 Drive Name of a USB Mass Storage Class-compliant Device is not Displayed Properly 322
26.13 CC-Link IE Field Network Basic Functions cannot be Used 322
26.14 Module Labels and Public Labels are not Refreshed Properly 322
26.15 Consistency with a Written Project is Unknown 323
26.16 A Delay Occurs in the Link Scan Time of CC-Link IE Field Network Basic 323

CHAPTER 27 ERROR CODE LIST 324
27.1 Error Code System 324
27.2 Operation When an Error Occurs 326
27.3 Error Clear 326
27.4 Error Code List 326

CHAPTER 28 EVENT LIST 346
28.1 Guide for Reference of Event List 346
28.2 Event List 348

APPENDIX 350
Appendix 1 External Dimensions 350
Appendix 2 Compliance with EMC and Low Voltage Directives 351
Appendix 3 Handling of Batteries 352

Transport guidelines . . . 352
Handling of batteries and devices with built-in batteries in EU member states . . . 352
Disposal of a battery . . . 352
Appendix 4 General Safety Requirements . . . 354
When the $\overline{\text{ERR.}}$ contact of a power supply module is not used . . . 354
When the $\overline{\text{ERR.}}$ contact of a power supply module is used . . . 356
Fail-safe measures when a C Controller system fails . . . 358
Appendix 5 Calculation Method for Heat Generation of a C Controller Module . . . 359
Appendix 6 Special Relay List . . . 360
Appendix 7 Special Register List . . . 362
Appendix 8 Buffer Memory . . . 376
Buffer memory list . . . 376
Buffer memory details . . . 377
Appendix 9 Processing Time of Functions . . . 380
C Controller module dedicated functions . . . 380
Appendix 10 Parameter List . . . 381
System parameters . . . 381
CPU parameters . . . 382
Module parameters . . . 383
Memory card parameters . . . 383
Appendix 11 VxWorks Component List . . . 384
Appendix 12 Communication Examples Using Serial Communication Module . . . 394
Appendix 13 How to Use MELSEC-Q Series Modules . . . 396
Screen change in CW Configurator . . . 396
Setting procedure . . . 399
Considerations . . . 400
Appendix 14 When the Parameters of Intelligent Function Modules are not Enabled . . . 402
Corresponding device list . . . 402
Appendix 15 Added and Changed Functions . . . 404

INDEX 407

REVISIONS . . . 409
CONDITIONS OF USE FOR THE PRODUCT . . . 410
WARRANTY . . . 411
INFORMATION AND SERVICES . . . 412
TRADEMARKS . . . 412
COPYRIGHTS . . . 412

RELEVANT MANUALS

The following manuals are relevant to this product.

Manual name [manual number]	Description
MELSEC iQ-R C Controller Module User's Manual [SH-082558ENG] (this manual)	Procedures before operation, specifications, devices, memory, functions, parameters, and troubleshooting of a C Controller module
MELSEC iQ-R Module Configuration Manual [SH-081262ENG]	The combination of the MELSEC iQ-R series modules, common information on the installation/wiring in the system, and specifications of the power supply module, base unit, SD memory card, and battery
MELSEC iQ-R C Controller Module Programming Manual [SH-081371ENG]	Programming specifications and dedicated function library of a C Controller module
MELSEC iQ-R C Controller Module/C Intelligent Function Module Programming Manual (Data Analysis) [SH-081756ENG]	Programming specifications and dedicated function libraries for analyzing the data of a C Controller module and a C intelligent function module
CW Workbench/CW-Sim Operating Manual [SH-081373ENG]	System configuration, specifications, functions, and troubleshooting of CW Workbench/CW-Sim
CW Configurator Operating Manual [SH-081382ENG]	System configuration, parameter settings, and operation methods for the online function of CW Configurator

This manual does not include detailed information on the following:

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

For details, refer to the following:

MELSEC iQ-R Module Configuration Manual

It should be noted that the terms in the left column shall be replaced with the one in the right column.

Term	Replaced term
Programmable controller	C Controller module
Programmable controller system	C Controller system

TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
Data analysis function	A dedicated function library offered by a C Controller module and a C intelligent function module. It is used for data analysis processing.
Intelligent function module	A module that has functions other than an input or output, such as an A/D converter module and D/A converter module
MELSEC data link function	A dedicated function library offered by a C Controller module. It is used to access another CPU module as a connection target via network or in a multiple CPU system.
SLMP	A SeamLess Message Protocol. This protocol is used to access an SLMP-compatible device or a programmable controller connected to an SLMP-compatible device from an external device.

GENERIC TERMS AND ABBREVIATIONS

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

Generic term/abbreviation	Description
CC-Link IE TSN master/local module	A generic term for RJ71GN11-T2 and RJ71GN11-SX
CC-Link IE TSN module	A generic term for the following modules: • RJ71GN11-T2 CC-Link IE TSN master/local module • RJ71GN11-SX CC-Link IE TSN master/local module • Motion module
CC-Link IE Controller Network module	A generic term for the following modules: • RJ71GP21-SX CC-Link IE Controller Network modules • RJ71GP21S-SX CC-Link IE Controller Network modules • RJ71EN71 (when a CC-Link IE Controller Network function is used)
CC-Link IE Field Network module	A generic term for the following modules: • RJ71GF11-T2 CC-Link IE Field Network master/local modules • RJ71EN71 (when a CC-Link IE Field Network function is used) • Simple motion modules
CC-Link IE module	A generic term for CC-Link IE Controller Network modules, CC-Link IE Field Network modules, and CC-Link IE TSN modules
CPU module	A generic term for MELSEC iQ-R series CPU modules
CW Configurator	A generic product name for SWnDND-RCCPU ('n' indicates its version.)
CW Workbench	An abbreviation for a C Controller module and C intelligent function module engineering tool, CW Workbench
CW-Sim	An abbreviation for VxWorks simulator that can operate and debug C Controller module programs and C intelligent function module programs on a personal computer with CW Workbench installed, without connecting to an actual device (target)
CW-Sim Standalone	An abbreviation for VxWorks simulator that can operate C Controller module programs and C intelligent function module programs even on a personal computer without CW Workbench installed
C intelligent function module	A generic term for MELSEC iQ-R series C intelligent function modules
C Controller module	A generic term for MELSEC iQ-R series C Controller modules
C Controller module dedicated function	A dedicated function library offered by a C Controller module. It is used to control a C Controller module.
GOT	An abbreviation for the Mitsubishi Graphic Operation Terminal
R12CCPU-V	An abbreviation for R12CCPU-V C Controller modules
RAS	Reliability, Availability, Serviceability. This term refers to the overall usability of automated equipment.
USB Mass Storage Class-compliant device	A USB device that is compliant with the standard for recognizing as a memory device (USB Mass Storage Class)
VxWorks	A product name for the real-time operating system manufactured by Wind River Systems, Inc.
Global label	A label which is valid for user programs of a C Controller module. Module labels (global labels) of a motion module can be used.
Network module	A generic term for the following modules: • CC-Link IE Controller Network modules • CC-Link IE Field Network modules • CC-Link IE TSN modules • MELSECNET/H network modules • CC-Link modules
Base unit	A generic term for main base units, extension base units, and RQ extension base units

Generic term/abbreviation	Description
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
Dedicated function library	A generic term for C Controller module dedicated functions, MELSEC data link functions, data analysis functions, statistical analysis functions, and motion module dedicated classes
Target device	A personal computer, GOT, or another CPU module to connect for data communication
Power supply module	A generic term for MELSEC iQ-R series power supply modules
Statistical analysis function	A dedicated function library offered by a C Controller module and a C intelligent function module. It is used for statistical analysis processing.
I/O module	A generic term for input modules, output modules, I/O combined modules, and interrupt modules

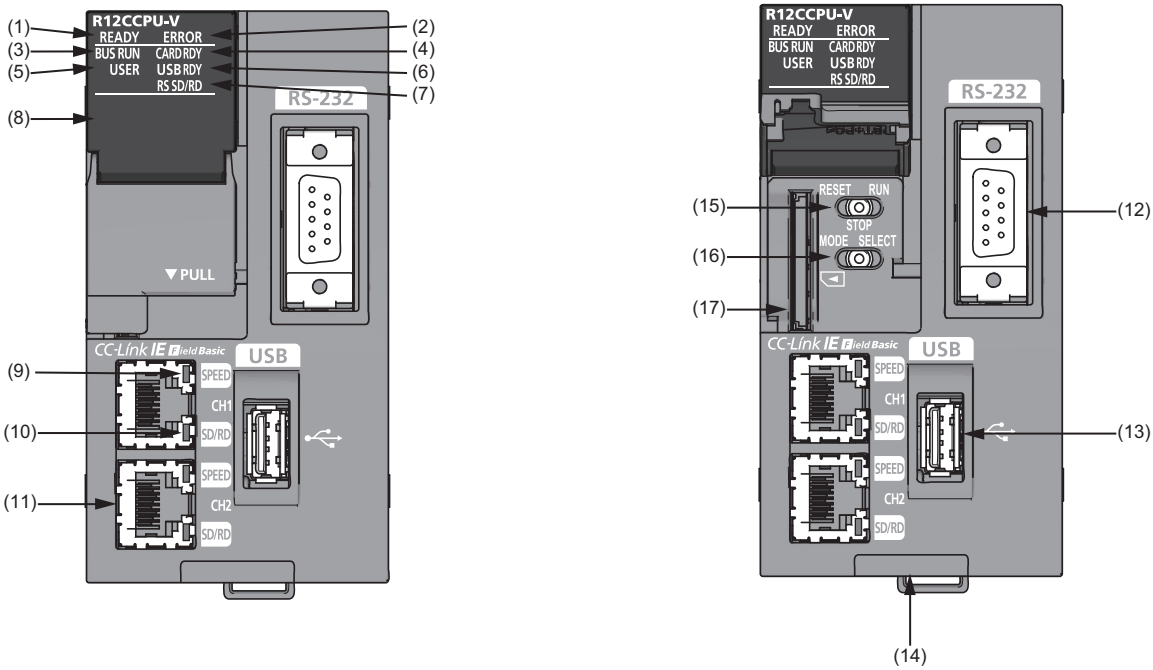
PART 1 PART NAMES

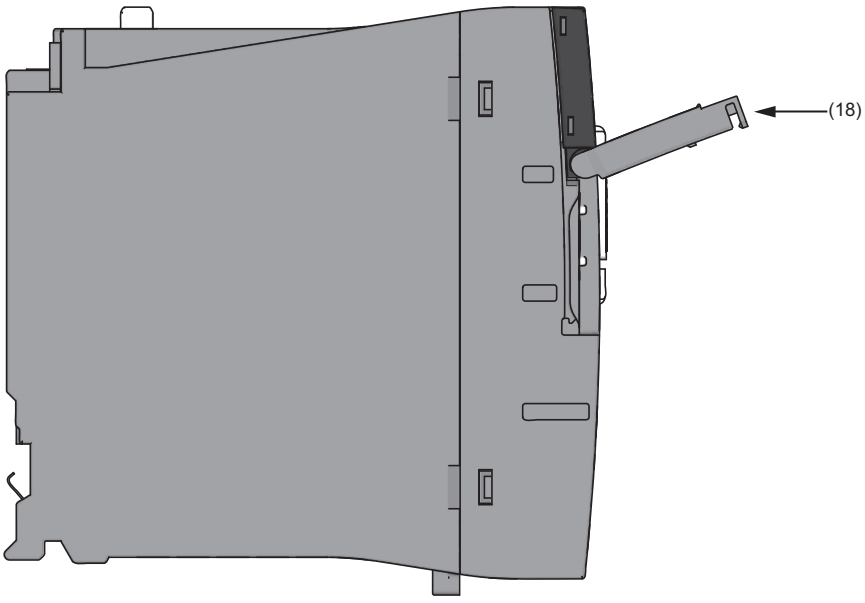
This part consists of the following chapter.

1 PART NAMES

1 PART NAMES

This chapter shows the part names of a C Controller module.





No.	Name	Description
(1)	READY LED	Indicates an operating mode. • ON: Normal operation mode • Flashing: Initializing (including the execution of the script file, "STARTUP.CMD") • OFF (for normal operation): Hardware failure or resetting • OFF (for diagnostics mode): Hardware diagnostics mode
(2)	ERROR LED	Indicates the error status. • ON: Continuation error or major error • Flashing (for normal operation): Stop error • Flashing (for diagnostics mode): Hardware diagnostics error or initialization error • OFF: Normal operation
(3)	BUS RUN LED	Indicates the operating status. • ON (for normal operation): RUN (state where output (Y) from a user program and writing to the buffer memory are permitted) • ON (for diagnostics mode): Hardware diagnostics mode • Flashing (low-speed): Performing the hardware diagnostics, configuring the module initialization setting, or performing initialization • Flashing (high-speed): Program/data memory shutdown complete • OFF : STOP/PAUSE (state where output (Y) from a user program and writing to the buffer memory are prohibited)
(4)	CARD RDY LED	Indicates the accessibility of an SD memory card. • ON: Accessible (mounted) • Flashing: Being mounted/unmounted • OFF: Inaccessible (not inserted/unmounted)
(5)	USER LED	The indication can be controlled with a user program.
(6)	USB RDY LED	Indicates the availability of a connected USB Mass Storage Class-compliant device. • ON: Accessible (mounted) • Flashing: Being mounted/unmounted • OFF: Inaccessible (not inserted/unmounted)
(7)	RS SD/RD LED	Indicates the data sending/receiving status for an RS-232 interface. • ON: Being sent/received • OFF: Not being sent/received
(8)	Dot matrix LED	Indicates the setting content or results for the operation selection mode or hardware diagnostics mode. The indication can be controlled with a user program during normal operation.
(9)	SPEED LED	Indicates the communication speed and the link status for Ethernet. • ON (orange): Linking-up (1 Gbps) • ON (green): Linking-up (100 Mbps) • OFF: Linking-down or linking-up (10 Mbps)
(10)	SD/RD LED	Indicates the data sending/receiving status in Ethernet. • ON: Being sent/received • OFF: Not being sent/received
(11)	Ethernet port	A port for connecting to an Ethernet device
(12)	RS-232 connector	A connector for connecting with an RS-232 supported device
(13)	USB connector	A connector for connecting with a USB-supported device
(14)	Product information marking	Displays the product information (16 digits) of a module.
(15)	RESET/STOP/RUN switch[*1]	A switch to control the hardware operating status. • RUN: Changes the operating status of the module to RUN. (State where output (Y) from a user program and writing to the buffer memory are permitted) • STOP: Changes the operating status of the module to STOP. (State where output (Y) from a user program and writing to the buffer memory are prohibited) • RESET: Resets the module. The switch operation for each operating status is described in the following sections. For initialization ☞ Page 28 Initialization For hardware diagnostics ☞ Page 30 Hardware Diagnostics For operation selection mode ☞ Page 37 Switch Operation
(16)	MODE/SELECT switch[*1]	A switch to select a hardware mode. The switch operation for each operating status is described in the following sections. For initialization ☞ Page 28 Initialization For hardware diagnostics ☞ Page 30 Hardware Diagnostics For operation selection mode ☞ Page 37 Switch Operation

No.	Name	Description
(17)	SD memory card slot	A slot for inserting an SD memory card (SD, SDHC standards-compliant: 2 GB (SD) to 16 GB (SDHC))
(18)	Slot cover	A cover of the SD memory card slot and the switches. Open this cover to insert/remove an SD memory card or to operate the switches. Close the cover unless inserting/removing an SD memory card or operating the switches to prevent foreign material intrusion such as dust.

*1 Operate the switches by a fingertip. Using tools such as a screwdriver may cause damage to the switches.

PART 2 PROCEDURE BEFORE OPERATION

This part consists of the following chapter.

2 STARTUP PROCEDURE

2 STARTUP PROCEDURE

This chapter shows the startup procedure and program execution of a C Controller module.
At the time of the first operation of a C Controller module, check that there is no error on the module using the hardware diagnostic, then start the system.

Performing hardware diagnostics

1. Mounting a C Controller module

Mount a power supply module and a C Controller module on a base unit. (MELSEC iQ-R Module Configuration Manual)

2. Initializing the C Controller module

Initialize the C Controller module. (☞ Page 28 Initialization)

3. Performing hardware diagnostics

Check the hardware status of the C Controller module. (☞ Page 30 Hardware Diagnostics)

Starting the C Controller system

1. Inserting an SD memory card

Insert an SD memory card to the C Controller module as necessary. (☞ Page 35 Inserting and Removing an SD Memory Card)

2. Mounting modules and wiring for each device

Mount modules on the base unit, and wire for each device. (MELSEC iQ-R Module Configuration Manual)
For wiring to a C Controller module, refer to the following:
☞ Page 64 WIRING SPECIFICATIONS

3. Turning the power of the system ON

Check the following items, and then turn ON the power supply.

- The wiring and supply voltage of the power supply are in the required range.
- The C Controller module is placed into the STOP state.

4. Creating a project

Create a project for the C Controller module to be used with a personal computer on which CW Configurator has been installed. (☞ Page 40 Creating a Project)

5. Connecting a personal computer to the C Controller module

Connect the personal computer on which CW Configurator has been installed to the C Controller module. (☞ Page 40 Connecting to a Personal Computer)

6. Setting parameters

Set the system parameter, CPU parameter, and module parameter. (☞ Page 41 Setting Parameters)
When using an SD memory card function, or when mounting an intelligent function module, also set parameters other than the above. (User's manual for each module)

7. Writing parameters to the C Controller module

Write the parameters set with CW Configurator to the C Controller module. (☞ Page 43 Writing Parameters to a C Controller Module)

8. Resetting the C Controller system

Reset the system by one of the following methods.

- Turning the power of the system OFF and ON
- Resetting the C Controller module (☞ Page 37 Switch Operation)

9. Checking errors

Check the READY LED and ERROR LED on the C Controller module. When any error occurred, perform the troubleshooting. When any errors caused by factors other than the C Controller module occurred, refer to the manual for each module.

10. Creating a user program

Create a user program

- Create a user program and debug it. (☞ Page 44 Creating a User Program)
- Create a script file. (☞ Page 49 Creating a Script File)
- Register the user program and the script file to the C Controller module. (☞ Page 51 Registering a User Program)

11. Resetting the C Controller system

Reset the system by one of the following methods.

- Turning the power of the system OFF and ON
- Resetting the C Controller module (☞ Page 37 Switch Operation)

12. Executing the program

Change the operating status of the C Controller module to RUN, and check that the BUS RUN LED turns ON. (☞ Page 37 Switch Operation)

2.1 Initialization

Initialize a C Controller module in the following cases.

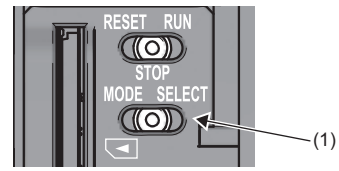
- First operation
- The C Controller module does not start by the execution of the script file registered in the program memory
- The user name/password set to the C Controller module has been forgotten

Point

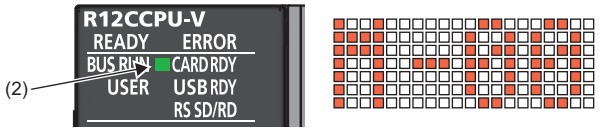
The data in the program memory, device/label memory, and data memory are deleted during the process of initialization. Back up all necessary data in advance.

Procedure for initialization

Check that the RESET/STOP/RUN switch is positioned at the center i.e. on the "Stop" position at first.



1. Put and hold the MODE/SELECT switch (1) on the MODE position.

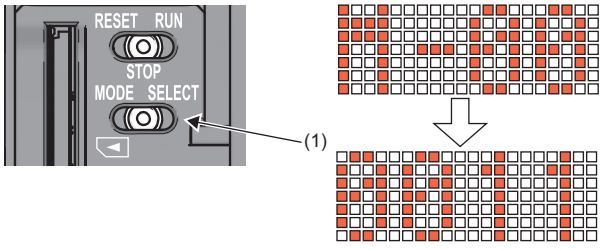


2. Turn the power OFF and ON, or reset the C Controller module. (☞ Page 37 Switch Operation)
The BUS RUN LED (2) turns ON, and "M-00" is displayed on the dot matrix LED.

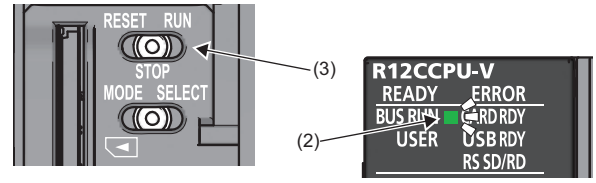
Point

When the power of the C Controller module is ON, the ERROR LED may flash and an error code may be displayed on the dot matrix LED.
(☞ Page 29 Abnormal completion of initialization process)

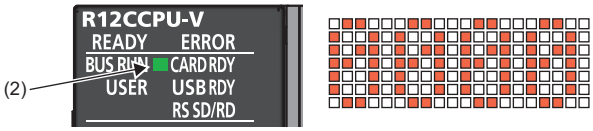
3. Release the MODE/SELECT switch (1) and put it back to the center position.



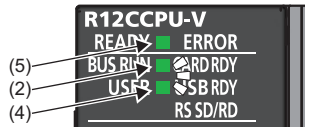
4. Set the MODE/SELECT switch (1) to the SELECT position.
Every time the switch is set to the SELECT position, the value of mode displayed on the dot matrix LED is changed.
Repeat this switch movement until "0011" is displayed on the dot matrix LED.

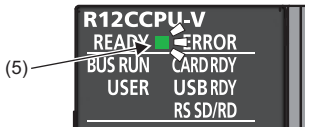


5. Set the RESET/STOP/RUN switch (3) to the RUN position.
The selected mode is executed. The BUS RUN LED (2) will be flashing during initialization.



6. Check that the BUS RUN LED (2) turns ON and "0000" is displayed on the dot matrix LED, then reset the C Controller module.
(☞ Page 37 RESET/STOP/RUN switch)





7. The initialization is performed by resetting the module. The READY LED (5) will turn ON, and the BUS RUN LED (2) and USER LED (4) will be flashing during the initialization.

8. Upon normal completion of the initialization, the BUS RUN LED (2) and USER LED (4) turn OFF, and the READY LED (5) starts flashing.

9. Reset the C Controller module.
(☞ Page 37 RESET/STOP/RUN switch)

■Mode selection

Mode	Dot matrix LED display	Description
10	0010	Default IP setting
11	0011	Module initialization setting

Point

Do not reset the C Controller module during initialization.
Perform the initialization again if the module has been reset in error.

Abnormal completion of initialization process

Upon abnormal completion of the initialization, the ERROR LED will be flashing, and the READY LED and USER LED turn ON. In the case of abnormal completion, perform the initialization again.

■The ERROR LED flashes, and the value other than '0000' is displayed on the dot matrix LED

The ERROR LED flashes, and the value corresponding to the error is displayed.

Dot matrix LED	Error name	Cause	Corrective action
E100	Initialization function execution error	The execution of the initialization function failed.	Execute the initialization again.
E101	Firmware update function execution error	For details on the firmware update function, refer to the following: MELSEC iQ-R Module Configuration Manual	
E102			

For an abnormal completion, check the error corresponding to the value displayed on the dot matrix LED, and take the appropriate corrective action. If the error occurs again, the C Controller module may have a hardware failure. Please contact your local Mitsubishi Electric sales office or representative.

Module status after initialization

After initialization, the status of the C Controller module will be as follows:

■Default IP setting

- The registered script file stops to be executed.*1
- The IP address for the C Controller module is set to the default. (☞ Page 64 Ethernet Port)

*1 The script file name is renamed as "STARTUP.BAK," and deregistered.

■Module initialization setting

- Default parameters are set to the data memory.
- The program memory, device/label memory, and data memory are formatted.
- A security password is initialized. (Default password: password)
- Values set with the following C Controller module dedicated functions are initialized.

Function name	Description
CCPU_SysClkRateSet	System clock rate (Default: 60 Hz)
CCPU_SetSysTimerMode	System timer mode (Default: Normal mode)

2.2 Hardware Diagnostics

Diagnose the hardware of a C Controller module.

Hardware diagnostics timing

Use the hardware diagnostics in the following cases.

- First operation
- Troubleshooting

Point

Do not turn the power OFF, or reset the C Controller module during the hardware diagnostics. Doing so may cause abnormal startup of the C Controller module. If it does not start normally, perform the initialization.

Diagnostics types

The modes of hardware diagnostics are shown below.

Mode	Dot matrix LED	Diagnostic item	Description
0	M-00	Diagnostic test for Mode 1 to Mode 6	To perform diagnostic test in the order from Mode 1 to Mode 6.
1	M-01	Program memory, data memory diagnostic test	To read data in the program memory and data memory, and perform error detection.
		Device/label memory diagnostic test	To write/read test data to/from the device/label memory, and check by verifying the data.
2	M-02	Ethernet diagnostic test	To diagnose the Ethernet port (CH1 and CH2) status.
3	M-03	SD memory card interface diagnostic test	To diagnose the SD memory card slot status.
4	M-04	RS-232 diagnostic test	To perform a self-loopback test for an RS-232 connector. The wiring for self-loopback is required.
5	M-05	USB diagnostic test	To diagnose the USB connector status.
6	M-06	Bus diagnostic test	To write/read/verify the memory of the internal bus and register.
7	M-07	Dot matrix LED test	To show the dot matrix LED test display.

Performing diagnostics

The following shows the procedure for hardware diagnostics.

Preparation

Perform the following before the hardware diagnostics.

1. Mount the modules.

Mount a power supply module and a C Controller module on a base unit.

Point

The hardware diagnostics can be performed even when a C Controller module is not mounted on the CPU slot.

To perform the hardware diagnostics with the module mounted on the slot other than the CPU slot, mount another CPU module on the CPU slot. Although an error may be detected on the mounted CPU module, the hardware diagnostics can be performed.

2. Check the wiring.

- Check that the power supply cable is wired properly.
- Do not connect any cable other than power supply cable.

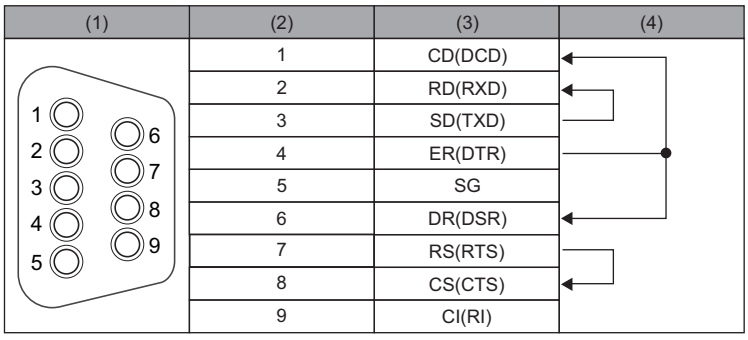
3. Make the necessary preparations for each diagnostics.

4. Power ON.

- Check that the power supply voltage is within the range of the specifications.
- Check that the RESET/STOP/RUN switch is put on the STOP position.

Preparation required for each diagnostics

Perform the following before starting each mode (0 to 7) of hardware diagnostics.

Mode	Description
0	Make all of the preparations required for Mode 1 to 6.
1	Back up the data in the program memory, data memory, and device/label memory.
2	Check that no cable is connected to the Ethernet port.
3	Check that no SD memory card is inserted.
4	Connect a cable to the RS-232 connector. The connector pin arrangement and cable connection are shown below.  (1) Connector (2) Pin number (3) Abbreviation of signal code (4) Cable connection
5	Check that no cable is connected to the USB connector.
6	No preparation is required for this mode.

Mode selection

The following shows how to select a mode.

1. Set the RESET/STOP/RUN switch to the RESET position, and keep it up to the step 4.
2. Check that all LEDs turned OFF.
3. Set the MODE/SELECT switch to the MODE position, and keep it up to the step 6.
4. Release the RESET/STOP/RUN switch and put it back to the STOP position.
5. The BUS RUN LED turns ON, and "M-00" is displayed on the dot matrix LED.
6. Release the MODE/SELECT switch and put it back to the center position.
7. Set the MODE/SELECT switch to the SELECT position, and select the desired diagnostics mode.

Every time the switch is set to the SELECT position, the value of mode displayed on the dot matrix LED is changed.
Repeat this switch movement until the dot matrix LED displays the desired diagnostics mode.

Mode execution

The following shows how to execute the selected mode.

■Executing Mode 0 to 6

1. Set the RESET/STOP/RUN switch to the RUN position.

- While Mode 0 or 1 is being executed, the BUS RUN LED will be flashing, and the current mode and its progress are displayed alternatively on the dot matrix LED.

(Example): "M-01" (diagnostics mode) ← displaying alternatively → "0050" (progress: %)

- While Mode 2 to 6 is being executed, the BUS RUN LED will be flashing, and the current mode is displayed on the dot matrix LED.

2. Check that the BUS RUN LED turns ON.

When the test is completed normally, "0000" is displayed on the dot matrix LED.

3. Turn the power OFF.

Point

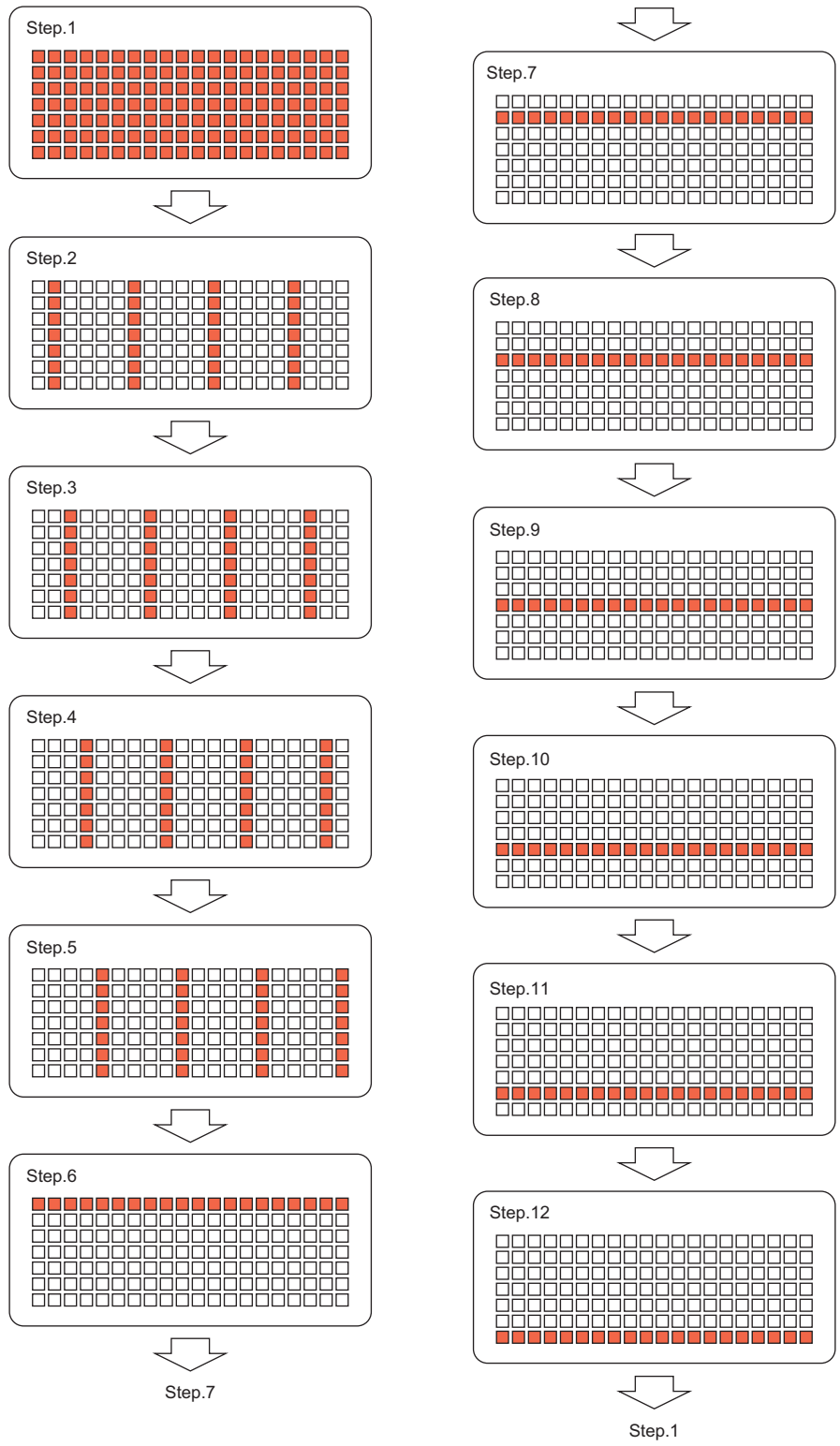
Before performing the steps of mode execution, put the RESET/STOP/RUN switch back to the STOP position, if the switch is in the RUN position.

■Executing Mode 7

1. Set the RESET/STOP/RUN switch to the RUN position.

2. Set the MODE/SELECT switch to the SELECT position.

Every time the switch is set to the SELECT position, the lighting status on the dot matrix LED switches.



■: ON
□: OFF

3. Turn the power OFF.

Point

Check visually that there is no inactive dot on the dot matrix LED.
If there is any inactive LED, the C Controller module may have a hardware failure. Please contact your local Mitsubishi Electric sales office or representative.

Operation at error detection

The error contents displayed at the time of error detection are shown below.

Dot matrix LED displays other than "0000"

The ERROR LED starts flashing and the value corresponding to the diagnostics on which an error occurred will be displayed at the time of error detection during diagnostics or setting. Only flashing of the ERROR LED without dot matrix LED display means a system error.

Mode	Dot matrix LED	Diagnostics on which an error occurred
1	E010	Program memory, data memory diagnostic test Device/label memory diagnostic test
	E020	
	E030	
	E040	
2	E050	Ethernet diagnostic test (CH1)
	E060	Ethernet diagnostic test (CH2)
3	E070	SD memory card interface diagnostic test
4	E080	RS-232 diagnostic test*1
5	E090	USB diagnostic test
6	E0A0	Bus diagnostic test
	E0B0	
	E0C0	
	E0D0	
	E0E0	

*1 An error may occur if a wiring cable is not connected properly. Connect the cable or wire the cable correctly again, and perform the diagnostics for Mode 4.

Point

The diagnostics or setting will be terminated at the time of abnormal completion. Retry the diagnostics or setting. If the error occurs again after the abnormal completion, the C Controller module may have a hardware failure. Please contact your local Mitsubishi Electric sales office or representative.

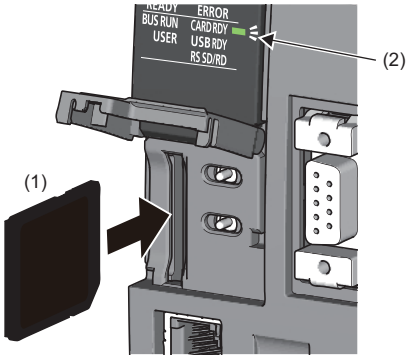
2.3 Inserting and Removing an SD Memory Card

Insert or remove an SD memory card by following the procedure below. Failure to do so may cause data corruption in the SD memory card.

Point

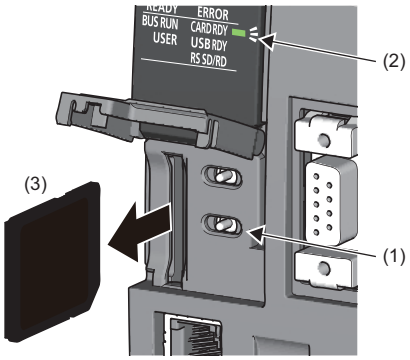
Change the MODE/SELECT switch operation to "SD UNMOUNT" or "SD/USB UNMOUNT" before removing the SD memory card. (☞ Page 37 Switch Operation)

Insertion procedure



1. Insert an SD memory card (1) straight into the SD memory card slot with its cutout pointed down. Make sure it is not uplifted after inserting it.
2. The CARD RDY LED (2) keeps flashing until the SD memory card is ready to be used.
3. Once the CARD RDY LED (2) turns ON, the card can be used.

Removal procedure



1. Check that no SD memory card is accessed.
2. Hold the MODE/SELECT switch (1) in the SELECT position to unmount the SD memory card. The CARD RDY LED (2) is flashing, which indicates that process of unmounting SD memory card is in progress, and later upon successful completion, the LED turns OFF.
3. Push the SD memory card (3) in once, and pull it out straight.

Point

An SD memory card can also be unmounted by using a user program with the C Controller module dedicated function (CCPU_UnmountMemoryCard).

Precautions

Do not perform the following operations while accessing the files in an SD memory card. Doing so may result in data corruption in the SD memory card or file system error.

- Turning the power of the system OFF
- Turning the power OFF and ON, or resetting the C Controller module
- Unmounting or removing the SD memory card

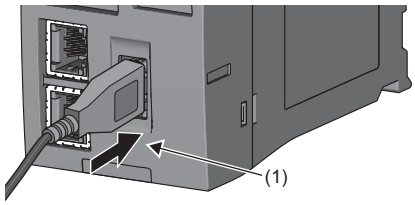
2.4 Connecting and Disconnecting a USB Mass Storage Class-compliant Device

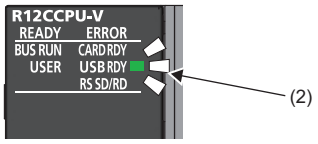
Connect or disconnect a USB Mass Storage Class-compliant device by following the procedure below. Failure to do so may cause data corruption in the USB Mass Storage Class-compliant device.

Point

Change the MODE/SELECT switch operation to "USB UNMOUNT" or "SD/USB UNMOUNT" before disconnecting the USB Mass Storage Class-compliant device. (☞ Page 37 Switch Operation)

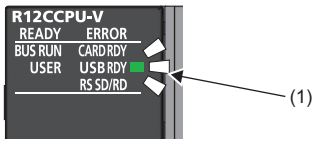
Connection procedure

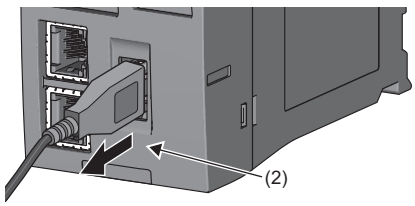




1. Connect a connector of USB Mass Storage Class-compliant device or USB cable to the USB connector (1). Firmly insert the connector to prevent it from falling out.
2. The USB RDY LED (2) keeps flashing until the USB Mass Storage Class-compliant device is ready to be used.
3. Once the USB RDY LED (2) turns ON, the USB Mass Storage Class-compliant device can be used.

Disconnection procedure





1. Check that no USB Mass Storage Class-compliant device is accessed.
2. Hold the MODE/SELECT switch in the SELECT position to unmount the USB Mass Storage Class-compliant device. The USB RDY LED (1) is flashing, which indicates that process of unmounting USB Mass Storage Class-compliant device is in progress, and later upon successful completion, the LED turns OFF.
3. Disconnect the USB Mass Storage Class-compliant device or USB cable that is inserted to the USB connector (2).

Point

A USB Mass Storage Class-compliant device can also be unmounted by using a user program with the C Controller module dedicated function (CCPU_UnmountMemoryCard).

Precautions

Do not perform the following operations while accessing the files in a USB Mass Storage Class-compliant device. Doing so may result in data corruption in the USB Mass Storage Class-compliant device or file system error.

- Turning the power of the system OFF
- Turning the power OFF and ON, or resetting the C Controller module
- Unmounting/ejecting the USB Mass Storage Class-compliant device

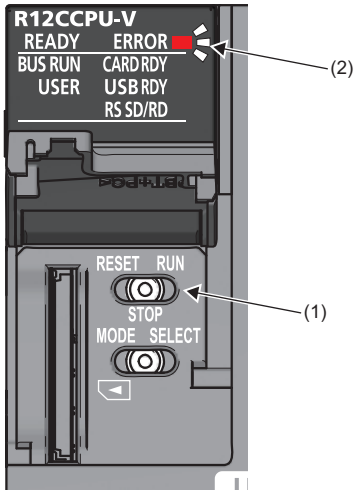
2.5 Switch Operation

This section explains the method for changing the operating status of a C Controller module.

RESET/STOP/RUN switch

The operating status of a C Controller module can be changed.

- By setting the switch to the RUN position, the operating status is changed to RUN.
  (Status where output (Y) from a user program and writing to the buffer memory are permitted)
- By setting the switch to the STOP position, the operating status is changed to STOP.
  (Status where output (Y) from a user program and writing to the buffer memory are prohibited)
- Operating the switch with the following procedure resets a C Controller module.



1. Hold the RESET/STOP/RUN switch (1) in the RESET position.
2. Check that all LEDs turn OFF after the ERROR LED (2) flashes several times.
3. Release the RESET/STOP/RUN switch (1) and put it back to the STOP position.

Point

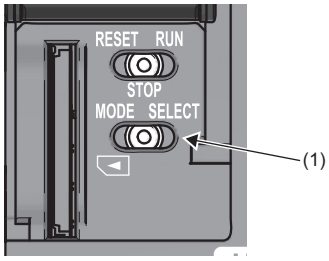
Operate the RESET/STOP/RUN switch by fingertip.
Using tools such as a screwdriver may cause damage on the switch.

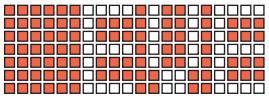
MODE/SELECT switch

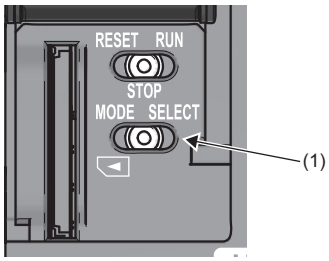
The mode can be switched between the operation selection mode and normal operation mode, and the operation selected in the operation selection mode can be performed.
The operation of 'notification/unmount' and 'information display on the dot matrix LED' can be selected in the operation selection mode. Make sure that the C Controller module is in normal operation before selecting its operation.

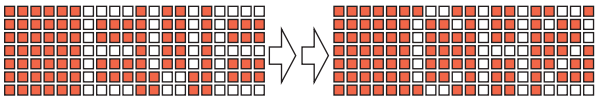
Selecting and performing an operation

The following shows how to select and perform the operation.









1. Hold the MODE/SELECT switch (1) in the MODE position.
2. The selectable operation is displayed on the dot matrix LED. (In the operation selection mode, the lighting status of LED display is reversed.)
3. Release the MODE/SELECT switch (1) and put it back to the center position.
4. Set the MODE/SELECT switch (1) to the SELECT position. Every time the switch is set to the SELECT position, the value of mode displayed on the dot matrix LED is changed. Repeat this switch movement until the desired operation is displayed on the dot matrix LED.
5. Hold the MODE/SELECT switch (1) in the SELECT position.
6. The selected operation is performed.

Point

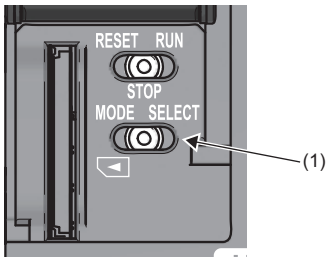
The mode is switched to the normal operation mode by holding the MODE/SELECT switch in the MODE position while selecting an operation.

LED display[*1]	Operation	Operation at execution
EVENT	An event is notified to the user program. (Default)	Notification/unmount[*2]
SD UNMOUNT	An SD memory card is unmounted forcibly.	
USB UNMOUNT	A USB Mass Storage Class-compliant device is unmounted forcibly.	
SD/USB UNMOUNT	An SD memory card and USB Mass Storage Class-compliant device are unmounted forcibly.	
USER[*3]	Contents specified by the user are displayed on the dot matrix LED. (Default)	Information display on the dot matrix LED
ERROR[*4]	An error code is displayed on the dot matrix LED.	
CH1[*5]	The IP address of Ethernet port (CH1) is displayed on the dot matrix LED.	
CH2[*5]	The IP address of Ethernet port (CH2) is displayed on the dot matrix LED.	

*1 The LED display scrolls horizontally (right to left) to display the portion not appeared on dot matrix LED.
*2 The selected operation is registered in the behavior caused when holding the switch in the SELECT position, and can be performed even in the normal operation mode.
*3 The character string output with the C Controller module dedicated function (CCPU_SetDotMatrixLED) is displayed. In addition, when this operation is selected again after switched to other operation, the previously shown character string will be displayed.
*4 The latest error is displayed. ("—" is displayed when no error occurred.)
*5 When an IP address is not set, "—.—.—.—" is displayed.

■Performing the 'notification/unmount' in the normal operation mode

The last selected operation (out of 'notification/unmount') in the operation selection mode is registered as the operation performable by holding the switch in the SELECT position. By holding the MODE/SELECT switch in the SELECT position, the registered operation can be performed even in the normal operation mode.



1. Hold the MODE/SELECT switch (1) in the SELECT position.
2. The operation selected in the operation selection mode is performed.

Point

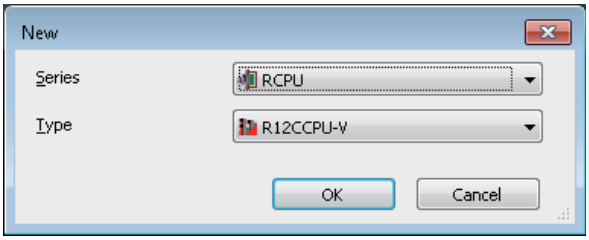
- By setting the MODE/SELECT switch to the SELECT position once during the normal operation, the operation that is currently selected is displayed on the dot matrix LED display.
- Even while the operation of the 'information display on the dot matrix LED' is being performed, the operation of the 'notification/unmount' can be performed.

2.6 Creating a Project

Create a project by starting CW Configurator.

[Project] ⇨ [New]

Creation procedure

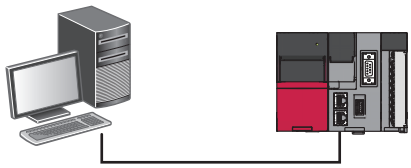


1. Select the C Controller module to be used in the "New" screen, and click the [OK] button.

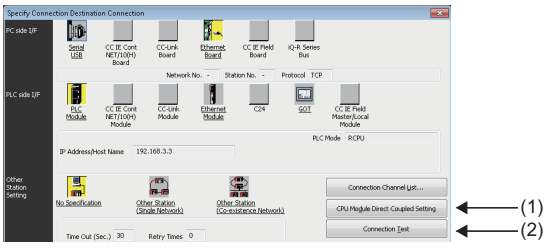
2.7 Connecting to a Personal Computer

This section shows the procedure to have direct connection between a personal computer and a C Controller module.

Connection procedure



1. Connect a personal computer with a C Controller module using an Ethernet cable. (☞ Page 64 Ethernet Port)



2. Select [Online] ⇨ [Current Connection Destination] from the menu of CW Configurator.
3. In the "Specify Connection Destination Connection" screen, click the [CPU Module Direct Coupled Setting] button (1).
4. Select the connection method to the C Controller module, and click the [Yes] button.
5. Click the [Communication Test] button (2) in the "Specify Connection Destination Connection" screen to check the connection to the C Controller module is available.

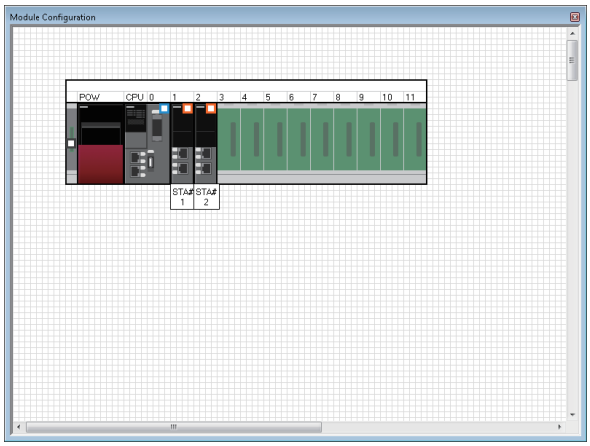
2.8 Setting Parameters

Set parameters for the system and modules.

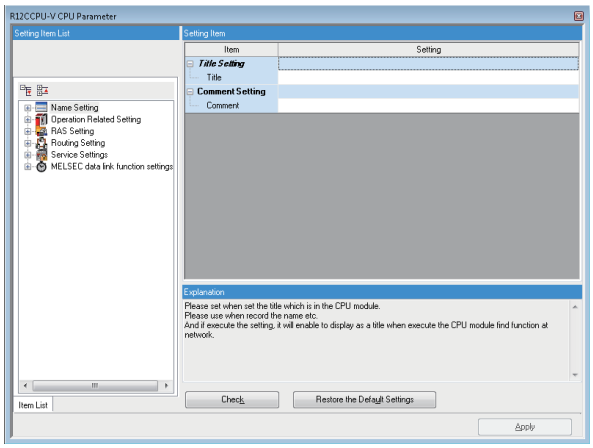
CW Configurator is connected to the actual system configuration

Read the actual system configuration to the module configuration diagram of CW Configurator to set parameters.

[Navigation window] ⇨ [Module Configuration]



1. Open the module configuration diagram, and select [Online] ⇨ [Read Module Configuration from PLC].
2. The system parameter has been set automatically, and the actual system configuration is displayed on the module configuration diagram.



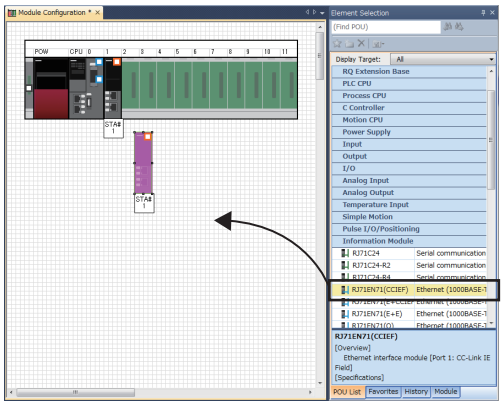
3. Double-click the CPU module, I/O module, and intelligent function module shown to display the parameter editor for the respective modules.
4. After setting the parameters, click the [Apply] button.

Precautions

An RQ extension base unit, MELSEC-Q series extension base unit, or modules mounted on these extension base units cannot be read. For the units and modules unable to be read, create the module configuration by dragging each unit/module from the element selection window and dropping it on the module configuration diagram.

CW Configurator is not connected to the actual system configuration

Create the module configuration manually to set parameters.



1. Select a base unit to be used from the element selection window, and drag and drop it onto the module configuration diagram.
2. Drag respective modules to be used and drop it on the base unit placed in the diagram.
3. Select [Edit] ⇨ [Parameter] ⇨ [Fix] from the menu.
4. Double-click the module to display the parameter editor for the respective modules.
5. After setting the parameters, click the [Apply] button.

Parameter settings from the navigation window

Set the following parameters from the navigation window.

■System parameter

To change the number of slots of a base unit or the number of occupied points for a module as well as to configure the multiple CPU setting or synchronization setting, the system parameter is set from the navigation window.

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

■Module parameter of a CPU module

To use a built-in Ethernet function for CPU module, setting the module parameter for CPU module is required.

[Navigation window] ⇨ [Parameter] ⇨ [(CPU module)] ⇨ [Module Parameter]

■Memory card parameter

To use any of the functions with SD memory card, setting the memory card parameter is required.

[Navigation window] ⇨ [Parameter] ⇨ [(CPU module)] ⇨ [Memory Card Parameter]

■Multiple module parameters and module extended parameter

Some intelligent function modules may require parameter settings for multiple modules or module extended parameter setting.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [(each of Intelligent function module)] ⇨ [Module Extended Parameter] or [Module Parameter]

Point

Settings with the module configuration diagram and settings by system parameter are used properly as follows depending on their usage.

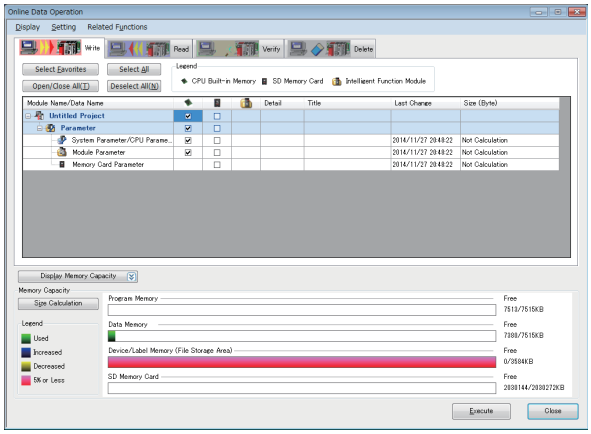
- Module configuration diagram: Module-specific information such as the number of occupied points is used without changing.
- System parameter: The number of slots of a base unit or the number of occupied points for a module is changed.

2.9 Writing Parameters to a C Controller Module

Write parameters to a C Controller module.

[Online] ⇨ [Write to PLC]

Operating procedure



1. Select "System Parameter/CPU Parameter" and "Module Parameter."
2. Click the [Execute] button.
3. After writing to a C Controller module is completed, click the [Close] button.

Point

The following parameters must be written: system parameters and CPU parameters to operate a C Controller module, and module parameters and module extended parameters to use an I/O module or an intelligent function module.

2.10 Creating a User Program

Create a user program with VxWorks standard API functions and functions offered by a C Controller module in accordance with the specification of VxWorks.

For programming method of VxWorks standard API functions, refer to the manual for VxWorks of the following version.

- VxWorks Version 6.9

For details on the dedicated function library of a C Controller module, refer to the following:

- 📖MELSEC iQ-R C Controller Module Programming Manual
- 📖MELSEC iQ-R C Controller Module/C Intelligent Function Module Programming Manual (Data Analysis)

Point

For details on sample programs, please contact your local Mitsubishi Electric sales office or representative.

Programming procedure

The following shows the procedure for creating a user program.

For details on project creation and programming method, refer to the following:

📖CW Workbench/CW-Sim Operating Manual

1. Install CW Workbench on a personal computer.
2. Start CW Workbench, and check that VxWorks image file version of the personal computer is the same as that of the C Controller module. (☞ Page 48 Checking VxWorks Image File)
3. Create a project of the user program.
4. Configure the setting to use the dedicated function library.
5. Program for a user program.
6. Debug the user program.
7. Create a script file to start the user program. (☞ Page 49 Creating a Script File)
8. Register the created user program and the script file to the C Controller module. (☞ Page 51 Registering a User Program)

Considerations for creating a user program

The following shows the considerations for creating a user program.

■Endian format (memory layout)

The endian format (memory layout) of a C Controller module is little endian. Compile a user program in the little endian format.

■VxWorks real-time process (RTP)

A C Controller module does not support applications running in VxWorks real-time process. Create a user program as an application based on the VxWorks kernel.

■Build property

Enter [-mlong-calls] to "Tool Flags," and then enter [-fsigned-char] to "Debug mode" and "Non Debugmode" to compile a user program.

[Project Explorer] ⇨ right-click an arbitrary project ⇨ [Properties] ⇨ [Build Properties] ⇨ [Tools] in CW Workbench

■User program execution

Execute a user program by starting a task from the script file. The system may malfunction if the user program is executed without the task being started.

■Startup of a task performing floating-point operations

Always specify the VX_FP_TASK option for the third argument of taskSpawn when activating the following tasks.

- A task performing floating-point operations
- A task calling a function that returns floating-point value
- A task calling a function that takes floating-point value as an argument
- A task calling a data analysis function
- A task calling a statistical analysis function

Activating the above task without the VX_FP_TASK option specified may cause the operating system runaway. For information on specifying the VX_FP_TASK option in a script file, refer to the following:

☞ Page 49 Creating a Script File

For details on the VX_FP_TASK option, refer to the manual for VxWorks.

■Task execution in STOP or PAUSE status

Even if the operating status of the C Controller module is STOP or PAUSE, the user program task does not stop. Use the C Controller module dedicated function (CCPU_GetCpuStatus) to split the user program processing according to the operating status of the C Controller module.

■Priority of execution tasks

Set the priority of a task to execute a user program as described below.

- When access is not made with FTP during user program execution, set the priority of the user program to 100 or more (100 to 255). Operating with the priority set to 0 to 99 may cause improper operation of the system.
- When access is made with FTP during user program execution, the actual FTP processing (task) of the C Controller module is performed at the priority of 200. Set the priority of the user program to 201 to 255. When setting the priority to 100 to 200, set a wait processing (such as taskDelay) in the user program to let the actual FTP processing operate.

■Communication with a target device via Ethernet

To communicate with a target device via Ethernet, check the port number being used in the C Controller module in advance using the VxWorks standard "netstat" command *1. Do not use any port number that has already been used. Normal communication may not be performed if used.

*1 Execute the "netstat" command using the Telnet tool.

■When no response is returned from a function

The response from a function may not be returned when an error occurs on a C Controller module. Design a user program take into account the possibility of no response from the function. The error occurrence status can be checked using the C Controller module dedicated function (CCPU_GetErrInfo).

■Memory fragmentation

Allocating and deallocating memory frequently may cause improper operation of the system due to memory fragmentation.

■Restrictions on MELSEC data link functions

A path for a MELSEC data link function does not support to be shared among multiple tasks. Make sure to manage a path opening/closing for each task.

■When changing system clock rate

- Use the C Controller module dedicated function (CCPU_SysClkRateSet) to change the system clock rate, and do not use the VxWorks standard API function (sysClkRateSet). Otherwise, the operation of VxWorks will be unstable.

■Settings of IP address

Set the IP address of a C Controller module using an engineering tool. If it is set with a user program, the network may be no longer functional properly.

■Access to USB Mass Storage Class-compliant device

To create a user program accessing a USB Mass Storage Class-compliant device, implement processing to check if accessing to the USB Mass Storage Class-compliant device is allowed and to retry. Although a USB Mass Storage Class-compliant device is automatically mounted when connected, it may take several minutes for the device to be accessible depending on its format type or capacity.

■Considerations for interrupt service routine

Fully understand the specifications on VxWorks, operating system of a C Controller module, before creating a routine which will be executed in an interrupt service routine. Setting an inappropriate value to an argument of functions or executing a function other than C intelligent function module dedicated functions for ISR may cause the VxWorks runaway.
In the dedicated function library (excluding data analysis functions and statistical analysis functions) offered by a C Controller module, C Controller module dedicated functions for ISR can be used. To execute other dedicated functions synchronously with an interrupt, implement the notification processing in a user program and perform the processing in a task.

■When using the spinlock, intCpuLock function, and taskCpuLock function of VxWorks

The following table shows task states in which a dedicated function library provided by a C Controller module is not available.
○: Available, ×: Not available

Task state	Dedicated function library	
	C Controller module dedicated function MELSEC data link function Motion module dedicated class	Data analysis function Statistical analysis function
Holding a spinlock	×	×
Prohibiting interrupts with the intCpuLock function		
Prohibiting switching of execution tasks with the taskCpuLock function		○

Do not use a dedicated function library in a task state in which the library is not available.
For details on the spinlock, the intCpuLock function, and the taskCpuLock function, refer to Help for CW Workbench or Wind River Workbench.

Considerations for CW Workbench connection

The following shows the considerations for CW Workbench connection.

■Error occurred during program download

When a user program is specified and downloaded to a target (C Controller module) with CW Workbench, the system failure/stop (such as user watchdog timer error) may occur on a C Controller module during the download if the program size is too large.

Take either of the following corrective actions in the case of a system failure/stop.

- Increase the value of the watchdog timer with CW Configurator
- Load the program and debug it in a C Controller module

■Interrupt processing delay

During remote debugging with CW Workbench, a C Controller module may be in the interrupt-disabled state. Any processing called from an interrupt routine (such as C Controller dedicated functions for ISR) is not executed during that time. Interrupts that are expected to occur at the fixed interval such as periodic timer interrupt may be delayed.

■Restarting a C Controller module

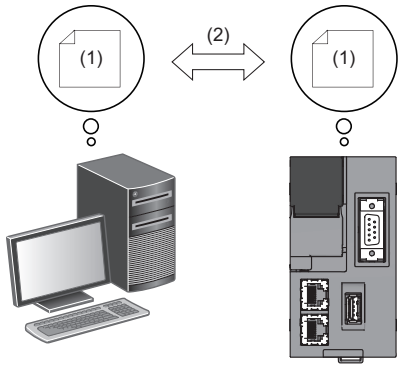
To restart a C Controller module, ensure that CW Workbench is disconnected. When the C Controller module is restarted with CW Workbench connected, CW Workbench automatically establishes the connection once the C Controller module has been started up. At that time, an error may occur since the system task which has been operating in the C Controller module is temporarily blocked.

■Message displayed on Host Shell

Message(s) issued by VxWorks during Host Shell connection may be displayed. For details on messages of VxWorks, refer to the manual for VxWorks or the help of CW Workbench.

2.11 Checking VxWorks Image File

Check that the VxWorks image file version in a personal computer and the one in a C Controller module are the same.
Specifying a different version of VxWorks image file prevents normal debugging.



(1) VxWorks image file
(2) Same version

Checking method

The checking methods for the VxWorks image file are shown below.

■C Controller module

Check the firmware version with the diagnostics function of CW Configurator or the rating plate on the side of the C Controller module.

■Personal computer

Check the file name of VxWorks image file specified with CW Workbench.

- R12CCPU-V_XX*1

*1 XX: Firmware version

Version of VxWorks image files are not identical

If the version of VxWorks image file is different, acquire the new image file stored in the C Controller module.

1. Connect the C Controller module with the personal computer. (☞ Page 64 WIRING SPECIFICATIONS)
2. Acquire the VxWorks image file *1 in the C Controller module using the FTP function.

*1 The file is stored in the system memory (/SYSTEMROM/OS_IMAGEFILE).

3. Reconfigure the target server setting with CW Workbench.

Point

The default settings of the FTP function are as follows:

- Login name: target
- Password: password

2.12 Creating a Script File

A script file, "STARTUP.CMD" is a file to describe the loading destination of a user program which starts at the time of a C Controller module startup and the starting order of tasks.

Considerations for creating a script file

■Starting a user program

Describe a command so that a user program is executed by activating the task[*1].

*1 Priority: 100 to 255

■Specifying VX_FP_TASK option

When specifying a VX_FP_TASK option for the third argument of taskSpawn, specify '0x1000000.'

Starting with the VX_FP_TASK option specified for "funcA" function

```
taskSpawn("TaskA",100,0x1000000,20000,funcA,0,0,0,0,0,0,0)
```

Script file storage location

A script file can be stored in the program memory and an SD memory card.

Store a script file in the root directory.

■A script file stored into both the program memory and an SD memory card

When a script file is stored in both memory, the script file in the SD memory card has priority, and the script file in the program memory is not executed.

Point

In the system which has been operated by a script file stored in the program memory, its processing can be switched for maintenance easily by installing an SD memory card in which a script file for maintenance work is stored.

Description of scripts

■Description of commands

- Only one command can be described in one line.
- Up to 12 arguments can be specified for one command.

■Description of comment statements

- Describe "//" at the beginning of a line.
- There is no limit on the number of characters in a statement.

■Executing C++ functions

Describe the function declaration portion on C++ source code as follows.

- extern "C" {Function declaration portion}

Point

Description of commands in a script file allows the following operations.

- Copying a user program
- Formatting the program memory

Commands described in a script file are the same as contents described in a startup script file for VxWorks. For details, refer to the manual for VxWorks.

Example of script file description

The following shows an example to load a user program in the program memory and SD memory card using a script file in the program memory.

■Storage file

Memory	File storage destination path	File name	Description
Program memory	/0	STARTUP.CMD	Script file to load a user program
	/0/DirA	fileA.out	User program ("funcA" function has already been included.)
SD memory card	/2/DirB	fileB.out	User program ("funcB" function has already been included.)

■Task contents

Activation order	Task name	Priority	Stack size	Function name	Argument specification
1	Default: tN (N=1, 2, ...)	Default: 100	Default: 20000 bytes	funcA	None
2	taskB	120	5000 bytes	funcB	Specified (First argument: 10)

■Example of script file (STARTUP.CMD) description

```
ld(1,0,"/0/DirA/fileA.out")                    /*(1)*/
ld(1,0,"/2/DirB/fileB.out")                    /*(2)*/
sp(funcA,0,0,0,0,0,0,0,0,0,0)                  /*(3)*/
taskSpawn("taskB",120,0,5000,funcB,10,0,0,0,0,0)   /*(4)*/
```

(1) Load the "fileA.out" file from "DirA" directory in the program memory.
(2) Load the "fileB.out" file from "DirB" directory in the SD memory card.
(3) Generate the "funcA" function under the default task name (t1). (The "funcA" function is already included in the fileA.out file.)
(4) Generate the "funcB" function under the task name, "taskB." (The "funcB" function is already included in the fileB.out file.)

Point

Up to 12 arguments can be specified for one command. Therefore, in the example of script file description above, up to seven arguments can be passed to the function entry specified for taskSpawn (example: funcB). If eight or more arguments are specified, the task will not be activated.

2.13 Registering a User Program

This section shows the procedure to register a user program.

1. Describe the loading destination of the user program which starts at the time of the C Controller module startup to the script file, "STARTUP.CMD".
2. Write the script file and user program file to the C Controller module.
3. Turn the power OFF and ON, or reset the C Controller module.

Point

- User program files can be stored in any memories that can store files.
- To register by overwriting, close the file before overwriting it. Overwriting with the file opened may fail.

How to write

The following methods are available to write a script file and a user program.

- Writing from a personal computer by using the FTP function
- Writing from an SD memory card by copying with a command in a script file

■When writing files using the FTP function from a personal computer

The following shows the procedure to write files using the FTP function from a personal computer.

1. Create the following files.

File name	Description
STARTUP.CMD	Script file to load a user program
file.out*1	User program

*1 Specify an arbitrary file name.

- STARTUP.CMD

```
ld(1,0,"/0/file.out")          /*(1)*/
sp(func,0,0,0,0,0,0,0,0,0)     /*(2)*/
```

(1) Load the "file.out" file from root directory in the program memory.
(2) Generate the "func" function under the default task name (t1). (The "func" function is already included in the file.out file.)

2. Turn the power of the C Controller module ON.
3. Connect the C Controller module with the personal computer.
4. Access the C Controller module using the FTP function.
5. Write the script file and user program to the program memory.
6. Turn the power OFF and ON, or reset the C Controller module.
7. Check that the READY LED starts flashing. ("STARTUP.CMD" is executed.)
8. Check that the READY LED turns ON. ("file.out" is executed.)

Point

The default settings of the FTP function are as follows:

- Login name: target
- Password: password

■When writing from an SD memory card by copying with a command in a script file

The following shows the procedure to copy files in an SD memory card by using a command in a script file.

Copy a file in the SD memory card in the following cases.

- When a personal computer and a C Controller module cannot be connected normally
- When writing a same user program in multiple C Controller modules

1. Create files.

File name	Description
STARTUP.CMD	Script file to copy files to the program memory
STARTUP.ROM*1	Script file to load a user program
file.out*2	User program

*1 Use any file name other than "STARTUP.CMD".

*2 Use arbitrary file name.

- STARTUP.CMD

```
copy("/2/file.out","/0/file.out")                    /*(1)*/
copy("/2/STARTUP.ROM","/0/STARTUP.CMD")    /*(2)*/
```

(1) Copy the user program, "file.out" in the SD memory card to the program memory.

(2) Rename the operational script file, "STARTUP.ROM" in the SD memory card to "STARTUP.CMD," and copy it to the program memory.

- STARTUP.ROM

```
ld(1,0,"/0/file.out")              /*(1)*/
sp(func,0,0,0,0,0,0,0,0,0)    /*(2)*/
```

(1) Load the "file.out" file from root directory in the program memory.

(2) Generate the "func" function under the default task name (t1). (The "func" function is already included in the file.out file.)

2. Write the created files to the SD memory card.

3. Insert the SD memory card to a C Controller module.

4. Turn the power OFF and ON, or reset the C Controller module.

5. Check that the READY LED starts flashing. ("STARTUP.CMD" is executed.)

6. Check that the READY LED turns ON. ("STARTUP.CMD" processing is completed.)

7. Turn the power of the C Controller module OFF, and remove the SD memory card.

8. Turn the power of the C Controller module ON.

9. Check that the READY LED starts flashing. (The file name is changed and "STARTUP.CMD" command which is copied to the program memory is executed.)

10. Check that the READY LED turns ON. ("file.out" is executed.)

PART 3 SYSTEM CONFIGURATION

This part consists of the following chapter.

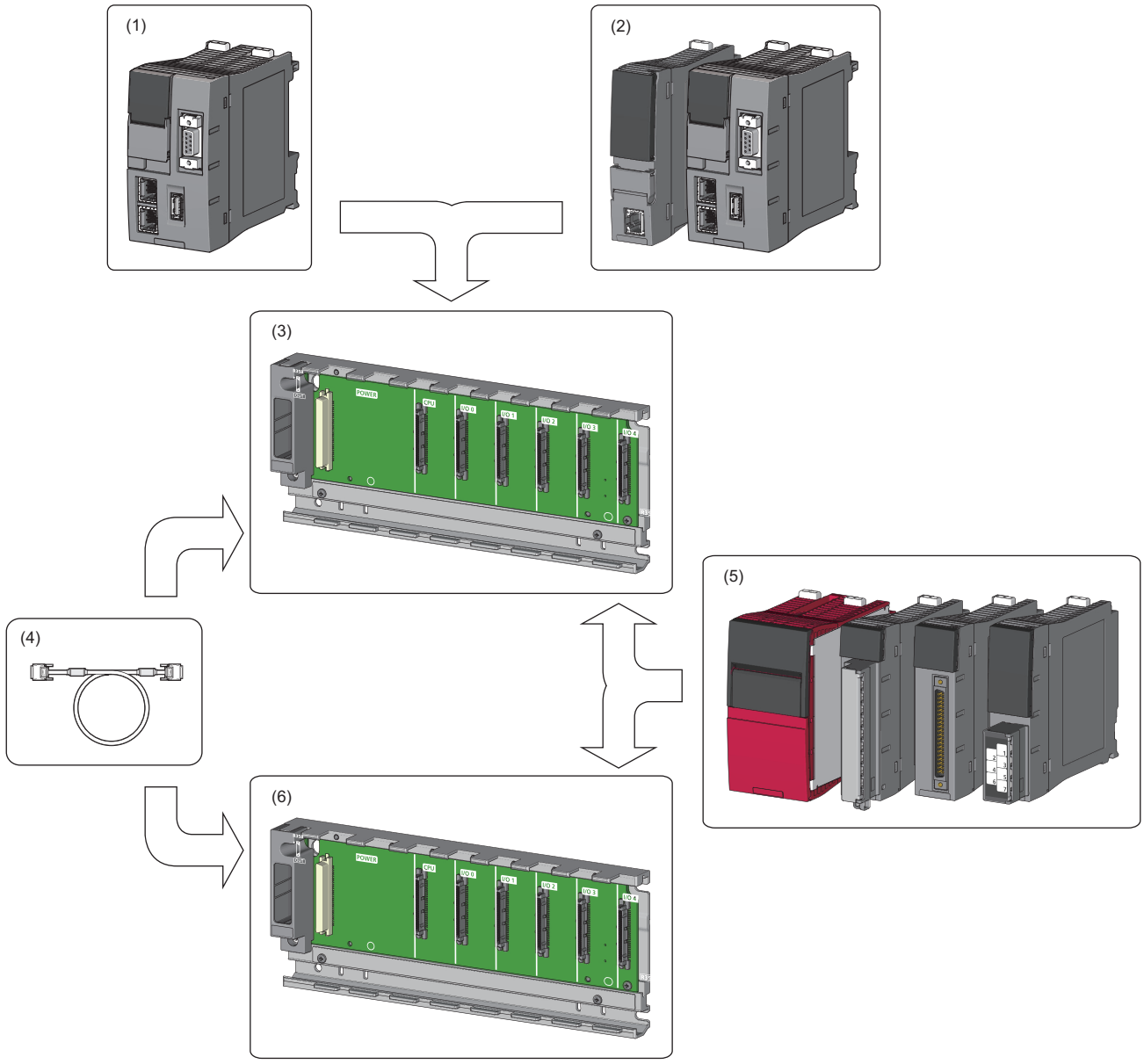
3 SYSTEM CONFIGURATION

3 SYSTEM CONFIGURATION

This chapter shows the overall configuration and considerations for system configuration of a C Controller system.
When configuring a C Controller system, refer to the following:
MELSEC iQ-R Module Configuration Manual

3.1 Overall Configuration

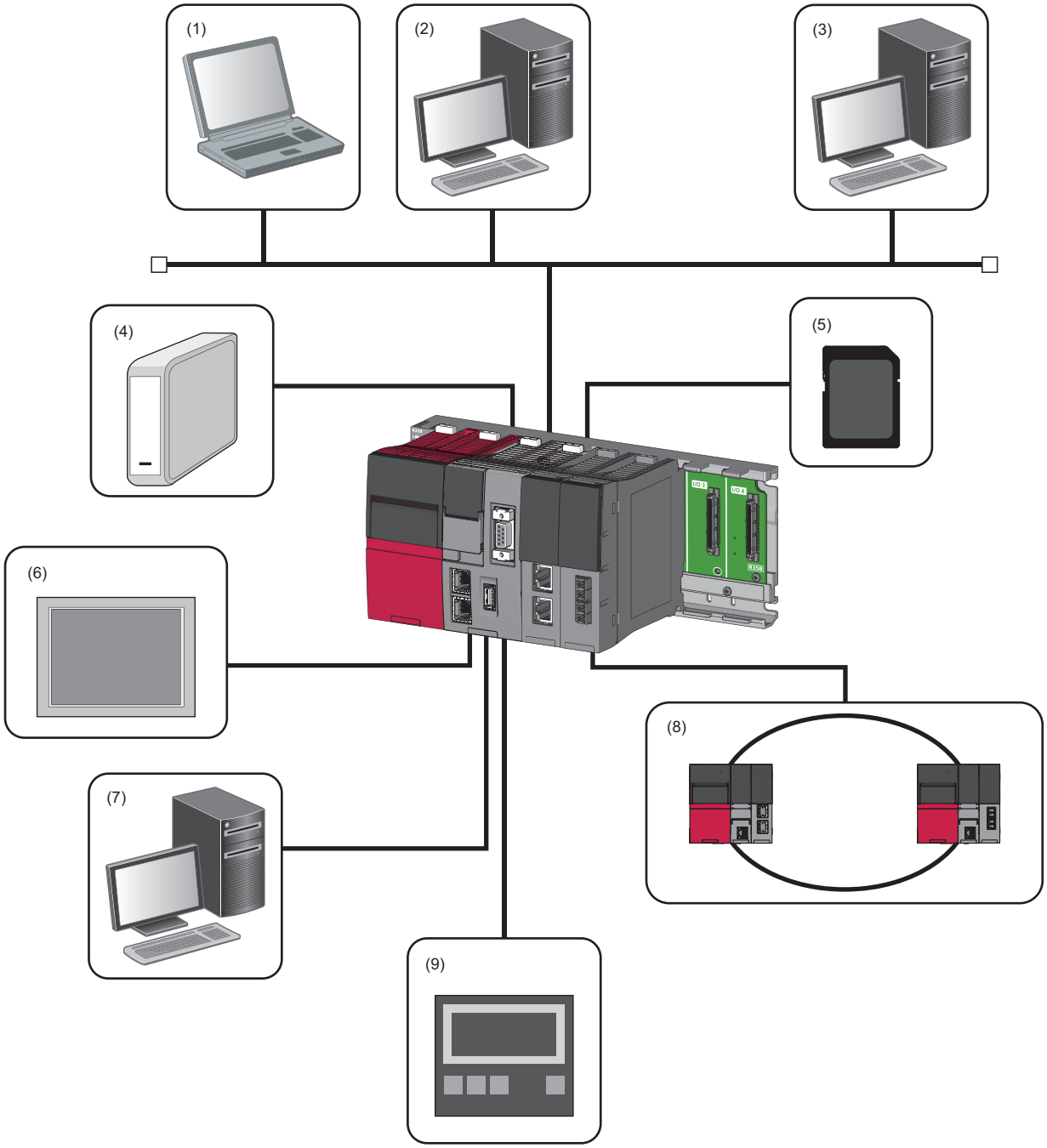
The overall configuration of a C Controller system is shown below.



(1) C Controller module
(2) Programmable controller CPU, process CPU, motion CPU, or C Controller module
(3) Main base unit
(4) Extension cable
(5) Power supply module, I/O module, or intelligent function module
(6) Extension base unit or RQ extension base unit

3.2 Peripheral Configuration

The configuration with peripherals is shown below.



(1) Personal computer for maintenance (Telnet function and FTP function)
(2) User program development environment (CW Workbench and CW-Sim)
(3) SNTP server
(4) USB Mass Storage Class-compliant device
(5) SD memory card
(6) Connection via a built-in Ethernet (HMI (GOT), SLMP supported device)
(7) CW Configurator
(8) Various networks via a network module (CC-Link IE Controller Network, CC-Link IE Field Network, CC-Link IE TSN, MELSECNET/H network, and CC-Link)
(9) Connection via a built-in Ethernet (CC-Link IE Field Network Basic device)

Point

- USB devices can be used for a C Controller module with firmware version '03' or later.
- Insert or connect peripherals to a C Controller module so that the specifications of both C Controller module and peripherals are met.
- For information on the access via each network module and the access using Ethernet communication, refer to the following:

☞ Page 145 Ethernet COMMUNICATION FUNCTIONS
☞ Page 165 CC-Link IE Field Network Basic FUNCTION
☞ Page 195 ACCESS FUNCTION USING NETWORK MODULE

Available software

The following software can be used for the system of MELSEC iQ-R C Controller modules. (📖 Manual for each software)

Software package		Version
CW Configurator	SW1DND-RCCPU-J	Version 1.00A or later
	SW1DND-RCCPU-E	
CW Workbench	SW1DND-CWWR-E/EZ/EVZ	Version 1.00A or later
CW-Sim	SW1DND-CWWSIMR-EZ	Version 1.00A or later
CW-Sim Standalone	SW1DND-CWWSIMSAR-E	Version 1.00A or later
Wind River Workbench	—	Version 3.3
GX Works3	SW1DND-GXW3-J	Version 1.007H or later
	SW1DND-GXW3-E	
GT Designer3	SW1DNC-GTWK3-J	Version 1.126G or later
	SW1DNC-GTWK3-E	
MT Works2	SW1DNC-MTW2-J	Version 1.110Q or later
	SW1DNC-MTW2-E	

SD memory card

Only one SD memory card can be inserted in a single C Controller module.

Available memory cards

The following Mitsubishi Electric Corporation's SD memory cards are available.

Model name	Description
NZ1MEM-2GBSD	SD memory card 2 GB
NZ1MEM-4GBSD	SD memory card 4 GB
NZ1MEM-8GBSD	SD memory card 8 GB
NZ1MEM-16GBSD	SD memory card 16 GB

For commercially available SD memory cards, refer to the following document. Before using any commercially available SD memory card, it is advised to check and ensure that the card has no impact on the control of the system.
TECHNICAL BULLETIN No. FA-A-0023

Precautions

- Use the format function of CW Configurator to format an SD memory card.
- If any SD memory card other than the one listed above is used, data in the SD memory card may be corrupted or a system shutdown may occur.
- If the power is turned OFF or a C Controller module is reset, or the SD memory card is removed while the card is being accessed, data in the SD memory card may be corrupted. Always turn the power OFF or reset a C Controller module, or remove an SD memory card after the access to the card has been stopped.
- If data is written or read to/from an SD memory card frequently, the writing speed to the memory card may be slower. In such case, data may fail to be written to the SD memory card when writing a large amount of data to it.
  If writing data to an SD memory card fails, take either of the following corrective actions:
  - Reduce data size to be written to the SD memory card.
  - Retry after a while.

MEMO

PART 4 SPECIFICATIONS

This part consists of the following chapters.

4 PERFORMANCE SPECIFICATIONS

5 DOT MATRIX LED DISPLAY SPECIFICATIONS

6 WIRING SPECIFICATIONS

4 PERFORMANCE SPECIFICATIONS

This chapter shows the performance specifications of a C Controller module.

Item			R12CCPU-V		
Hardware	Endian format		Little endian		
	MPU		ARM® Cortex-A9 Dual Core		
	Memory capacity	Work RAM	256 MB[*1]		
		ROM	16 MB (program memory: 8 MB, data memory: 8 MB)		
		Backup RAM	4096 KB (File storage area for device/label memory[*2]: 3584 KB, For system: 512 KB)		
Software	Operating system		VxWorks Version 6.9		
	Programming language		C language (C/C++)		
SD memory card slot	Interface		SD/SDHC memory card (up to 16 GB)		
	Power supply		+3.3 VDC, up to 200 mA		
Ethernet port	Number of channels		2		
	Interface		1000BASE-T	100BASE-TX	10BASE-T
	Data transmission rate		1 Gbps	100 Mbps	10 Mbps
	Number of cascade connections[*3]		—	2 levels maximum	4 levels maximum
	Maximum segment length		100 m (length between a hub and a node)		
	Communication method		Full-duplex/half-duplex		
	Transmission method		Base band		
	Applicable connector for external wiring		RJ45		
	Supported function		Auto-negotiation function (automatic recognition of the communication speed/ communication method) Auto-MDI/MDI-X (automatic recognition of a straight/crossover cable)		
	IP version		IPv4 supported		
RS-232 connector	Number of channels		1		
	Interface		RS-232-compliant		
	Communication method		Full-duplex/half-duplex		
	Synchronization method		Asynchronous communication		
	Transmission rate		9600, 14400, 19200, 28800, 38400, 57600, 115200 bps		
	Transmission distance		Up to 15 m		
	Data format	Start bit	1		
		Data bit	7/8		
		Parity bit	1/None		
		Stop bit	1/2		
	Parity check		Yes (Even/Odd)/None		
	Sum check code		Yes/None		
	Transmission control		Flow control (RS/CS control)		
USB connector	Interface		USB 2.0-compliant		
	Connector		Type A		
	Transfer rate		480 Mbps (High Speed)		
	Power supply		Bus power +5 VDC, up to 500 mA[*4]		
	Complied standard		USB Mass Storage Class (up to 2 TB)		
Number of occupied I/O points			4096 points		
Clock function	Displayed information		Year, month, day, hour, minute, second, day of week (automatic leap year detection)		
	Precision		Daily error : -5.86 to +3.35 seconds (0 to 55 ℃) Daily error: -1.71 to +3.35 seconds (25 ℃)		
Allowable momentary power failure time			According to the power supply module		
5 VDC internal current consumption			1.26 A[*5]		
External dimensions	Height		106 mm		
	Width		56 mm		
	Depth		110 mm		

Item	R12CCPU-V
Weight	0.35 kg

*1 Device area of device/label memory and module label area are secured in the work RAM area.
For details on the memory configuration, refer to the following:
☞ Page 74 Memory Configuration

*2 For details on the file storage area, refer to the following:
☞ Page 77 Files

*3 For a repeater hub.
For a switching hub, consult the manufacturer of the hub used.

*4 Select an appropriate USB Mass Storage Class-compliant device to be connected to observe the current consumption limit.

*5 The current consumption of a USB is not included.

5 DOT MATRIX LED DISPLAY SPECIFICATIONS

This chapter shows the display specifications of dot matrix LED.

Operation selection

LED display	Operation	Reference
EVENT	An event is notified to the user program.	☞ Page 38 Selecting and performing an operation
SD UNMOUNT	An SD memory card is unmounted forcibly.	
USB UNMOUNT	A USB Mass Storage Class-compliant device is unmounted forcibly.	
SD/USB UNMOUNT	An SD memory card and USB Mass Storage Class-compliant device are unmounted forcibly.	
USER	Contents specified by the user are displayed on the dot matrix LED.	
ERROR	An error code is displayed on the dot matrix LED.	
CH1	The IP address of Ethernet port (CH1) is displayed on the dot matrix LED.	
CH2	The IP address of Ethernet port (CH2) is displayed on the dot matrix LED.	

Hardware diagnostics

LED display	Description	Reference
M-00	Diagnostic test for Mode 1 to Mode 6	☞ Page 30 Hardware Diagnostics
M-01	Program memory, data memory diagnostic test	
	Device/label memory diagnostic test	
M-02	Ethernet diagnostic test	
M-03	SD memory card interface diagnostic test	
M-04	RS-232 diagnostic test	
M-05	USB diagnostic test	
M-06	Bus diagnostic test	
M-07	Dot matrix LED test	

Initialization

LED display	Description	Reference
0010	Default IP setting	☞ Page 28 Initialization
0011	Module initialization setting	

Firmware update

LED display	Description	Reference
0020	Firmware update setting	📖MELSEC iQ-R Module Configuration Manual

MEMO

6 WIRING SPECIFICATIONS

This chapter shows the wiring specifications of a C Controller module.

Point

The bend radius of the cable near the connector or port should be at least four times longer than the cable's outside diameter.

6.1 Ethernet Port

This section shows the specification of a usable Ethernet cable and its wiring.

Ethernet cable

The following shows the specifications of the cables that can be used for connection with peripherals by using the Ethernet ports (CH1 and CH2). Cables compliant with IEEE802.3 10BASE-T/100BASE-TX/1000BASE-T standards can be used.

Transmission rate	Unshielded twisted pair cable (UTP cable) Shielded twisted pair cable (STP cable)	
	Straight cable	Crossover cable
1 Gbps	Category 5e or higher	Category 5e
100 Mbps	Category 5 or higher	Category 5 or 5e
10 Mbps	Category 3 or higher	Category 3 to 5e

Connection procedure

1. Check the insertion direction, and insert an Ethernet cable into the Ethernet port on the C Controller module until it clicks.
2. Check if the cable is securely connected by pulling it slightly.
3. Check that the SPEED LED of the Ethernet port is ON.

Point

- The time required from when an Ethernet cable is connected to when the SPEED LED turns ON may vary. Normally, it turns ON in a few seconds. However, it may take longer because the linking-up processing is repeated due to the device condition on the line.
- When the SPEED LED does not turn ON, check if the connected Ethernet cable has any failure.
- The SPEED LED is turned OFF when connecting with an Ethernet device on the network of which the transmission rate is 10 Mbps. Check the communication state by executing the PING test, etc.

Disconnection procedure

1. Pull out the Ethernet cable while pinching a clip on the connector.

IP address setting

To use the Ethernet port(s), set the IP address with CW Configurator.

[Navigation window] ⇨ [Parameter] ⇨ [(CPU module)] ⇨[Module Parameter] ⇨ [Basic Settings] ⇨ [Own Node Settings] ⇨ [IP Address]

The default value is set to the IP address for the C Controller module before shipment and after initialization.

Ethernet port	Default IP address
Ethernet port (CH1)	192.168.3.3
Ethernet port (CH2)	No setting

■Checking method of IP address

The IP address set to a C Controller module can be checked with the operation selection mode or the diagnostics function of CW Configurator.

For details on the operation selection mode, refer to the following:

☞ Page 37 Switch Operation

Precautions

■Considerations for IP address setting

The following shows the considerations for setting IP address.

- Set the same value for the network portion of the IP addresses for the target device to be connected and the Ethernet port.
- Set the different value for the network portion of CH1 and CH2 on the Ethernet port.
- A message to the target device (including response packets such as ping) is sent (responded) from the Ethernet port that has IP address of which network portion is the same.
- Do not set the IP address for a C Controller module with a user program.

■Considerations for Ethernet device connection

The following shows the considerations for connecting Ethernet devices.

- When the C Controller module has been replaced and also IP address has been changed, then reset the Ethernet device too. If the Ethernet device holds the Ethernet address (MAC address) of the communication target, continuous communication may not be performed since the module replacement will change the Ethernet address (MAC address).
- Perform the troubleshooting in accordance with the manual for the Ethernet device when an error occurred on the Ethernet device.
- When Telnet or Shell of CW Workbench is connected to a C Controller module, an event/error message may be displayed once VxWorks detects a network error. For details on the displayed event/message, refer to the manual for VxWorks. Please ask Wind River Systems, Inc. for any event/message which cannot be handled.

■Considerations for 1000BASE-T/100BASE-TX connection

In a high-speed data communication (1 Gbps/100 Mbps) via 1000BASE-T/100BASE-TX connection, a communication error may occur due to the effect of high frequency noise generated from the equipment other than C Controller system, depending on the installation environment. Take the following countermeasures on a C Controller module to eliminate the effect of high frequency noise when constructing a network system.

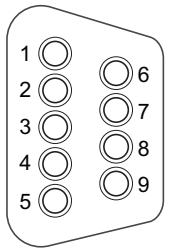
- Do not install the twisted pair cables together with the main circuit or power lines, or bring them close to each other.
- Make sure to place the twisted pair cable in a duct.
- In the environment where the cable is susceptible to noise, use a shielded twisted pair cable (STP cable).
- In an environment where the system is susceptible to noise, include a retry processing in a user program.
- Change the target device connected with the C Controller module to one which communicates at 10 Mbps, and decrease the data transmission rate.

6.2 RS-232 Interface

This section shows the specification of an RS-232 interface and its wiring.

RS-232 connector

The RS-232 interface specifications to connect to a target device are shown below.

Connector shape	Pin No.	Signal code	Signal name	Signal direction (R12CCPU-V ⇔ RS-232 device)
	1	CD(DCD)	Data carrier detect	←
	2	RD(RXD)	Received data	←
	3	SD(TXD)	Transmitted data	→
	4	ER(DTR)	Data terminal ready	→
	5	SG	Signal ground	↔
	6	DR(DSR)	Data set ready	←
	7	RS(RTS)	Request to send	→
	8	CS(CTS)	Clear to send	←
	9	CI(RI)	Ring indicator	←

Use the following product as a connection cable connector.

Connector type	Manufacturer name	Model name
D-sub connector (Solder-connection type)	DDK Ltd.	17JE-13090-02(D8C)(-CG)

The specifications of fixing screws are as follows:

- Connector mating screw: #4-40UNC
- Tightening torque range : 0.15 to 0.20 N·m

Precautions

■Considerations for wiring RS-232 cable

The following shows the considerations for wiring RS-232 cable.

- Make sure that RS-232 connection cable shield is single-point grounded.
- Do not short-circuit the FG and SG signals of the RS-232 connection cable. When the FG and SG signals are connected inside of the peripheral device, do not connect the FG signal to a C Controller module.
- For connection method of peripheral device, check the specifications of the peripheral device.

■Considerations for connecting a target device

The following shows the considerations for connecting a target device.

- A reception error may occur on the target device connected to a C Controller module when the power for the C Controller module or the target device is turned ON or OFF.
- A reception error occurs on the target device when the system on the target device side is booting while a C Controller module is transmitting data to the target device.
- When an error occurred on the target device, take corrective actions in accordance with the manual for the target device.

6.3 USB Connector

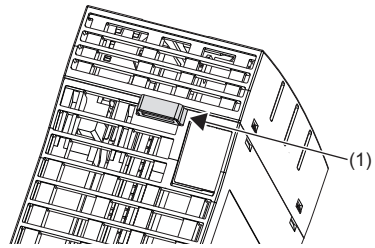
This section shows the specification of a usable USB cable and its wiring.

USB cable

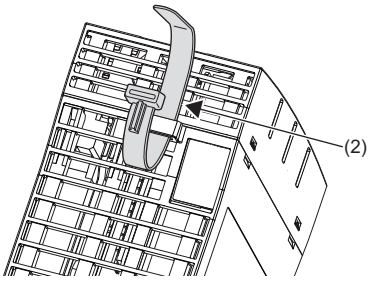
Use a USB cable supplied with a USB Mass Storage Class-compliant device.
The type of cable connector connectable with the USB connector is type A.

Preventing USB cable from falling out

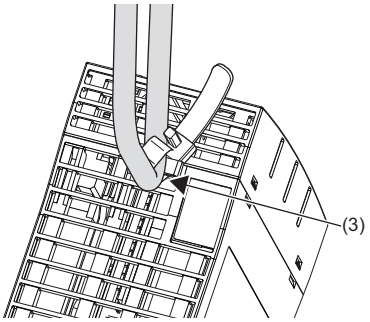
Connected USB cable can be fixed to the hole for fixing band of the module by using a fixing band (recommended specification: width: 6 to 9 mm, thickness: 1 mm or less).



1. A through hole for a fixing band (1) is situated on the bottom of a C Controller module.



2. Pass the fixing band (2) through the hole.



3. Pass the USB cable (3) through the fixing band to fix.

6.4 HMI (GOT)

This section shows the connection method of GOT to a C Controller module.

Connection route

A C Controller module can be connected to GOT by any of the following routes.

- Connection using Ethernet port(s)
- Connection via a network module

Connection using Ethernet port(s)

Connect to GOT by using the Ethernet port(s) (CH1 and CH2).

Connection via a network module

Connection to GOT is available via any of the following network modules.

- CC-Link IE Controller Network
- CC-Link IE Field Network
- CC-Link (via an intelligent device station, G4)

For the connection methods via a network, refer to the manual for the network module used.

PART 5 C CONTROLLER MODULE OPERATION

This part consists of the following chapters.

7 EXECUTING PROGRAMS

8 OPERATION PROCESSING IN C CONTROLLER MODULE

9 MEMORY CONFIGURATION OF C CONTROLLER MODULE

7 EXECUTING PROGRAMS

This chapter explains the program execution of a C Controller module.

7.1 Execution Order

The following shows the execution order of the programs in a C Controller module.

1. Initial processing
2. Operation processing of a program

7.2 Initial Processing

Initial processing is pre-processing for executing a program.
This processing is executed when turning ON or resetting a C Controller module. Completing the processing indicates that the C Controller module is set by the RESET, STOP, or RUN switch.

Processing to be executed*1

- Booting*2 from a memory card
- Checking*3 for each parameter
- Checking multiple CPU system parameter consistency
- Assigning I/O number for the mounted module
- Initializing each module and setting information*4
- Script file execution

*1 When a memory card is inserted and memory card access is enabled, it takes time for the initial processing. Therefore, a C Controller module takes time to respond to peripheral devices.
In addition, the initial processing time may be further extended by the following conditions:
· Using parameters in a memory card according to the service settings and memory card parameter settings
· A memory card status (e.g. the memory card contains the excess number of files or reaches (exceeds) its service life.)

*2 For details on the boot function, refer to the following:
☞ Page 94 Boot Function of Parameters

*3 This processing may take time. Therefore, a C Controller module may take time to respond to peripheral devices.

*4 This processing may take time. Therefore, a C Controller module may take time to respond to peripheral devices.

Precautions

The startup time of a C Controller module may vary depending on the firmware version.
To make a system program run on the startup of a C Controller module, configure an interlock using special relays and special registers. (☞ Page 360 Special Relay List, Page 362 Special Register List)

Special relay/special register	Description
CPU operating status (SD203)	Available to check the completion status of initial processing with operating status. If the operation status is in RUN, it indicates that the initial processing is completed.
LED status (SD201)	Available to check the activation status of a script file with the READY LED status. If the READY LED is flashing, it indicates that a script file is activated.
Special relays or special registers of each network module	Available to check the communication status with a target station.

7.3 I/O Access Timing

This section shows the timing for reading input (X) and writing output (Y).

Timing for reading input (X)

The timing for reading input (X) is as follows:

- When a dedicated function library (such as CCPU_X_In_BitEx/mdRandREx) is executed in a user program
- When input (X) data is read out from a peripheral device (Device/buffer memory batch monitoring of CW Configurator is performed.)

Timing for writing output (Y)

The timing for writing output (Y) is as follows:

- When a dedicated function library (such as CCPU_Y_Out_BitEx/mdRandWEx) is executed in a user program
- When output (Y) data is written to a peripheral device (Device/buffer memory batch monitoring of CW Configurator is performed.)

8 OPERATION PROCESSING IN C CONTROLLER MODULE

This chapter explains the operation processing of a C Controller module.

8.1 Operation Processing Depending on Operating Status

The operating statuses of a C Controller module are as follows:

- RUN
- STOP
- PAUSE

Point

- Even if the operating status of the C Controller module is in RUN, STOP, or PAUSE, the output (Y) can be performed and data can be written to the buffer memory from CW Configurator, SLMP communication, and MELSEC data link functions.
- Even if the operating status of the C Controller module is in RUN, STOP, or PAUSE, the operation processing of a user program is performed. When splitting the program processing according to the operating status, use the C Controller module dedicated function (CCPU_GetCpuStatus) for programming.

Operation processing in RUN state

RUN is a state where performing output (Y) to each module and writing data to the buffer memory from a user program*1 are enabled.

*1 A program in which a C Controller module dedicated function is used

■Output when entering the RUN state

C Controller module outputs the following depending on the setting of the output (Y) when the module state is changed from STOP to RUN. (☞ Page 93 Output Mode Setting Function at STOP to RUN)

- The status of output (Y) which is saved in STOP state

Operation processing in STOP state

STOP is a state where performing output (Y) to each module and writing data to the buffer memory from a user program*1 are disabled.

The operating status can be changed with the RESET/STOP/RUN switch or the remote operation function.

If the stop error occurred in a C Controller module, the module will be in STOP state.

*1 A program in which a C Controller module dedicated function is used

■Output when entering the STOP state

C Controller module saves the output status right before the C Controller module is in STOP state, and clears all outputs (Y) to OFF.

Operation processing in PAUSE state

PAUSE is a state where performing output (Y) to each module and writing data to the buffer memory from a user program are disabled with the ON/OFF of the output (Y) retained.

Considerations

The following processing is performed in a C Controller module even if its operating status is STOP or PAUSE.

- Refreshing label memory and motion module buffer memory
- Refreshing devices and link devices (link refresh)

Therefore, when user programs which perform the following processing are running, values are output to a motion module and network even if a C Controller module operating status is STOP or PAUSE.

- Writing data to the label memory
- Writing data to devices to be refreshed to link devices

When splitting the program processing according to the operating status of the C Controller module, use the C Controller module dedicated function (CCPU_GetCpuStatus) for programming.

8.2 Operation Processing at Momentary Power Failure

When the input power voltage fed to the power supply module is lower than the specified range, a C Controller module detects a momentary power failure and performs the following processes.

When a momentary power failure less than the allowable time has occurred

When a momentary power failure occurs, a C Controller module suspends processing with its output status retained.
After power is recovered, error information is registered to the event history file. (Only at the first detection)

■When the momentary power failure is recovered

After the momentary power failure is recovered, a C Controller module continues processing.

■Check for the number of momentary power failure detection

Since a C Controller module retains number of momentary power failure inside the module, it can be checked with the special register (SD53) or the C Controller module dedicated function (CCPU_GetPowerStatus).

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

A C Controller module continues measuring watchdog timer if the operation is stopped due to the momentary power failure. For example, if a momentary power failure of 15 ms has occurred when the fixed cycle processing time is 190 ms while the monitoring time of the system watchdog time is set to 200 ms, a watchdog timer error occurs.

When a power failure longer than the allowable time has occurred

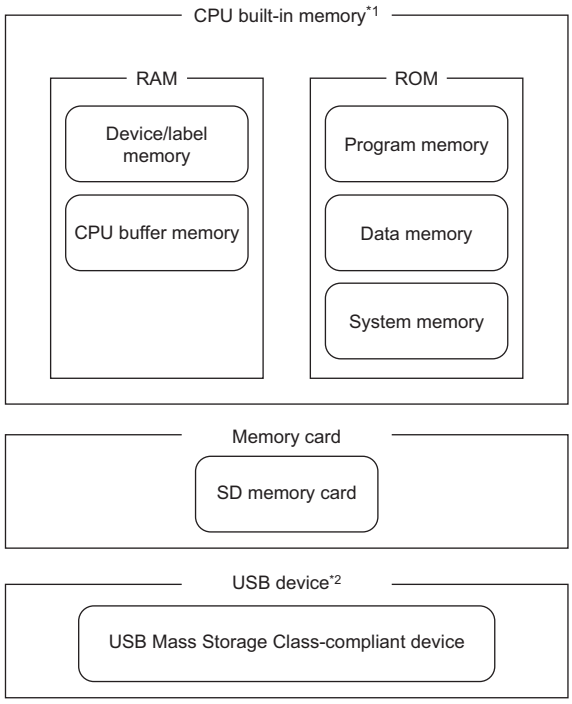
The initial start is performed and the operation processing will be the same as when the powered ON or reset the C Controller module.

9 MEMORY CONFIGURATION OF C CONTROLLER MODULE

This chapter explains the memory configuration of a C Controller module.

9.1 Memory Configuration

This section shows the memory configuration of a C Controller module.



*1 CPU built-in memory is a generic term for the built-in memory in a C Controller module.

*2 When using a USB device, check the firmware version of the C Controller module. (☞ Page 404 Added and Changed Functions)

Point

The usage of memory can be checked from CW Configurator. (📖CW Configurator Operating Manual)

Program memory

Program memory is a memory to store files such as user programs and script files.

System memory

System memory is a memory to store system files.

Writing files to the system memory is not available.

CPU buffer memory

CPU buffer memory is a memory that is used for data communication among multiple CPUs.

Device/label memory

Device/label memory has the following areas.

Device area
Module label area
File storage area

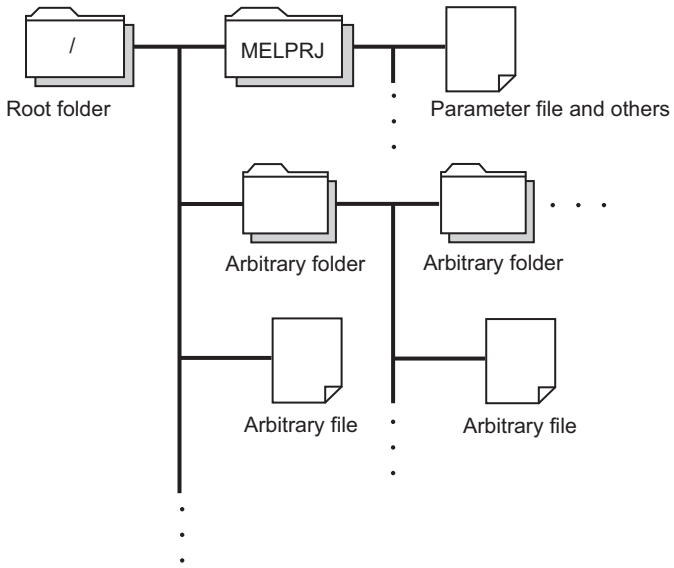
Data to be allocated

The following table shows the data to be allocated to each area.

Area	Application
Device area	User device
Module label area	Module label
File storage area	File register

Data memory

Data memory is a memory to store the data such as parameter files or arbitrary folders/files. Parameter files written with CW Configurator are stored in the "MELPRJ" folder. (☞ Page 78 Access to the "MELPRJ" folder)



SD memory card

SD memory card is a memory that stores the folders/files created by a function using SD memory card as well as the arbitrary folder/file. The folder configuration is the same as the data memory. (☞ Page 78 Access to the "MELPRJ" folder)
It can be accessed from an FTP, a Telnet, and a user program.

USB Mass Storage Class-compliant device

USB Mass Storage Class-compliant device is a memory that stores user programs and arbitrary folders/files.
It can be accessed from an FTP, a Telnet, and a user program.

Point

For details on how to create and delete user folders and files, refer to the following:
CW Configurator Operating Manual

9.2 Memory Operation

Memory can be initialized and values can be cleared in a C Controller module with CW Configurator.
For details on the memory operation, refer to the following:
CW Configurator Operating Manual

Item			Description
Initialization	Formatting program memory/data memory		Deletes all the folders and files in the program memory and data memory.*1
	Formatting an SD memory card		Deletes all the folders and files in an SD memory card.*2
Clearing value	Device, label	Zero clear	Clears devices (X, Y, M, B, D, W) and contents of module labels to zero.
	File register		Clears all the file registers (ZR) to zero.

*1 The "MELPRJ" folder is created in the data memory, and the default parameters are set.
*2 The "MELPRJ" folder is created.

Point

- If the power is turned OFF or the module is reset while initializing each memory or clearing values, the memory will be partly initialized or the value is partly cleared. In this case, perform the memory operation again.
- If the power is turned OFF or the module is reset while accessing each memory, data corruption in the memory or file system error may occur. Shut down the program memory and data memory with the C Controller module dedicated function (CCPU_ShutdownRom), and then power OFF or reset the module.

9.3 Memory Life

This section shows the number of writable times (life) of program memory and data memory.

Program memory and data memory life

The life is represented as a write count index value and data can be written until it reaches to 100,000 times. However, since the system increases the life for writing of program memory and data memory, the write count index value differs from the actual write count.
If the write count index value exceeds 100,000 times, the following symptoms may occur. Replace the C Controller module.

- Decrease of writing speed to program memory and data memory
- Unable to write data to program memory and data memory

Method for checking write count index value

The write count index value can be acquired with the C Controller module dedicated function (CCPU_GetCpuStatus).
If the value exceeds 100,000 times, it is registered in the event history. (☞ Page 108 RAS Function)

Point

- Since the operation of the program memory and data memory is checked at factory acceptance test, the initial value of the write count index is not 0.
- Since a C Controller module always checks data in the program memory and data memory, the write count index value may be increased.

9.4 Files

This section shows the files of a C Controller module.

Drive names and file systems

The following table shows drive names and file systems that correspond to each memory.

Type	Name	Drive name	File system
CPU built-in memory	Program memory	/0	FAT16
	Data memory	/4	FAT16
	System memory	/SYSTEMROM	FAT16
Memory card	SD memory card	/2	FAT16/FAT32
USB device	USB Mass Storage Class-compliant device	/USB0	FAT16/FAT32

File type and storage destination memory

The following table shows file types and storage destination memory.

○: Storable, ×: Not storable

File type	CPU built-in memory				SD memory card	USB Mass Storage Class-compliant device	File name and extension
	Program memory	Device/ label memory	Data memory	System memory			
User program	○	×	×	×	○	○	Any
Script file	○	×	×	×	○	×	STARTUP.CMD
System parameter	×	×	○	×	○	×	SYSTEM.PRM
CPU parameter	×	×	○	×	○	×	CPU.PRM
Module parameter	×	×	○	×	○	×	UNIT.PRM
Module extended parameter	×	×	○	×	○	×	UEXmmmnn.PRM*1
CC-Link IE TSN device station parameter file	×	×	○	×	○	×	SLAVEmmmnnnxxxx.NSP*2
Memory card parameter	×	×	×	×	○	×	MEMCARD.PRM
Event history file	×	×	○	×	○	×	EVENT.LOG
Arbitrary folder/file	×	×	○	×	○	○	Any

*1 The 'mmm' represents the start input/output number of the module (the first three digits in four-digit hexadecimal representation). Note that the number of the CPU module is fixed to 3FFH. Additionally, the 'nn' represents the serial number (two-digit hexadecimal representation) of module extension parameter files or module-specific backup parameter files of each module.

*2 The 'mmm' represents the start input/output number (the first three digits in four-digit hexadecimal representation) of the master station of CC-Link IE TSN, the 'nnn' represents the number of units in network configuration setting, and the 'xxxx' represents the serial number (in four-digit hexadecimal representation) of the parameter.

File and folder configuration

The following table shows the configurations of files and folders.

○: Access allowed/Valid, ×: Access not allowed/Invalid, —: No folder

Drive	Folder	File		FTP access	Factory setting	After initialization
/0				○	×	×
/4	MELPRJ*1			○	○	○
/SYSTEMROM	OS_IMAGEFILE	R12CCPU-V_XX*2	VxWorks image file	○	○	○
	INCLUDE	CCPUFunc.h	C Controller module dedicated function header file	○	○	○
		MDFunc.h	MELSEC data link function header file	○	○	○
		DANLFunc.h	Data analysis function/ statistical analysis function header file	○	○	○
		MCFBFunc.h	Motion module dedicated class header file	○	○	○
	—	prjParams.h	VxWorks component list file	○	○	○
	—	DriveNameInfo.txt	Drive name list file	○	○	○
/2*3	MELPRJ*1			○	×	×
/USB0*4				○	×	×

*1 The folder is created after initialization or writing parameters.
*2 The suffixed 'XX' indicates the upper two digits of the serial number.
*3 The drive is created when an SD memory card is inserted.
*4 The drive is created when a USB Mass Storage Class-compliant device is connected.

Considerations on file operation

The following explains the restriction on memory and drive operations of a C Controller module.

■Writing files

Before writing files with an FTP or a Telnet during user program operation, make sure that the files to be written will not affect the running user program.

■Access to the "MELPRJ" folder

The "MELPRJ" folder manages data written with CW Configurator. Do not access it for a purpose other than backup or restoration.

If a portion of the files stored in the folder is changed, the C Controller module may not operate properly.

■Access to the same file

For C Controller modules, a file being written cannot be accessed and a file being accessed cannot be written.

■Number of files

Up to 512 files can be stored in the program memory and data memory of a C Controller module.*1 (A folder is included in the number of files.) However, the maximum number may be reduced depending on a file name length and character types.

When using an SD memory card and a USB Mass Storage Class-compliant device, note that the number of files which can be stored differs depending on the memory capacity and a file system format.

The number of files in a folder must be 500 or less. Storing more than 500 files may significantly increase the file access time.

*1 This number applies when storing a file which is composed of a file name within 8 characters and an extension within 3 characters, and which includes alphanumeric characters (not including lower-case characters) only.

■Creating folders

Do not create a folder in the root ("/") of a C Controller module. If a folder is created, an error may occur or an unintended folder may be created. (These errors also occur when transferring a folder to the root ("/") from FTP.)

■File names and folder names

Use alphanumeric characters and special characters (excluding \, /, *, ?, <, >, |, :, ", $) for a file name and folder name to be stored. Otherwise, the following symptoms may occur.

- Garbled file name and folder name
- Loss of file and folder

When only a special character (.) is used for a folder name, a folder cannot be created.

- A special character (.) following the usable character codes is not included in the folder name.
  A folder, "a" is created when a folder name starts with "a...".

When a file name is composed of lower-case characters, it may be displayed all in upper-case characters when referred from FTP or Telnet.

■File writing destination

Do not write files to the program memory and data memory using a user program.

Since the number of writable times (life) of program memory and data memory is limited, the available period of a C Controller module is shortened.

When writing files using a user program, write files to an SD memory card, a USB Mass Storage Class-compliant device, or a network device*1 (via FTP/NFS/netDrv), etc.

*1 For details, refer to the manual for VxWorks.

■Uninstallation of external memory devices

If the following memory devices are unmounted while writing data to user files in the memory devices, data corruption or file system error may occur. When writing data to user files, close the user files before unmounting.

- SD memory card
- USB Mass Storage Class-compliant device

■Writing data to an SD memory card

If data is written or read to/from an SD memory card frequently, the writing speed to the memory card may be slower. In such case, data may fail to be written to the SD memory card when writing a large amount of data to it.

If writing data to an SD memory card fails, take either of the following corrective actions:

- Reduce data size to be written to the SD memory card.
- Retry after a while.

MEMO

PART 6 FUNCTIONS

This part consists of the following chapters.

10 FUNCTION LIST

11 BASIC FUNCTIONS

12 MOTION MODULE CONTROL FUNCTION

13 Ethernet COMMUNICATION FUNCTIONS

14 CC-Link IE Field Network Basic FUNCTION

15 SECURITY FUNCTION

16 ACCESS FUNCTION USING NETWORK MODULE

17 MULTIPLE CPU SYSTEM FUNCTION

18 FIRMWARE UPDATE FUNCTION

10 FUNCTION LIST

This chapter shows the functions of a C Controller module.

Function			Description
Basic functions	Device access function		To read/write data from/to devices and buffer memory of an intelligent function module controlled by a CPU module or a C Controller module by using the dedicated function library.
	Label communication function		To read/write data from/to labels stored in CPU modules of other stations.
	Data analysis function		To perform data analysis processing such as fast Fourier transform, digital filter operation, calculation of a cross point between a wave and a specified value, and calculation of a standard deviation.
	Interrupt function to a C Controller module		To execute a routine registered with the C Controller module dedicated function (CCPU_EntryInt) as an interrupt routine when an interrupt request is issued to a C Controller module from an input module, interrupt module, intelligent function module, and another CPU module.
	MELSEC data link function setting function		To set parameters required for using the MELSEC data link function.
	Output mode setting function at STOP to RUN		To set the output (Y) mode when the operating status of a C Controller module is switched from STOP to RUN.
	Boot function of parameters		To read parameters stored in an SD memory card when turning the power of a C Controller module OFF and ON, or resetting the module.
	Remote operation function		To control the operating status of a C Controller module remotely.
	Clock function		To set the time for the built-in clock of a C Controller module.
	RAS function	Watchdog timer (WDT) function	To monitor and detect errors on hardware and user programs by using an internal timer of a C Controller module (system watchdog timer and user watchdog timer).
		Self-diagnostics function	To diagnose whether there are any errors in a C Controller module.
		Error clear function	To clear all the existing continuation errors in the system of a C Controller module in a batch.
		Event history function	To collect information from each module including errors detected by a module, operations performed for a module, and errors occurred on the network and save them. The saved information can be checked in chronological order of occurrence.
	Fixed cycle processing function		To perform operations such as the refresh processing with a network module and the self-diagnostics processing in a fixed cycle.
	Inter-module synchronization function		To adjust the input or output timing of modules to be synchronized to the inter-module synchronization cycle by matching the control timing of signals between multiple modules.
Motion module control function			To control a motion module controlled by a C Controller module.
Ethernet communication function	Connection with MELSOFT product or GOT		To monitor a C Controller module with CW Configurator, and to monitor and test a C Controller module from GOT via Ethernet.
	SLMP communication function		To read/write device data and perform the operation monitoring and data analysis in a C Controller module by accessing the module from an external device.
	FTP function		To access a file in a C Controller module from a target device with the FTP client function. (The FTP server function is supported by a C Controller module.)
	Time setting function		To set the time on a C Controller module automatically by acquiring time information from the time information server (SNTP server) on the network.
	Telnet function		To execute the Shell command with a Telnet tool in a personal computer for a TCP/IP network.
CC-Link IE Field Network Basic function	Cyclic transmission function		To perform data communication periodically between a master station and remote stations by using link devices.
Security function	Individual identification information read function		To read the individual identification information of a C Controller module with the C Controller module dedicated function (CCPU_GetIDInfo).
	File access restriction function		To set a file attribute to a file stored in the memory.
	IP filter setting function		To set security for the access using Ethernet.
	Service setting function		To set the services for a C Controller module.
	Lockout function		To prevent a brute force attack from unauthorized users. If password authentication fails for a certain number of times, it is rejected (locked out) for a fixed period of time.

Function		Description
Access function using network module	Cyclic transmission function	To perform data communication periodically between stations on the same network by using link devices.
	Transient transmission function	To perform data communication with other stations when a communication request is issued. Communication with a different network can also be performed.
Multiple CPU system function	Out-of-group I/O fetch	To access a module which is not controlled by a CPU module to import the input/output (X/Y) or access the buffer memory of an intelligent function module.
	Data communication between CPU modules	To send/receive data between CPU modules in a multiple CPU system. The direct access communication enables data writing or reading between CPU modules.
	Interrupt from another CPU	To restart a user program waiting for an interrupt event by an interrupt from another CPU module with the C Controller module dedicated function (CCPU_WaitEvent).
	Issuing an interrupt to another CPU	To issue an interrupt request to another CPU (C Controller module and motion CPU) from a C Controller module.
Firmware update function		To update the firmware of a C Controller module. (MELSEC iQ-R Module Configuration Manual)
System timer mode switching function		To change the system timer mode of a C Controller module.

11 BASIC FUNCTIONS

This chapter explains the basic functions of a C Controller module.

11.1 Device Access Function

This function reads/writes data from/to devices and buffer memory of an intelligent function module controlled by a CPU module or a C Controller module by using the dedicated function library.
For accessible modules and routes, refer to the following:
MELSEC iQ-R C Controller Module Programming Manual

Point

Data can be read from/written to devices and buffer memory of a C Controller module by using a peripheral device (such as CW Configurator).

Function list

The following table shows the functions used for accessing devices.

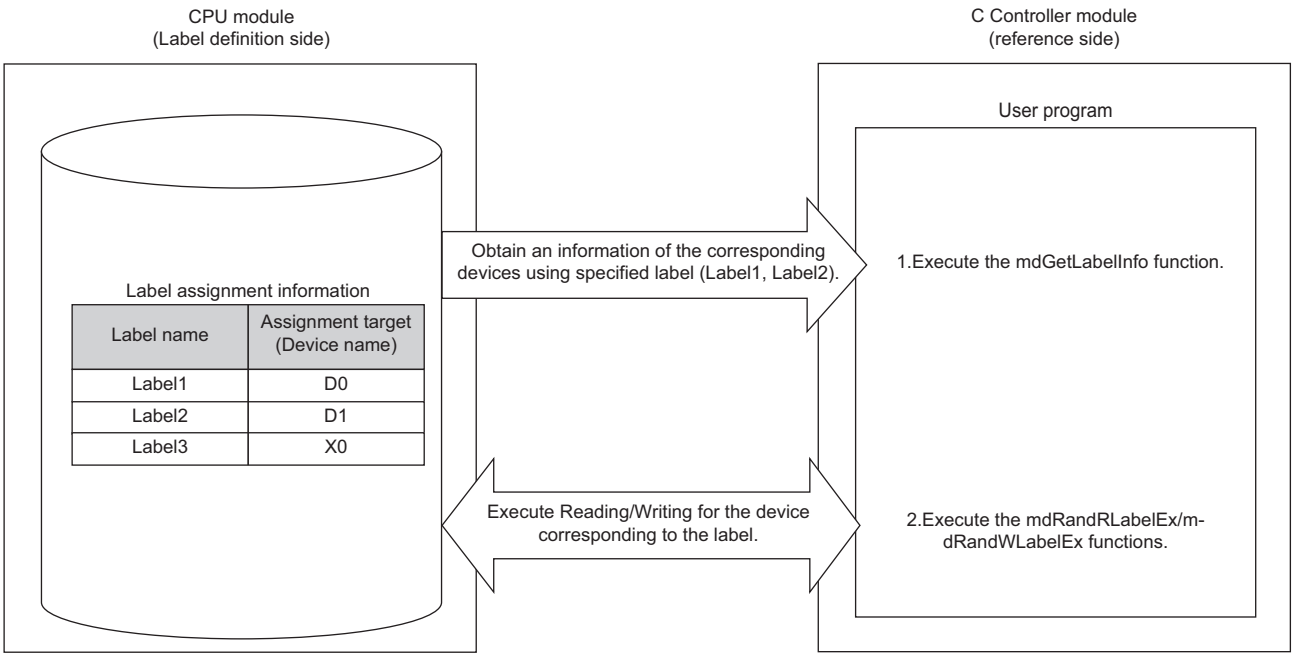
Function name	Description
CCPU_FromBuf	To read data from the CPU buffer memory of the CPU module or buffer memory of the intelligent function module which is mounted on the specified module position. (FROM instruction)
CCPU_FromBuf_ISR	
CCPU_ToBuf	To write data to the CPU buffer memory of the CPU module (host station) or the buffer memory of the intelligent function module which is mounted on the specified module position. (TO instruction)
CCPU_ToBuf_ISR	
CCPU_X_In_BitEx	To read an input signal (X) in bit (1-point) units.
CCPU_X_In_WordEx	To read an input signal (X) in word (16-point) units.
CCPU_X_In_Word_ISR	
CCPU_Y_In_BitEx	To read an output signal (Y) in bit (1-point) units.
CCPU_Y_In_WordEx	To read an output signal (Y) in word (16-point) units.
CCPU_Y_In_Word_ISR	
CCPU_Y_Out_BitEx	To output an output signal (Y) in bit (1-point) units.
CCPU_Y_Out_WordEx	To output an output signal (Y) in word (16-point) units.
CCPU_Y_Out_Word_ISR	
mdDevRstEx	To reset bit devices.*1
mdDevSetEx	To set bit devices.*1
mdRandREx	To read devices randomly.*1
mdRandWEx	To write devices randomly.*1
mdReceiveEx	To read devices in a batch.
mdSendEx	To write devices in a batch.

*1 CPU buffer memory cannot be accessed.

11.2 Label Communication Function

This function reads/writes data from/to labels stored in CPU modules of other stations.

Label communication flow



1. Acquire label assignment information (device information) of the specified label with the MELSEC data link function (mdGetLabelInfo).

2. Read/write data from/to a device based on the acquired label assignment information (device information) with the MELSEC data link functions (mdRandRLabelEx/mdRandWLabelEx).

Point

- In the label communication, a CPU module can be accessed without changing a user program by acquiring label assignment information again even if the label assignment information of the CPU module is changed.
- The label assignment information (device information) acquired with the MELSEC data link function (mdGetLabelInfo) does not need to be acquired for each MELSEC data link function (mdRandRLabelEx/mdRandWLabelEx) execution. However, if the label assignment information (device information) stored in a CPU module is changed, acquire it again with the MELSEC data link function (mdGetLabelInfo). (Otherwise, an error response is returned.)

■User program functions

The following table shows the functions used for label communication.

Function name	Description
mdGetLabelInfo	To acquire device information corresponding to label names.
mdRandRLabelEx	To read devices corresponding to labels randomly.
mdRandWLabelEx	To write devices corresponding to labels randomly.

Accessible CPU modules

The following table shows the accessible CPU modules.

Product name	Model name
Programmable controller CPU	R04CPU, R04ENCPU, R08CPU, R08ENCPU, R16CPU, R16ENCPU, R32CPU, R32ENCPU, R120CPU, R120ENCPU
Process CPU	R08PCPU, R16PCPU, R32PCPU, R120PCPU

Label types which can be referred to

The following table shows the label types that can be referred to from a C Controller module.

○: Applicable, ×: Not applicable, —: Not available

Label type	"Access from External Device" is selected or not selected.	Availability*1
Global label	Selected	○
	Unselected	×
Local label	—	
System label		

*1 The availability of the label differs depending on the device type assigned to the label.
For the device type, refer to the following:
MELSEC iQ-R C Controller Module Programming Manual

Point

For referring to a label, select "Access from External Device" in GX Works3.
(GX Works3 Operating Manual)

11.3 Data Analysis Function

This function performs data analysis processing such as fast Fourier transform, digital filter operation, calculation of a cross point between a wave and a specified value, and calculation of a standard deviation.
This function enables the detection of machining errors by monitoring current wave and the preventive maintenance of devices by analyzing vibrations.
For data analysis functions and statistical analysis functions, refer to the following:
MELSEC iQ-R C Controller Module/C Intelligent Function Module Programming Manual (Data Analysis)

Function list

The following table shows the functions used for the data analysis function.

■Data analysis function

Function name	Description
DANL_SetOpCondition	To set operating conditions for data analysis.
DANL_GetOpCondition	To acquire operating conditions for data analysis.
DANL_DigitalFilter	To perform digital filter operation for the specified wave.
DANL_EnvelopeCalculation	To calculate the envelope of the specified wave.
DANL_FFTSpectrum	To perform spectrum calculation using fast Fourier transform (FFT) for the specified wave.
DANL_FindCrossPoint	To calculate the number of cross points of the specified wave and a reference value for the number of cross points specified to the maximum number of cross points.
DANL_Peak	To calculate the peak values (maximum and minimum) of the specified wave.
DANL_RMS	To calculate an RMS (root mean square) of the specified wave.
DANL_BoundCompareTest	To compare the specified wave and a check value to check an upper/lower limit.
DANL_AryBoundCompareTest	To compare the specified wave and a check value to check an upper/lower limit of the wave.

■Statistical analysis function

Function name	Description
DANL_LeastSquare	To calculate a coefficient and a constant of a polynomial, and a multiple correlation coefficient by using a least-squares method for the specified array.
DANL_MovingAverage	To calculate a moving average of the specified array.
DANL_StandardDeviation	To calculate a standard deviation of the specified array.
DANL_Variance	To calculate a variance of the specified array.
DANL_MTUnit	To determine a unit space that is used in the MT method based on the specified normal data.
DANL_MTMahalanobisDistance	To calculate a Mahalanobis distance of the specified input data.
DANL_MultipleRegression	To calculate a coefficient, constant, and regression statistics for multiple regression analysis.

11.4 Interrupt Function to a C Controller Module

This function executes a routine registered with the C Controller module dedicated function (CCPU_EntryInt) as an interrupt routine when an interrupt request is issued to a C Controller module from an input module, interrupt module, intelligent function module, and another CPU module.
With the C Controller module dedicated function (CCPU_WaitUnitEvent), a user program being executed is placed into the state of waiting for an interrupt event so that the user program can be restarted when an interrupt request is issued from an input module, interrupt module, or intelligent function module. (☞ Page 90 Restarting user programs)

Point

When executing an interrupt request, module parameters of each module need to be set.
For details on the setting, refer to the user's manual for each module.

Interrupt pointer number and interrupt factor

The following table shows the interrupt factor for each interrupt pointer number.

Factor	Interrupt pointer number	Description
Interrupt from module	I0 to I15	A pointer used in the module which has an interrupt function
Inter-module synchronous interrupt	I44	A fixed cycle interrupt pointer used for the inter-module synchronization function
Multiple CPU synchronous interrupt	I45	A fixed cycle interrupt pointer used for the multiple CPU synchronization function
Interrupt from module	I50 to I1023	A pointer used in the module which has an interrupt function

For details on an interrupt program using interrupt pointers (I44, I45), refer to the following:
☞ Page 123 Inter-module synchronous interrupt
☞ Page 248 Multiple CPU synchronous interrupt

Interrupt priority

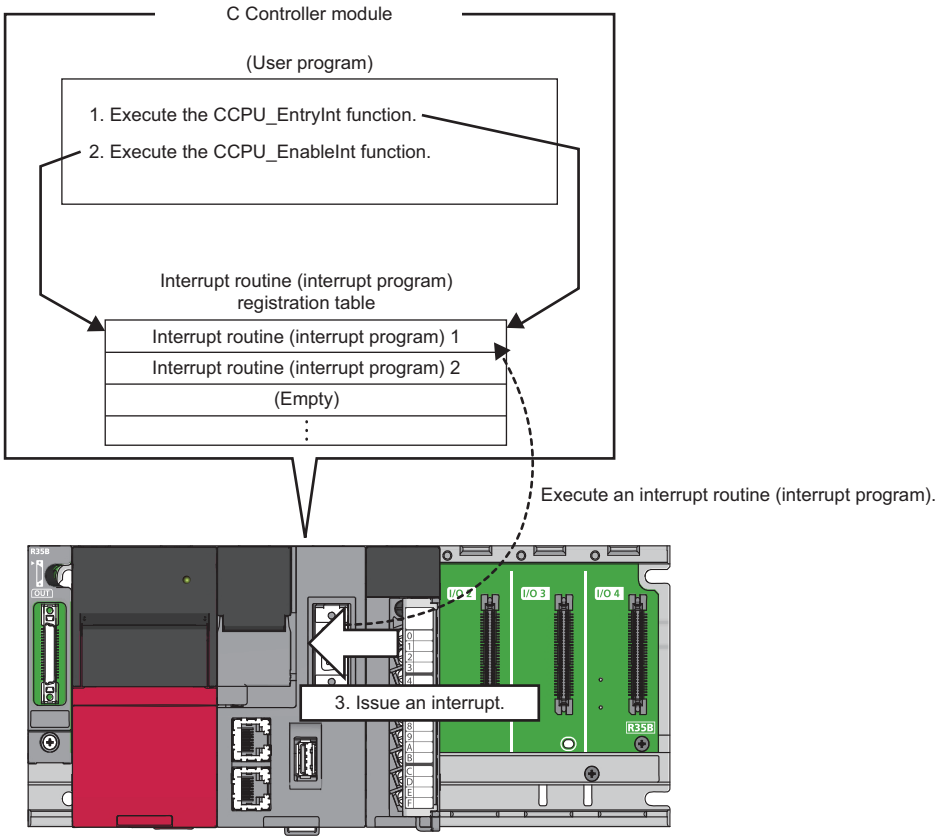
The priority has been set to each interrupt pointer. If the priority of an interrupt program of which execution conditions have been satisfied is higher than that of an interrupt program that is currently being executed, the program is executed in accordance with the priority. If the priority of an interrupt of which execution conditions have been satisfied is lower, it will be in a wait state until an interrupt program that is currently being executed is completed.

Interrupt priority		Interrupt pointer	Execution order at simultaneous occurrence
High	3	Inter-module synchronous interrupt (I44), multiple CPU synchronous interrupt (I45)	I45 → I44
Low	8	Interrupt from module (I0 to I15, I50 to I1023)	I0 → I1 → I2 → to → I1021 → I1022 → I1023

Interrupt procedure

The following explains the interrupt procedure using interrupt from a module (I0 toI15, I50 to I1023).

Executing interrupt routines



1. With the C Controller module dedicated function (CCPU_EntryInt), register a routine (user program) to be called as an interrupt routine (interrupt program) when an interrupt is requested.
2. Enable the registered interrupt routine (interrupt program) with the C Controller module dedicated function (CCPU_EnableInt). If it is disabled, the interrupt routine (interrupt program) will not be executed.
3. When an interrupt request is issued from the module, the interrupt routine (interrupt program) is executed.

Point

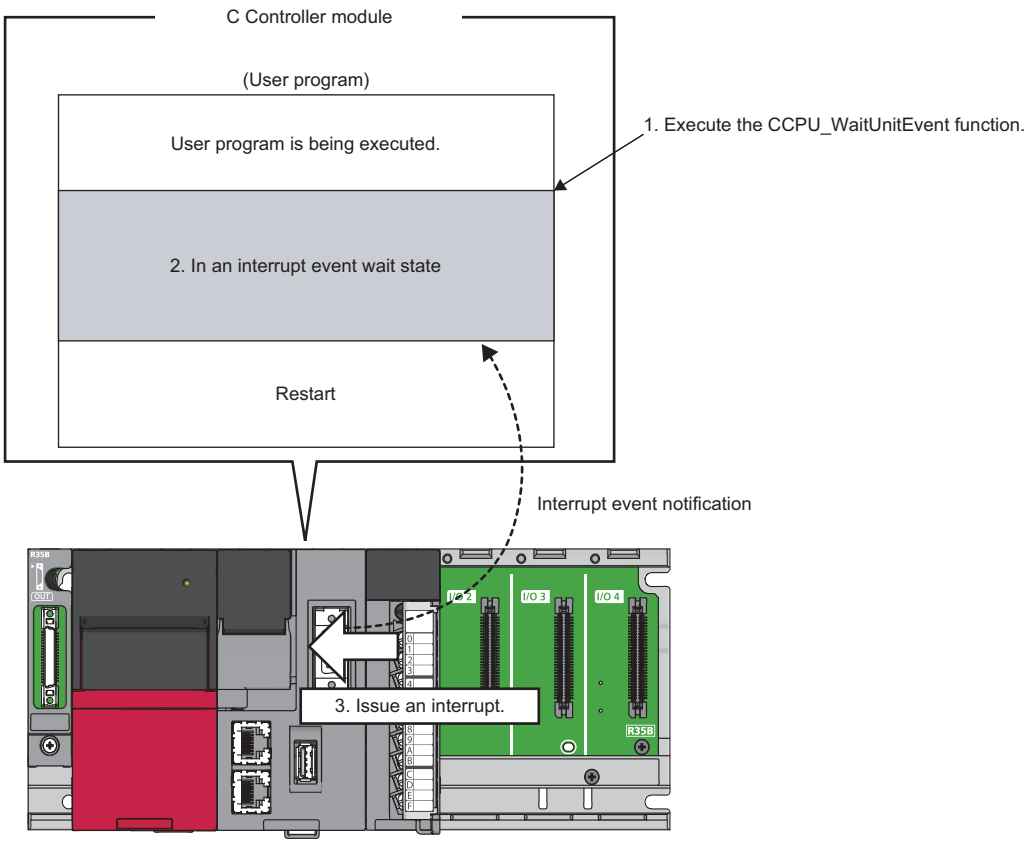
When an interrupt request is issued to the routine disabled with the C Controller module dedicated function (CCPU_DisableInt), the interrupt request is ignored.

■Function list

The following table shows the functions used for interrupting a C Controller module.

Function name	Description
CCPU_EntryInt	To register a routine to be called when an interrupt occurs.
CCPU_EnableInt	To enable the routine registered with the CCPU_EntryInt function.
CCPU_EnableInt_ISR	
CCPU_DisableInt	To disable the routine registered with the CCPU_EntryInt function.
CCPU_DisableInt_ISR	

Restarting user programs



1. Execute the C Controller module dedicated function (CCPU_WaitUnitEvent) while executing a user program.
2. The user program is placed into a state of waiting for an interrupt event notification from the module.
3. When an interrupt request is issued, the user program restarts.

■Function list

The following table shows the functions used for interrupting a C Controller module.

Function name	Description
CCPU_WaitUnitEvent	To wait for an interrupt event notification from a module.

Precautions

The following shows the considerations for using the C Controller module dedicated function (CCPU_WaitUnitEvent).

■When an interrupt event has already been notified

When an interrupt event has already been notified at the time of executing the C Controller module dedicated function (CCPU_WaitUnitEvent), a user program restarts from a state of waiting for an interrupt event at the same time as the execution of the function.

In addition, when multiple interrupt events have been notified to the same interrupt event number at the time of executing the C Controller module dedicated function (CCPU_WaitUnitEvent), a user program performs processing as a single interrupt event notification.

■When using the function in multiple user programs

Do not specify a same interrupt event (interrupt pointer number) in multiple user programs.

Otherwise, a user program to which a specified interrupt event (interrupt pointer number) is notified will be undefined.

11.5 MELSEC Data Link Function Setting Function

This function sets parameters required for using the MELSEC data link function.

Channel No.

Set a channel number to be used when opening a communication line (a channel) with the MELSEC data link function (the mdOpen function).

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ a target module ⇨ [Required Settings] ⇨ [Channel Number]

For details on channel numbers, refer to the following:

MELSEC iQ-R C Controller Module Programming Manual

Point

A bus interface channel number is not required to be set because the number is fixed to '12.'

Precautions

Make sure the set channel numbers are not duplicated.

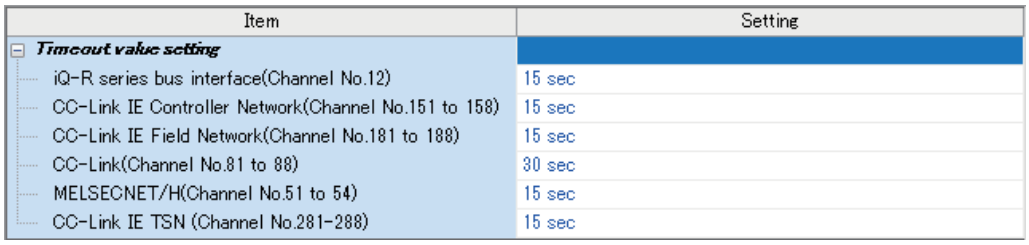
Timeout value

Set a timeout value to be used in the MELSEC data link function for each network.

A timeout value indicates the timeout value for one communication processing with an access target.

[CPU Parameter] ⇨ [MELSEC data link function settings] ⇨ [Timeout value setting]

Window



Displayed items

Item	Description	Setting range	Default
iQ-R series bus interface(Channel No.12)	Set the MELSEC data link function timeout value of the iQ-R series bus interface (Channel No.12).	1 to 360 sec (1 second units)	15 sec
CC-Link IE Controller Network(Channel No.151 to 158)	Set the MELSEC data link function timeout value of the CC-Link IE Controller Network (Channel No.151 to 158).		15 sec
CC-Link IE Field Network(Channel No.181 to 188)	Set the MELSEC data link function timeout value of the CC-Link IE Field Network (Channel No.181 to 188).		15 sec
CC-Link(Channel No.81 to 88)	Set the MELSEC data link function timeout value of the CC-Link (Channel No.81 to 88).		30 sec
MELSECNET/H(Channel No.51 to 54)	Set the MELSEC data link function timeout value of the MELSECNET/H (Channel No.51 to 54).		15 sec
CC-Link IE TSN(Channel No.281 to 288)	Set the MELSEC data link function timeout value of the CC-Link IE TSN (Channel No.281-288).		15 sec

Target settings

Set the settings for using a logical station number in the MELSEC data link function.

For details on logical station numbers, refer to the following:

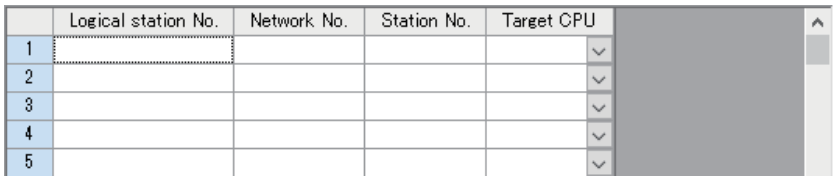
MELSEC iQ-R C Controller Module Programming Manual

Point

- An access target can be changed without changing programming.
- Other CPUs in a multiple CPU system of an access target (except for the CPU which controls an access target network module) can be accessed.

[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ a target module ⇨ [Application Settings] ⇨ [Target settings]

Window



Displayed items

—: Not set

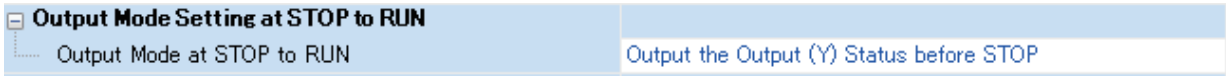
Item	Description	Setting range	Default
Local station No.	Set number of the logical station.	65 to 239	—
Network No.	Set network number.	1 to 239	
Station No.	Set station number.	0 to 120	
Target CPU	Select the target CPU.	• CPU No.1 • CPU No.2 • CPU No.3 • CPU No.4	

11.6 Output Mode Setting Function at STOP to RUN

This function sets the mode of the output (Y) when C Controller module operating status is switched from STOP to RUN.

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

Window



Displayed items

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

Output the Output (Y) Status before STOP

After the output (Y) status before the operating status turns into STOP state is output, the user program is executed.

Clear the Output (Y)

The output (Y) is turned OFF, and the output (Y) status is output after the program operations are executed.

Precautions

The following shows the considerations on outputs when changing the status from STOP to RUN after forcing the output (Y) ON at STOP status.

■"Output the Output (Y) Status before STOP" is selected

- When the output (Y) is forced ON when the operating status is STOP, the status before it stopped is output.
- If the output (Y) is OFF before entering the STOP state, the ON state is not retained.

■"Clear the Output (Y)" is selected

- When the output (Y) is forced ON when the operating status is STOP, the ON state is retained.

11.7 Boot Function of Parameters

This function reads parameters stored in an SD memory card when turning the power of a C Controller module OFF and ON, or resetting the module. By writing parameters created in an engineering tool, the boot of which is enabled, to the CPU built-in memory (data memory) and inserting an SD memory card to which other parameters are written to a C Controller module, parameters of the C Controller module can be changed without using an engineering tool. (☞ Page 94 Boot procedure)
Parameters to use (either one stored in the CPU built-in memory (data memory) or in an SD card memory) can be set in "Service Settings" of CPU parameters and in memory card parameters.

Boot target

Boot target parameters are as follows:

- System parameters
- CPU parameters
- Module parameters

Boot procedure

The following explains the procedure for booting parameters.

Operating procedure

1. Set "Enable" for the following items in "Service Settings" of CPU parameters. (☞ Page 192 Service Setting Function)
- Memory card access
- Memory card parameter execution

2. Write CPU parameters to the CPU built-in memory (data memory).

Point

Data stored in the CPU built-in memory is read by setting the "Service Settings" of CPU parameters. To use the boot function of parameters, settings for "Service Settings" must be written to the CPU built-in memory (data memory) in advance.

3. Insert an SD memory card to a C Controller module.
4. Specify parameters to be booted in memory card parameters (☞ Page 95 Memory card parameters)
5. Write memory card parameters and parameters to be booted on the SD memory card.
6. Turn the power OFF and ON, or reset the C Controller module.

Precautions

- When either of "Memory card access" or "Memory card parameter execution" is set to "Disable," parameters of the CPU built-in memory (data memory) will be read. To read parameters in the SD memory card, set both of the items to "Enable."
- If a file specified in "Boot File Setting" does not exist in the SD memory card, an error will occur.

Memory card parameters

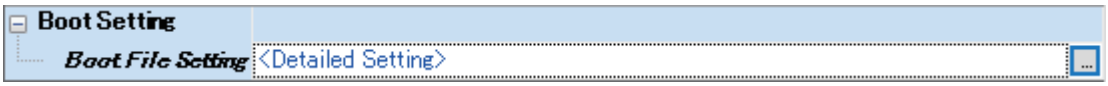
■Boot setting

Set parameters to be booted.

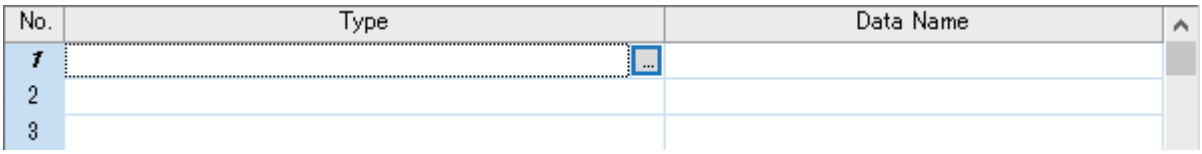
[Memory Card Parameter] ⇨ [Boot Setting]

Operating procedure

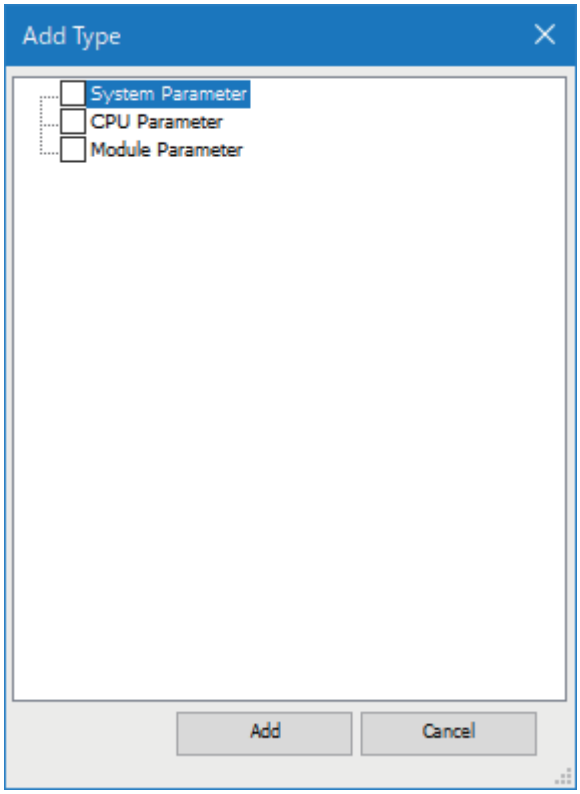
1. Double-click "<Detailed Setting>" in "Boot File Setting."



2. Click on the "Type" column.



3. Select the type for the boot file (available to select two or more parameters), and click [Add] button.



Displayed items

Item	Description	Setting range	Default
Boot File Setting	Set boot transfer target file from memory card.	• System Parameter • CPU Parameter • Module Parameter	—

■Enabling/disabling the use of file/data on memory card

Set whether to use files/data stored on a memory card.

[Memory Card Parameter] ⇨ [Setting of File/Data Use or Not in Memory Card]

Window



Displayed items

Item	Description	Setting range	Default
Module Extended Parameter	Set whether to 'Use' or 'Not Use' module extension parameter stored in memory card. Set 'Use' when using the module extended parameter stored in memory card. Operates with the module extended parameter stored in CPU built-in memory when 'Not Use' is set. Module extension parameter is the target.	• Not Use • Use	Not Use
Device Station Parameter	Set whether to 'Use' or 'Not Use' the parameters of device station saved in memory card. To use the parameters of device station saved in memory card, set 'Use.' To use the parameters of device station saved in CPU built-in memory, set 'Not Use.' Set parameters of device station in Network Configuration Settings of CC-Link IE TSN.	• Not Use • Use	Not Use

11.8 Remote Operation Function

This function controls the operating status of a C Controller module remotely.
It can be performed with CW Configurator, a user program, or SLMP.
The following types of remote operation are available:

- Remote RUN/STOP
- Remote PAUSE
- Remote RESET

Remote RUN/STOP

Change the C Controller module status to RUN/STOP externally while the RESET/STOP/RUN switch is on the RUN position. Use this function to change the C Controller module status to RUN/STOP when the module is located in an inaccessible place such as in the control panel by using an external signal.

Executing remote operations

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

■Using CW Configurator

Refer to the following:
CW Configurator Operating Manual

■Using a user program

Execute the C Controller module dedicated function (CCPU_Control) to perform remote RUN/STOP.
MELSEC iQ-R C Controller Module Programming Manual

■Using SLMP

Refer to the following:
SLMP Reference Manual

Precautions

Since a C Controller module gives priority to STOP, observe the following considerations.

■STOP timing

The C Controller module is stopped when remote STOP is performed from either of a user program or CW Configurator.

■To change the remote STOP state to RUN again

Perform remote RUN.

■Output (Y) status during remote STOP

When "Output Mode Setting at STOP to RUN" is set to "Output the Output (Y) Status before STOP" in CW Configurator, the output (Y) state is output when the state is changed from RUN to STOP at remote RUN.

■User program during remote STOP

The user program does not stop the execution even if the C Controller module is placed into the remote STOP state. To change the processing according to the operating status of the C Controller module, use the C Controller module dedicated function (CCPU_GetCpuStatus) in the program.

Remote PAUSE

Change the C Controller module status to PAUSE externally while the RESET/STOP/RUN switch is on the RUN position. Use this function to retain the output (Y) of the C Controller module in the RUN state in such system as process control.

Executing remote operations

The following methods are available to execute remote PAUSE:

■Using CW Configurator

Refer to the following:

CW Configurator Operating Manual

■Using a user program

Execute the C Controller module dedicated function (CCPU_Control) to perform remote PAUSE.

MELSEC iQ-R C Controller Module Programming Manual

■Using SLMP

Refer to the following:

SLMP Reference Manual

Precautions

The user program does not stop executing even if the C Controller module is in the remote PAUSE state. To change the processing according to the operating status of the C Controller module, use the C Controller module dedicated function (CCPU_GetCpuStatus) in the program.

Remote RESET

Reset the C Controller module externally while the module is in the STOP state (including the case in which the module is stopped due to an error). Even when the RESET/STOP/RUN switch is on the RUN position, the C Controller module can be reset while the module is in STOP state.

Enabling remote RESET

Before performing remote RESET, enable the remote RESET.

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

Window



Displayed items

Item	Description	Setting range	Default
Remote Reset	Set whether to enable remote RESET.	• Disable • Enable	Disable

Executing remote operations

The following methods are available to execute remote RESET:

■Using CW Configurator

Refer to the following:

CW Configurator Operating Manual

■Using a user program

Execute the C Controller module dedicated function (CCPU_Reset) to perform remote RESET.

MELSEC iQ-R C Controller Module Programming Manual

■Using SLMP

Refer to the following:

SLMP Reference Manual

Precautions

■Before performing remote RESET

Close each user program in the C Controller module before performing remote RESET. Performing remote RESET while the user program is being operated may damage the user program and data files.

■Status after the completion of remote RESET

After performing the remote RESET operation, the operating status of the C Controller module or programmable controller CPU will be in the status set with the switch.

■Remote RESET at error stop

If remote RESET is performed when a C Controller module in a single CPU system or CPU No.1 in a multiple CPU system has stopped due to an error, the C Controller module or programmable controller CPU will be in the status set with the switch.

■Remote RESET with CW Configurator

After the remote RESET is performed, the communication between CW Configurator and C Controller module will be disconnected. In this case, reconnect the connection with CW Configurator.

■Remote RESET when CPU No.1 is a programmable controller CPU in a multiple CPU system

For remote RESET of programmable controller CPU, refer to the manual for the programmable controller CPU to be used.

■The host CPU is other than bus master CPU (CPU No.1)

- The parameter ("Enable" is set to "Remote Reset") is set in the bus master CPU (No.1): (Unset: Error)
- The bus master CPU (No.1) is STOP: (CPU is RUN/PAUSE: Error)
- The bus master CPU (No.1) is a programmable controller CPU

Point

The programmable controller CPU (bus master CPU (CPU No.1) cannot be reset when other peripherals (such as GX Works3) performs remote STOP to a programmable controller CPU (bus master CPU (CPU No.1)). Therefore, cancel the remote STOP with the peripheral device performing the remote STOP. Then, place the programmable controller CPU (bus master CPU (CPU No.1)) into STOP state with its RESET/STOP/RUN switch, the remote operation with CW Configurator, or a user program (the mdControl function) of the C Controller module.

- To perform remote RESET with CW Configurator, change the C Controller module status to STOP using CW Configurator.
- To perform remote RESET with a user program, change the C Controller module status to STOP using a user program.

Remote operation and operating status of a C Controller module

The following shows the relationship between remote operations and operating status changes of a C Controller module.

Status change by switch operation and at stop error

The following table shows the operating status of a C Controller module by switch operation and at a stop error.

—: No status change

Before changing the operating status			After changing the operating status		
Factor to determine CPU operating status	Operating status	RESET/STOP/RUN switch status	RESET/STOP/RUN switch operation		Stop error occurred
			RUN	STOP	
RESET/STOP/RUN switch operation	STOP	STOP	RUN	—	—
	RUN	RUN	—	STOP	STOP
Operation with CW Configurator or SLMP	STOP	STOP	—	—	—
	PAUSE (Actual status: STOP)	STOP	PAUSE	—	—
	RUN (Actual status: STOP)	STOP	RUN	—	—
	STOP	RUN	—	—	—
	PAUSE	RUN	—	STOP	STOP
	RUN	RUN	—	STOP	STOP
Execution of the C Controller module dedicated function	STOP	STOP	—	—	—
	PAUSE (Actual status: STOP)	STOP	PAUSE	—	—
	RUN (Actual status: STOP)	STOP	RUN	—	—
	STOP	RUN	—	—	—
	PAUSE	RUN	—	STOP	STOP
	RUN	RUN	—	STOP	STOP
Stop error occurred	STOP	STOP	— (Error STOP)	—	—
	STOP	RUN	—	— (Error STOP)	—

Status change by remote operations with CW Configurator and SLMP

The following table shows the operating status of a C Controller module by remote operations with CW Configurator and SLMP.

—: No status change

Before changing the operating status			After changing the operating status			
Factor to determine CPU operating status	Operating status	RESET/STOP/RUN switch status	Remote operation with CW Configurator and SLMP			
			RUN	STOP	PAUSE	RESET*1
RESET/STOP/RUN switch operation	STOP	STOP	—	—	—	RESET
	RUN	RUN	—	STOP	PAUSE	—
Operation with CW Configurator or SLMP	STOP	STOP	—	—	—	RESET
	PAUSE (Actual status: STOP)	STOP	—	—	—	RESET
	RUN (Actual status: STOP)	STOP	—	—	—	RESET
	STOP	RUN	RUN	—	PAUSE	RESET
	PAUSE	RUN	RUN	STOP	—	—
	RUN	RUN	—	STOP	PAUSE	—
Execution of the C Controller module dedicated function	STOP	STOP	—	—	—	RESET
	PAUSE (Actual status: STOP)	STOP	—	—	—	RESET
	RUN (Actual status: STOP)	STOP	—	—	—	RESET
	STOP	RUN	RUN	—	PAUSE	RESET
	PAUSE	RUN	RUN	STOP	—	—
	RUN	RUN	—	STOP	PAUSE	—
Stop error occurred	STOP	STOP	— (Error STOP)	— (Error STOP)	— (Error STOP)	RESET
	STOP	RUN	— (Error STOP)	— (Error STOP)	— (Error STOP)	RESET

*1 Setting "Remote Reset Setting" to "Enable" of the CPU parameters is required.

Status change at the execution of the C Controller module dedicated function

The following table shows the operating status of a C Controller module at the execution of the C Controller module dedicated function (CCPU_Control).

—: No status change

Before changing the operating status			After changing the operating status			
Factor to determine CPU operating status	Operating status	RESET/STOP/RUN switch status	Execution of the C Controller module dedicated function			
			RUN	STOP	PAUSE	RESET*1
RESET/STOP/RUN switch operation	STOP	STOP	—	—	—	RESET
	RUN	RUN	—	STOP	PAUSE	—
Operation with CW Configurator or SLMP	STOP	STOP	—	—	—	RESET
	PAUSE (Actual status: STOP)	STOP	—	—	—	RESET
	RUN (Actual status: STOP)	STOP	—	—	—	RESET
	STOP	RUN	RUN	—	PAUSE	RESET
	PAUSE	RUN	RUN	STOP	—	—
	RUN	RUN	—	STOP	PAUSE	—
Execution of the C Controller module dedicated function	STOP	STOP	—	—	—	RESET
	PAUSE (Actual status: STOP)	STOP	—	—	—	RESET
	RUN (Actual status: STOP)	STOP	—	—	—	RESET
	STOP	RUN	RUN	—	PAUSE	RESET
	PAUSE	RUN	RUN	STOP	—	—
	RUN	RUN	—	STOP	PAUSE	—
Stop error occurred	STOP	STOP	— (Error STOP)	— (Error STOP)	— (Error STOP)	RESET
	STOP	RUN	— (Error STOP)	— (Error STOP)	— (Error STOP)	RESET

*1 Setting "Remote Reset Setting" to "Enable" of the CPU parameters is required.

11.9 Clock Function

This function sets the time for the built-in clock of a C Controller module.
The clock data is used to manage the time for the functions performed by the system such as date for the event history.

Point

- The clock is running continuously using the internal battery of the C Controller module while the power of the module is OFF or a power failure longer than the allowable momentary power failure time occurred.
- For the time stamp of a file, the time of the operating system is used. Since the time of the C Controller module built-in clock and that of operating system may differ, correct the time of the operating system using a user program.

Clock data setting

The following shows the method for setting clock data.

Clock data change

The clock data can be set by either of the following methods:

- CW Configurator
- C Controller module dedicated function

Point

When the clock data is changed, the following operations are performed:

- The millisecond clock is reset to '0.'
- "Clock setting" (event code: 24000) is saved in the event history file.

■Using CW Configurator

[Online] ⇨ [Set Clock]

■With a C Controller module dedicated function

Write the clock data with the C Controller module dedicated function (CCPU_SetRTC).

Point

A C Controller module sets the time of its built-in clock to that of the operating system at the startup of the module after turning the power ON or resetting it. Use a user program in order to set the time for a running operating system.

Clock data reading

Read the clock data with the C Controller module dedicated function (CCPU_GetRTC).

Precautions

■When using this function for the first time

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

■When modifying the clock data

Even if changing a portion of the clock data, be sure to write all data to the C Controller module again.

■When changing clock data with a user program

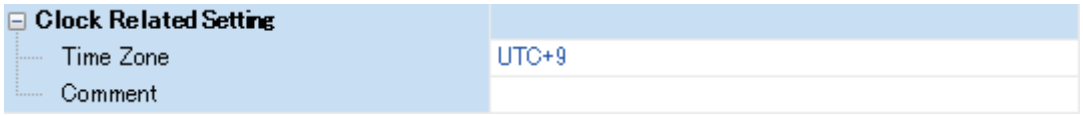
To change the clock data with a user program, be sure to use the C Controller module dedicated function (CCPU_SetRTC). If other clock data setting function is used, the accurate clock data will not be set in the C Controller module.

Time zone setting

Set a time zone used in a C Controller module. Specifying the time zone enables the clock in the C Controller module 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	Set a time zone used in a C Controller module.	• UTC+13 • UTC+12 • UTC+11 • UTC+10 • UTC+9:30 • UTC+9 • UTC+8 • UTC+7 • UTC+6:30 • UTC+6 • UTC+5:45 • UTC+5:30 • UTC+5 • UTC+4:30 • UTC+4 • UTC+3:30 • UTC+3 • UTC+2 • UTC+1 • UTC • UTC-1 • UTC-2 • UTC-3 • UTC-3:30 • UTC-4 • UTC-4:30 • UTC-5 • UTC-6 • UTC-7 • UTC-8 • UTC-9 • UTC-10 • UTC-11 • UTC-12	UTC+9
Comment	Enter a comment such as a name of a city for the time zone.	1 to 32 characters	—

Point

- To apply the time zone setting on a C Controller module, turn the power OFF and ON or reset the module after writing parameters.
- In a multiple CPU system, the time zone set in the CPU No.1 is used for other CPU modules. If the time zone is set to the CPUs from No.2 to No.4, the setting is not applied.

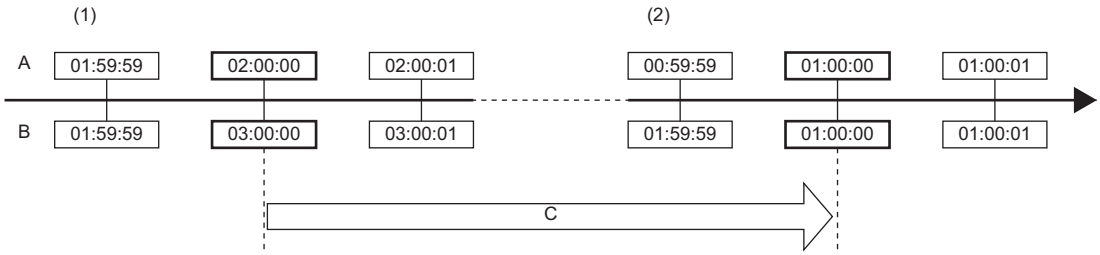
Precautions

Once the module is initialized, the parameters will be restored to the default setting even when overseas time zone has been set. Changing the clock of the module is required when the time zone is set again.

Daylight saving time function

The time of a C Controller module can be adjusted to daylight saving time.
This function advances the time of a C Controller module by one hour on the start date and time, and reverses it by one hour on the end date and time.

- When daylight saving time starts at 2:00 on the second Sunday in March (1), and ends at 2:00 on the first Sunday in November (2)



A: Before adjustment
B: After adjustment
C: Daylight saving time

Point

Before using daylight saving time, check the version of a C Controller module and an engineering tool.
(☞ Page 404 Added and Changed Functions)

Timing of daylight saving time adjustment

Daylight saving time is adjusted at the following timing:

- At the start and end of daylight saving time
- Turn the power OFF and ON, or reset the C Controller module

Operation check of the daylight saving time function

The operation of the daylight saving time function can be checked by the following:

■C Controller module dedicated function

Whether the date lies inside or outside the daylight saving time period can be checked with the C Controller module dedicated function (CCPU_GetRTC).

■Event history

The history for the start and end of daylight saving time can be checked in the event history for the date set in the daylight saving time setting.

Operation of other functions using clock data

The following table shows the operation of the functions using clock data in a C Controller module during the daylight saving time period.

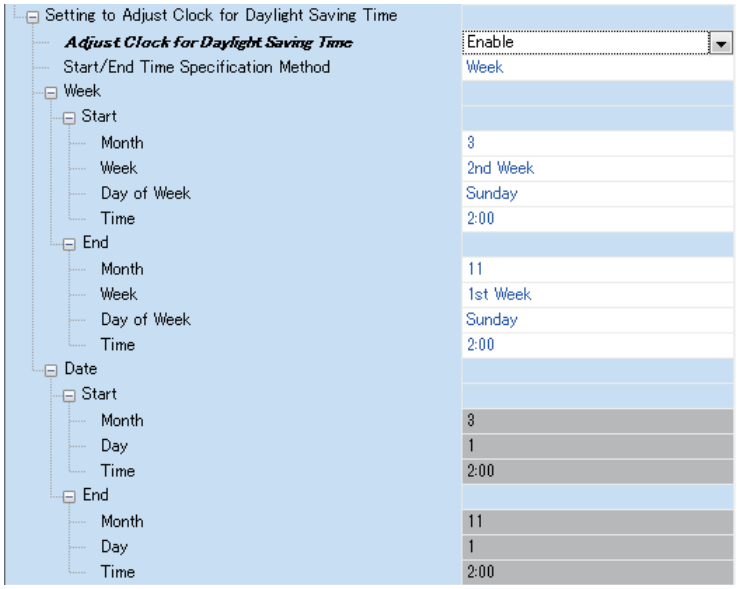
Item	Description
Clock data reading	Read clock data after adjusting the time for daylight saving time.
Clock data writing	Write as clock data after adjusting the time for daylight saving time.

Daylight saving time settings

Set a start date and time, and an end date and time for daylight saving time.

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

Window



Displayed items

Item			Description	Setting range	Default
Adjust Clock for Daylight Saving Time			Set whether to enable the daylight saving time setting.	• Enable • Disable	Disable
Start/End Time Specification Method			Set the timing for switching to daylight saving time to the specified week or the specified date.	• Week • Date	Week
Week	Start	Month	Set a date and time to start daylight saving time.	1 to 12	3
		Week		The Last Week, 1st Week to 4th Week	2nd Week
		Day of Week		Sunday to Saturday	Sunday
		Time		0:00 to 23:00	2:00
	End	Month	Set a date and time to end daylight saving time.	1 to 12	11
		Week		The Last Week, 1st Week to 4th Week	1st Week
		Day of Week		Sunday to Saturday	Sunday
		Time		0:00 to 23:00	2:00
Date	Start	Month	Set a date and time to start daylight saving time.	1 to 12	3
		Day		The Last Date, 1 to 31	1
		Time		0:00 to 23:00	2:00
	End	Month	Set a date and time to end daylight saving time.	1 to 12	11
		Day		The Last Date, 1 to 31	1
		Time		0:00 to 23:00	2:00

Precautions

- The clock data cannot be changed to data less than one hour from the start date and time of daylight saving time. When the daylight saving time setting is enabled for the CPU No.1 in a multiple CPU system, the clock data in the CPU No.2 to 4 also cannot be changed in the same way.
- For the period less than one hour from the start date and time or less than one hour until the end date and time of daylight saving time, the function that is triggered by time may not perform or may perform twice.
- Functions perform based on date information after adjustment for daylight saving time when the daylight saving time setting is enabled, and therefore there are times when the date output by functions using clock data is (before adjustment) ≥ (after adjustment), and the order (No.) in which events occur and the sorting order for the date on which events occur do not match. Consequently, when checking output results in chronological order, sort not in the order for the date on which events occur, but in the order (No.) in which events occur.

11.10 RAS Function

This section explains the RAS function of a C Controller module.

Watchdog timer (WDT) function

This function monitors and detects errors on hardware and user programs by using an internal timer of a C Controller module (system watchdog timer and user watchdog timer).

Item	Description
System watchdog timer	Used to detect an error in hardware and system software of a C Controller module. If system processing is suspended for a long time because of a hardware failure, interrupt program execution, etc., a timeout will occur.
User watchdog timer	Used to detect an error in a user program. If a user program cannot complete processing within the time specified with the C Controller module dedicated function (CCPU_StartWDT), and also cannot reset with the C Controller module dedicated function (CCPU_ResetWDT), a timeout will occur.

Timeout of watchdog timer

When the watchdog timer times out, an error indicating that the monitoring time set in the watchdog timer setting has been exceeded (watchdog timer error) occurs. If a user watchdog timer error occurs, the BUS RUN LED turns OFF and the ERROR LED starts flashing. If a system watchdog timer error occurs, the READY LED turns OFF.

Precautions

When using the following functions, a user watchdog timer error occurs easily since the CPU utilization by a system task with high priority increases.

- Shell command
- Connection with CW Workbench and Wind River Workbench
- Mounting and unmounting an SD memory card
- Unmounting the USB Mass Storage Class-compliant device
- File access
- Ethernet communication
- NFS server communication

Monitoring time setting and reset

The following shows the setting and resetting method for the monitoring time for the watchdog timer.

■System watchdog timer

Set a monitoring time for the system watchdog timer within the range of 20 to 2000 ms (in 10 ms units).

The system of a C Controller module resets the system watchdog timer while executing fixed cycle processing.

[CPU Parameter] ⇨ [RAS Setting] ⇨ [WDT (Watchdog Timer) Setting]

Window



Displayed items

Item	Description	Setting range	Default
Monitoring time	Set the execution monitoring time to the system WDT.	20 to 2000 ms (10 ms units)	1000 ms

■User watchdog timer

Set a monitoring time for the user watchdog timer within the range of 100 ms to 10000 ms (in 10 ms units) with the C Controller module dedicated function (CCPU_StartWDT).

Monitoring starts with the C Controller module dedicated function (CCPU_StartWDT), and the monitoring time is reset with the C Controller module dedicated function (CCPU_ResetWDT).

Self-diagnostics function

This function diagnoses whether there are any errors in a C Controller module.

Self-diagnostics timing

If an error occurred when the C Controller module is powered ON or while it is in the RUN/STOP state, the detected error information is displayed, and the operation is stopped.

Point

However, depending on the error occurrence status or the instruction to execute, the C Controller module may not be able to detect the error. Configure safety circuits external to the system to ensure that the entire system operates safely even in such a case.

Error checking methods

The following shows the error checking methods.

■Checking with special relays or special registers

When C Controller module detects any error, the special relays (SM0, SM1) turns ON, and an error code corresponding to the error is stored in the special register (SD0). If more than one error is detected, the latest error code is stored in the special register (SD0). Use the special relay (SM0, SM1) and special register (SD0) in the program for the C Controller module or mechanical interlock. In addition, up to 16 error codes (latest errors occurred on the system) will be stored in the special registers (SD10 to SD25). (The error codes for the 17th and onwards will not be stored.)

■Checking with CW Configurator

The error occurrence of the entire system, latest errors occurred, and event history can be checked in the module diagnostics screen. (📖CW Configurator Operating Manual)

- In the C Controller module, maximum 16 latest errors occurred can be displayed. In addition, if an additional error occurs after a stop error, the error information is not updated.
- Error logs can be checked using the event history function. (☞ Page 116 Event history function)

Point

The maximum number of continuation errors and stop errors to be displayed is 15 and 2 respectively. If already 15 continuation errors have been displayed, then next continuation error will not be displayed. If the error of the same code has been displayed, the date and time of occurrence and the detailed information about the relevant error is updated.

Operation at error detection of an intelligent function module

If an error is detected with the self-diagnostics function, a C Controller module operates in accordance with the setting of "CPU Module Operation Setting at Error Detected" as follows. (☞ Page 112 Operation setting when an error is detected in an intelligent function module)

■When "Stop" is selected in "CPU Module Operation Setting at Error Detected"

When an error is detected, the operation differs depending on the output mode settings at error in the module parameter of each module.

- When "Clear" is set: Output to the corresponding module is turned OFF.
- When "Hold" is set: Output to the corresponding module is retained.

Point

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

■When "Continue" is selected in "CPU Module Operation Setting at Error Detected"

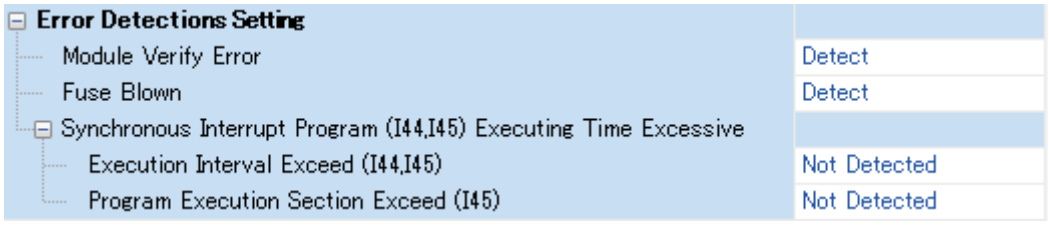
The operation of the C Controller module does not stop.

Error detection setting

Set whether to detect errors.

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

Window



Displayed items

Item		Description	Setting range	Default
Module Verify Error*1		Set whether to detect the module verification error.	• Detect • Not Detected	Detect
Fuse Blown		Set whether to detect the fuse blown in the controlled module.		
Synchronous Interrupt Program (I44, I45) Executing Time Excessive	Execution Interval Exceed (I44, I45)	Set whether to detect the excessive execution time (I44, I45).		Not Detected
	Program Execution Section Exceed (I45)	Set whether to detect the program execution section exceed errors (I45).		

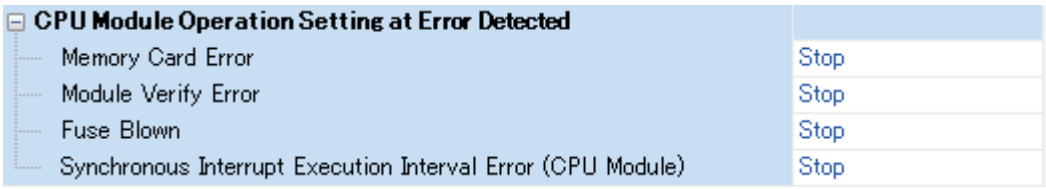
*1 If an operating module in which "Not Detected" is selected in the parameter is removed, a module verification error is not detected; however, a stop error may occur if accessing the removed module by using programs, etc. Since the removed module is not accessible even if it is mounted again, accessing the module by changing the module status from STOP to RUN may result in a stop error.

Operation setting when an error is detected

Set the operation of a C Controller module when an error is detected.

[CPU Parameter] ⇨ [RAS Setting] ⇨ [CPU Module Operation Setting at Error Detected]

Window



Displayed items

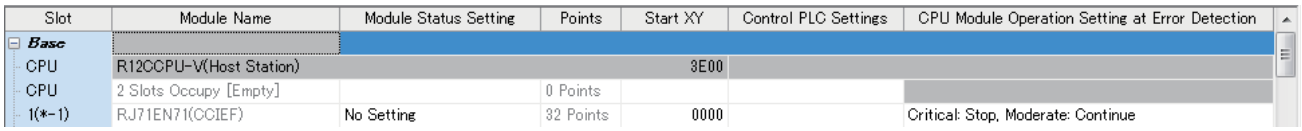
Item	Description	Setting range	Default
Memory Card Error	Set the C Controller module operation upon a memory card error.	• Stops • Continues	Stops
Module Verify Error	Set the C Controller module operation upon a module verification error.		
Fuse Blown	Set the C Controller module operation upon fuse blown error.		
Synchronous Interrupt Execution Interval Error (CPU Module)	Set the operation of CPU module at a synchronous interrupt execution interval error.		

Operation setting when an error is detected in an intelligent function module

The operation of a C Controller module when an error occurred in an intelligent function module will be determined with the parameter setting of "CPU Module Operation Setting at Error Detected" under "I/O Assignment Setting."

[System Parameter] ⇨ [I/O Assignment Setting] ⇨ [I/O Assignment Setting]

Window



Displayed items

Item	Description	Setting range	Default
CPU Module Operation Setting at Error Detected	Set the operation of a C Controller module at the detection of a major error or moderate error in the configured module.	• Critical: Stop, Moderate: Continue • Critical: Stop, Moderate: Stop • Critical: Continue, Moderate: Continue	Critical: Stop, Moderate: Continue

LED display setting

Sets whether to turn the ERROR LED ON/OFF.

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

Window



Displayed items

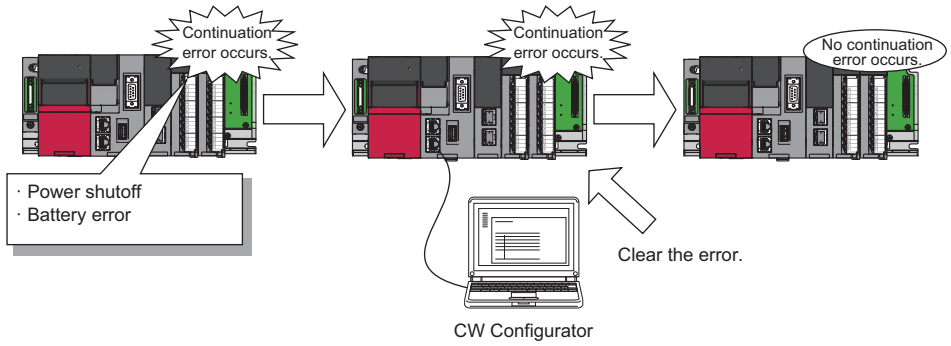
Item		Description	Setting range	Default
ERROR LED	Minor Error (Continue Error)	Set whether to 'display' or 'do not display' ERROR LED when minor error (continue error) was detected.	• Display • Do Not Display	Display

Operation setting when a stop error occurred in a multiple CPU function

Set whether to stop all CPUs when a major or moderate error occurs on each CPU in a multiple CPU configuration. (☞ Page 231 Stop setting)

Error clear function

This function clears all the existing continuation errors in the system of a C Controller module in a batch.



Error that can be cleared

Only the following continuation errors can be cleared.

Error code	Error name
1000H	Power interruption
1080H	ROM write count error
1100H	Memory card access error
1120H	SNTP clock setting error
1124H	Default gateway/gateway IP address error
1128H	Own node port number error
1129H	Open specification port number error
112DH	Specified IP address error
112EH	Connection establishment failed
1133H	Socket communications response send error
1134H	TCP connection timeout
1152H	IP address error
1155H	Connection number acquisition error
1157H	Receive buffer securement error
1165H	UDP/IP send failed
1166H	TCP/IP send failed
1167H	Unsend data send error
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
1830H	Receive queue full
1831H	Receive processing error
1832H	Transient data error
1840H	Memory Card Error
1843H	Internal battery failure
1846H	Refresh cycle exceeded
2120H, 2121H	Memory Card Error
2400H, 2401H	Module Verify Error
2420H	Fuse blown error
2441H, 2442H	Module major error
2450H	Detection of module major error
2461H, 2462H	Another CPU module major error
2470H	Another CPU module major error
2610H	Inter-module synchronization signal error

Error code	Error name
2630H	Multiple CPU synchronization signal error

Method for clearing error

The following shows how to clear errors.

■Using CW Configurator

Clear errors by performing the module diagnostics in CW Configurator. (CW Configurator Operating Manual)

■Using user programs

1. Check the continuation error detected with the C Controller module dedicated function (CCPU_GetErrInfo).
2. Clear the cause of the currently detected continuation errors.
3. Execute the C Controller module dedicated function (CCPU_ClearError).

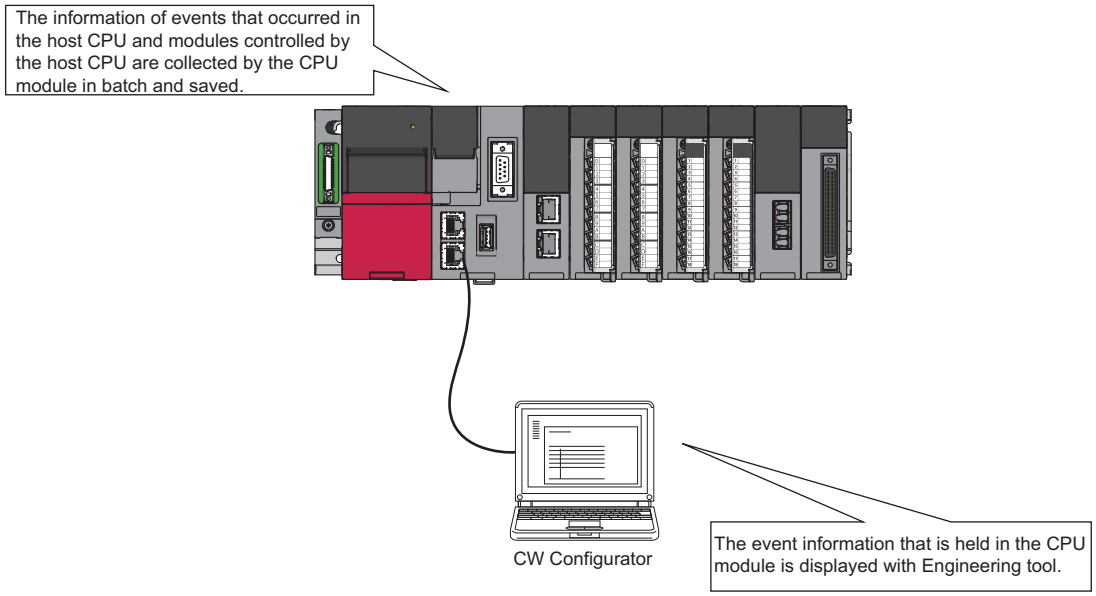
Precautions

- Since this function clears all the detected continuation errors at once, unintended errors may also get cleared.
- This function does not remove the cleared errors from the event history.
- Any errors occurred in a module other than the C Controller module cannot be cleared by this function.

Event history function

This function collects information from each module including errors detected by a module, operations performed for a module, and errors occurred on the network and saves them. The saved information can be checked in chronological order of occurrence.

Using this function enables to investigate the problems that have occurred in the equipment/devices, check the update related to control data in a C Controller system, and detect any unauthorized access.



Point

The event history information is constantly collected regardless of the operating state of the C Controller module. In some cases; however, the event history information may not be collected due to a major error in a module, a base unit error, a cable failure, or others.

Event history settings

Under normal circumstances, the event history function can be used with its default settings and doesn't need to be manually configured. The storage memory and file size of the event history file can be changed if desired.

[CPU Parameter] ⇨ [RAS Setting] ⇨ [Event History Setting]

Window



Displayed items

Item	Description	Setting range	Default
Save Destination	Set save destination of event history file. Set the 'memory card' to set each file save volume which exceed the volume of CPU Built-in memory (data memory).	• Data Memory • Memory Card	Data Memory
Set Save Volume of Per File	Set the capacity to save per event history file. Setting more capacity enables you to save more events.	1 to 2048 K Byte	128 K Byte

■Registration from a user program

Event logs can be registered from a user program with the C Controller module dedicated function (CCPU_RegistEventLog). In this case, the event type will be "Operation."

Saving event history

■Modules from which event history information is collected

Event history information is collected from the C Controller module and other modules mounted on the same base unit (such as the main base unit and 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. For more information including the coverage of event history collection regarding devices on the network, refer to the manual for each module. For a multiple CPU system, each CPU logs only events detected on the modules under its control.

■Events to be saved

The detailed information such as the operation initiator information are saved for troubleshooting purpose when the event history is saved. For the event to be saved as the event history by C Controller module, refer to the event list. (☞ Page 116 Event history function)

Event history file

The storage destination memory and file size for event history files can be changed in the event history setting. (☞ Page 117 Event history settings)

■Storage destination memory

The storage destination memory is set to either the data memory or SD memory card.

When an SD memory card is selected, disable the write protect switch of the SD memory card. If it is enabled, an event history will not be stored. (Reading the event history file in the SD memory card using CW Configurator is possible.)

If the write protect switch of an SD memory card is enabled, the write error will occur, because, during system operation, an attempt to write an event occurred will failed as the write protect switch is enabled. An error can be checked with the module diagnostics immediately after the error occurred, but the same information cannot be checked after turning the power OFF and ON, or resetting the module since errors are not saved.

Point

For a system for which file write occurs frequently or the state of which frequently changes because of the unstable communication, the file size of the event history must be larger enough to store a greater number of events. In this case, using an SD memory card is recommended as the storage memory.

■File size

If the specified size is exceeded, the records are deleted from the oldest one and the latest one is stored.

A file size of the event history can be obtained by the following formula.

- File size = File header size + Event history management information size + Number of records × Size per event history record

The sizes of each element are as follows:

Element name	Size
File header size	20 bytes
Event history management information size	12 bytes
Size per event history record*1	40 to 1112 bytes

*1 Since the events which overlaps multiple records exist depending on the event to be saved, a file size per one record will be changed.

■Conditions for collecting events

A C Controller module collects event history regardless of the operating status (RUN/STOP/PAUSE/STOP error); however, the event history may not be collected under the following conditions.

- Major error
- Base unit error
- Cable failure

■File creation timing

An event history file is created when:

- When the power is turned OFF and ON (no event history file exists/after the change of event history settings)
- When a C Controller module is reset (no event history file exists/after the change of event history settings)
- At initialization of memory
- At registration of event history (no event history file exists)

Point

When a new event history file is created, the event that indicates a new file creation is saved.

The following shows the operations of the event history when the storage memory is an SD memory card.

Operation	Operation of event history
Removal of the SD memory card	When the memory initialization event occurs, the event history is stored in the internal memory. If the internal memory reaches the maximum allowable number for saving event history, all subsequent events will be dropped.
Insertion of an 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 C Controller module continues to store the event history information. If the file size is different, the C Controller module removes the existing event history file and creates a new event history file.

■Parameter application timing

The changed parameters are enabled at any of the following timing.

- When the power is turned OFF and ON
- When the C Controller module is reset

Point

If the C Controller module operating status is changed from STOP to RUN after writing the changed parameters, the changed parameters will not be enabled. The changed parameters are enabled only after turning the power OFF and ON, or resetting the C Controller module.

Event dropping

If either of the following conditions are met, some event histories may be lost.

■When events are detected frequently (including events occurred in a module other than a C Controller module)

- The event code (*HST LOSS*) indicating event loss is displayed.
- A corresponding module is displayed in "Source" and "Start I/O No."

■When a C Controller module is powered OFF or reset while collecting event histories

- The event code (*HST LOSS*) indicating event loss is displayed.
- "Source" and "Start I/O No." are blank.

Precautions

When resetting a C Controller module using RESET/STOP/RUN switch while the module is in the following conditions, the event code (*HST LOSS*) may be recorded in the event history: The C Controller module is in the RUN or PAUSE status and the RESET/STOP/RUN switch is in the RUN position.

To reset the module using the RESET/STOP/RUN switch, ensure to set the RESET/STOP/RUN switch to the STOP position first, and then to the RESET position.

Displaying event history

The event history can be displayed using the menus of CW Configurator. For details on the operating procedures and how to read the displayed information, refer to the following:

CW Configurator Operating Manual

Clearing event history

The event history can be cleared on the event history window. Once the event history is cleared, all the event history stored in the specified storage memory is deleted. For details on the operating procedures, refer to the following:

CW Configurator Operating Manual

11.11 Fixed Cycle Processing Function

This function performs operations such as the refresh processing with a network module and the self-diagnostics processing in a fixed cycle.

It is processed as follows:

No.	Processing
❶	Refresh processing with network modules (link refresh)
❷	Refresh processing for module labels and motion module public labels*1
❸	Self-diagnostics processing
❹	Reset processing of watchdog timer
❺	Completion processing of dedicated instructions

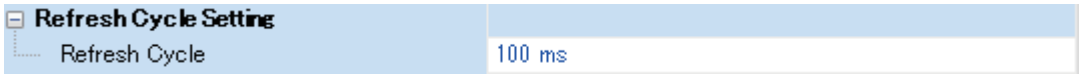
*1 The refresh from module labels to motion module public labels will be stopped if a major error occurs.

Setting a fixed cycle processing interval

The following shows the setting method for the cycle that the fixed cycle processing function operates.

[CPU Parameter] ⇨ [Operation Related Setting] ⇨ [Refresh Cycle Setting]

Window



Displayed items

Item	Description	Setting range	Default
Refresh Cycle	Set the interval that the fixed cycle processing function operates.	1 to 2000 ms (1 ms units)	100 ms

Point

Set the value that satisfies the following relational expression for the cycle of the fixed cycle processing. An error will occur when the following expression is not satisfied.

• Setting time of the watchdog timer (WDT) function > Cycle of the fixed cycle processing

Checking the interval

The following shows the method to check the interval (maximum value/minimum value/current value) at which the fixed cycle processing function operates.

Checking with a function

The interval can be acquired with the C Controller module dedicated function (CCPU_GetConstantProcessStatus).

Checking with special registers

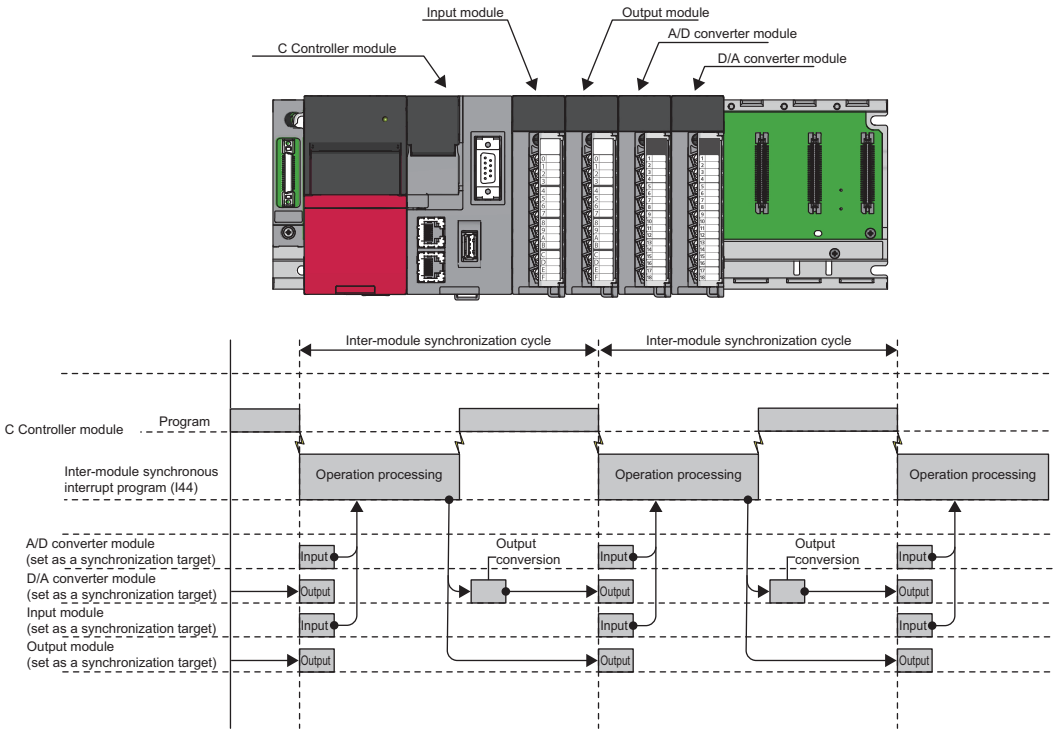
The interval at which the function operates is stored in special registers (SD520/SD522/SD524). (☞ Page 362 Special Register List)

11.12 Inter-module Synchronization Function

This function adjusts the input or output timing of modules to be synchronized to the inter-module synchronization cycle by matching the control timing of signals between multiple modules.

For details on the inter module synchronization function, refer to the following:

MELSEC iQ-R Inter-Module Synchronization Function Reference Manual



Fixed cycle synchronization function

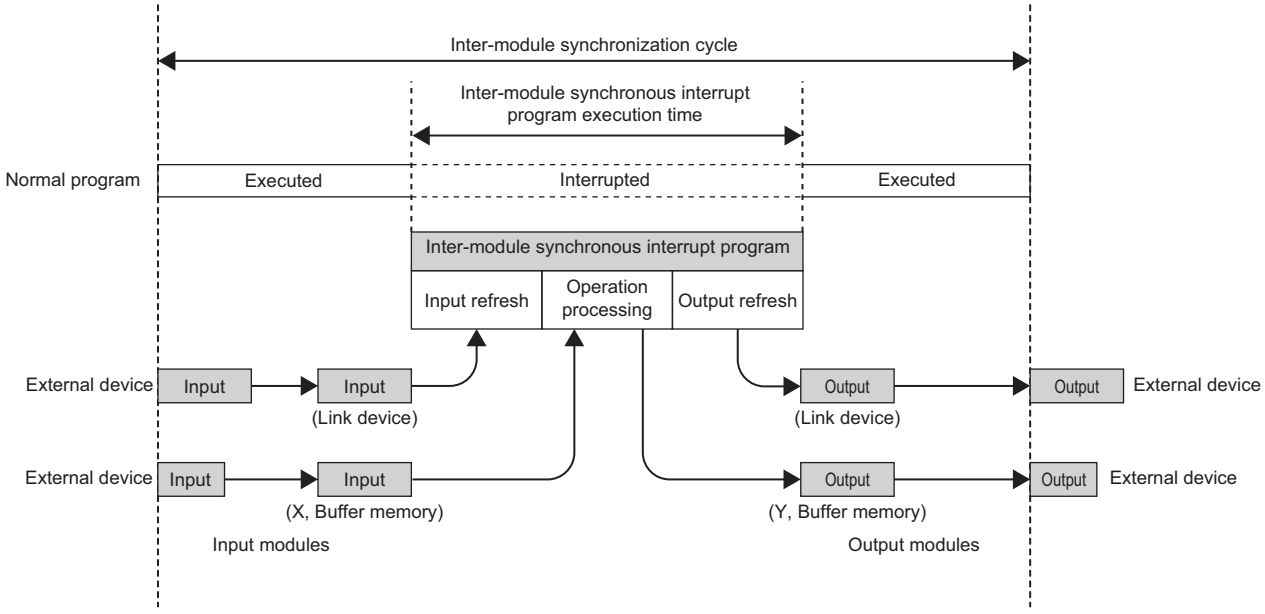
The fixed cycle synchronization function is a function to synchronize multiple modules at the timing of the inter-module synchronization cycle to perform data communication and input/output control at a fixed cycle. Using this function enables the accurate speed to be acquired by the encoder input at the fixed cycle, and also enables the highly accurate model prediction control by the accurately-tracked input/output timing.

Point

Although the shorter interval of the inter-module synchronization cycle provides the more accurate synchronous control, the size of program executable in the inter-module synchronous interrupt program will be smaller. By lengthening the interval of the inter-module synchronization cycle, the program with bigger size can be executed. However, the accuracy of the synchronous control will be reduced.
To use the fixed cycle synchronization function, consider the program size to be executed.

Synchronization timing of a C Controller module

A C Controller module executes the inter-module synchronous interrupt program (I44) at every inter-module synchronization cycle. Synchronization between a C Controller module and each module is performed at the timing of refresh before and after the inter-module synchronous interrupt program (I44). This will enable the C Controller module to import the input data and to write the output data at the timing of the inter-module synchronization cycle.



Inter-module synchronous interrupt

Interrupt programs are executed at the timing of the inter-module synchronization cycle set with the parameters. The interrupt programs executed at every inter-module synchronization cycle is referred to as the inter-module synchronous interrupt program (I44).

Point

- Describe the control programs to be synchronized in the inter-module synchronous interrupt program (I44).
- The operation when an interrupt factor arises and the program creation method are the same as normal interrupt program.

■Execution timing

The inter-module synchronous interrupt program (I44) is executed at the timing of the inter-module synchronization cycle. The inter-module synchronous interrupt program (I44) is registered using the interrupt function form module. (☞ Page 88 Interrupt Function to a C Controller Module)

Processing of an inter-module synchronous interrupt program (I44)

The following shows the processing of an inter-module synchronous interrupt program (I44).

■Input refresh

At the input refresh, link devices (RX and RWr) are refreshed from the CC-Link IE Field Network module or CC-Link IE TSN module to be synchronized.

■Operation processing

- To import input from the buffer memory and input signal (X), use the C Controller module dedicated functions (CCPU_X_In_Word_ISR, CCPU_FromBuf_ISR).
- To execute output to the buffer memory and output signal (Y), use the C Controller module dedicated functions (CCPU_Y_Out_Word_ISR, CCPU_ToBuf_ISR).

Point

A program that is registered in inter-module synchronous interrupt (I44) with the C Controller module dedicated function (CCPU_EntryInt) is executed only when the program is enabled with the C Controller module dedicated function (CCPU_EnableInt). (☞ Page 88 Interrupt Function to a C Controller Module)

■Output refresh

At the output refresh, link devices (RY and RWw) are refreshed to the CC-Link IE Field Network module or CC-Link IE TSN module to be synchronized.

Precautions

■Operation for inter-module synchronous interrupt program processing

In C Controller modules, when the inter-module synchronization setting function is set to 'Use,' input refresh and output refresh of an inter-module synchronous interrupt program will be performed during the initial processing of the modules. When a C Controller module operates normally after the completion of C Controller module initial processing, different processing will be performed depending on the status of an inter-module synchronous interrupt program.

Item	Start status of C Controller module	Inter-module synchronous interrupt program		Execution status	Operation timing
		Registration status*1	Enabled status*2		
Input refresh	During initial processing*3	The status of inter-module synchronous interrupt program does not affect.		Executed (Refreshed regardless of the status of inter-module synchronous interrupt program.)	At every inter-module synchronization cycle
	At normal operation*4				
Operation processing	During initial processing*3	The status of inter-module synchronous interrupt program does not affect.		Not executed (Not executed regardless of the status of inter-module synchronous interrupt program.)	No operation
	At normal operation*4	Registered	Enabled	Executed	At every inter-module synchronization cycle
		Registered	Disabled	Not executed	No operation
		Not registered	Enabled		
		Not registered	Disabled		
Output refresh	During initial processing*3	The status of inter-module synchronous interrupt program does not affect.		Executed (Refreshed regardless of the status of inter-module synchronous interrupt program.)	At every inter-module synchronization cycle
	At normal operation*4				

*1 A status of whether a program is registered in inter-module synchronous interrupt (I44) with the C Controller module dedicated function (CCPU_EntryInt)
*2 A status of whether a program for inter-module synchronous interrupt (I44) is enabled with the C Controller module dedicated function (CCPU_EnableInt)
*3 For initial processing, refer to the following:
☞ Page 70 Initial Processing
*4 Even if the C Controller module is in the STOP state, the inter-module synchronous interrupt program will continue to be executed. For the operation in the STOP state, refer to the following:
☞ Page 125 Refresh timing

■Synchronizing with CC-Link IE TSN master/local module

In C Controller modules, a processing time for inter-module synchronous interrupt program may vary due to the effect of other processing. Depending on refresh settings or cyclic transmission time settings of a CC-Link IE TSN master/local module, it may cause cyclic transmission skip error (1D10H) in the CC-Link IE TSN master/local module because of the variations in processing time.
Take corrective actions according to the following table.

Item	Corrective action
When a program has been registered in inter-module synchronous interrupt (I44) and enabled even once	Take a corrective action according to the action described in the cyclic transmission skip error (1D10H).
When the cyclic transmission skip error occurs during the start of C Controller module, or any cases other than above are applied	• Reduce the total number of points of devices in the refresh settings of a CC-Link IE TSN master/local module. • Lengthen communication cycle and cyclic transmission time of a CC-Link IE TSN master/local module.

Refresh timing

The following shows the refresh timing depending on the operating status of a C Controller module.

■Operation in the STOP state (RUN to STOP)

- C Controller module turns OFF the output when it is in the STOP state due to user operation (such as user program or switch operation). Refresh is performed continuously during the STOP state.
- A C Controller module turns the output OFF and also stops the refresh when a stop error which is caused by any failure of the C Controller module or any other modules occurred.

Point

The inter-module synchronous interrupt program (I44) does not stop even if C Controller module is in the STOP state.

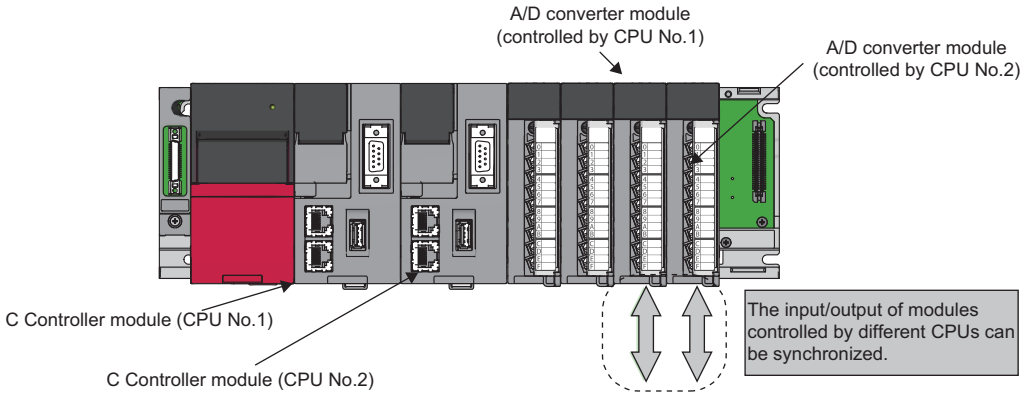
Precautions

When the system parameter of the inter-module synchronization function is rewritten while the C Controller module is in the STOP state, it will not applied by switching to the RUN state, and the C Controller module operates with the parameter before the rewrite.

Interaction with cycle of the fixed cycle communication of the multiple CPU system function

The cycle of the fixed cycle communication of the multiple CPU system function can be matched with the inter-module synchronization cycle.

By adjusting the cycle of the fixed cycle communication to the inter-module synchronization cycle, the input/output of modules having different control CPUs can be synchronized. The interaction with the multiple CPU system function is set with the parameter. (☞ Page 239 Fixed scan communication setting)



Operation of program

Once the inter-module synchronization cycle is interacted with the cycle of the fixed cycle communication, a multiple CPU synchronous interrupt program (I45) and an inter-module synchronous interrupt program (I44) are executed in each inter-module synchronization cycle.

The multiple CPU synchronous interrupt program (I45) is executed first, and then the inter-module synchronous interrupt program (I44) is executed. Two inter-module synchronization cycles are required for data acquired by the host CPU to reach to another CPU, and delaying the output timing of the controlled module of the host CPU and the controlled module of another CPU for two cycles is required for the output synchronized among CPU modules.

Point

For a C Controller module, only the link devices of CC-Link IE Field Network modules (RX, RY, RWr, and RWw) are refreshed at the 'input refresh' and 'output refresh' in an inter-module synchronous interrupt program. To import the I/O signals (X, Y) and the values from the buffer memory, use a C Controller module dedicated function.

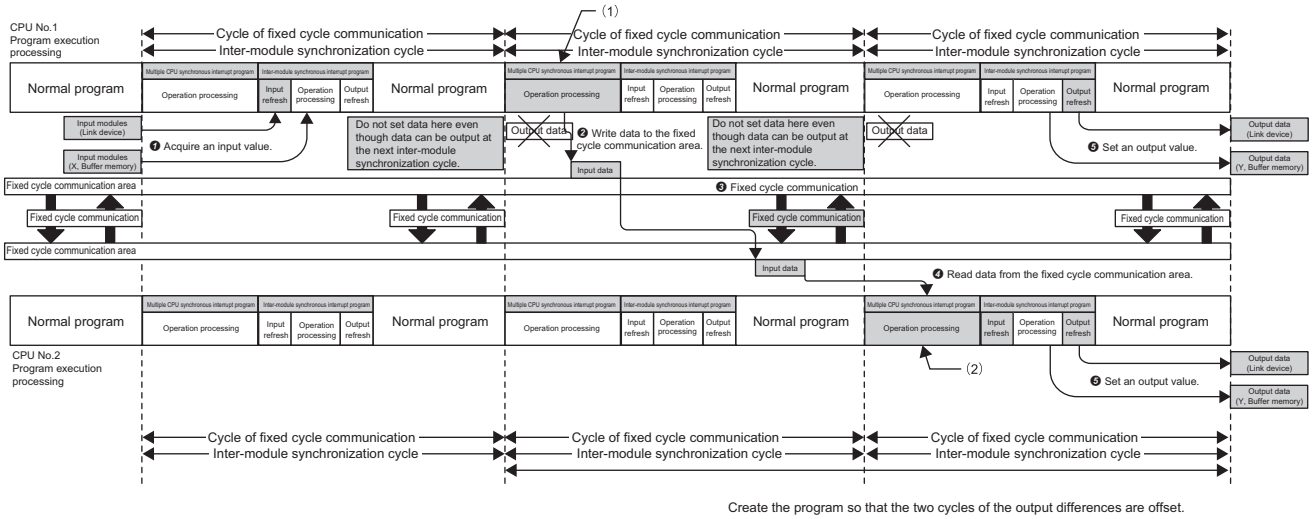
For details on processing of each interrupt program of a C Controller module, refer to the following:

- Inter-module synchronous interrupt program (I44)

☞ Page 123 Fixed cycle synchronization function

- Multiple CPU synchronous interrupt program (I45)

☞ Page 248 Multiple CPU synchronous interrupt



(1): Write the input value to the fixed cycle communication area of CPU No.1.

(2): Set the output value by reading data from the fixed cycle communication area of CPU No.1.

Precautions

The following shows the considerations for interacting with cycle of fixed cycle communication.

- Selecting "Use" for the inter-module synchronization function enable "Cooperate" to be selected for "Fixed Scan Communication Function and Inter-module Synchronization Function." It cannot be selected when "Not use" is selected for the inter-module synchronization function.
- Even if "I/O Setting Outside Group" is enabled in a program of another CPU, importing the input or output from the module to be synchronized is not available. Although it can be read with a direct specification such as DX, DY, Un\Gn, or the C Controller module dedicated function (CCPU_FromBuf), data inconsistency may occur.
- To adjust the start of the inter-module synchronization function among all CPUs, confirm that the inter-module synchronization function is available by checking the ready flag of CPU No.n from SM220 to SM223 turns ON. By using SM220 to SM222 in an interlock program, the start of the inter-module synchronization function can be matched among all CPUs.

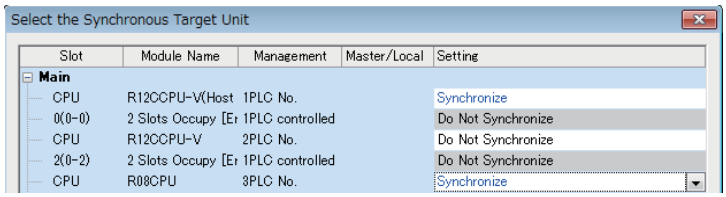
Parameter setting

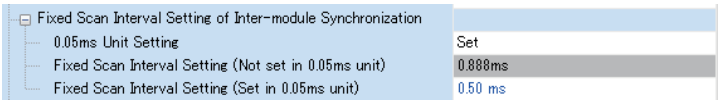
Set a module configuration diagram with CW Configurator to configure the inter-module synchronization setting.

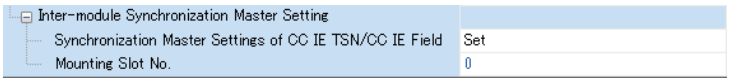
[System Parameter] ⇨ [Inter-module Synchronization Setting] ⇨ [Inter-module Synchronization Setting]

Operating procedure









1. Select "Use" for "Use Inter-module Synchronization Function in System."
2. Click "Detailed Setting" for "Select Inter-module Synchronization Target Module."
3. Select "Synchronize" for the module to be synchronized.
4. Set the inter-module synchronization cycle in "Fixed Scan Interval Setting of Inter-module Synchronization."
5. When writing parameters, write both of the system parameter and module parameter.
6. Configure the setting for the synchronous master when a master or local station of CC-Link IE TSN module or a local station of CC-Link IE Field Network module is to be synchronized. (The setting is not required for a master station.)

Displayed items

Item		Description	Setting range	Default
Use Inter-module Synchronization Function in System		Set whether to 'Use' or 'Not Use' inter-module synchronization function in system.	• Not Use • Use	Not Use
Select Inter-module Synchronization Target Module	Detailed Setting	Set target module which executing inter-module synchronization function.	• Do not Synchronize • Synchronize	Do not Synchronize
Fixed Scan Interval Setting of Inter-module Synchronization*1	0.05ms Unit Setting	Set whether to 'Set' or 'Not Set' fixed scan interval for inter-module synchronization in 0.05 ms unit.	• Not Set • Set	Not Set
	Fixed Scan Interval Setting (Not Set in 0.05ms unit)	Set the fixed scan interval of inter-module synchronization.	• 0.222 ms • 0.444 ms • 0.888 ms • 1.777 ms • 3.555 ms • 7.111 ms	0.888 ms
	Fixed Scan Interval Setting (Set in 0.05ms unit)	Set the fixed scan interval of synchronous within the modules by 0.05 ms.	0.10 to 10.00 ms (in 0.05 ms units)	0.50 ms
Inter-module Synchronization Master Setting	Synchronization Master Settings of CC IE TSN/CC IE Field	Set whether to 'Set' or 'Not Set' CC-Link IE TSN/ Field as synchronization master.	• Not Set • Set	Not Set
	Mounting Slot No.	Set the mounted slot No. of CC-Link IE TSN/Field to set as a synchronization master.	0 to 11	0

*1 The setting range for the inter-module synchronization cycle differs depending on modules. (Manual for respective modules)

Point

To interlink with the cycle of the fixed cycle communication of the multiple CPU system function, set "Fixed Scan Communication Function and Inter-module Synchronization Function" under "Fixed Scan Communication Setting" to "Cooperate." (☞ Page 239 Fixed scan communication setting)

11.13 System Timer Mode Switching Function

This function changes the system timer mode of a C Controller module.
The system timer is a timer used for functions (except for the fixed cycle processing function) of a C controller module.
When setting the high system clock rate, the system timer precision may decrease due to high priority tasks on an operating system.
By switching the system timer to the high-precision mode, it can be operated without losing the precision even when the system clock rate has been changed.

Point

The current system timer mode can be checked with the C Controller module dedicated function (CCPU_GetSysTimerMode).
Use the high-precision mode in the following cases:

- When setting the high system clock rate
- When using the CCPU_WaitTimerEvent function and enabling the module to operate by minimizing variations in timer event occurrence cycle
- When using the CC-Link IE Field Network Basic function and enabling the module to operate by minimizing variations in link scan time

System timer mode list

The system timer mode list is as follows:

Mode	Description
Normal mode (default)	Default mode
High-precision mode	A mode that operates by minimizing variations in a system timer compared to the normal mode

Usage

The system timer mode can be changed by executing the C Controller module dedicated function (CCPU_SetSysTimerMode) and turning the power OFF and ON, or resetting the C Controller module.

Precautions

When using the high-precision mode, the following values are affected; therefore, check the operation before use.

- Link scan time of CC-Link IE Field Network Basic
- Timer event occurrence cycle specified with the C Controller module dedicated function (CCPU_WaitTimerEvent)

Function list

The following table shows the function list.
For details on functions, refer to the following:
MELSEC iQ-R C Controller Module Programming Manual

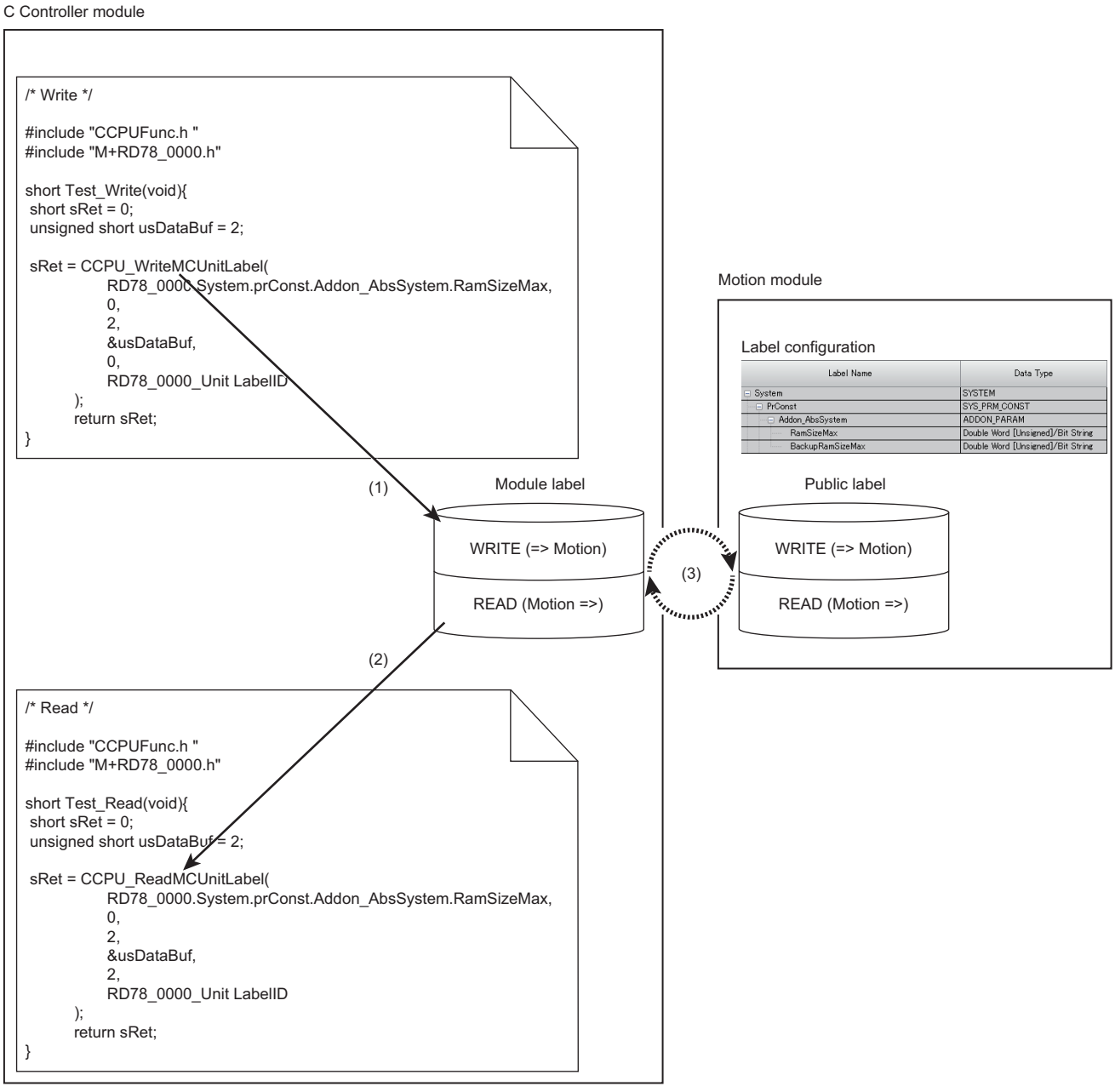
Function name	Description
CCPU_SetSysTimerMode	To change the system timer mode.
CCPU_GetSysTimerMode	To acquire the system timer mode that is currently in operation.

12 MOTION MODULE CONTROL FUNCTION

This function controls a motion module controlled by a C Controller module.
The motion module is controlled by a motion control function block (hereinafter referred to as MCFB) or module labels.

12.1 Module Labels

The motion module control function accesses public labels of a motion module with C Controller module dedicated functions.
The function manages axes and input and output signals, which are controlled by a motion module, as module labels.



(1) Data is written to a module label with the C Controller module dedicated function (CCPU_ReadMCUnitLabel or CCPU_ReadMCUnitLabelBit).
(2) Data is read from a module label with the C Controller module dedicated function (CCPU_WriteMCUnitLabel or CCPU_WriteMCUnitLabelBit).
(3) Refreshing

Accessing module labels

The motion module control function accesses (writes to or reads from) module labels with C Controller module dedicated functions.

Information of module labels (within the module label area) in a C Controller module is refreshed along with public labels (buffer memory) of a motion module according to the refresh cycle.

Module labels are accessed when a C Controller module dedicated function is executed in a user program.

■Function list

The following shows functions to use for accessing the label memory.

Function name	Description
CCPU_ReadMCUnitLabel	To read data from module labels of C Controller module in word units.
CCPU_WriteMCUnitLabel	To write data to module labels of C Controller module in word units.
CCPU_ReadMCUnitLabelBit	To read data from module labels of C Controller module in bit units.
CCPU_WriteMCUnitLabelBit	To write data to module labels of C Controller module in bit units.

Precautions

When creating a user program, specify a module label ID for the argument of C Controller module dedicated function. If a module label written to a C Controller module and a module label ID specified for the module label access function are different, the module label cannot be accessed.

■Refreshes with public labels of a motion module

Module labels of a C Controller module and public labels of a motion module are refreshed at the fixed cycle processing. (☞ Page 121 Fixed Cycle Processing Function)

To refresh the labels at the fixed cycle processing, set "At the Execution Time of END Instruction" for the refresh timing of motion module.

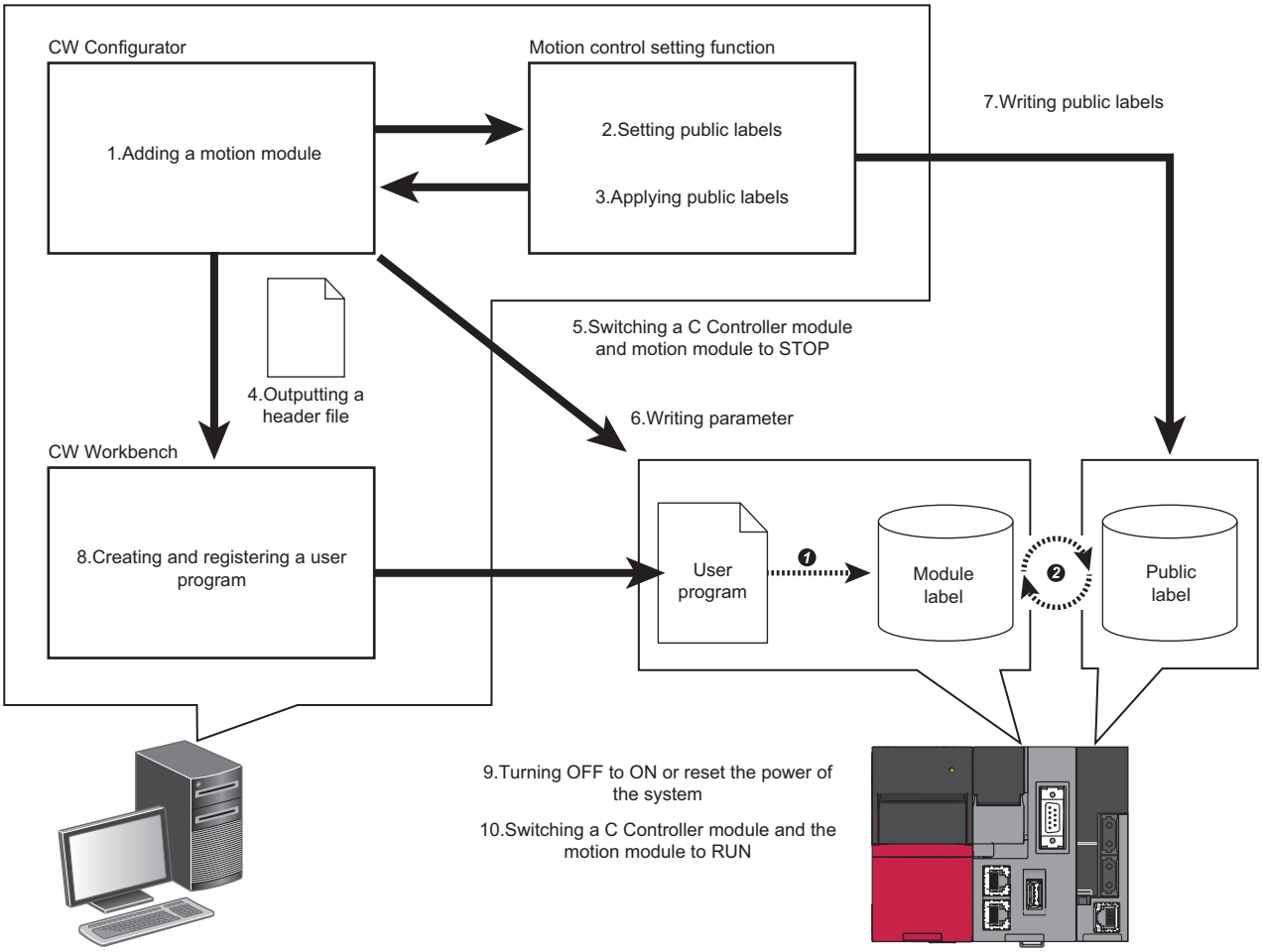
[Module Parameter (Motion)] ⇨ [Refresh Setting] ⇨ [Refresh by the Set Timing] ⇨ [Refresh Timing]

Usage

The following explains the method for controlling a motion module using module labels.

Precautions

To control a motion module by module labels, perform the operation procedure from '1. Adding a motion module' to '10. Switching a C Controller module and the motion module to RUN' described below as a series of operations. Otherwise, an inconsistency occurs in module label information between a C Controller module and motion module, which may cause an unintended operation.



❶ A user program accesses module labels.

❷ Buffer memories and motion control are refreshed (updated) according to the refresh cycle.

Operating procedure

The following shows the procedure for controlling a motion module by module labels.

1. Adding a motion module

Add a motion module in CW Configurator.

2. Setting public labels

Start the motion control setting function from CW Configurator.

Register labels for motion module and set them to disclose to the CPU module side.

For a procedure to create a program using the public label function, refer to the following:

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

3. Applying public labels

Apply public labels to CW Configurator.

Public labels are registered as global labels (module labels) in CW Configurator.

For a procedure to create a program using the public label function, refer to the following:

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

4. Outputting a header file

Output label data, which is necessary for programming, as a header file.

For output methods of a header file, refer to the following:

CW Configurator Operating Manual

Point

- Setting information on a module label is managed by a module label ID generated when outputting a header file. If the procedure 'applying public labels' is not performed after performing 'outputting a header file,' the module label ID will not be changed.
- When using multiple motion modules, output a header file for the each module. The module label ID is the same for all header files because module label areas of all motion modules are centrally managed.

5. Switching a C Controller module and motion module to STOP

Switch a C Controller module to STOP and check that the preparation completion flag of the motion module is turned OFF (axis control is stopped) and output to a network is stopped.

Precautions

If operations described after this procedure are performed with leaving the C Controller module and motion module in the RUN status, inconsistency occurs in module label information between the C Controller module and motion module when writing public labels to the motion module. Therefore, it may cause an unintended operation.

However, if "Hold" is set for "Output Hold/Clear Setting during CPU STOP" in module parameters of the motion module, output to a network will continue to be performed even when switching the operation status to STOP. Therefore, it may cause an unintended operation.

Take any of the following corrective actions to stop output to a network.

- Stop all the programs running in the C Controller module.
- Remove an Ethernet cable connected to the motion module.

For operation processing depending on the operation status of a C Controller module, refer to the following:

Page 72 Operation Processing Depending on Operating Status

6. Writing parameter

Write module labels to a C Controller module.

[Online] ⇨ [Write to PLC]

Information on public labels is stored in refresh parameters and is written to the C Controller module.

7. Writing public labels

Write public labels to a motion module.

For a procedure to create a program using the public label function, refer to the following:

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

8. Creating and registering a user program

Create a user program in CW Workbench using the header file output in the step 4.

Register the created user program to the C Controller module.

For methods for creating and registering a user program, refer to the following:

☞ Page 44 Creating a User Program

CW Configurator Operating Manual

9. Turning OFF to ON or reset the power of the system

Turn OFF to ON or reset the power of the C Controller system.

10. Switching a C Controller module and the motion module to RUN

Switch a C Controller module to RUN and turn ON the PLC READY of the motion module.

Precautions

- When a module label is updated, the module label ID will be changed; therefore, a header file must be output again. The updated module label is displayed in red in the navigation window.
- The module label ID written to a C Controller module will not be a target for project verification.
  Therefore, to check consistency of setting information on a module label, make sure to check if the module label ID written to a C Controller module matches the module label ID of a header file output from a project.
  The module label ID written to a C Controller module can be checked with 'Module label ID' (SD1596 to 1597). (☞ Page 362 Special Register List)
- Detailed setting information of a module label is not written to parameters which are written to a C Controller module.
  Therefore, even if parameters are read, the same project (having the same module label ID) cannot be restored. Make sure to save a project for which the procedure of 'outputting a header file' is performed.

Header file

A header file which is output from CW Configurator is as follows:

```
/*M+RD78_0000*/                                                    (1)

#define RD78_0000_UnitLabelID 1156UL                               (2)

typedef struct{                                                    (3)
 struct{
  struct{
   struct{
    unsigned long RamSizeMax;
    unsigned long BackupRamSizeMax;
   } Addon_AbsSystem;
  } PrConst;
 } System;
} RD78_0000_TAG;  <-- (4)

static const RD78_0000_TAG RD78_0000 = { { { { 0x100000, 0x100020 } } } };  (5)
```

(1) A label of motion module
(2) Module label ID
(3) Definition of a structure
(4) Label definition of motion module
(5) Module label address

Ex.

Outputting a header file of a motion module (RD78G4)

Label data

- M+RD78_0020 (global label)

Label name	Data type
RD78_0020	M+RD78_0020

- M+RD78_0020 (structure)

Label name	Data type
RD78_0020	M+RD78_0020
uLabel1	Word [Unsigned]/Bit string [16-bit]
uLabel2	Word [Unsigned]/Bit string [16-bit] (0..7)*1
stLabel3	M+RD78_0020_ST_LABEL1
stLabel4	M+RD78_0020_ST_LABEL1(0..1)*2

*1 Basic data array
*2 Structured array

- M+RD78_0020_ST_LABEL1 (structure)

Label name	Data type
bLabel1	Bit
uLabel2	Word [Unsigned]/Bit string [16-bit]
wLabel3	Word [signed] (0..3)*1

*1 Label array

Export file

• RD78_0020.h

```
/*M+RD78_0020*/

#define RD78_0020_UnitLabelID 123456789UL

typedef struct{
    unsigned long uLabel1;
    unsigned long uLabel2;*1
    struct{
        unsigned long bLabel1;
        unsigned long uLabel2;
        unsigned long wLabel3;
    } stLabel3;
    struct{
        unsigned long bLabel1;
        unsigned long uLabel2;
        unsigned long wLabel3;
    } stLabel4[2];*2
} RD78_0020_TAG;

static const RD78_0020_TAG RD78_0020 = {
0x180000,*3
0x180010,*4
{ 0x1800C0, 0x1800D0, 0x1800E0 }*5,
{ { 0x180140, 0x180150, 0x180160 }*6,
{ 0x1801C0, 0x1801D0, 0x1801E0 };*7 } };
```

*1 The number of array elements is not added to the array label of basic data.
*2 The number of array elements is defined for a structure array label.
*3 Assignment address of uLabel1
*4 Assignment address of uLabel2 (the first address only)
For the second array or more arrays that come next, a C Controller module accesses the first assignment address with the following value being added.
· Label data type address size*array number
(Example: Word address size = 1, Double-word address size = 2)
*5 Assignment address of stLabel3
*6 Assignment address of stLabel4[0]
*7 Assignment address of stLabel4[1]

12.2 I/O Signals

The motion module control function can access I/O signals of a motion module with the C Controller module dedicated functions.

For the specifications of I/O signals when controlling a motion module from a control CPU, refer to the following:

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

12.3 Buffer Memory

The motion module control function can access the buffer memory of a motion module with the C Controller module dedicated functions.

For the specifications of the buffer memory when controlling a motion module from a control CPU, refer to the following:

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

12.4 MCFB

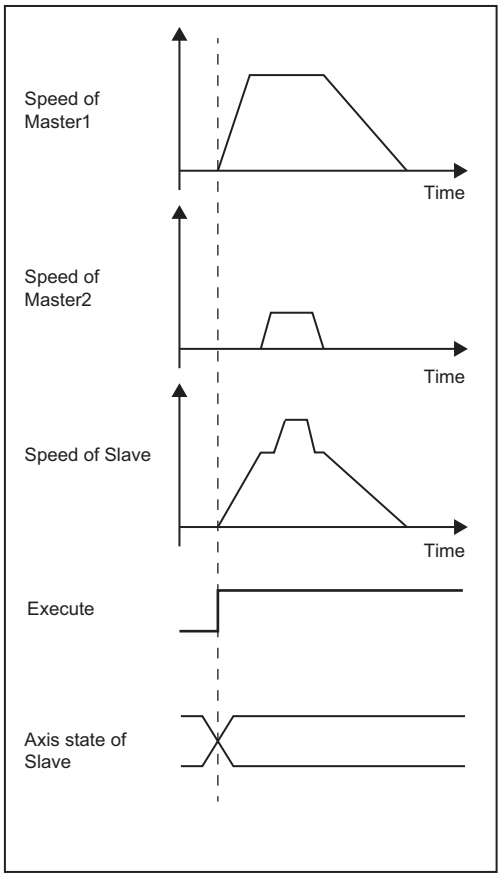
The motion module control function executes MCFBs using motion module dedicated classes of C Controller module.
By doing so, motion control (such as positioning control, continuous control and synchronous control) can be performed.

C Controller module

```
/*****************************************/
/*Execute addition/subtraction positioning*/
/*****************************************/

// Set I/O variables and input variables of a member variable
mcCombineAxes.Master1.StartIO = START_IO;
mcCombineAxes.Master1.AxisNo = 2;
mcCombineAxes.Master2.StartIO = START_IO;
mcCombineAxes.Master2.AxisNo = 3;
mcCombineAxes.Slave.StartIO = START_IO;
mcCombineAxes.Slave.AxisNo = 1;
mcCombineAxes.CombineMode = mcCombineAxes.mcAddAxes;
mcCombineAxes.GearRatioNumeratorM1 = 1000000;
mcCombineAxes.GearRatioDenominatorM1 = 2000000;
mcCombineAxes.GearRatioNumeratorM2 = 1000000;
mcCombineAxes.GearRatioDenominatorM2 = 2000000;
mcCombineAxes.MasterValueSourceM1 = mcCombineAxes.mcSetValue;
mcCombineAxes.MasterValueSourceM2  = mcCombineAxes.mcSetValue;
mcCombineAxes.BufferMode = mcCombineAxes.mcAborting;

// Set I/O variables and input variables in a motion module and execute MCFB
sRet = mcCombineAxes.SetExecute(TRUE);
if( sRet != 0 ){
    return;
}

// Refresh I/O variables, output variables, and public variables
while(1){
    sRet = mcCombineAxes.RefreshLabels();

    // Check if the MCFB is completed
    if( sRet == 0 ){
        if( mcCombineAxes.Error == FALSE){
            if(mcCombineAxes.InSync == TRUE){
                break;
            }
        }else{
            break;
        }
    }else{
        return;
    }
    taskDelay(1);
}
```

(1)

Motion module



(1) The MCFB (homing) is executed.

Correspondence table for MCFBs and instruction classes

The following table shows instruction classes of motion module dedicated classes that correspond to MCFBs.

For details on functions of a motion module and MCFBs, refer to the following:

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

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

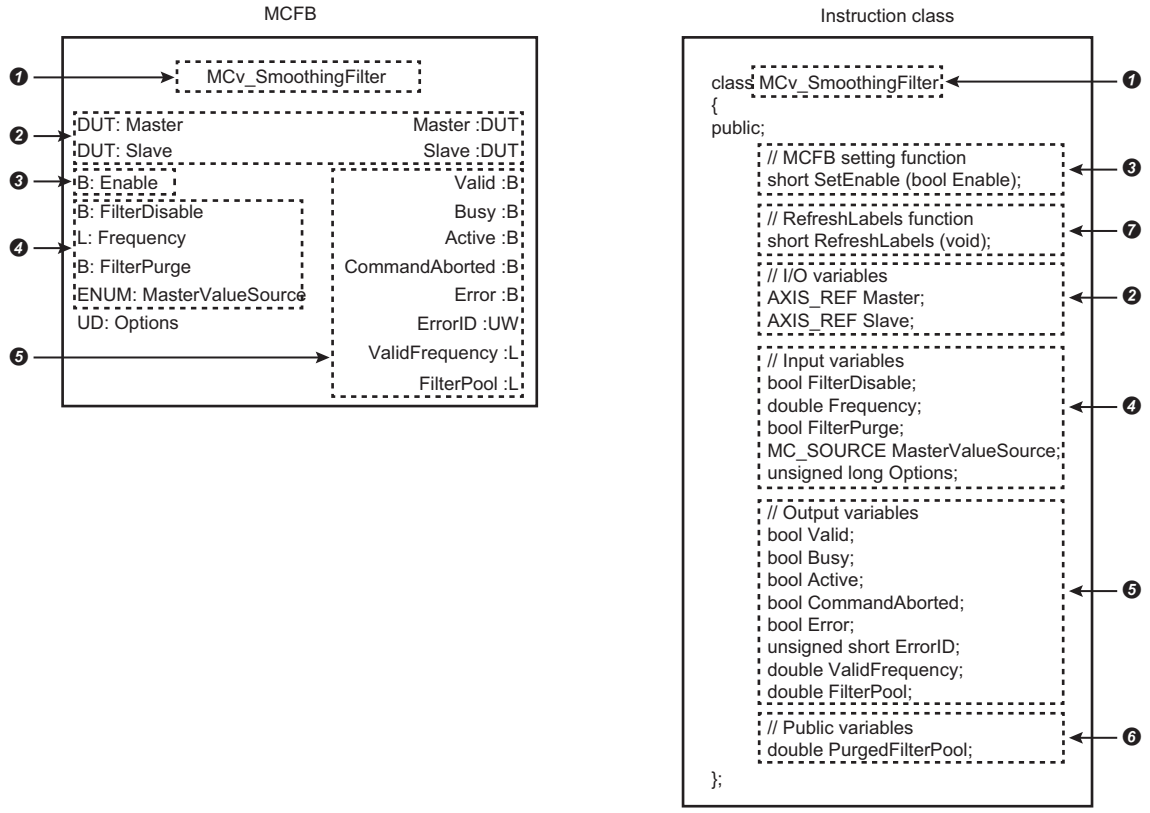
MCFB	Corresponding instruction class	Instruction class type
MC_AbortTrigger	MC_AbortTrigger	Execute type
MC_CamIn	MC_CamIn	Execute type
MC_CamTableSelect	MC_CamTableSelect	Execute type
MC_CombineAxes	MC_CombineAxes	Execute type
MC_GearIn	MC_GearIn	Execute type
MC_GroupDisable	MC_GroupDisable	Execute type
MC_GroupEnable	MC_GroupEnable	Execute type
MC_GroupReset	MC_GroupReset	Execute type
MC_GroupStop	MC_GroupStop	Execute type
MC_Home	MC_Home	Execute type
MC_MoveAbsolute	MC_MoveAbsolute	Execute type
MC_MoveRelative	MC_MoveRelative	Execute type
MC_MoveVelocity	MC_MoveVelocity	Execute type
MC_Reset	MC_Reset	Execute type
MC_SetPosition	MC_SetPosition	Execute type
MC_Stop	MC_Stop	Execute type
MC_TorqueControl	MC_TorqueControl	Execute type
MC_TouchProbe	MC_TouchProbe	Execute type
MC_WriteParameter	MC_WriteParameter	Execute type
MCv_ChangeCycle	MCv_ChangeCycle	Execute type
MCv_MotionErrorReset	MCv_MotionErrorReset	Execute type
MCv_MoveCircularInterpolateAbsolute	MCv_MoveCircularInterpolateAbsolute	Execute type
MCv_MoveCircularInterpolateRelative	MCv_MoveCircularInterpolateRelative	Execute type
MCv_MoveLinearInterpolateAbsolute	MCv_MoveLinearInterpolateAbsolute	Execute type
MCv_MoveLinearInterpolateRelative	MCv_MoveLinearInterpolateRelative	Execute type
MCv_ReadProfileData	MCv_ReadProfileData	Execute type
MCv_SetTorqueLimit	MCv_SetTorqueLimit	Execute type
MCv_SpeedControl	MCv_SpeedControl	Execute type
MCv_SpeedLimitFilter	MCv_SpeedLimitFilter	Execute type
MCv_WriteProfileData	MCv_WriteProfileData	Execute type
MC_GroupSetOverride	MC_GroupSetOverride	Enable type
MC_Power	MC_Power	Enable type
MC_ReadParameter	MC_ReadParameter	Enable type
MC_SetOverride	MC_SetOverride	Enable type
MCv_AllPower	MCv_AllPower	Enable type
MCv_BacklashCompensationFilter	MCv_BacklashCompensationFilter	Enable type
MCv_DirectionFilter	MCv_DirectionFilter	Enable type
MCv_Jog	MCv_Jog	Enable type
MCv_SmoothingFilter	MCv_SmoothingFilter	Enable type

Definition of an instruction class

The following explains an instruction class of motion module dedicated classes.

Relationship between an MCFB and an instruction class

The following shows the relationship between an MCFB and an instruction class.



Item			Definition in MCFB	Definition in instruction class
❶	An FB to be used		Defined as an FB.	Defined as a class that has the same name as the MCFB.
❷	I/O variables		Defined as I/O variables.	Defined as member variables that have the same name as I/O variables of the MCFB.
❸	Input variables	Execution instruction input variables*1	Defined as input variables	Not defined (Set with an MCFB setting function)
❹		Other than execute commands		Defined as member variables that have the same name as I/O variables of the MCFB.
❺	Output variables		Defined as output variables.	Defined as member variables that have the same name as output variables of the MCFB.
❻	Public variables		Defined as local labels.	Defined as member variables that have the same name as public variables of the MCFB.
❼	Update of I/O variables, output variables, and public variables		Updated by the END scan.	Updated by executing the RefreshLabels function.

*1 MCFBs have input variables (execution instruction input variables) to instruct execution.
Execution instruction input variables include execute command (Execute), enable (Enable), positive rotation JOG command (JogForward), and reverse rotation JOG command (JogBackward).
For the setting method with an MCFB setting function, refer to the following:
☞ Applying member variable settings (input variables and I/O variables) to a motion module

Member variable usage

The following explains methods for using member variables.

Item		How to set or update in instruction class
I/O variables		■Setting • Assign values to member variables. ■Update • Updated by executing the RefreshLabels function.
Input variables	Execution instruction input variables	■Setting Set by executing an MCFB setting function.
	Other than above	■Setting Assign values to member variables.
Output variables		■Update Updated by executing the RefreshLabels function.
Public variables		

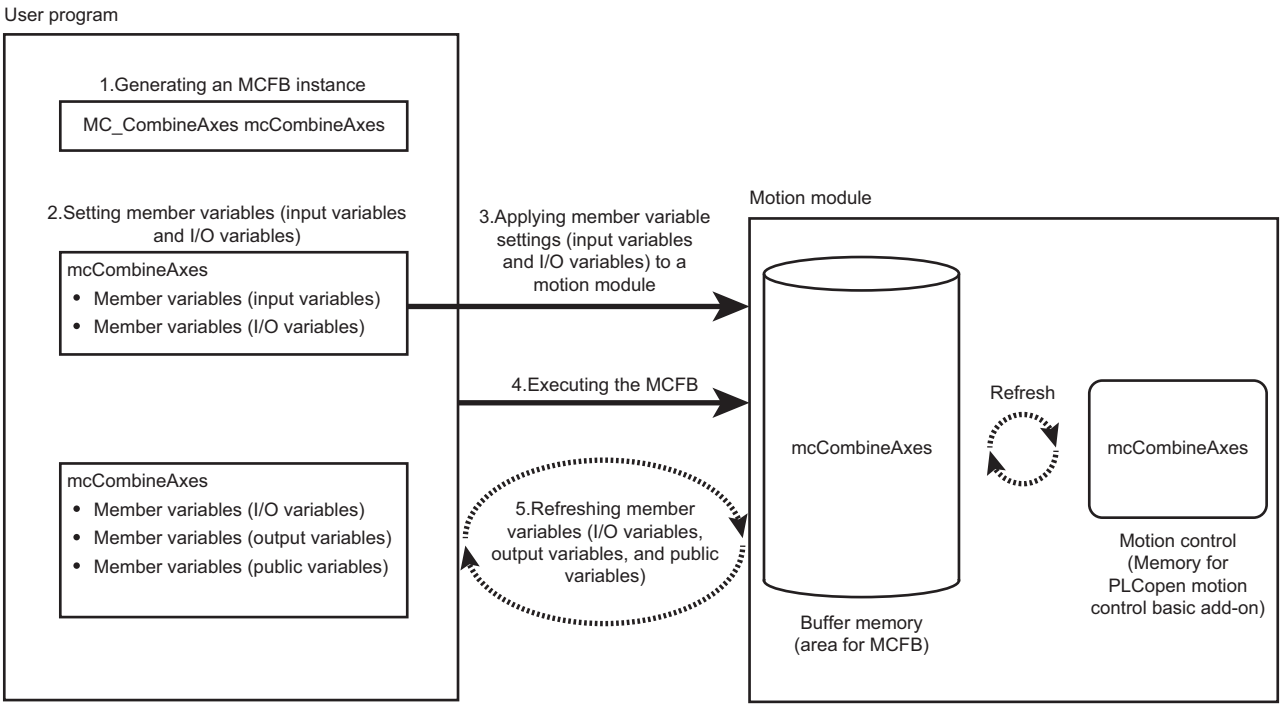
Usage

The following explains the method for executing an MCFB using a user program.

For methods for creating an MCFB instance and using motion module dedicated classes, refer to the following:

📖MELSEC iQ-R C Controller Module Programming Manual

The following shows the data flow when executing an MCFB using the MC_CombineAxes class.



1. Generating an MCFB instance

Generate an instance of an instruction class which corresponds to an MCFB.

Specify an MCFB instance as follows in the user program to generate it.

```
MC_CombineAxes mcCombineAxes;
```

For a corresponding instruction class, refer to the following:

☞ Page 139 Correspondence table for MCFBs and instruction classes

2. Setting member variables (input variables and I/O variables)

Set specified values to execute in the MCFB for input variables and I/O variables of member variables.

Specify arguments of the member variables as follows in the user program.

```
mcCombineAxes.Master1.StartIO = 0x000;
mcCombineAxes.Master1.AxisNo = 2;
```

Precautions

Execution instruction input variables of instruction classes are not defined in member variables. Execute an MCFB with an MCFB setting function. (☞ Applying member variable settings (input variables and I/O variables) to a motion module)

If execution instruction input variables are specified in a user program as follows, an error occurs when compiling the program.

```
mcCombineAxes.Execute = TRUE;
```

3. Applying member variable settings (input variables and I/O variables) to a motion module

Apply member variable settings to a motion module with an MCFB setting function.

An MCFB setting function that corresponds to each instruction class type is as follows.

Instruction class type		Execution instruction input variable	MCFB setting function of instruction class
Execute type		Execute	SetExecute
Enable type	Other than MCv_Jog class	Enable	SetEnable*1
	MCv_Jog class	• JogForward • JogBackward	SetEnableJog

*1 MCv_Jog class do not support the SetEnable function.

Specify the MCFB setting function as follows in the user program to execute it.

```
sRet = mcCombineAxes.SetExecute(TRUE);
```


Precautions

- At the initial execution of an MCFB setting function, a C Controller module creates an internal instance for sending/receiving data to/from a motion module. In addition, a C Controller module communicates with a motion module when creating an internal instance. At that time, an area for sending/receiving data is generated on the motion module side, and therefore it may take time for the MCFB setting function to be completed. (☞ Page 144 Generating buffer memory (area for MCFB) and memory for motion control add-on)
- At the initial execution of an MCFB setting function, an MCFB is executed at the same time when settings are applied. When executing the function for the second time or after, an MCFB is executed at the execution of the RefreshLabels function. (☞ Executing the MCFB)

4. Executing the MCFB

Execute the MCFB.

The execution timing of an MCFB is different depending on whether the MCFB is executed for the first time or for the second time or after in the same instance of each instruction class.

MCFB execution count	Execution timing
First time	An MCFB is executed at the initial execution of an MCFB setting function.
Second time or after	An MCFB is executed at the execution of the RefreshLabels function.

5. Refreshing member variables (I/O variables, output variables, and public variables)

Execute the MCFB and refresh I/O variables, output variables and public variables using the RefreshLabels function.

Precautions

- At the initial execution of an MCFB setting function, only the refresh of I/O variables, output variables, and public variables will be performed even if the RefreshLabels function is executed.
- Depending on motion module specifications, the RefreshLabels function needs to be executed for multiple times to start processing on the motion module side. After executing an MCFB, make sure to check values of I/O variables, output variables, and public variables by executing the RefreshLabels function repeatedly until the processing of the MCFB is completed.

  For the completion status of MCFBs, check specifications of each MCFB. (📖MELSEC iQ-R Motion Module User's Manual (Application))

Buffer memory refresh on the motion module side

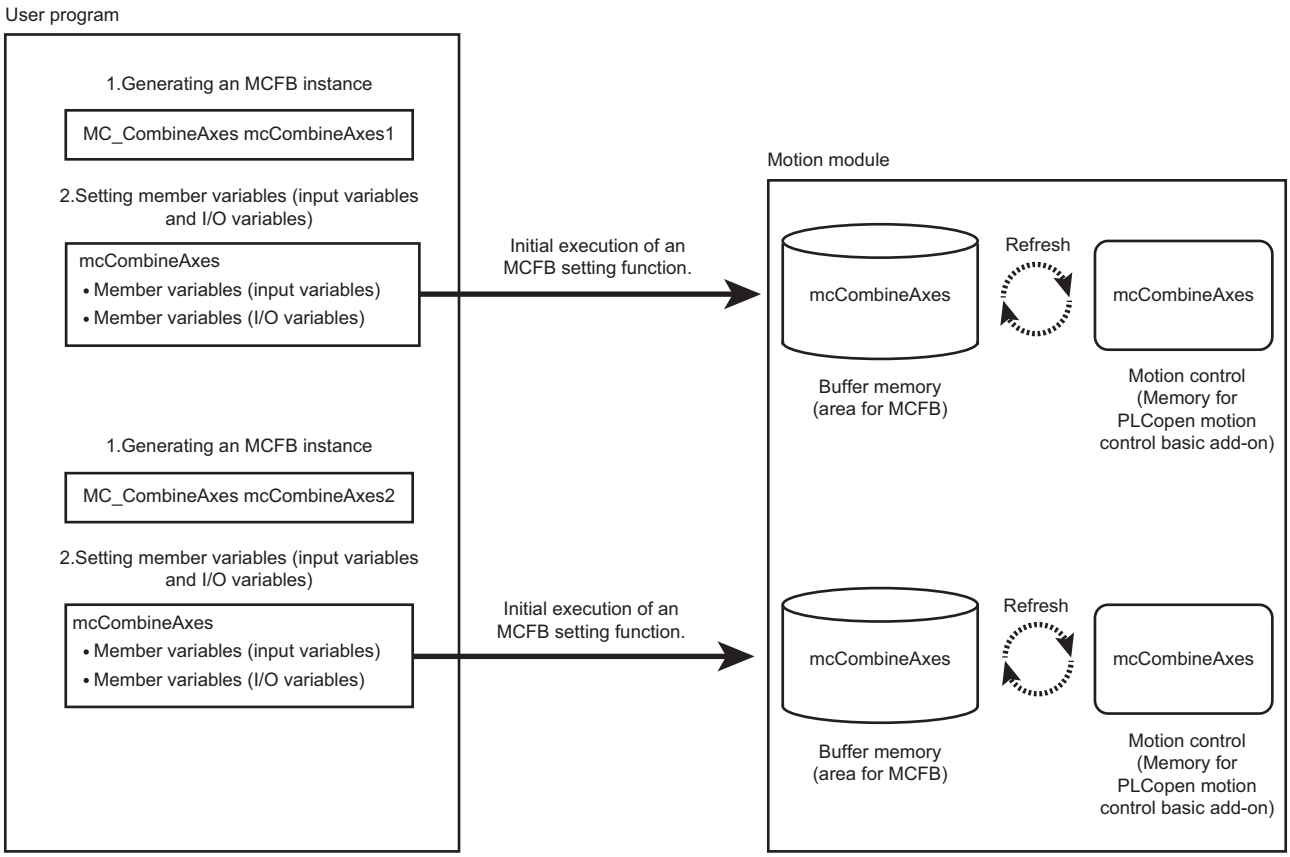
Buffer memory on the motion module side is refreshed at the refresh cycle set for the buffer memory refresh cycle.

For methods for setting the buffer memory refresh cycle of a motion module, refer to the following:

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

Generating buffer memory (area for MCFB) and memory for motion control add-on

The buffer memory (area for MCFB) and memory for motion control add-on are generated in a motion module at the initial execution of an MCFB setting function.



The buffer memory (area for MCFB) and memory for motion control add-on are generated for an instance of each instruction class.

In addition, the memory area of a motion module is used when executing an MCFB using each instruction class. Make sure that the number of instruction class instances does not exceed each memory area size.

Name	Occupied size	Checking method of available size
Buffer memory (area for MCFB)	For the occupied size, refer to the 'List of FBs' in the following manual. MELSEC iQ-R Motion Module User's Manual (Application)	System.Md.BuffermemoryFreeMcfbArea
Memory for PLCopen motion control basic add-on		System.PrConst.Addon_MotionControl_General.RamSizeMax System.Md.Addon_MotionControl_General.RamUsage

Point

- The buffer memory (area for MCFB) and memory for motion control add-on are hold when an MCFB setting function is executed for the first time. If the free space is in short, the error "insufficient memory capacity" (349AH) will occur when the MCFB is called for the first time.
- Even if each instruction class instance is deleted, the memory on the motion module side is not released. Therefore, if each instruction class instance is added or deleted repeatedly to/from user programs, the memory area of the motion module side will be in short and the error "memory shortage" (349AH) may occur. In this case, reset the C Controller module once.

13 Ethernet COMMUNICATION FUNCTIONS

This chapter explains the Ethernet communication functions.

13.1 Connection with MELSOFT Product or GOT

A C Controller module can be monitored with CW Configurator, and can be monitored/tested from a GOT via Ethernet. This enables remote operations with long-distance connection and high-speed communication using Ethernet.
The section shows how to connect a C Controller module with a GOT or MELSOFT product (such as CW Configurator, MX Component).

Connection method	Description
Connection via a hub (specifying an IP address)	• Connecting with a C Controller module that has no network number and station number • Connecting with multiple MELSOFT products or GOTs
Direct connection (without specifying an IP address, network number, and station number)	• Connecting with a target device on 1:1 basis by using one Ethernet cable without a hub • Connecting with a C Controller module of which IP address is unknown

Connection via a hub

The following shows the settings when performing connection via a hub.

Setting method

■Settings on C Controller module

1. Set the IP address of the C Controller module in "Own Node Settings" under "Basic Settings" of the module parameter. (☞ Page 159 Own node settings)

When connecting devices with specifying the network number and station number, specify them in "Own Node Settings" under "Basic Settings."

2. Set the connection configuration in "External Device Configuration" under "Basic Settings." (☞ Page 160 External device configuration)

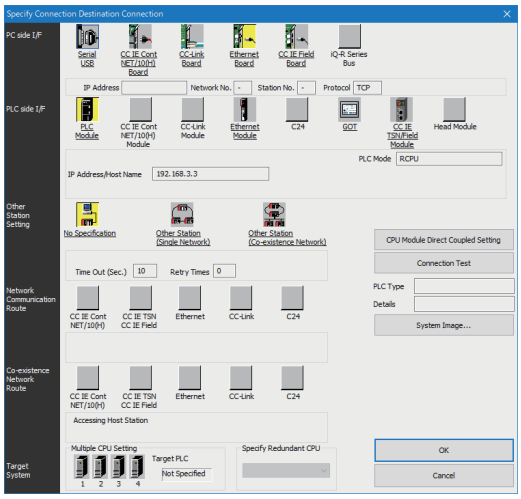
Point

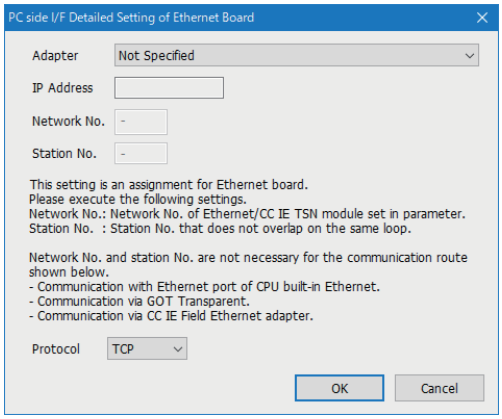
When connecting multiple MELSOFT products with TCP/IP connection, drag "MELSOFT Connection Module" from the "Module List" to "List of devices" or "Device map area" in "External Device Configuration" under "Basic Settings." (☞ Page 150 Using multiple MELSOFT connections (TCP/IP) simultaneously to connect a target device (with the same IP address as this product) to this product)

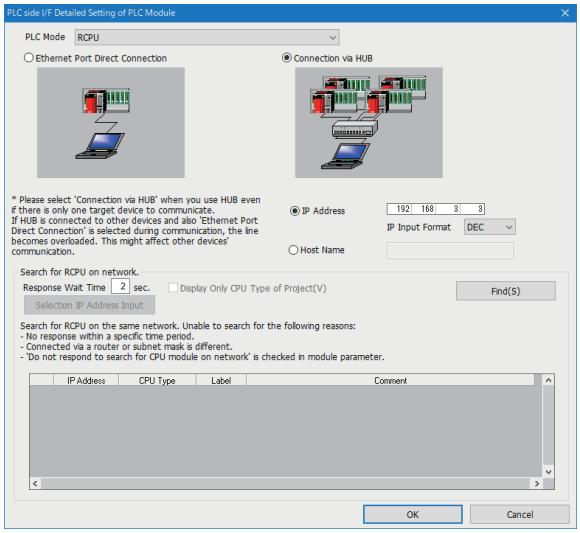
■Settings on CW Configurator side

Specify the connection route to a C Controller module in the "Specify Connection Destination Connection" screen.

[Online] ⇨ [Current Connection Destination]







1. Set "PC side I/F" to "Ethernet Board."
2. Double-click "Ethernet Board," and open the "PC side I/F Detailed Setting of Ethernet Board" screen.
3. Set the protocol.
4. Set the module to be connected to "PLC side I/F."
5. Double-click the icon of the set module to display the PLC side I/F detailed setting screen.
6. Select "Connection via HUB" for the connection method, and enter the station number and IP address or host name of the C Controller module.
7. Click the [OK] button once the setting is completed.
8. Set "Other Station Setting" and "Network Communication Route" as necessary.

Direct connection

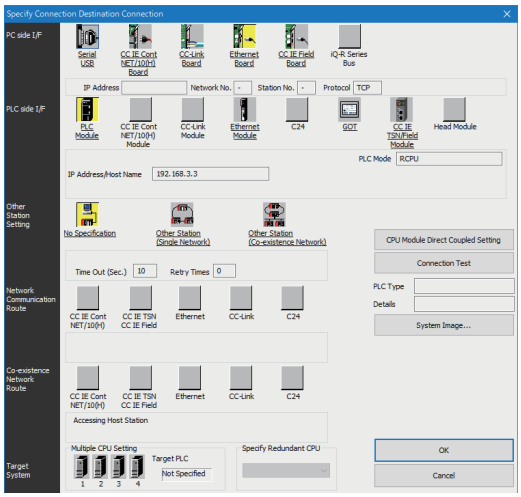
A C Controller module and an engineering tool can be directly connected with one Ethernet cable without using a hub. By connecting them directly, communication can be performed without setting an IP address or host name for the connection destination specification.

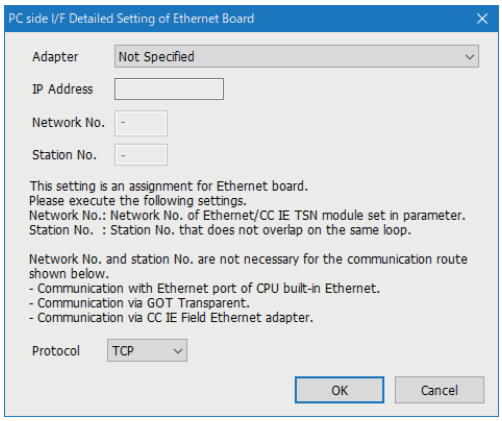
Point

To prohibit the direct connection using an Ethernet cable, set "Disable" in "Disable Direct Connection with MELSOFT" under "Security" under "Application Settings."

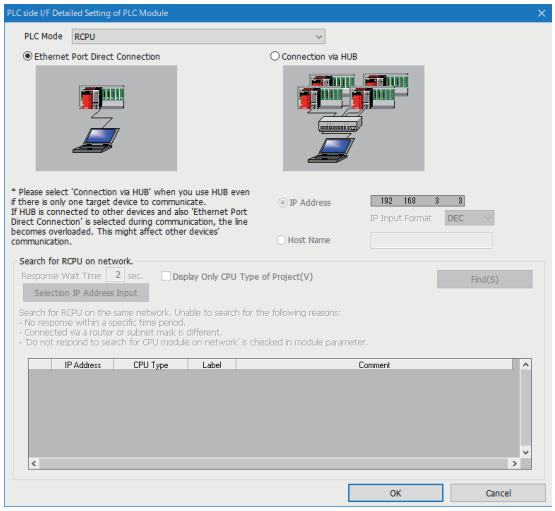
Setting method

[Online] ⇨ [Current Connection Destination]





1. Set "PC side I/F" to "Ethernet Board."
2. Double-click "Ethernet Board," and open the "PC side I/F Detailed Setting of Ethernet Board" screen.
3. Set the protocol.



4. Set the module to be connected to "PLC side I/F."
5. Double-click the icon of the set module to display the PLC side I/F detailed setting screen.
6. Select "Ethernet Port Direct Connection" for the connection method.
7. Click the [OK] button once the setting is completed.

Point

When connecting directly with a C Controller module, the setting above can also be performed by clicking the [CPU Module Direct Coupled Setting] button in the "Specify Connection Destination Connection" screen.

Precautions

■Connection with LAN

Do not connect with LAN and set the direct connection. Since the data will be sent to all the target devices on the LAN, this may increase the line load and affects communication with other target devices.

■Connections that are not a direct connection

- Do not use a configuration in which C Controller module and target devices are connected via hub. Direct connection will not be established when the devices are connected via hub.
- If two or more Ethernet ports are set to "Enable" in the network setting on the personal computer, a direct connection cannot be established. Review the setting of the personal computer so that only the Ethernet port for the direct connection is set to "Enable," and the other Ethernet ports are set to "Disable."

■Conditions that cannot communicate directly

A direct connection may not be established if any of the following conditions are met. In this case, review the settings.
The examples of the error occurrence are as follows:

- When all the bits of the C Controller module-side IP address that correspond to 0 part of the personal computer-side subnet mask are ON or OFF

Item	Description
IP address of a C Controller module	64.64.255.255
IP address of a personal computer	64.64.1.1
Subnet mask of a personal computer	255.255.0.0

- When all the bits of the C Controller module-side IP address that correspond to the host address of each class of the personal computer-side IP address are ON or OFF

Item	Description
IP address of a C Controller module	64.64.255.255
IP address of a personal computer	192.168.0.1
Subnet mask of a personal computer	255.0.0.0

Point

IP addresses of each class are as follows:

- Class A: 0.0.0.0 to 127.255.255.255
- Class B: 128.0.0.0 to 191.255.255.255
- Class C: 192.0.0.0 to 223.255.255.255

Host addresses of each class are the '0' parts below.

- Class A: 255.0.0.0
- Class B: 255.255.0.0
- Class C: 255.255.255.0

■Using multiple MELSOFT connections (TCP/IP) simultaneously to connect a target device (with the same IP address as this product) to this product

Target devices with the same IP address may not be able to connect to a C Controller module even when setting multiple MELSOFT connections (TCP/IP) in "External Device Configuration."

To perform this connection, set the following setting for the connection destination of MELSOFT products.

Condition	Setting method
All the MELSOFT products to be connected simultaneously can specify an Ethernet board adapter as a connection destination.	Specify the same Ethernet board adapter for the connection destination.
A MELSOFT product which cannot specify an Ethernet board adapter for a connection destination is included in the MELSOFT products to be connected simultaneously.	Specify "Not Specified" for the Ethernet board adapter.*1

*1 For MELSOFT products which cannot specify an adapter, the setting will be the same as that of when "Not Specified" is specified.

13.2 SLMP Communication Function

This function reads or writes device data, and performs the operation monitoring and data analysis in a C Controller module by accessing the module from an external device.
For details on SLMP, refer to the following:
SLMP Reference Manual

Accessible ranges

- The connected C Controller module can only be accessed.
- In a multiple CPU system, the access to other CPUs which are not connected with Ethernet, and communication with other stations via C Controller module cannot be performed.

Data communication frames/data codes

The available data communication frames/data codes that can be used for C Controller modules are as follows.
○: Applicable, ×: Not applicable

Frame	Data code	Applicability
4E frame	ASCII code	×
	Binary code	×
QnA compatible 3E frame	ASCII code	×
	Binary code	○
A compatible 1E frame	ASCII code	×
	Binary code	×

Setting method

The setting method is as shown below.
Set the connection configuration in "External Device Configuration" under "Basic Settings." (☞ Page 160 External device configuration)

1. Select "SLMP Connection Module" from the "Module List," and drag and drop it to "List of devices" or "Device map area."
2. Set the other items to the connection as necessary.

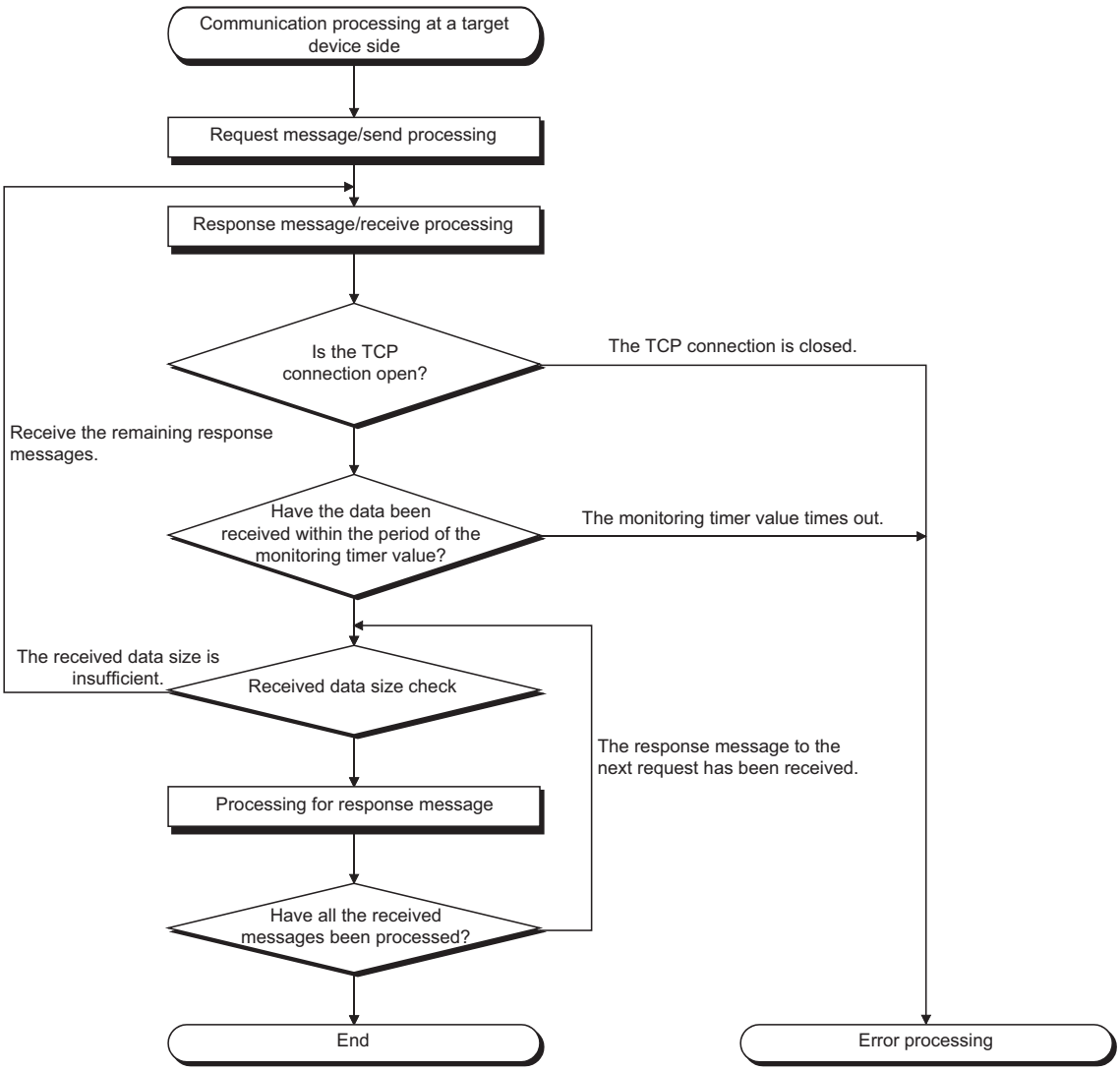
Point

- If a new request message is sent before the response message has been returned for the previous request message to one UDP port, the new request message will be discarded.
- When multiple own station port numbers are set for UDP, it will be regarded as one setting. When performing communication with the same own station port number as multiple target devices, select TCP as a protocol.

Error codes at communication

For the error codes, refer to the following:
SLMP Reference Manual

Receive processing of response message



Point

For Ethernet communication, the TCP socket function (socket function) is used.
When data has been sent from the sending side by calling the SEND function once, the receiving side is required to call one or more RECV function. (The ratio of the SEND function and the RECV function is not 1 to 1.) Therefore, the receive processing above is necessary for the program processing for the target device.

Command list

The available commands for C Controller modules are as follows.

Item		Command	Subcommand	Description
Type	Operation			
Device	Read	0401	00□1	Reads values from bit devices (consecutive device number) in 1-point units.
			00□0	• Reads values from bit devices (consecutive device number) in 16-point units. • Reads values from word devices (consecutive device number) in 1-word units.
			00□3	Reads values from bit devices (consecutive device number) in 1-point units.
			00□2	• Reads values from bit devices (consecutive device number) in 16-point units. • Reads values from word devices (consecutive device number) in 1-word units.
	Write	1401	00□1	Writes values to bit devices (consecutive device number) in 1-point units.
			00□0	• Writes values to bit devices (consecutive device number) in 16-point units. • Writes values to word devices (consecutive device number) in 1-word units.
			00□3	Writes values to bit devices (consecutive device number) in 1-point units.
			00□2	• Writes values to bit devices (consecutive device number) in 16-point units. • Writes values to word devices (consecutive device number) in 1-word units.
	Read Random	0403	00□0	Reads values from word devices in 1-word units or 2-word units by specifying device number. Nonconsecutive device numbers can be specified.
			00□2	Reads values from word devices in 1-word units or 2-word units by specifying device number. Nonconsecutive device numbers can be specified.
	Write Random	1402	00□1	Writes values to bit devices in 1-point units by specifying device number. Nonconsecutive device numbers can be specified.
			00□0	• Writes values to bit devices in 16-point units by specifying device number. Nonconsecutive device numbers can be specified. • Writes values to word devices in 1-word units or 2-word units by specifying device number. Nonconsecutive device numbers can be specified.
			00□3	Writes values to bit devices in 1-point units by specifying device number. Nonconsecutive device numbers can be specified.
			00□2	• Writes values to bit devices in 16-point units by specifying device number. Nonconsecutive device numbers can be specified. • Writes values to word devices in 1-word units or 2-word units by specifying device number. Nonconsecutive device numbers can be specified.
	Read Block	0406	00□0	Reads data by specifying multiple blocks as regarding one block as n-point of word devices or bit devices (1-point is 16-bit). Nonconsecutive device numbers can be specified.
			00□2	
	Write Block	1406	00□0	Writes data by specifying multiple blocks as regarding one block as n-point of word devices or bit devices (1-point is 16-bit). Nonconsecutive device numbers can be specified.
			00□2	
Extend Unit	Read	0601	0000	Reads data in the buffer memory of an intelligent function module.
	Write	1601	0000	Writes data in the buffer memory of an intelligent function module.
Remote Control	Remote Run	1001	0000	Performs remote RUN to the access target module.
	Remote Stop	1002	0000	Performs remote STOP to the access target module.
	Remote Pause	1003	0000	Performs remote PAUSE to the access target module.
	Remote Reset	1006	0000	Performs remote RESET to the access target module.
	Read Type Name	0101	0000	Reads model and model code of the access target module.
File	Read Directory/ File	1810	0040	Reads information list of the file.
	Search Directory/ File	1811	0040	Reads fine number, file size, and existence of the specified file.
	New File	1820	0040	Reserves the storage area in the specified file.
	Delete File	1822	0040	Deletes files.
	Copy File	1824	0040	Copies the specified file.
	Change File State	1825	0040	Changes the file attribute.
	Change File Date	1826	0040	Changes the file creation date.
	Open File	1827	0040	Locks the file in order that the file content is not changed form other devices.
	Read File	1828	0000	Reads the content of a file.
	Write File	1829	0000	Writes content to a file.
	Close File	182A	0000	Unlocks a file with open processing.

■Processing points for Test (random write)

Set the processing points within the value which can be acquired with the following formula.

(Number of word access points) × 12 + (Number of double word access points) × 14 ≤ 1920

- For bit devices, 1-point equals 16-bit for word access. As for double word access, 1-point equals 32-bit.
- For word devices, 1-point equal 1-word for word access. As for double word access, 1-point equal 2-word.

Available devices

The available devices are as follows.

Device		Device code*1	Device number range	
I/O device	Input	9CH	0H to FFFH	Hex
	Output	9DH	0H to FFFH	Hex
Internal user device	Internal relay	90H	Device number within the range in the C Controller module of the target station can be specified.	Dec
	Data register	A8H		Dec
	Link relay	A0H		Hex
	Link register	B4H		Hex
Internal system device	Special relay	91H	0 to 2047	Dec
	Special register	A9H	0 to 2047	Dec

*1 ASCII code cannot be used.

13.3 FTP Function

This function accesses a file in a C Controller module from a target device with the FTP client function since the FTP server function is supported by a C Controller module.

File operation

The following operations can be performed for a file in a C Controller module from a target device with the FTP client function.

■Reading files (Download)

Use this when storing a file in a C Controller module on the target device side.

■Writing files (Upload)

Use this when registering a file stored on the target device side to a C Controller module.

■Browsing and deleting files and folders

Use this when browsing and deleting a file and folder in a C Controller module from the target device side.

FTP function usage

For using the FTP function, the FTP server needs to be set.

☞ Page 162 FTP server settings

Precautions

■Specifications of FTP client

For the specifications of the FTP client installed on the target device, refer to the manual for the target device.

■Operation while accessing file

Do not perform any of the following operations while accessing a file. The file may be corrupted.

- Turning the power OFF and ON, or resetting the C Controller module
- Inserting and removing an SD memory card

■Reconnection after timeout

If a timeout error occurs during file transfer, the TCP connection will be closed (disconnected).

Log in to the C Controller module again with the FTP client before restarting the file transfer.

■File transmission time

The file transfer processing time will differ depending on the following causes.

- Load rate of Ethernet line (line congestion)
- Number of connections to be used simultaneously (processing of other connections)
- System configuration

■Number of simultaneous connections

- Up to 10 target devices (FTP client) can log in to a C Controller module. If connecting from the 11th FTP client in the state where 10 target devices have logged in, an error will occur without establishing the connection.
- If UDP communication is performed during file transfer with FTP, an error such as timeout may occur. Either communicate after the file transfer or communicate with TCP.

■File write

- Files with the read-only attribute and files that are locked from other devices/functions cannot be written. Doing so may cause a write error.
- The write files cannot be transferred if the SD memory card is write-protected. Doing so may cause a write error.
- When writing a large file to the SD memory card, change the CPU operating status to STOP.

■File deletion

Determine the timing for deleting the files for the entire system including the C Controller module and peripheral devices.

13.4 Time Setting Function

This function sets the time on a C Controller module automatically by acquiring time information from the time information server (SNTP server) on the network.

Point

- The time information acquired from the time information server is Coordinated Universal Time (UTC). The acquired UTC time information is adjusted according to the time zone setting of the CPU module, and is set to the CPU module.
- In the multiple CPU environment, set the time only for the CPU No.1. If the time is set for the CPU modules other than the CPU No.1, the clock data in the CPU No.1 is set automatically to CPU No.2 to No.4.

Execution timing of time setting

The time setting is performed at any of the following timing.

- At turning the power OFF and ON, or resetting the C Controller module
- At specified time intervals
- At specified time
- At arbitrary timing using a program*1

*1 Acquire the clock information with the VxWorks function (sntpcTimeGet), and then execute the C Controller module dedicated function (CCPU_SetRTC).

Setting method

Set "Time Setting" under "Application Settings." (☞ Page 163 Time setting)

Precautions

■Time information server

Install an SNTP server on the LAN to be connected.

■Delay by communication time

The time set in the module may be delayed by the time required to communicate with the SNTP server. For a high-accuracy time setting, specify an SNTP server on the network that is as close to the module as possible.

■Available period

The period within the range from 1980 to 2079 can be used.

13.5 Telnet Function

This function executes the Shell command with a Telnet tool in a personal computer for a TCP/IP network.
This allows simple remote debugging (such as task information display and memory dumping) of a C Controller module without using CW Workbench.

Telnet function usage

For using the Telnet function, the Telnet server needs to be set.
☞ Page 164 Telnet server settings

Remote debugging with serial communication

Remote debugging using serial communication can be performed same as Telnet connection.
The standard I/O destination can be switched to the serial communication side by executing the following commands from Telnet Shell or with STARTUP.CMD.

```
fd = open("/tyCo/0",2,0)
ioctl(fd,4,115200)
ioctl(fd,3,0x7F)
ioGlobalStdSet(0,fd)
ioGlobalStdSet(1,fd)
ioGlobalStdSet(2,fd)
```

Precautions

■Available Shell commands

Shell commands of CW Workbench can be used. For details on the Shell commands, refer to the manual for VxWorks.

■Number of connections

The same C Controller module cannot be connected by using multiple Telnet tools. Connect a Telnet tool to a C Controller module on a 1:1 basis. When connecting another Telnet tool, make sure to close (disconnect) the Telnet tool being connected.

■Shell command

Shell commands entered from the Telnet tool operate on a task of priority 1.
Note that the following, when executing the command. System errors/stop (such as watchdog timer error) may occur in a C Controller module.

- Make sure to check the command specifications before executing commands which occupy the CPU processing.
- For rebooting VxWorks, reset the C Controller module, or turn the power OFF and ON. Do not reboot VxWorks by executing the command (reboot) of VxWorks or pressing Ctrl + X.
- Before executing a command in which arguments are included, make sure to check the command specifications/argument specifications. (When executed without specifying those arguments, with the result that 0 is specified to an argument.) Do not execute the close command with no argument specified. By doing so, a resource that is reserved in the VxWorks system will be closed. When a command that shows the status of the module, such as the show command, is executed, the module will be in the interrupt-disabled state for a long period of time, and any processing called from an interrupt routine is not executed. As a result, an interrupt which occurs at the fixed interval may be delayed.

■Message display on Shell

A message issued by VxWorks during Telnet connection may be displayed on Shell. For the message of VxWorks, refer to the manual and help for VxWorks.

■Timeout

When the line is disconnected during Telnet connection, it will take 30 seconds before Telnet connection (TCP) times out on the C Controller module side. Telnet cannot be reconnected until it times out.

Timeout time can be changed by the command provided by VxWorks.

```
ipcom_sysvar_set("iptcp.KeepIdle", "XX", 1);
ipcom_sysvar_set("iptcp.KeepIntvl", "YY", 1);
ipcom_sysvar_set("iptcp.KeepCnt", "ZZ", 1);
ipcom_ipd_kill ("iptelnets");
ipcom_ipd_start ("iptelnets");
```

Timeout time = iptcp.KeepIdle value + (iptcp.KeepIntvl value × iptcp.KeepCnt value)

- iptcp.KeepIdle: Interval from line disconnection to the first retry
- iptcp.KeepIntvl: Interval between retries
- iptcp.KeepCnt: Number of retries
- XX, YY: Specify the time (in seconds). (When '0' is specified, no timeout will occur.)
- ZZ: Specify the number of retries.

The following shows the procedure to change the timeout time of a C Controller module in operation.

1. Establish a Telnet connection to a C Controller module with a Telnet tool.
2. Execute the commands above from the shell command of the Telnet tool to change the timeout value.
3. Reboot the Telnet server.
4. Close (disconnect) the Telnet connection.
5. Establish a Telnet connection to a C Controller module with a Telnet tool again.

When changing the timeout time at the startup of a C Controller module, follow the procedure shown below.

1. Describe the commands above in the script file (STARTUP.CMD).
2. Turn the power of the C Controller module ON.

13.6 Parameter Settings

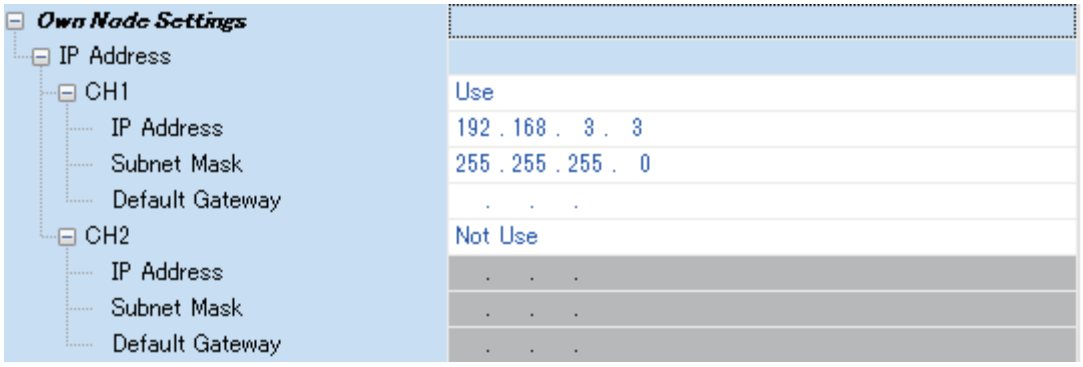
This section shows the settings required for communication with a target device in a C Controller module.

Own node settings

Set the own node required for Ethernet communication.

[Module Parameter] ⇨ [Basic Settings] ⇨ [Own Node Settings]

Window



Displayed items

—: Not set

Item			Description	Setting range	Default
IP Address	CH1	—	Set whether to use the Ethernet port (CH1).	• Not Use • Use	Use
		IP Address	Set the same class and subnet address as the target device to be communicated with. Set the IP address so that CH1 belongs to the different network from CH2.	• 0.0.0.1 to 223.255.255.254	192.168.3.3
		Subnet Mask	Set the subnet mask pattern of the default gateway when IP address of the default gateway is set and perform communication with a target device on another network via a router. All the devices on a sub network should have the same subnet mask. This setting is not required for communication in a single network.	• 128.0.0.0 to 255.255.255.252	255.255.255.0
		Default Gateway	Set the IP address for the device (default gateway) which is relayed for the access to the target device on other network. Set the value that satisfies the following conditions. • The IP address class is any of A, B, and C. • The subnet address of the default gateway is the same as that of the C Controller module on the own station. • All host addresses are neither '0' nor '1.'	• — (Blank) • 0.0.0.1 to 223.255.255.254	—
	CH2	—	Set whether to use the Ethernet port (CH2).	• Not Use • Use	Not Use
		IP Address	Same as CH1	• 0.0.0.1 to 223.255.255.254	—
		Subnet Mask		• 128.0.0.0 to 255.255.255.252	
		Default Gateway		• — (Blank) • 0.0.0.1 to 223.255.255.254	

Point

When using both CH1 and CH2, the IP address and subnet mask should be set with different values for CH1 and CH2, respectively.

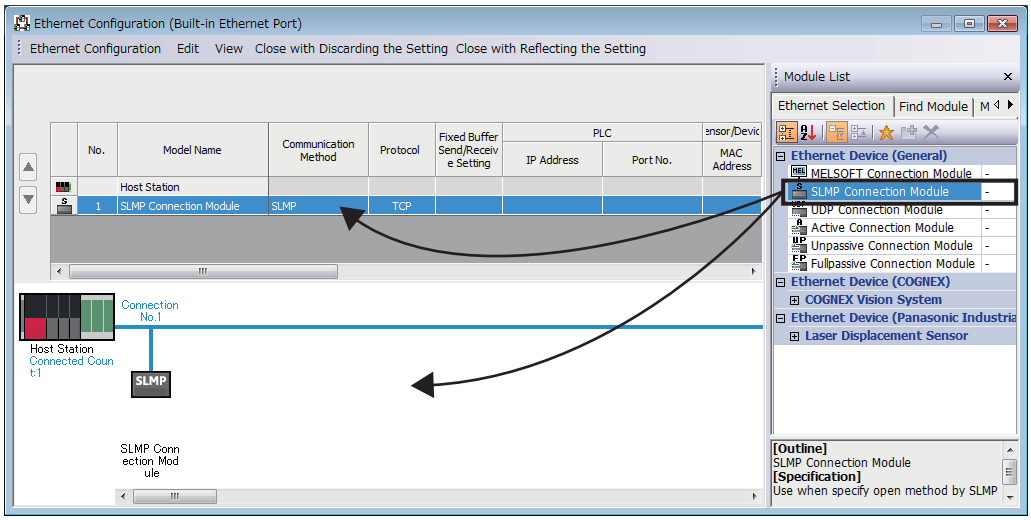
External device configuration

Select the method and protocol used for communication with a target device.

[Module Parameter] ⇨ [Basic Settings] ⇨ [External Device Configuration] ⇨ [<Detailed Setting>]

Setting method

1. Select the target device to be connected in "Module List" and drag it to "List of devices" or "Device map area."



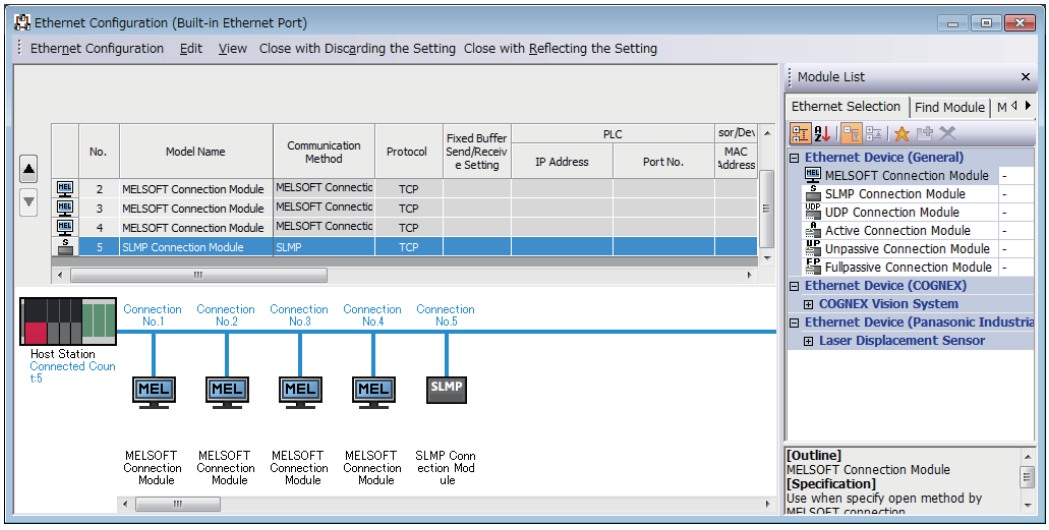
2. Set each item.

3. Select [Close with Discarding the Setting] and close "External Device Configuration."

Point

In setting "External Device Configuration," setting the target device in order starting from the connection No.1 is required. To use the specific connection number, set "MELSOFT Connection Module" to the connection number which is not used.

• When using only the connection No.5



Restriction

C Controller module does not support the Active Connection Module, Unpassive Connection Module, Fullpassive Connection Module, or UDP Connection Module.

Setting items

The following shows the setting items for "External Device Configuration."

Item		Description	Setting range
No.		Connection numbers for distinguishing the settings for each user connection.	It is set in ascending order from 1 within the following range. C Controller module: 1 to 16
Model Name		Names of target devices are displayed.	—
Communication Method		Set the communication method with the target device.	• Broadcast Send • Broadcast Receive • Fixed Buffer (Procedure Exist)*1 • Fixed Buffer (No Procedure)*1 • Random Access Buffer*1 • Predefined Protocol*2 • Socket Communication • MELSOFT Connection • SLMP
Protocol		Select the communication protocol for the target device.	• TCP*3 • UDP*4
Fixed Buffer Send/Receive Setting		Not required.	—
Programmable controller	IP Address	Not required.	—
	Port No.*5	Set the port number for each connection of Ethernet-equipped module.	1 to 4999, 5010 to 65534 (Default: blank)
Sensor/Device	MAC Address	Not required.	—
	Host Name	Not required.	—
	IP Address	Set the IP address of the target device.	0.0.0.1 to 223.255.255.254 (Default: blank)
	Port No.	Set the port number of the target device. To receive data from all the port numbers, set '65535.'	1 to 65534, 65535 (Default: blank)
	Subnet Mask	Not required.	—
	Default Gateway	Not required.	—

*1 C Controller module cannot set this item.

*2 Although it may be set as the communication method, C Controller module does not support it.

*3 The module can be connected up to the number equivalent to ((the maximum number of connections in the setting for external device configuration) - (the set number of connections) + 1).

*4 As simultaneous communication from multiple connected target devices can overload network, communication may not be established in such overloaded network conditions.

*5 The own station port number 1 to 1023 are the number for reserved in general (WELL KNOWN PORT NUMBERS), and the port number 61440 to 65534 are the number used for other communication functions. Using the port number within the range of 1024 to 4999 or 5010 to 61439 is recommended.

■Alive check with the KeepAlive function

When the protocol setting is TCP, perform the alive check using the KeepAlive function.

A message for alive check is sent after 22 seconds have passed since the last message received from a target device, and whether or not the response is returned from the target device is checked. When no response is returned, a message for alive check is sent every one second. When no response is confirmed for eight seconds (30 seconds since the last received message), the connection is terminated as being determined that the target device does not exist.

Point

If the target device does not support the TCP KeepAlive function (response to ACK message for KeepAlive), the connection may be terminated.

■TCP retransmission processing

In TCP connection, the retransmission processing is performed for the number of retransmission and at the retransmission interval shown below when the TCP protocol ACK response is not returned from a target device against transmission. When no TCP protocol ACK response is returned in 60 seconds after the last retransmission, the connection is terminated as being determined the target device has an error.

- Number of retransmission: 12 times[*1]
- Retransmission interval[*2]: (The number of retransmission × the number of retransmission × RTO[*3]) ÷ 1024 (seconds)

*1 After 30 seconds from establishment of the connection, destination device error is assumed and the connection is terminated even if the number of resends does not reach to 12 times.

*2 The maximum value of the retransmission time is 60 seconds.

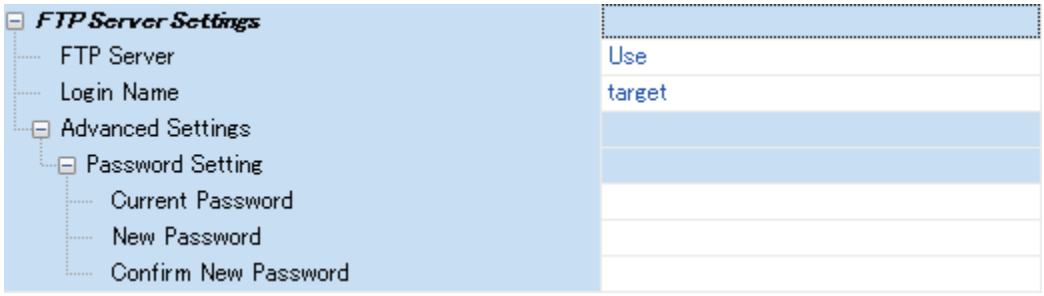
*3 RTO (retransmission timeout) is a value increases exponentially based on RTT (round-trip time).

FTP server settings

Set the FTP function.

[Module Parameter] ⇨ [Application Settings] ⇨ [FTP Server Settings]

Window



Displayed items

Item		Description	Setting range	Default
FTP Server		Select whether to use the FTP function of a C Controller module.	• Not Use • Use	Use
Login Name		Set the login name to be used for file transfer request (login) from the target device.	Up to 12 characters	target
Advanced Settings	Password Setting	Set the password to be used for file transfer request (login) from the target device.	8 to 32 characters	password

Password setting

■Current password

Set a password when logged in to an FTP server (C Controller module) from an FTP client.

Point

Change the password when using the FTP function.
Otherwise, unauthorized access may be caused.

■New password and confirmation password

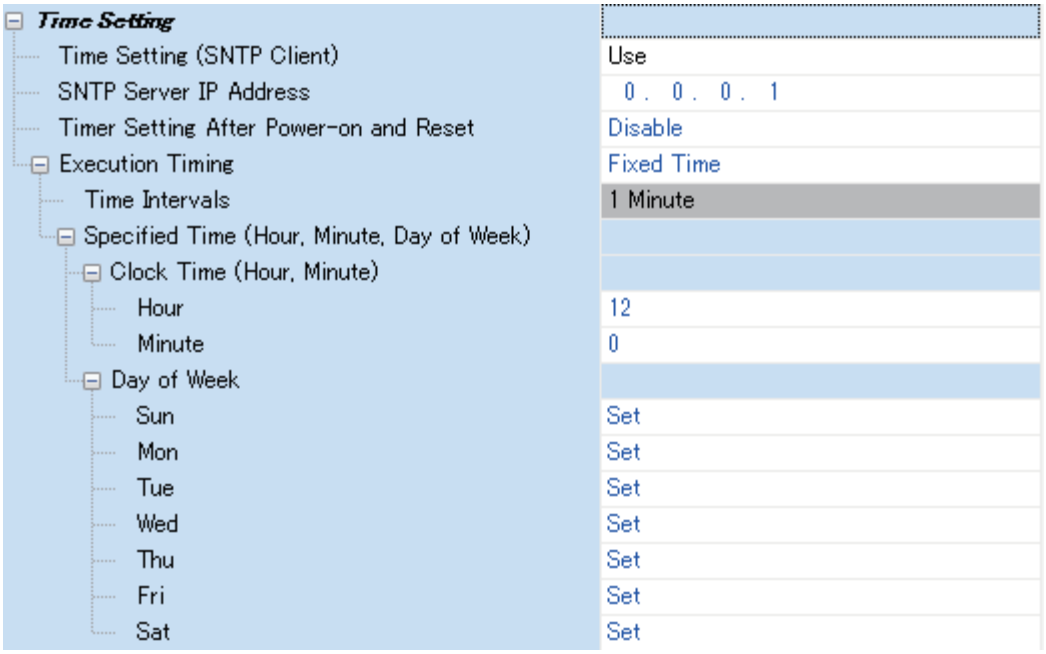
To change the password, enter a new password in "New Password" and "Confirm New Password."

Time setting

Set the time setting function (SNTP client).

[Module Parameter] ⇨ [Application Settings] ⇨ [Time Setting]

Window



Displayed items

Item		Description	Setting range	Default
Time Setting (SNTP Client)		Select whether to use the FTP function of a C Controller module.	• Not Use • Use	Not Use
SNTP Server IP Address		Set the IP address of the SNTP server.	0.0.0.1 to 223.255.255.254	0.0.0.1
Timer Setting After Power-on and Reset		Select whether to execute the time setting function after turning the power ON or resetting the C Controller module.	• Disable • Enable	Disable
Execution Timing	—	Set the execution timing of the time setting.	• Fixed Time • Time Intervals	Fixed Time
	Specified Time Intervals	Set the time interval (minute) to execute time setting if you select the "Specified Time Intervals."	1 to 1440	1 minute
	Specified Time (Hour, Minute, Day of Week)	Set the time (hour/minute/day of the week) to execute time setting if you select the "Specified Time."	—	

Point

- Connect only one SNTP server on the system.
- The output time will be the same even if multiple modules acquires the time from the same SNTP server.

Execution timing

■Specified time (hour, minute, day of week)

Set an execution time within the following range in "Clock Time (Hour, Minute)."

Unit	Setting range
Hour	0 to 23
Minute	0 to 59

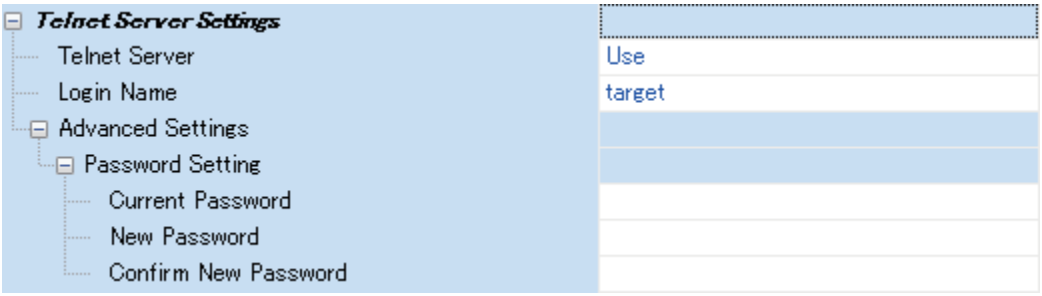
To specify the day of the week to perform the time setting, select "Not Set" for the day on which time setting is not performed under "Day of Week." Make sure that at least one day of the week is set to "Set." An error occurs if "Not Set" is selected for all the days.

Telnet server settings

Set the Telnet server.

[Module Parameter] ⇨ [Application Settings] ⇨ [Telnet Server Settings]

Window



Displayed items

Item		Description	Setting range	Default
Telnet Server		Set whether to use the Telnet function of a C Controller module.	• Not Use • Use	Use
Login Name		Set the login name to be used for file transfer request (login) from the target device.	Up to 12 characters	target
Advanced Settings	Password Setting	Set the password to be used for file transfer request (login) from the target device.	8 to 32 characters	password

Password setting

■Current password

Enter the current password for login to a C Controller module.

Point

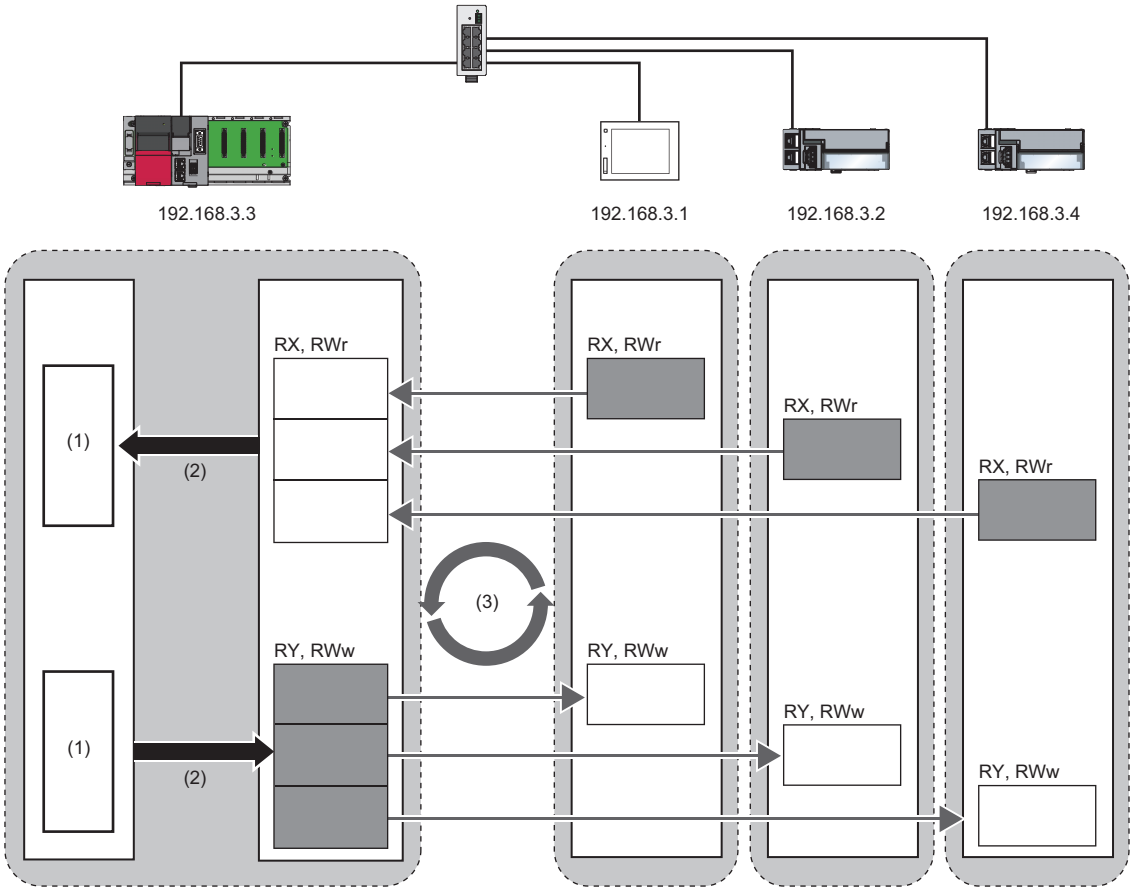
Change the password when using the Telnet function.
Otherwise, unauthorized access may be caused.

■New password and confirmation password

To change the password, enter a new password in "New Password" and "Confirm New Password."

14 CC-Link IE Field Network Basic FUNCTION

This function performs data communication with a device connected to CC-Link IE Field Network Basic.



(1) Device
(2) Link refresh
(3) Link scan

Point

This manual does not describe the following information that is related to the CC-Link IE Field Network Basic function.

- Specifications
- Function list
- Procedure before operation
- System configuration

For details, refer to the following:

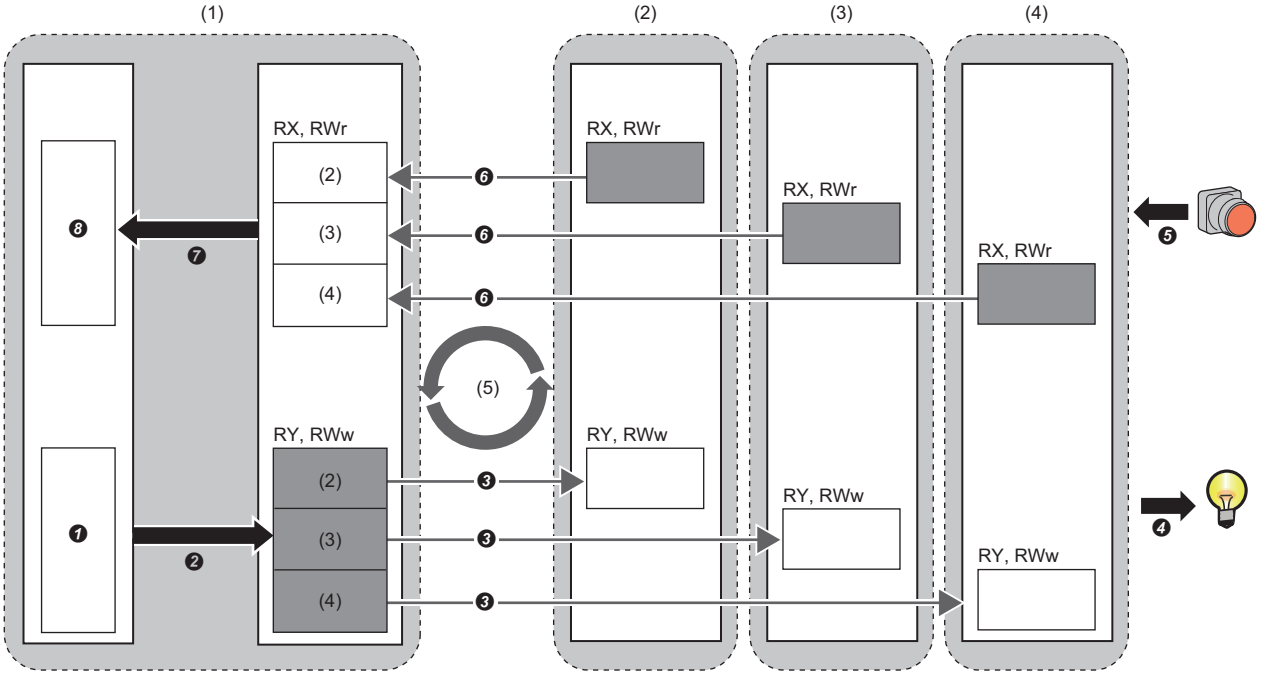
CC-Link IE Field Network Basic Reference Manual

14.1 Cyclic Transmission Function

This function performs data communication periodically between a master station and remote stations by using link devices.

Data flow and link device assignment

The following figure shows the data flow between a master station and remote stations using link devices.



: Area where data is sent to other stations

(1) Master station
(2) Remote station 1*1
(3) Remote station 2*1
(4) Remote station 3*1
(5) Link scan

*1 The remote station order is the same as the order set in the network configuration setting. (☞ Page 179 Network configuration settings)

• Output from a master station

❶ A device of the master station turns ON.

❷ The device status of the master station is stored to link devices (RY and RWw) of the master station by link refreshes.

❸ The status of link devices (RY and RWw) of the master station is stored to link devices (RY and RWw) of remote stations by link scans.

❹ The status of the link devices (RY and RWw) of the remote stations is output to an external device.

• Input from remote stations

❺ The status of the external device is stored to the link devices (RX and RWr) of the remote stations.

❻ The status of the link devices (RX and RWr) of the remote stations is stored to the link devices (RX and RWr) of the master station by link scans.

❼ The status of the link devices (RX and RWr) of the mater station is stored to the link devices of the master station by link refreshes.

❽ The device of the master station turns ON.

Setting method

Assign link devices in "Network Configuration Settings." (☞ Page 179 Network configuration settings)
Assign link refreshes in "Refresh Settings." (☞ Page 182 Refresh settings)

Point

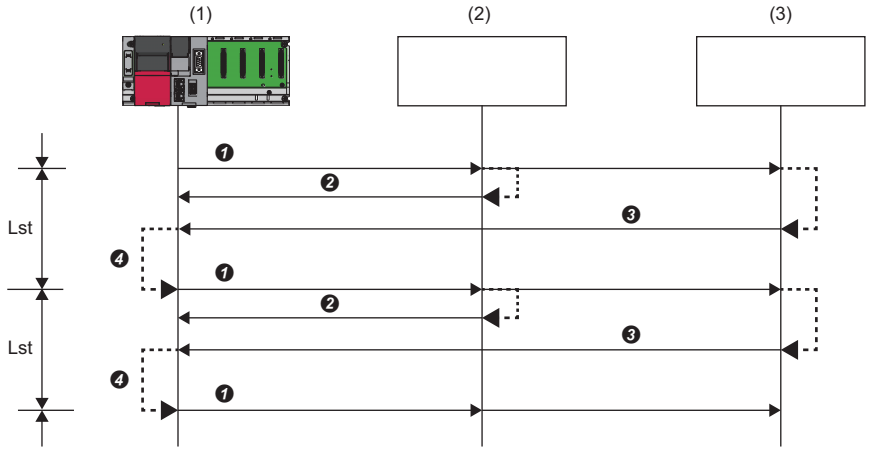
Cyclic transmission is performed for a group which consists of up to 16 stations.
For assignments of link devices and the one for link refreshes, however, the group number is not required to be taken into consideration.

Data flow of cyclic transmission

The following figure shows the data flow of cyclic transmission.

■Basic operation

A master station sends a request to all remote stations (including Ethernet devices within the same network address).
After sending requests to all remote stations, the master station starts another link scan after the set link scan time elapsed.
The link scan refers to the operation from sending a request to sending another request, and the link scan time refers to the time required for a link scan. (☞ Page 171 Operation of link scans)



Lst: Link scan time (link scan)
(1) Master station
(2) Remote station 1
(3) Remote station 2

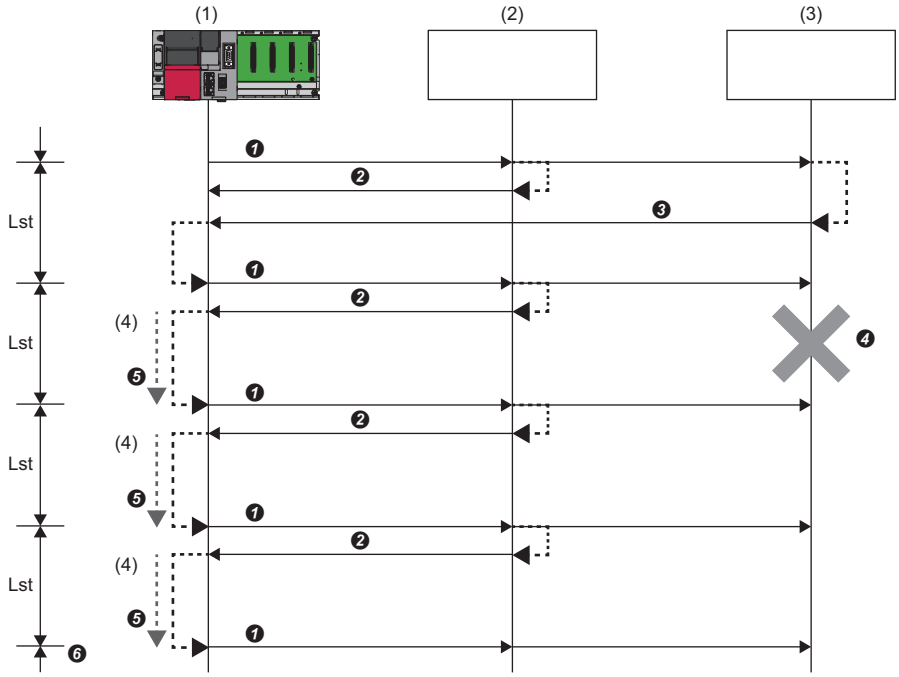
❶ The master station sends a request to the remote station 1 and remote station 2.
❷ The remote station 1 returns a response to the master station.
❸ The remote station 2 returns a response to the master station.
❹ After the link scan time set is elapsed, the master station starts sending another request.

Point

- When the master station sends a request (when a cyclic transmission is started) to all remote stations, 'Cyclic transmission status' (SM1536) turns ON, and when the master station receives responses from each remote station, a bit corresponding to the station number of remote stations, which have sent a response of 'Cyclic transmission status of each station' (SD1536 to SD1539), turns ON.
- When the master station starts cyclic transmission, if no response has been received from a remote station, the remote station is not regarded as an error station. ('Data link status' (SM1540) does not turn ON.) In addition, the transmission status of CC-Link IE Field Network Basic diagnostics becomes "Unfixed."
- When remote stations are divided into groups in the group number setting, each of the groups performs cyclic transmission. (☞ Page 172 Group number setting)

■No response received from a remote station

If the master station is unable to receive a response from a remote station due to power interruption or cable disconnection of the remote station, the master station waits for a response from the remote station within the timeout time set in the link scan setting. If no response is received within the specified number of times for disconnection detection, the master station disconnects the remote station.



(Remote station disconnection detection setting: Timeout time = 500 ms, number of times = 3 times)
Lst: Link scan time (link scan)
(1) Master station
(2) Remote station 1
(3) Remote station 2
(4) Additional 100 ms

❶ The master station sends a request to the remote station 1 and remote station 2.
❷ The remote station 1 returns a response to the master station.
❸ The remote station 2 returns a response to the master station.
❹ The remote station 2 cannot return a response to the master station due to an error such as power interruption or cable disconnection of the remote station.
❺ The master station waits for a response from the remote station 2 until the timeout time elapses, and then sends a request to the remote station 1 and remote station 2 because the master station cannot receive a response from the remote station 2. Note that the link scan time is extended by the timeout time.
❻ The master station disconnects the remote station 2 because the master station cannot receive a response from the remote station 2 within the specified number of times (three times) for the disconnection detection.
The operations of special relays and special registers when a remote station which did not send a response is disconnected are as follows:

Special relay/special register	Operation
'Cyclic transmission status' (SM1536)	The device remains ON.
'Cyclic transmission status of each station' (SD1536 to SD1539)	The bit corresponding to the station number of the disconnected remote station turns OFF.
'Data link status' (SM1540)	The device turns OFF and ON.
'Data link status of each station' (SD1540 to SD1543)	The bit corresponding to the station number of the disconnected remote station turns ON (indicating that the remote station is an error station).

Link scan time after timeout

If a timeout occurs due to the failure of a remote device, the link scan time will be changed depending on the setting value of the timeout time.

- When the timeout time is shorter than the link scan time, the module operates with the value set for "Link Scan Time" in "CC-Link IEF Basic Setting." (☞ Page 178 CC-Link IEF Basic setting)
- When the timeout time is longer than the link scan time, the module operates with a link scan time which is equivalent to the time set for "Time-out Period" in "Link Scan Setting." (☞ Page 180 Link scan setting)

Point

- To lessen the impact on communications with normal remote stations by transmission delay, modify a timeout time. (☞ Page 180 Link scan setting)
  For example, using the CC-Link IE Field Network Basic diagnostics, check the current link scan time (when all remote stations are operating normally), and then set a value, which is about five times as long as the link scan time, for the timeout time. (When the current link scan time is 10 ms, set 50 ms for the timeout time.)
- The delay in the link scan time caused by a timeout can be checked in 'Accumulated number of timeouts' (Un\G1063). (☞ Page 184 Acquiring diagnostic information of remote stations)

■An error response received from a remote station

The master station immediately disconnects the remote station regardless of the timeout time and the number of times for disconnection detection set in the link scan setting.

The operations of special relays and special registers when a remote station is disconnected are as follows:

Special relay/special register	Operation
'Cyclic transmission status' (SM1536)	The device remains ON.
'Cyclic transmission status of each station' (SD1536 to SD1539)	The bit corresponding to the station number of the disconnected remote station turns OFF.
'Data link status' (SM1540)	The device turns OFF and ON.
'Data link status of each station' (SD1540 to SD1543)	The bit corresponding to the station number of the disconnected remote station turns ON (indicating that the remote station is an error station).

Point

Some remote stations are equipped with a function that makes them disconnected without being an error station. For details, refer to the manual for the remote station used. (Since the remote station does not become an error station, 'Data link status' (SM1540) and 'Data link status of each station' (SD1540 to SD1543) do not change.)

Link refresh

This function automatically transfers data between devices and link devices of the master station.
Link refresh is performed for each link scan time. (☞ Page 178 Parameter Settings)

Setting method

A refresh target device is set in "Refresh Settings" under "CC-Link IEF Basic Setting." (☞ Page 182 Refresh settings)

Precautions

■Latched devices of a C Controller module

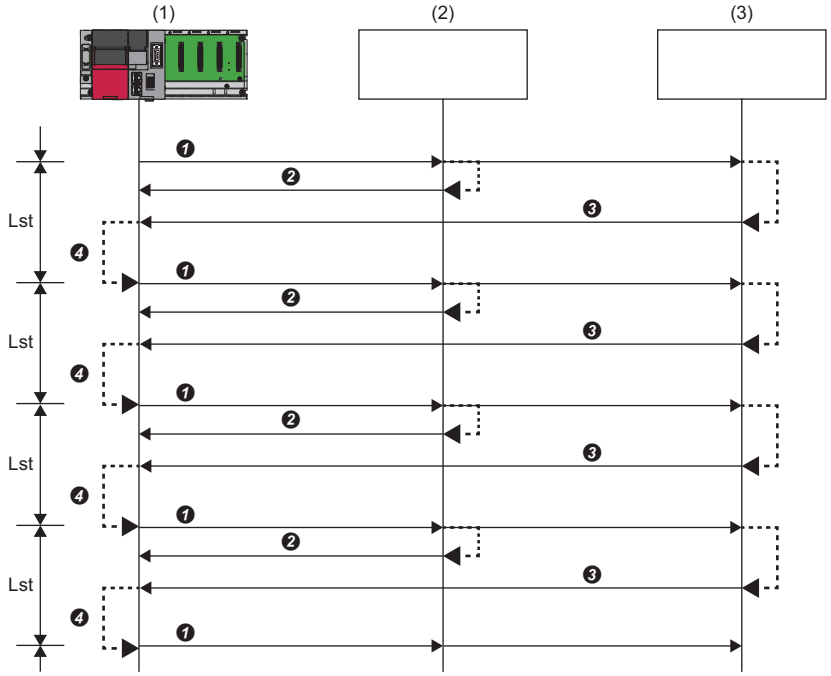
If the device data of a C Controller module which is being latched is cleared to '0' by turning the power OFF and ON or resetting the module, depending on the timing of link scan and link refresh, the latched data is output without clearing to '0' .
To prevent outputting device data of a C Controller module which is being latched, perform the following operation.

Item	Description
File register (R, ZR)	Clear the device 0 using the initial device value.

Operation of link scans

After sending requests to all remote stations and subsequently receiving responses from all the remote stations, the master station starts another link scan.

Link refresh is performed for each link scan time. (☞ Page 178 Parameter Settings)



Lst: Link scan time (link scan)
(1) Master station
(2) Remote station 1
(3) Remote station 2

❶ The master station sends a request to the remote station 1 and remote station 2.

❷ The remote station 1 returns a response to the master station.

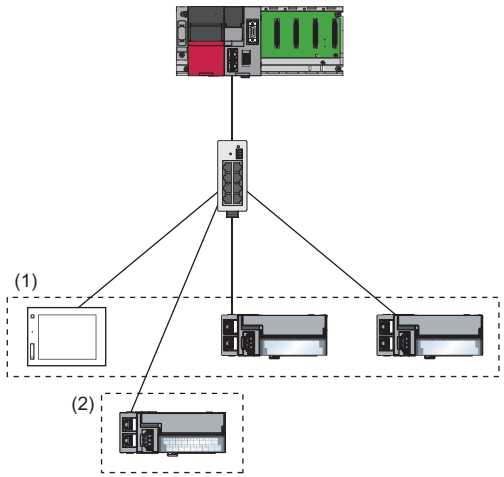
❸ The remote station 2 returns a response to the master station.

❹ After the link scan time set is elapsed, the master station starts sending another request.

Group number setting

This function divides remote stations into groups by setting a group number to each remote station and each of groups performs cyclic transmission.

By organizing groups separating remote stations with shorter response processing time from ones with longer response processing time, the differences of the reference response times of each remote station do not affect the cyclic transmission.



(1) Group No.1
(2) Group No.2

Point

- The total number of occupied stations for one group is 16 maximum.
- Remote stations can be divided into up to four groups.

How to organize groups

Organize groups considering the following.

■Dividing remote stations into groups

- Organizing two or more groups can configure a network with remote stations that occupy 17 or more stations in total.
- By dividing remote stations into groups with similar reference response time, the gap of the response time of each remote station does not badly affect the cyclic transmission. For details on the reference response time, refer to the manuals for remote stations used. Link scan times vary from group to group. The link scan time of each group is affected by a remote station that has the longest reference response time in a group. (☞ Page 167 Data flow of cyclic transmission)

■Merging remote stations into one group

- To perform operation among remote stations in cooperation, merge them into the same group.
- When the line load is high, merging remote stations into the fewest number of groups as possible according to the number of remote stations connected to the master station is recommended. For example, merge remote stations into one group if the remote stations are 16 or less. When two or more groups are organized, the master station sends requests to each of them. Since the packets of the cyclic transmissions performed for each group are sent on the line, the more groups are organized, the higher the line load becomes.

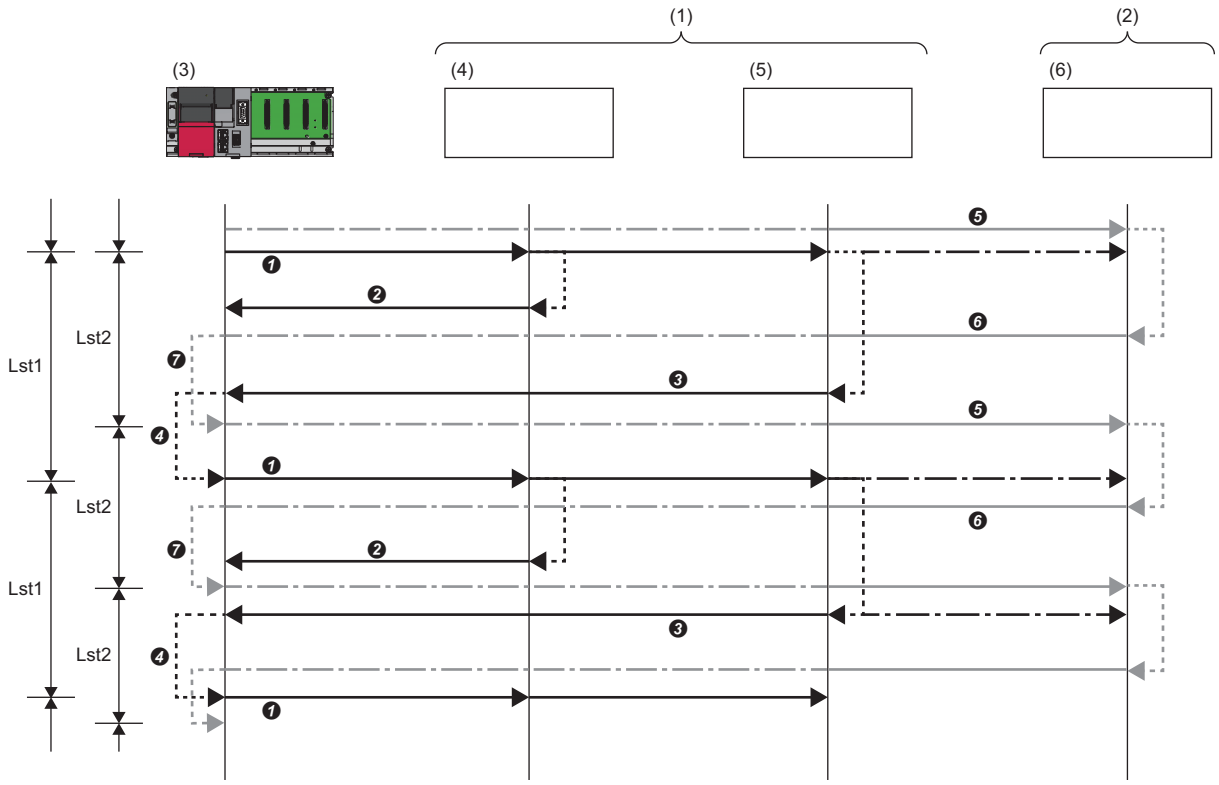
Data flow of cyclic transmission

The following figure shows the data flow of cyclic transmission when a group number is set.

■Basic operation

The master station sends requests, that vary from a group to group, to all remote stations. The remote stations receive multiple request messages from the master station; however, each remote station handles a request message for a group where each remote station belongs to.

After the link scan time set for each group elapsed, the master station starts sending another request to the group. Since the link scan setting can be configured for each group, the setting can be configured according to the response processing times of each group. (☞ Page 180 Link scan setting)



Lst1: Link scan time of group 1 (link scan)
Lst2: Link scan time of group 2 (link scan)
(1) Group No.1
(2) Group No.2
(3) Master station 1
(4) Remote station 1
(5) Remote station 2
(6) Remote station 3

❶ The master station sends requests to the remote station 1 and remote station 2 that belong to group 1.
❷ The remote station 1 returns a response to the master station.
❸ The remote station 2 returns a response to the master station.
❹ After the link scan time set to the group 1 is elapsed, the master station starts sending another request.
❺ The master station sends a request to the remote station 3 that belongs to group 2.
❻ The remote station 3 returns a response to the master station.
❼ After the link scan time set to the group 2 is elapsed, the master station starts sending another request.

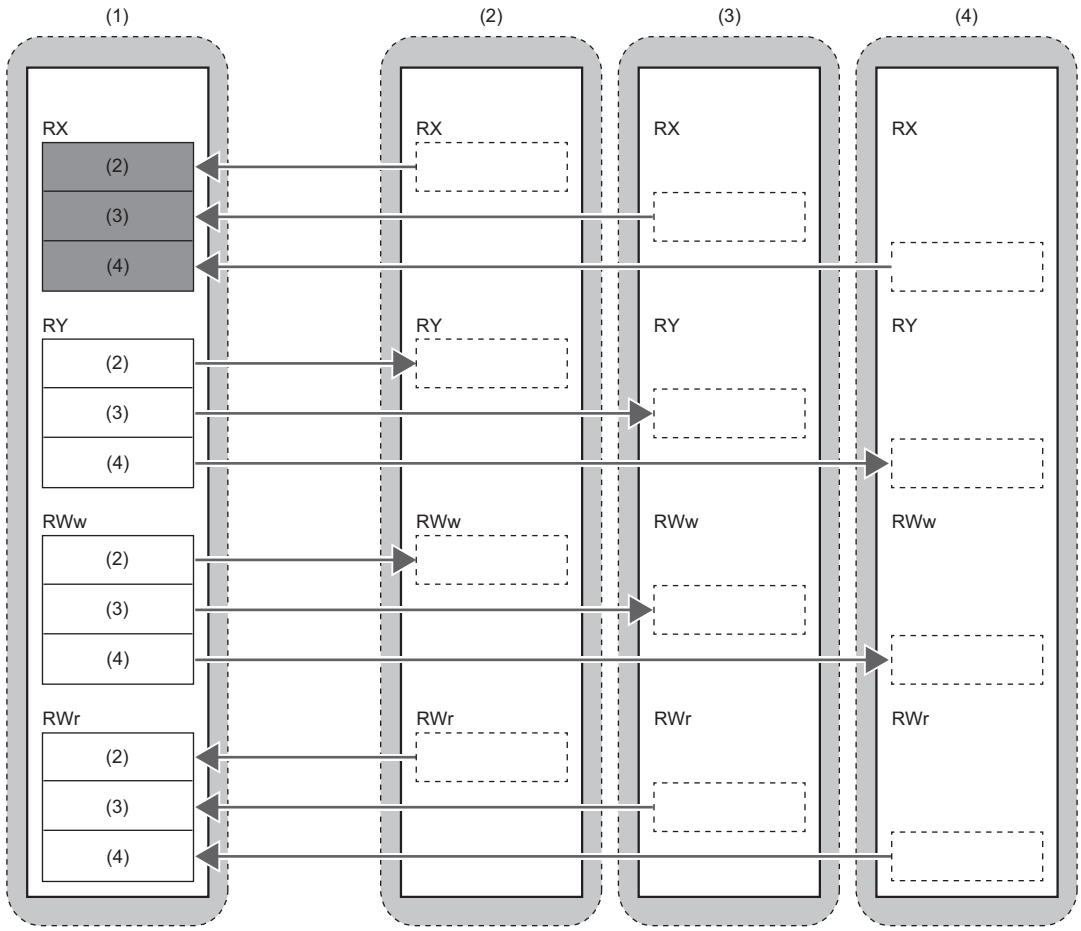
Setting method

Group numbers can be set in "Network Configuration Settings." (☞ Page 179 Network configuration settings)

Input and output status when an error occurs

This section shows the status of input from a data link error station, and output status of cyclic data when a stop error occurs in a C Controller module.

Status	Operation
Stats of input from a data link error station	RX is cleared. For RWr, the data before an error occurs is held.
Cyclic data output when a stop error occurs in a C Controller module	Data is held.



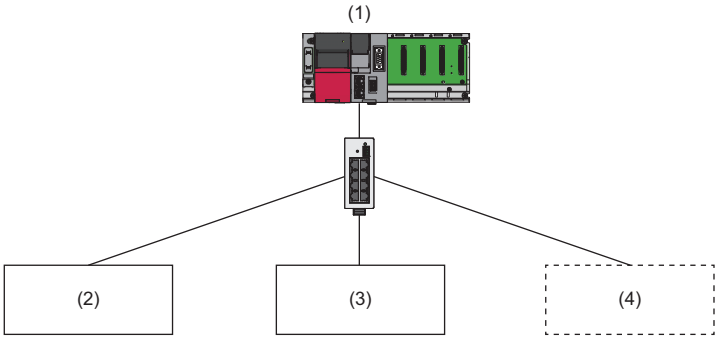
: Area where an input from an error station is cleared
: Area where data is held
: Area where the operation depends on the settings on a remote station side
(1) Master station
(2) Remote station 1
(3) Remote station 2
(4) Remote station 3

Output status when a C Controller module is in STOP state

The cyclic data output is retained when a C Controller module is in STOP state.

Reserved station specification

This functions reserves a station (a station which is not actually connected but counted as a connected station) for future use. A reserved station is not detected as an error station even if it is not actually connected.



(1) Master station
(2) Remote station 1
(3) Remote station 2
(4) Remote station 3 (reserved station which is not actually connected)

Point

A reserved station is also included in the refresh range.

Setting method

Specify a remote station as a reserved station in the network configuration settings. (☞ Page 179 Network configuration settings)

Point

Items such as the number of occupied stations and IP address can be set for a reserved station.

14.2 Programming

This section shows the programming of CC-Link IE Field Network Basic when using a C Controller module.

Interlock program for cyclic transmission

When creating a cyclic transmission program, configure an interlock so that the processing is performed when normal cyclic transmission between the master station and remote stations is performed.

Point

The cyclic transmission status between a master station and a remote station can be checked with a special relay and a special register.
Check the cyclic transmission status of each station, then create a program so that the module communicates only with a station which is performing cyclic transmission.
☞ Page 360 Special Relay List, Page 362 Special Register List

Function list

The following shows the functions used for CC-Link IE Field Network Basic.

Function name	Description
CCPU_ChangeCCIEFBCycPrm	To change the operation parameter of the cyclic transmission of the CC-Link IE Field Network Basic function.
CCPU_EndCCIEFBDataAssurance	To end data assurance for one link scan of CC-Link IE Field Network Basic.
CCPU_EntryCCIEFBRefEndFunc	To register a routine to be called when the link scan of CC-Link IE Field Network Basic is completed.
CCPU_GetCCIEFBDiagnosticInfo	To acquire the diagnostic information of CC-Link IE Field Network Basic.
CCPU_RestoreDefaultCCIEFBCycPrm	To restore the operation parameter of cyclic transmission of CC-Link IE Field Network Basic to the default value (which is set in the parameter).
CCPU_StartCCIEFBDataAssurance	To start data assurance for one link scan of CC-Link IE Field Network Basic.
CCPU_WriteDevice	To write data to devices and internal system devices of a C Controller module.
CCPU_ReadDevice	To read data from devices and internal system devices of a C Controller module.

Programming example

The following shows an example of cyclic transmission with C Controller module dedicated functions.

■Cyclic transmission program which is synchronized with link scans

1. Register a cyclic data processing routine (A) for link refreshes with the C Controller dedicated function (CCPU_EntryCCIEFBRefEndFunc).
2. The registered routine (A) is executed by every link refreshes.

Operation of routine (A)
❶ Check if a cyclic transmission is operating normally with the C Controller module dedicated function (CCPU_ReadDevice). ❷ The sampled cyclic data is processed (refresh target device is accessed).

■Cyclic transmission program which is not synchronized with link scans

1. Start data assurance for one link scan with the C Controller dedicated function (CCPU_StartCCIEFBDataAssurance).
2. Configure an interlock in a program with the C Controller module dedicated function (CCPU_ReadDevice) to check if a cyclic transmission is operating normally.
3. The sampled cyclic data is processed (refresh target device is accessed).
4. End data assurance for one link scan with the C Controller dedicated function (CCPU_EndCCIEFBDataAssurance).

Point

With the C Controller dedicated function (CCPU_StartCCIEFBDataAssurance), data inconsistency among acquired device data can be prevented.

Precautions

During the assurance of one link scan data, link refresh is controlled. When the control period of link refresh exceeded the set link scan time, the actual link scan time may exceed the set link scan time. (Consequently, link scan time is extended.) Whether or not a link scan time is exceeding the set time can be checked by checking the maximum link scan time.

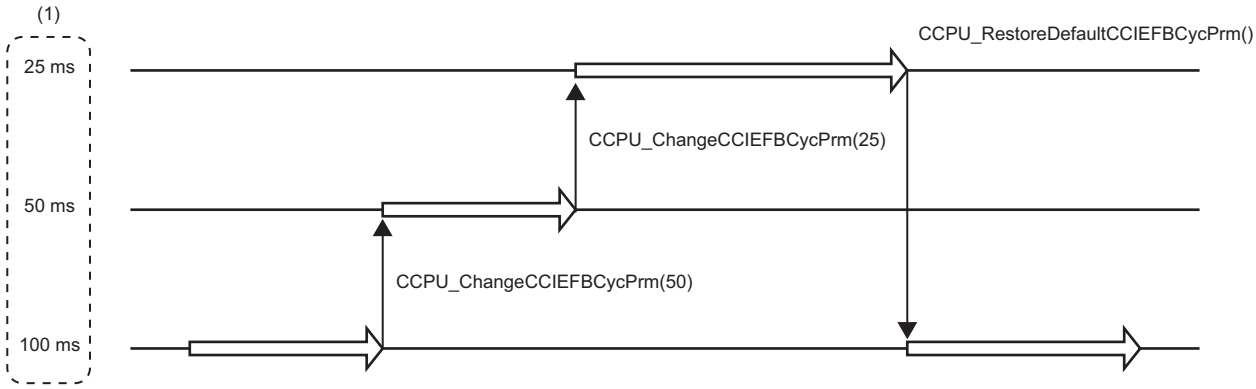
Parameter change example

The following shows an example to change parameters with a C Controller module dedicated function.

■Changing a cyclic transmission operation parameter

A link scan time and link refresh parameter can be changed by calling the C Controller module dedicated function (CCPU_ChangeCCIEFBCycPrm). The changed link scan time and link refresh parameter can be restored to the default value (which is set in the parameter) with the C Controller module dedicated function (CCPU_RestoreDefaultCCIEFBCycPrm).

- Changing link scan time (1) (parameter setting value: 100 ms)



14.3 Parameter Settings

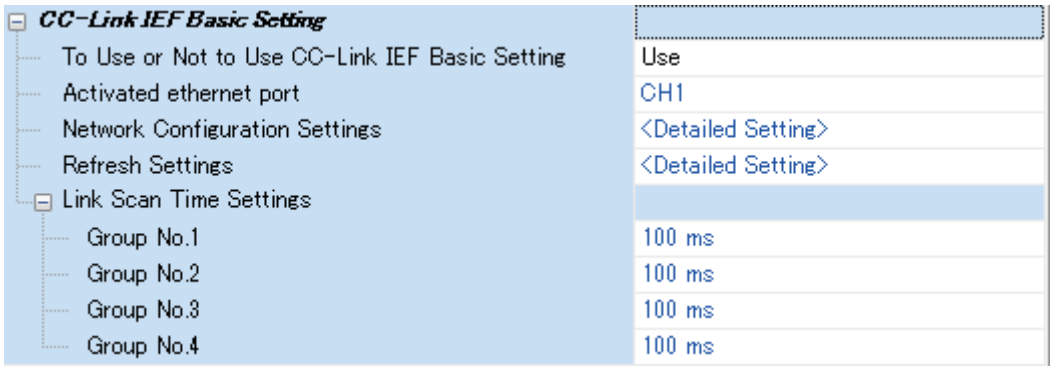
This section shows the parameter settings for a master station.
Set the parameters of the master station with CW Configurator.

CC-Link IEF Basic setting

Configure the basic settings such as whether to use CC-Link IE Field Network Basic.

Window

[Module Parameter] ⇨ [Basic Settings] ⇨ [CC-Link IEF Basic Setting]



Displayed items

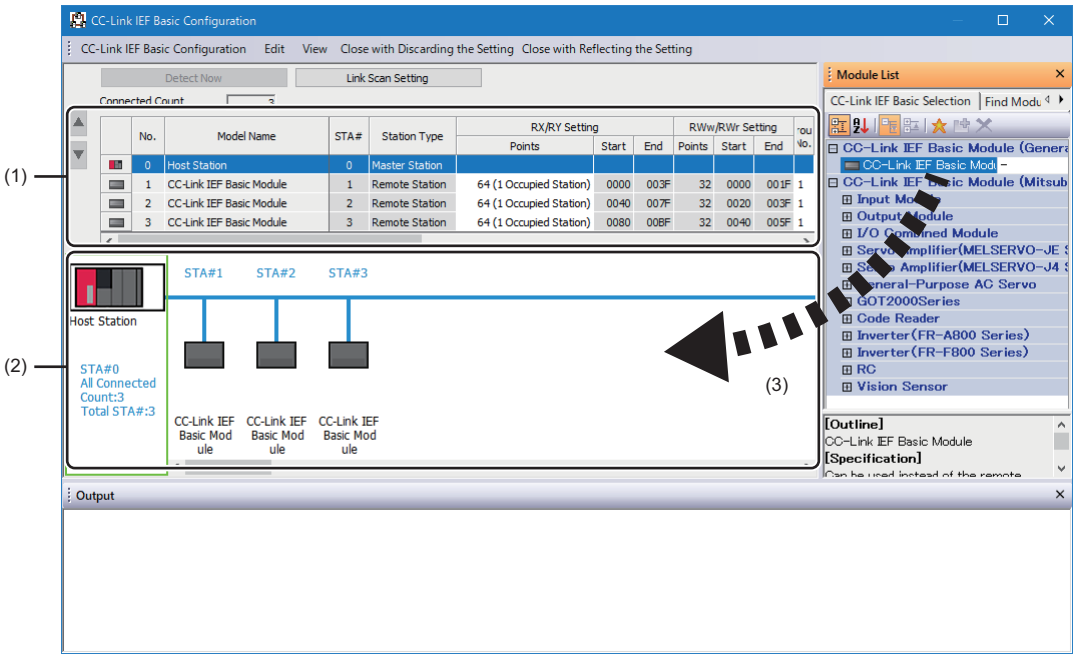
Item	Description	Setting range	Default
To Use or Not to Use CC-Link IEF Basic Setting	Set whether or not to use CC-Link IE Field Network Basic.	• Not Use • Use	Not Use
Activated ethernet port	Set an Ethernet port used for CC-Link IE Field Network Basic. When the own node is CH1, set "CH1." As for CH2, set "CH2."	• CH1 • CH2	CH1
Network Configuration Settings	Set the information to perform cyclic transmission. Moreover, set the station number, occupied station, IP address, and subnet mask of a remote station. When setting CC-Link IE Field Network Basic, set both network configuration and refresh setting.	—	—
Refresh Settings	Configure refresh settings. Configure the settings to automatically transfer data between link devices (RX/RY/RWr/RWw) and CPU devices (used device or file register). When setting CC-Link IE Field Network Basic, set both network configuration and refresh setting.	—	—
Link Scan Time Settings*1	Configure the link scan time setting. The interval of link scan time is kept constant with the interval set for the group number. If the set link scan time is longer than the actual ones, the module operates with the set link scan time. Note that, this setting does not keep transmission delay time constant.	0 to 10000 [ms] (1 ms units)	100 ms

*1 When the link scan time is set to 0 ms, the module operates as a best-effort scheme.

Network configuration settings

Set the network configuration settings.

Window



(1) List of stations
(2) Device map area
(3) Drag and drop

Displayed items

Item		Description	Setting range	Default
[Detect Now] button		Click this to detect connected devices automatically. (This function cannot be used for this product.)	—	—
[Link Scan Setting] button		Click this to configure the link scan setting. (☞ Page 180 Link scan setting)	—	—
Connected Count		The total number of connected remote stations is displayed.	—	—
No.		The station number of the remote station is displayed.	—	—
Model Name		Module model name is displayed. When there is no module information, "Module With No Profile Found" is displayed.	—	—
STA#		The start station number of the remote station is displayed.	—	—
Station Type		The station type (master station/remote station) is displayed.	—	—
RX/RY Setting	Points	Set the assignment of the number of points for RX/RY in 64-point units.	64 (1 Occupied Station) 128 (2 Occupied Station) 192 (3 Occupied Station) 256 (4 Occupied Station)	64 (1 Occupied Station)
	Start	The start number of RX/RY is displayed.	—	—
	End	The end number of RX/RY is displayed.		
RWw/RWr Setting	Points	The number of points for the number of stations in 32-point units is displayed.	—	—
	Start	The start number of the RWw/RWr is displayed.		
	End	The end number of RWw/RWr is displayed.		
Group No.		Set the group number of remote stations.	1 to 4*1	1
RSVD STA		Set whether to set the remote station as a reserved station.	• No setting • RSVD STA	No setting

Item	Description	Setting range	Default
IP Address	Specify the IP address of a remote station.	0.0.0.1 to 223.255.255.254	• First to third octet: first to third octet of the IP address of the master station • Fourth octet: Automatically numbered from the number not in use from 1 to 254 in ascending order
Subnet Mask	Specify the subnet mask of the remote station.	0.0.0.1 to 255.255.255.255	Subnet mask of the master station
MAC Address	The MAC address of the remote station is displayed.	—	—
Comment	The information entered in "Comment 1" in the "Properties" screen displayed by right-clicking the module in the list of stations or device map area is displayed.	32 characters	(Blank)

*1 Groups do not need to be numbered serially. For example, setting group No.2 only (number of groups: one) and setting group No.1 and No.3 (number of groups: two) are both possible.

Restriction

A C Controller module does not perform automatic detection of connected devices, reflection of the communication setting of remote stations, and parameter processing of remote stations. Set the communication setting such as an IP address and subnet mask, and remote station specific parameters on each remote station. The remote devices on which parameters cannot be set cannot be used.

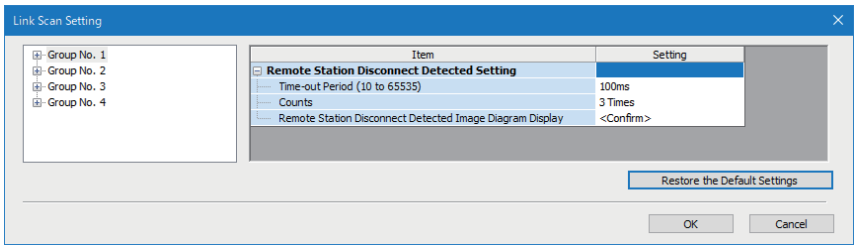
■Link scan setting

Set timeout time and number of retries for remote station disconnection detection.

Point

The parameter is used as an initial parameter. With a C Controller dedicated function, the initial parameter can be changed dynamically. (☞ Page 177 Parameter change example)

Window



Displayed items

Item		Description	Setting range	Default
Remote Station Disconnection Detection Setting	Time-out Period (10 to 65535)	Set the timeout time (ms) for remote station disconnection detection.*1	10 to 65535 ms	100 ms
	Counts	Set the number of retries for remote station disconnection detection.*1,*2	3 Times, 5 Times, 10 Times	3 Times
	Remote Station Disconnection Detection Image Display	The operation image regarding the remote station disconnection detection period is displayed. Refer to this at the setting of "Time-out Period."	—	—

*1 Timeout time and the number of times for disconnection detection are counted for each remote station.

*2 Disconnection occurs in the event that no response is received from the remote station for the specified number of times in succession within the timeout time.

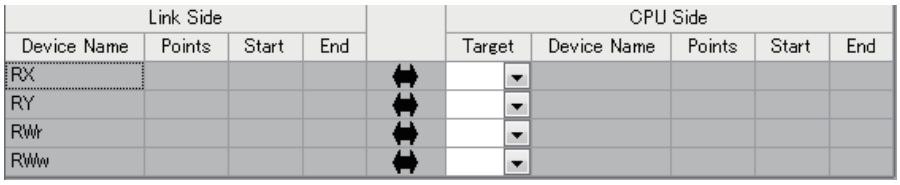
Point

- Set an adequate value for the timeout time according to the actual system used.
- Time-out Period and Counts can be set for each group.

Refresh settings

Set refresh parameters.

Window



Displayed items

Item		Description	Setting range	Default
Link Side		The number of points for the link devices (RX/RY, RWr/RWw) for the number of occupied stations and start/end device number set in the network configuration settings are displayed.	—	—
CPU Side	Target	The target for link refresh is displayed.	Specify Device	(Blank)
	Device Name	Set the device of the link refresh target.	M, B, D, W, ZR	(Blank)
	Points	The number of device points for the link refresh target is displayed. (The same value as the number of points on the link side is displayed.)	—	—
	Start	Set the start device number within the link refresh range.	Use the device range of C Controller module. (☞ Page 264 DEVICES)	(Blank)
	End	The end device number within the link refresh range is displayed.	—	—

14.4 CC-Link IE Field Network Basic Diagnostics

Perform troubleshooting by performing the CC-Link IE Field Network Basic diagnostics and checking the network status and error details.

Usage

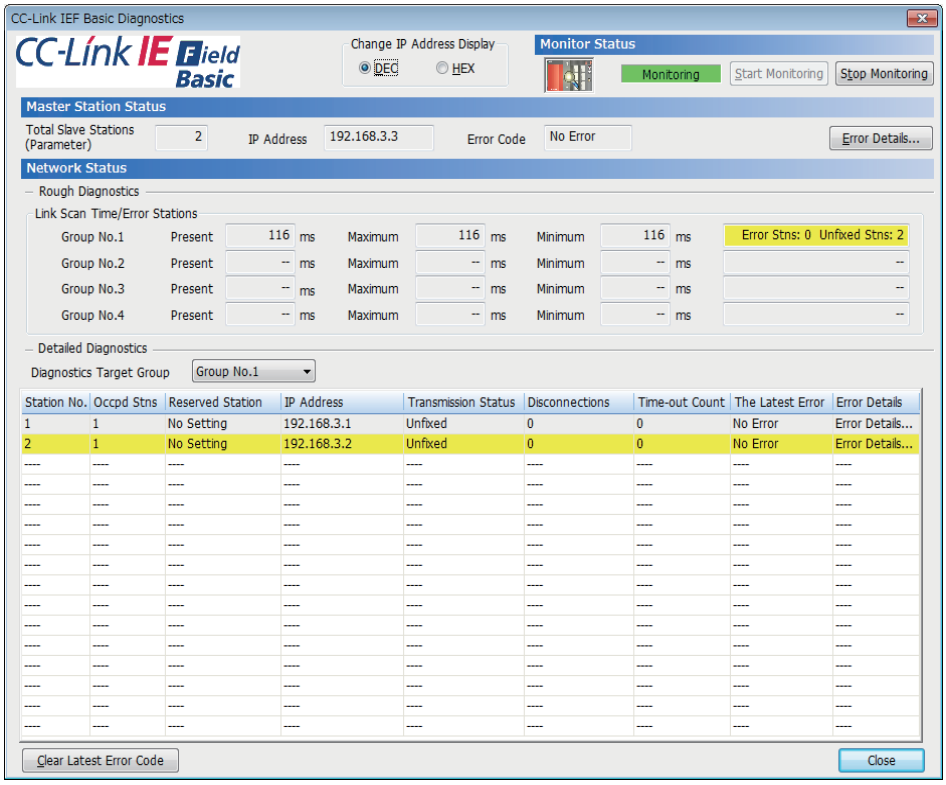
1. Connect CW Configurator to a C Controller module on the master station.
2. Start the CC-Link IE Field Network Basic diagnostics.

[Diagnostics] ⇨ [CC-Link IEF Basic Diagnostics]

Diagnostics screen

The status of the master station is checked in "Master Station Status."
The network status including remote stations is checked in "Network Status."
For details on the "CC-Link IEF Basic Diagnostics" screen, refer to the following:
CC-Link IE Field Network Basic Reference Manual

Window



Acquiring diagnostic information of remote stations

If an error occurs in a remote station or the cyclic data cannot be read or written correctly, check the status of each remote station with the following method for acquiring diagnostic information of remote stations.

- C Controller module dedicated function (CCPU_GetCCIEFBDiagnosticInfo)
- Buffer memory

Procedure to acquire diagnostic information with buffer memory

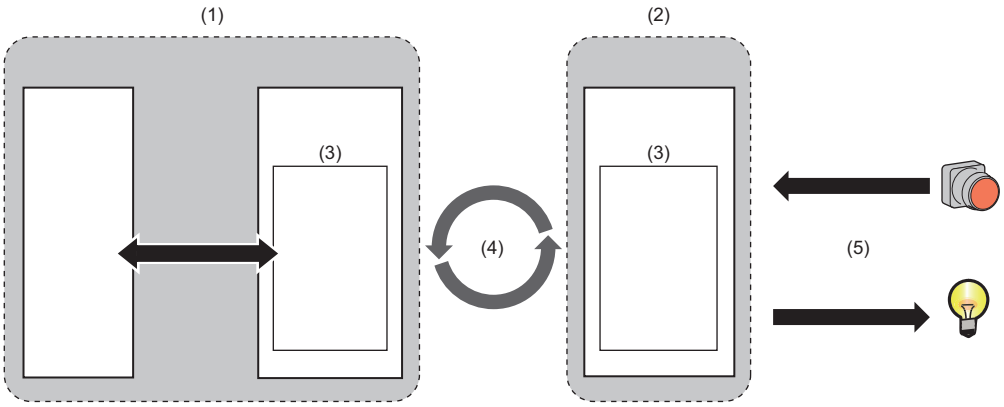
Acquire diagnostic information by buffer memory operation. (☞ Page 376 Buffer Memory)

1. Set a station number to acquire diagnostic information for 'Diagnostic request information' (Un\G1051).
2. By turning 'Diagnostic information display request' (Un\G1050.0) OFF and ON, diagnostic information is stored to 'Diagnostic information 1' (Un\G1053 to Un\G1067) and 'Diagnostic information 2' (Un\G1068 to Un\G1083). ('Diagnostic information display request' (Un\G1050.0) turns OFF when the diagnostic information is stored.)

14.5 Processing Time

The processing time of CC-Link IE Field Network Basic consists of the time components below.

- Link scan time + remote station response processing time = transmission delay time



(1) Master station
(2) Remote station
(3) Link device
(4) Link scan time
(5) Response processing time of a remote station

Link scan time

The link scan time of CC-Link IE Field Network Basic operates in accordance with the setting of "Link Scan Time Settings" under "CC-Link IEF Basic Settings." (☞ Page 178 Parameter Settings)

Link scan time check

The maximum value, minimum value, and current value of a link scan time can be checked with the following method.

- CC-Link IE Field Network Basic diagnostics
- Buffer memory

Transmission delay time

A transmission delay time includes an input transmission delay time and an output transmission delay time.

Input transmission delay time

The input transmission delay time indicates the following time.

- Time between when a signal (RX) is input to a remote station and when a device of the master station turns ON or OFF
- Time between data (RWr) is input to a remote station and when the data is stored to a device of the master station

■Calculation formula for input transmission delay time

- Input transmission delay time (maximum) = (Ls × 2) + SS [ms]

Item	Description
Ls	Link scan time [ms]
SS	Processing time required for the remote station to reflect the input [ms] (Manual for the remote station used)

Output transmission delay time

The output transmission delay time indicates the following time.

- Time between when a device of the master station turns ON or OFF and when the output (RY) of a remote station turns ON or OFF
- Time between when data is set to a device of the master station and when the data (RWw) is output to a remote station

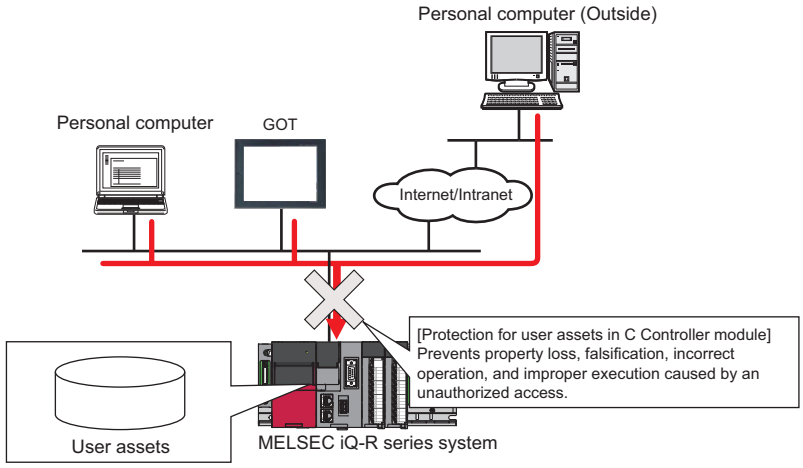
■Calculation formula for output transmission delay time

- Output transmission delay time (maximum) = Ls + SS [ms]

Item	Description
Ls	Link scan time [ms]
SS	Processing time required for the remote station to reflect the output [ms] (Manual for the remote station used)

15 SECURITY FUNCTION

This function prevents assets stored in a personal computer or a C Controller module in the MELSEC iQ-R series system from being stolen, falsified, operated incorrectly, and executed improperly due to unauthorized access from a third party.
Apply an appropriate security function in accordance with the situation.



Point

The security function is one of the methods for preventing unauthorized access (such as program or data corruption) from an external device. However, this function does not prevent unauthorized access completely. Incorporate measures other than this function if the C Controller system's safety must be maintained against unauthorized access from an external device. Mitsubishi Electric Corporation cannot be held responsible for any system problems that may occur from unauthorized access.
Examples of measures for unauthorized access are shown below.

- Install a firewall.
- Install a personal computer as a relay station, and control the relay of send/receive data with an application program.
- Install an external device for which the access rights can be controlled as a relay station. (For details on the external devices for which access rights can be controlled, consult the network provider or equipment dealer.)

15.1 Individual Identification Information Read Function

The individual identification information of a C Controller module can be read with the C Controller module dedicated function (CCPU_GetIDInfo). By implementing an activation function with a user program, a user program, which does not run in C Controller modules with other individual identification information, can be created.
For C Controller module dedicated functions, refer to the following:
MELSEC iQ-R C Controller Module Programming Manual

15.2 File Access Restriction Function

A file attribute can be set for the files stored in the memory. By setting a file attribute, access to a target file can be restricted, and falsification by an unauthorized user and data leakage to outside can be prevented.

- Program memory
- Data memory
- SD memory card
- USB Mass Storage Class-compliant device

Point

- When an SD memory card or a USB Mass Storage Class-compliant device is inserted to a peripheral device other than a C Controller module (such as a personal computer), files to which the access restriction is set can be operated. If the access restriction is set for the file in the SD memory card and the USB Mass Storage Class-compliant device, take appropriate measures so that the SD memory card and the USB Mass Storage Class-compliant device cannot be removed from the C Controller module at will.
- Access restriction cannot be set for folders.

File access restriction setting

Change a file attribute handled in a C Controller module by using the attrib() command. A security password is required to change a file attribute.

For details on the attrib() command, refer to the manual for VxWorks.

■Setting file attribute

Set a file attribute to a file to be restricted by using the attrib() command.

The file attributes that can be handled in a C Controller module are as follows.

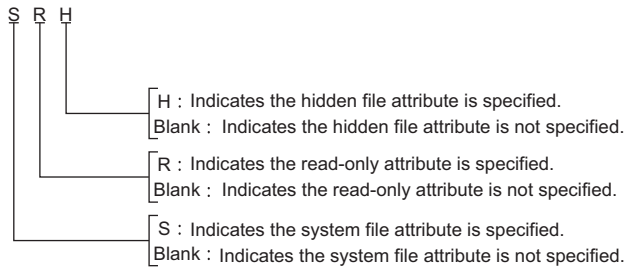
Attribute		Description
S	System file attribute	File operations can be prohibited.
R	Read-only attribute*1	File deletion and data write can be prohibited.
H	Hidden file attribute*2	A file is not listed by using the ls command and it is not displayed at FTP connection.

*1 This attribute is not supported by the file access restriction function. However, if it is set, file deletion and file write can be prohibited.

*2 When a file is opened by specifying a file name, the file can be operated. To prohibit file operations, make sure to set a system file attribute.

■Checking file attribute

A file attribute which is set can be checked by using the attrib() command.



Checking file access restriction status

File access restriction status can be checked with the Shell command or the C Controller module dedicated function (CCPU_GetFileSecurity).

Point

File access restriction status cannot be checked by using the script file (STARTUP.CMD).

Canceling/re-setting file access restriction

Change the file access restriction status by using the Shell command, the script file (STARTUP.CMD), or a user program. The security password set with CW Configurator is required.

■Changing system file attribute

For operating a file with a system file attribute attached, cancel the file access restriction temporarily with the C Controller module dedicated function (CCPU_ChangeFileSecurity). The canceled setting can be set again by setting the file access restriction with the C Controller module dedicated function (CCPU_ChangeFileSecurity) or resetting the C Controller module.

Point

- When accessing a file to which a system file attribute is attached in the script file "STARTUP.CMD," cancel the access restriction in the script file. In this case, add the system file attribute to the script file in order to prevent the leakage of the password.
- Do not use the files with the system file attribute attached in the script file (STARTUP.CMD) in an SD memory card in order to prevent the leakage of password.

15

Precautions

■When maintaining the safety against unauthorized access from external parties

To maintain the safety of a C Controller system against unauthorized access from external parties, take appropriate measures. Note the following when setting a security password to prevent the leakage of the security password.

- Avoid settings with only simple alphanumeric characters.
- Set a complex password with symbols.

■Settable characters for a security password

Characters that can be set are single byte alphanumeric characters and symbols. (Security password is case-sensitive.)

■If the security password has been forgotten

Initialize the C Controller module.

For the procedure for initialization, refer to the following:

☞ Page 28 Initialization

15.3 IP Filter Setting Function

This function sets security for the access using Ethernet.
Access from an illegal IP address is prevented by identifying the IP address of communication target. Set the IP address of the target device to be allowed or denied in the parameters to restrict the access from target devices.
Use of this function is recommended when using C Controller module in an environment which is connected to LAN.

Setting method

1. Set the IP address for the IP address to be allowed or denied in "Security" under "Application Settings."
2. Write the module parameters to the C Controller module.
3. The IP filter function is enabled after turning the power OFF and ON, or resetting the C Controller module.

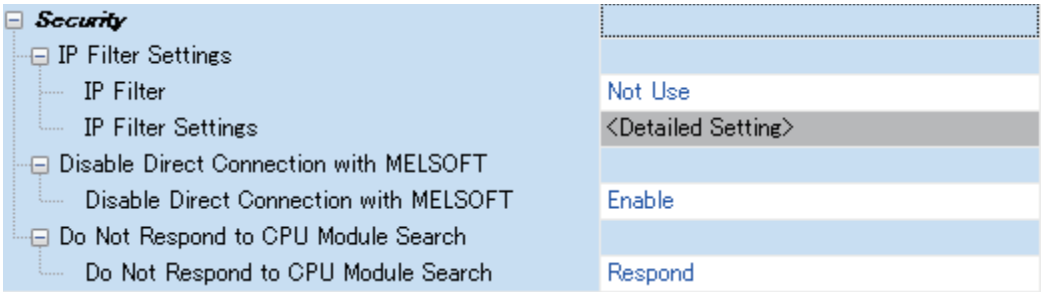
Point

Even if the connection is specified by the setting of "External Device Configuration" of the C Controller module or the program, the access from the target device is either allowed or denied according to the IP filter settings. If the IP address set to "External Device Configuration" of the C Controller module is set to be denied in "IP Filter Settings," the IP filter setting is enabled and communication with the target device is denied.

Window

Set the security settings.

[Module Parameter] ⇨ [Application Settings] ⇨ [Security]



Displayed items

Item		Description	Setting range	Default
IP Filter Settings	IP Filter	Set whether to use the IP filter function.	• Not Use • Use	Not Use
	IP Filter Settings	Set the IP addresses to be allowed/denied.	—	
Disable Direct Connection with MELSOFT		Enables/disables the direct connection with CW Configurator.	• Disable • Enable	Enable
Do Not Respond to CPU Module Search		Select whether to respond to the search from the other nodes performed by 'Searching CPU Modules on the Network' in engineering tool.	• Do Not Respond • Respond	Respond

■IP filter settings

Up to 32 IP addresses can be set to allowed/denied using the IP filter function.

The range of IP address and IP addresses to be excluded from the specified range can be set at one setting.

Item	Description	Setting range	Default
Access from IP address below	Set whether to allow/deny the access from the specified IP addresses.	• Allow • Deny	Allow
Range Setting	Select this when specifying IP address range.	—	Unselected
IP Address	Set the IP addresses to be allowed/denied. When a checkbox of "Range Setting" is selected, set both start IP address and end IP address within the range.	0.0.0.1 to 223.255.255.254	—
IP Address Excluded from Range	When a checkbox of "Range Setting" is selected, set the IP address to be excluded from the range in "IP Address." Up to 32 IP addresses can be set within the specified range.	0.0.0.1 to 223.255.255.254	—

Precautions

If there is a proxy server on the LAN, deny the IP address of the proxy server. Otherwise, the access from the personal computers that can access the proxy server cannot be prevented.

15.4 Service Setting Function

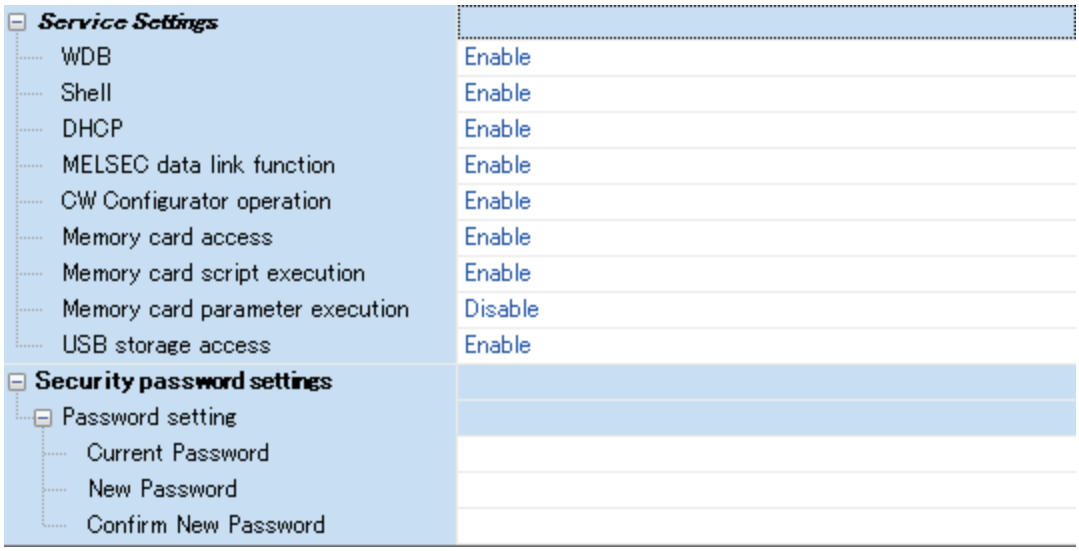
The services for a C Controller module can be set.

By restricting the services, unauthorized access from other users can be prevented.

A security password is required for changing service settings.

[CPU Parameter] ⇨ [Service Settings]

Window



Displayed items

Item		Description	Setting range	Default
Service Settings	WDB	Required for the connection with CW Workbench.	• Disable • Enable	Enable
	Shell	Required for the execution of commands.		
	DHCP*1	Required when using a function which assigns the network settings automatically.		
	MELSEC data link function	Required when performing MELSEC communication.		
	CW Configurator operation	A service required for the operation of CW Configurator. If this service is disabled, the following operations cannot be performed. • Writing data to a C Controller module • Reading data from a C Controller module • Verifying data with a C Controller module • Deleting data in a C Controller module • CPU memory operation (Initialization)		
	Memory card access	Required when accessing a memory card.		
	Memory card script execution	Required when executing a script file "STARTUP.CMD" stored in a memory card.		
	Memory card parameter execution*2	Required when performing parameters stored in a memory card.		Disable
	USB storage access	Required when accessing USB devices.		Enable
Security password settings	Password setting	A security password setup is set up.	8 to 16 characters	password

*1 A function which assigns the network settings automatically will not be available when disabling the service.
To use the function, use VxWorks standard API functions.
For details on VxWorks standard API functions, refer to the manuals for VxWorks with the following version.
.VxWorks Version 6.9
If a function which assigns the network settings automatically is used, the IP address set in the parameter will be disabled.

*2 When changing the service, write parameters to the data memory.
If the parameters are written to the memory card, the service setting is not changed.

Password setting

■Current password

Enter the current security password.

■New password and confirmation password

To change the security password, enter a new security password in "New Password" and "Confirm New Password."

Precautions

■CW Configurator operation

When the CW Configurator operation is disabled, the parameters cannot be set. To enable the service, initialize C Controller module.

■Memory card parameter execution

To select an SD memory card as a target memory of the parameters, enable "Memory card parameter execution." ("Disable" is selected as a default.)

■Stopping multiple services

When the memory card access is disabled, the memory card script execution will be disabled as well.

15.5 Lockout Function

If the password authentication failed for several times, the password authentication will be denied (locked out) for a period of time. This prevents a brute force attack from unauthorized users.

Lockout time

The lockout time is as follows:

Number of password input error*1	Lockout time
1st time to 5th time	0 minutes
6th time	1 minute
7th time	5 minutes
8th time	15 minutes
9th time or later	60 minutes

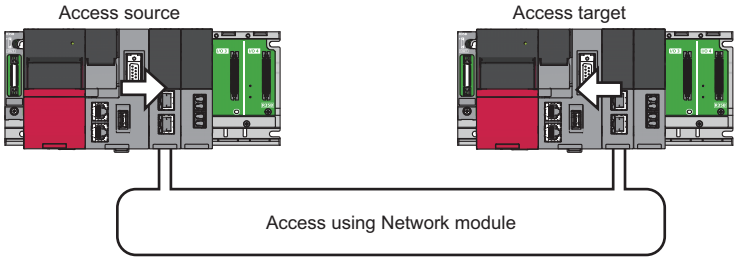
*1 Once the correct password is entered, the number of password input error will be cleared.

Point

- The password input error will not be counted during lockout. Therefore, the lockout time will not be extended additionally by one minute even if the 7th input error occurs before one minute has passed since the 6th.
- When the security setting is configured with the C Controller module dedicated function (CCPU_ChangeFileSecurity), the password authentication will not be locked out.

16 ACCESS FUNCTION USING NETWORK MODULE

Data communication with a device connected to the network can be performed via a network module controlled by a C Controller module.



Point

In a multiple CPU system configuration, the access via a network module which is controlled by another CPU is not available.

Available network modules

Network modules that can be controlled by C Controller module are as follows:

- CC-Link IE Controller Network module
- CC-Link IE Field Network module
- CC-Link IE TSN module
- MELSECNET/H network module
- CC-Link module

For module names, refer to the following:

MELSEC iQ-R Module Configuration Manual

Network parameter

To use a network module, the setting of "Module Parameter" for the network module is required. To perform this settings, refer to the user's manual for the respective network modules.

16.1 Data Communication via Network

This section shows how to perform data communication using C Controller modules on each network.

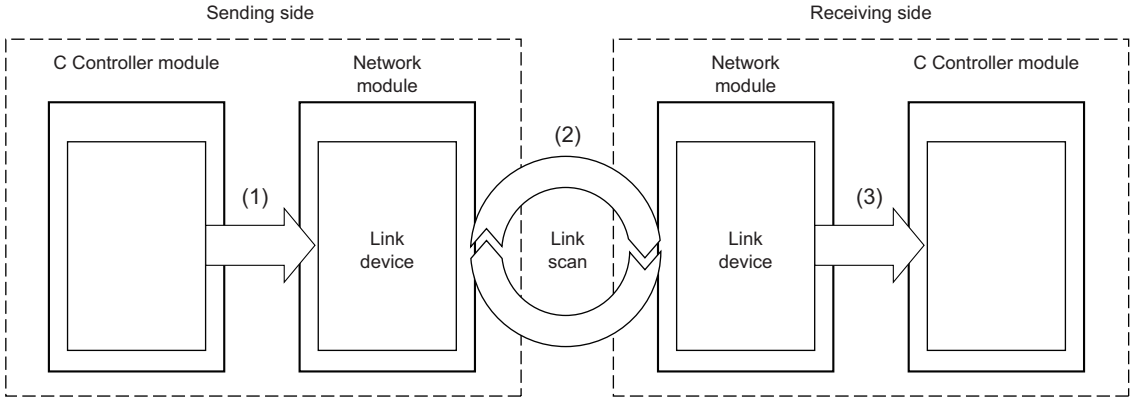
Transmission type	Communication method
Cyclic transmission function	Uses link devices of a network module on the own station controlled by a C Controller module.
Transient transmission function	Uses devices or buffer memory of a network module of another station via a network.

16.2 Cyclic Transmission Function

This function performs data communication periodically between stations on the same network by using link devices.
This section shows the specifications of the cyclic transmission when using a C Controller module.

Data flow

The following shows the data flow at cyclic transmission.



(1) The sending side C Controller module writes data to a link device of the network module.
(2) Data in the sending side link device is stored to the receiving side link device by link scan.
(3) The receiving side C Controller module reads data from a link device of the network module.

■Link device

Link devices are used for sharing data in a network module with other stations on the network. The data in each station is updated every link scan.
○: Applicable, ×: Not applicable

Link device		CC-Link IE Controller Network	CC-Link IE Field Network	CC-Link IE TSN	MELSECNET/H network	CC-Link
Link input	LX	○	×	×	○	×
Link output	LY	○	×	×	○	×
Link relay	LB	○	×	○	○	×
Link register	LW	○	×	○	○	×
Remote input	RX	×	○	○	×	○
Remote output	RY	×	○	○	×	○
Remote register	RWw	×	○	○	×	○
	RWr	×	○	○	×	○
Link special relay	SB	○	○	○	○	○
Link special register	SW	○	○	○	○	○

■Link scan and link scan time

In cyclic transmission, each station on the network transmits data in the specified send area of the own station within the defined time interval. The processing of data transmission of each station is called as link scan. The authority of data transmission is given to each station during every link scan. Time required for one-cycle of data transmission i.e. one link scan is called as link scan time.

To perform link scan on the network, network range assignment settings are required.

■Network range assignment

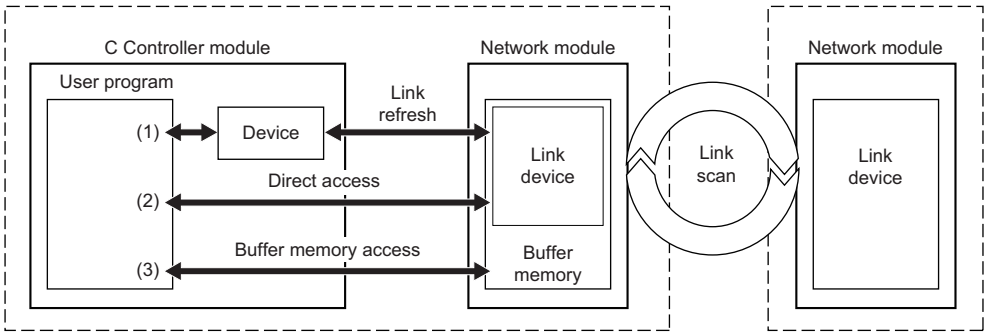
Network range assignment settings are applicable to following networks:

Network name	Setting item
CC-Link IE Controller Network	"Network Range Assignment Setting" of the control station
CC-Link IE Field Network	"Network Configuration Settings" of the master station
CC-Link IE TSN	"Network Configuration Settings" of the master station
MELSECNET/H network	"Network Range Assignment Setting" of the control station
CC-Link	"Network Configuration Settings" of the master station

■Link device access function

The link devices can be accessed by using a user program with a function of dedicated function library.

The following figure shows how to access a link device in the controlled network module from a C Controller module.



Transmission type		Description	Data
(1)	Access by link refresh	A method to access devices of a C Controller module from a user program. The device data communicates with link devices in a network module by link refresh.	• Frequently used link device • Link device that requires station-based block data assurance setting
(2)	Direct access	A method for direct access to link devices in a network module from a user program.	• Rarely used link device
(3)	Buffer memory access	A method to access the buffer memory in a network module from a user program.	

Point

Excluding rarely used link devices from link refresh device area and not using access by link refresh may reduce link refresh time.

■Available access methods

The available methods to access each controlled network module from a C Controller module are shown below.

○: Applicable, ×: Not applicable

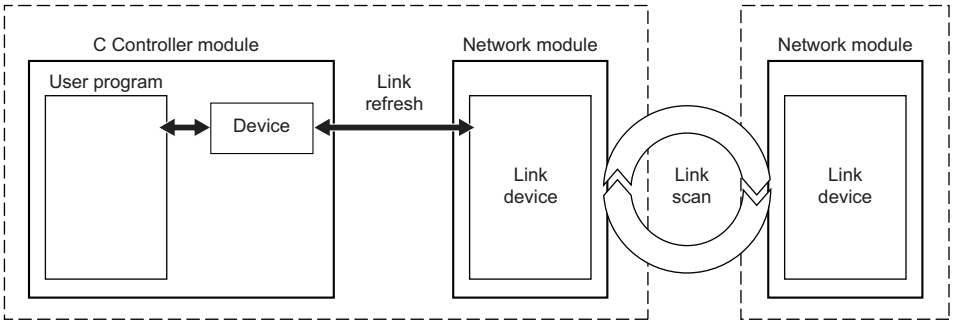
Transmission type	CC-Link IE Controller Network	CC-Link IE Field Network	CC-Link IE TSN	MELSECNET/H network	CC-Link
Access by link refresh	○	○	○	○	×
Direct access	○	○	○	○	×
Buffer memory access	×	○	○	×	○

Access by link refresh

This method is used to access the devices of a C Controller module from a user program using link refresh.

Data flow

The following figure shows the data flow of link refresh.



■Devices

C Controller modules use the following devices to share the data with network modules.

Device		Number of points	Range of use
Internal relay	M	61440 points	M0 to 61439
Link relay	B	655360 points	B0 to 9FFFF
Data register	D	4184064 points	D0 to 4184063
Link register	W	1048576 points	W0 to WFFFFF
File register	ZR	1835008 points	ZR0 to 1835007

■Link refresh and its refresh cycle

The processing of data communication between devices of a C Controller module and link devices of a network module is called as link refresh. The link refresh is performed in every refresh cycle of a C Controller module. (☞ Page 121 Fixed Cycle Processing Function)

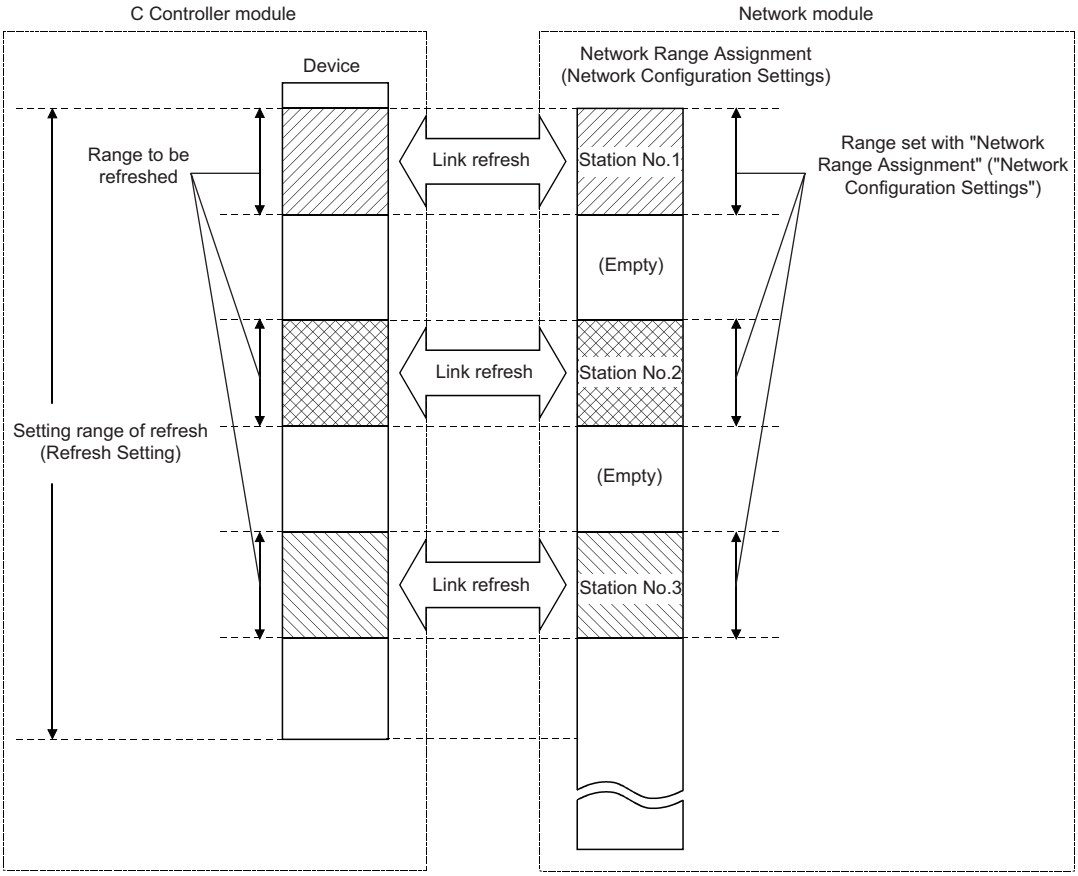
■Parameter setting

When performing link refresh, set the following parameters.

- "Refresh Setting" and "Network Configuration Settings" of each network module
- Station-based block data assurance setting

■Refresh range

Refresh is performed for the range set in "Refresh Setting" and "Network Range Assignment" ("Network Configuration Settings"). For the access by link refresh, specify the devices within the refresh range.

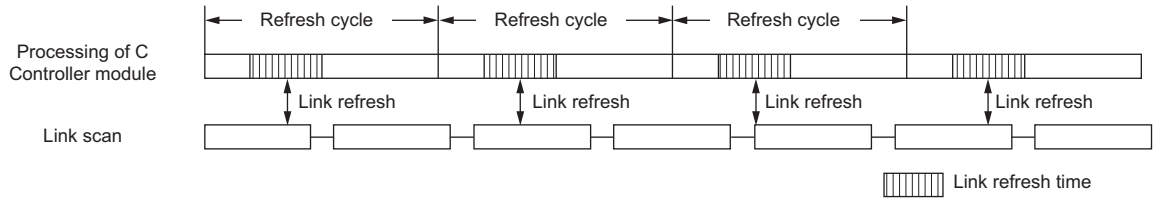


■Assurance of cyclic data consistency (station-based block data assurance)

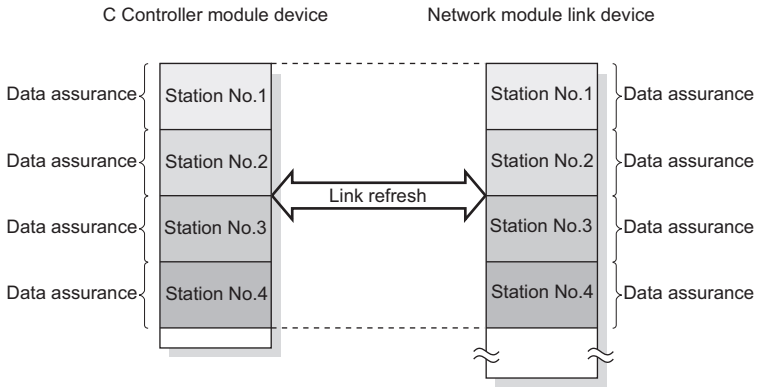
The station-based block data assurance function prevents overlapping of previous link scan data and new link scan data in one station.

Point

Link scan is performed asynchronous to link refresh in the C Controller module. Therefore, if 32-bit or more cyclic data is handled, new data and old data may overlap due to the timing of link refresh.



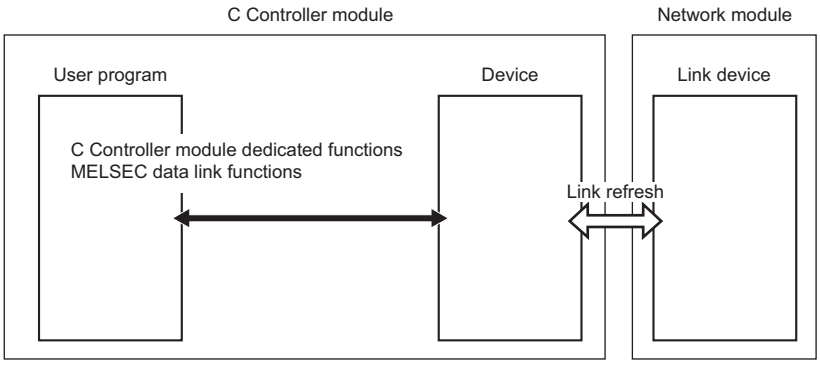
When the station-based block data assurance is set, cyclic data consistency is assured in station units since the C Controller module does refresh cycle with a network module by handshaking.



For details of the station-based block data assurance function and the other data assurance functions, refer to the user's manual for each network module.

■User program functions

The devices can be accessed by using a user program with a function of dedicated function library.



The following shows the functions used to access devices by link refresh.

Dedicated function library	Description
CCPU_WriteDevice	To write data to devices and internal system devices of a C Controller module.
CCPU_ReadDevice	To read data from devices and internal system devices of a C Controller module.
mdDevRstEx	To reset (turn OFF) bit devices.
mdDevSetEx	To set (turn ON) bit devices.
mdRandREx	To read devices randomly.
mdRandRLabelEx	To read labels or devices randomly.
mdRandWEx	To write devices randomly.
mdRandWLabelEx	To write labels or devices randomly.
mdReceiveEx	To read devices in a batch.
mdSendEx	To write devices in a batch.

16

Precautions

When the station-based block data assurance function is enabled, access the devices using the CCPU_WriteDevice/CCPU_ReadDevice function. If the mdSendEx/mdReceiveEx/mdRandWEx/mdRandREx/mdRandRLabelEx/mdRandWLabelEx function is used, data inconsistency may occur.

■Devices specified with a function

Devices corresponding to each link device are shown below.

In the dedicated function library, specify the device type with the device name defined for each function.

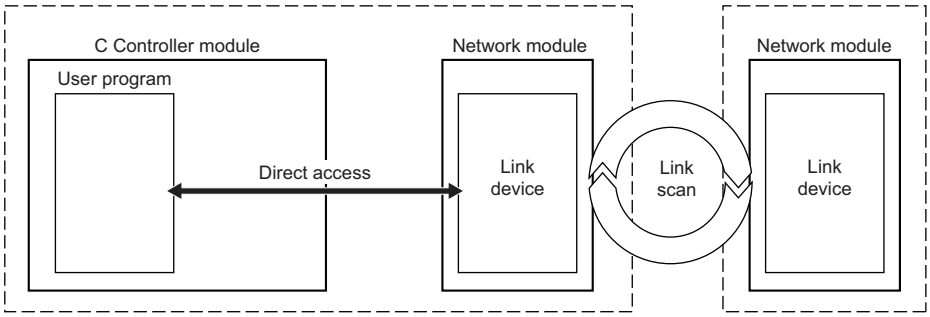
Link device		Device	C Controller module dedicated function	MELSEC data link function
Link input	LX	M, B, D, W, ZR	Dev_CCPU_M Dev_CCPU_B Dev_CCPU_D Dev_CCPU_W Dev_CCPU_ZR	DevM DevB DevD DevW DevZR
Link output	LY			
Link relay	LB			
Link register	LW			
Remote input	RX			
Remote output	RY			
Remote register	RWw			
	RWr			
Link special relay	SB	The access by link refresh is not allowed. Use the direct access.		
Link special register	SW			

Direct access

This method is used for direct access to link devices in a network module from a user program.

Data flow

The following figure shows the data flow for direct access to link devices of a network module.



■User program functions

The following shows the functions used for direct access to network module link devices.

Dedicated function library	Description
CCPU_WriteLinkDevice	To write data to link devices of a network module directly.
CCPU_ReadLinkDevice	To read data from link devices of a network module directly.

■Link devices specified with a function

To directly access link devices, access them as link direct devices (J□\□) depending on network module specifications.

The following shows the specification method for accessing link direct devices (J□\□) with C Controller module dedicated functions.

In the dedicated function library, specify the device type with the device name defined for each function.

J□¥□
(1) (2)

Link device*1		Specification method for C Controller module dedicated function		
		(1)	(2)	
		Module position (usIoNo)	Device type (sDevType)	Start device number (ulDevNo)
Link input	Jn\X	Specify an access target network module.	DevLX	Specify the start device number of an access target.
Link output	Jn\Y		DevLY	
Link relay	Jn\B		DevLB	
Link register	Jn\W*2		DevLW	
Link special relay	Jn\SB		DevLSB	
Link special register	Jn\SW		DevLSW	

*1 Accessible link devices vary depending on network modules. For the accessible devices, refer to the manual for a network module.

*2 For CC-Link IE TSN modules and CC-Link IE Field Network modules, access target devices (RWw, RWr, and LW) vary depending on a device number to be specified. Therefore, a device number which is actually accessed may be different from the specified device number.

Ex.

Example: Accessing 'J1\W4000' (When an access target is a CC-Link IE TSN module with the start I/O number 0x20)

- Module position (usIoNo): 0x20 ÷ 16 = 0x2
- Device type (sDevType): DevLW
- Start device number (ulDevNo): 4000

LW0 of the CC-Link IE TSN module with the start I/O number 0x20 is accessed.

■Link direct device mode setting

When using either of the following modules, "Extended Mode (iQ-R Series Mode)" needs to be specified for the link direct device setting in CPU parameters. (The default is "Q Series Compatible Mode.")*1

- CC-Link IE TSN master/local module
- Motion module
- CC-Link IE Controller Network module when setting the link points extended setting*2

*1 To use the mode, check the version of a C Controller module and an engineering tool. (☞ Page 404 Added and Changed Functions)

*2 Applies when the "Link points extended setting" of the module parameter is set to "Extend" and the extended link device is specified. (📖MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application))

Point

When specifying a link direct device, if any of the following modules (even single module) exists among the network modules controlled by a C Controller module, set the link direct device setting to "Extended Mode (iQ-R Series Mode)."

- CC-Link IE TSN master/local module
- Motion module
- CC-Link IE Controller Network module when setting the link points extended setting

Note that modules, except for the above modules, operate in either of the modes.

The device range available for the link direct device setting is as follows:

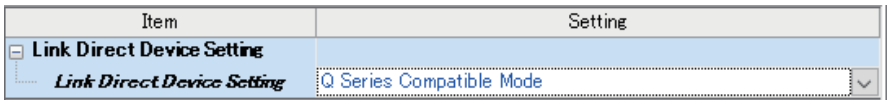
Setting	Device range
Q series compatible mode	• Link input: 16K points • Link output: 16K points • Link relay: 32K points • Link special relay: 512 points • Link register: 128K points • Link special register: 512 points
Extended mode (iQ-R series mode)	• Link input: 160K points • Link output: 160K points • Link relay: 640K points • Link special relay: 5120 points • Link register: 2560K points • Link special register: 5120 points

For the devices that can be used for C Controller modules, refer to the following:

☞ Page 264 DEVICES

Window

[CPU Parameter] ⇨ [Memory/Device Setting] ⇨ [Link Direct Device Setting]



Displayed items

Item	Description	Setting range	Default
Link Direct Device Setting	Set 'Q Series Compatible Mode' or 'Extended Mode (iQ-R Series Mode)' for link direct device. Set 'Extended Mode (iQ-R Series Mode)' when using any one of the modules shown below as a network module to control. • CC-Link IE TSN master/local module • Motion module • CC-Link IE Controller Network-equipped module when setting the Link points extended setting	• Q Series Compatible Mode • Extended Mode (iQ-R Series Mode)	Q Series Compatible Mode

Precautions

The range of link direct devices that can be accessed by either of the following methods varies depending on the firmware version of a C Controller module.

- Accessing with MELSEC data link functions or C Controller module dedicated functions
- Using the device access function from another product

When executing the firmware update of a C Controller module, note that the operation of a user program may be affected by changing the firmware version.

When setting a CC-Link IE Controller Network module as an access target, the accessible range of link direct devices will be as follows regardless of the "Link points extended setting" of module parameters in the module.

■Setting "Extended Mode (iQ-R Series Mode)" in the link direct device setting

CC-Link IE Controller Network module	Firmware version of a C Controller module	
	'15'	'16' or later
A firmware version that supports the number of send points extension by using the link points extended setting or later*1	• Link input: 16K points • Link output: 16K points • Link relay: 32K points • Link special relay: 512 points • Link register: 128K points • Link special register: 512 points	• Link input: 16K points • Link output: 16K points • Link relay: 64K points • Link special relay: 512 points • Link register: 256K points • Link special register: 512 points
A firmware version that does not support the number of send points extension by using the link points extended setting*1	• Link input: 16K points • Link output: 16K points • Link relay: 32K points • Link special relay: 512 points • Link register: 128K points • Link special register: 512 points	

*1 For the firmware version of a CC-Link IE Controller Network module, refer to the following:
(MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application))

■Setting "Q Series Compatible Mode" in the link direct device setting

CC-Link IE Controller Network module	Firmware version of a C Controller module	
	'15'	'16' or later
A firmware version that supports the number of send points extension by using the link points extended setting or later*1	• Link input: 16K points • Link output: 16K points • Link relay: 32K points • Link special relay: 512 points • Link register: 128K points • Link special register: 512 points	
A firmware version that does not support the number of send points extension by using the link points extended setting*1		

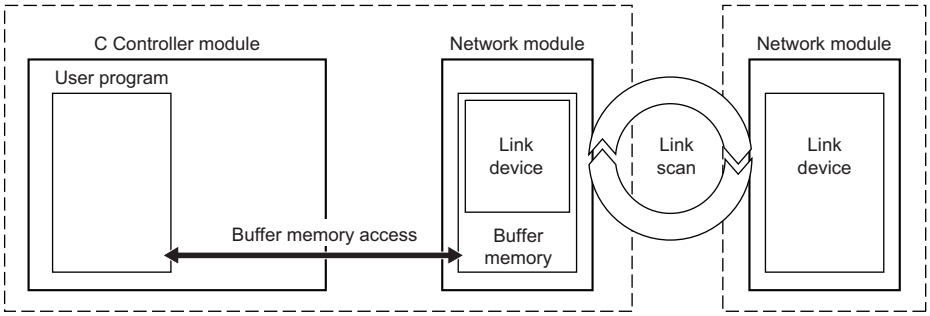
*1 For the firmware version of a CC-Link IE Controller Network module, refer to the following:
(MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application))

Buffer memory access

This method is used to access the buffer memory of a network module from a user program.

Data flow

The following figure shows the data flow to access buffer memory of a network module.



■User program functions

The following shows the functions used to access buffer memory of a network module.

Dedicated function library	Description
CCPU_ToBuf	To write data to the CPU buffer memory and intelligent function module buffer memory in the module on the specified module position.
CCPU_ToBuf_ISR	
CCPU_FromBuf	To read data from the CPU buffer memory and intelligent function module buffer memory in the module on the specified module position.
CCPU_FromBuf_ISR	

■Link devices specified with a function (buffer memory)

The following link devices are assigned to each specific address in the buffer memory.

In the dedicated function library, specify the device type with the device name defined for each function.

Link device		C Controller module dedicated function
Link relay*1	LB	DevSPG(0) to DevSPG(255)
Link register*1	LW	
Remote input	RX	
Remote output	RY	
Remote register	RWw	
	RWr	
Link special relay	SB	
Link special register	SW	

*1 Can be accessed when using CC-Link IE TSN.

When using a C Controller module dedicated function, specify the offset of the buffer memory.

For details of buffer memory addresses for each link device, refer to the user's manual for the network module used.

16.3 Transient Transmission Function

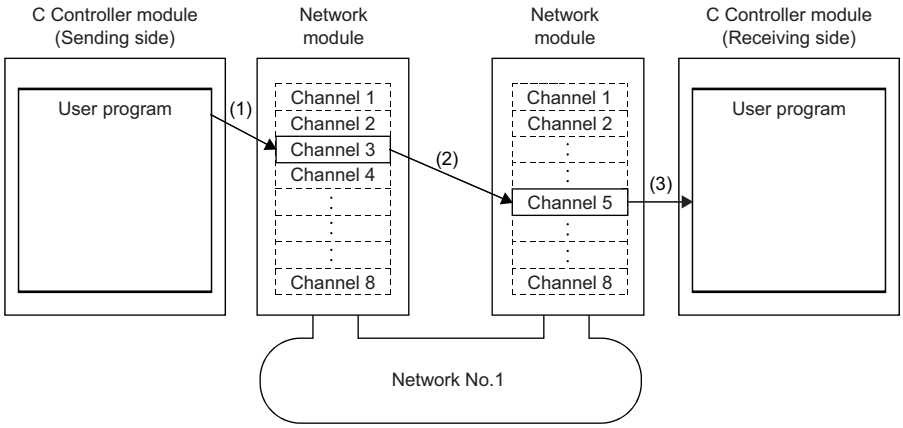
This function performs data communication with other stations when a communication request is issued. Communication with a different network can also be performed.
This section shows transient transmission using a C Controller module.
There are two methods available for transient transmission: message transmission using the channel of the network module which is controlled by a C Controller module, and direct access to devices of another station.

Message communication

This method is used to read/write data using the channel of a controlled network module from a C Controller module.
The message communication function allows data communication similar to link dedicated instructions i.e. SEND/RECV instructions.

Data flow

The following figure shows the data flow for message communication.



(1): The sending side C Controller module sends data by specifying the channel used in the own station and the channel used in the target station with the message send function of a user program.
(2): From the sending side channel, data is transmitted to the receiving side channel.
(3): The receiving side C Controller module (another station) reads messages from the receiving side channel with the message receive function of a user program.

Point

- The message communication function can be used even if CPU module on sending station or receiving station is other than C Controller module. For the target CPU module, use the SEND/RECV link dedicated instruction or a function equivalent to it.
- When sending data to the same channel of the receiving station with arrival confirmation, make sure to send the data after the receiving station reads the previously sent data with the message receive function (or the RECV instruction). Otherwise, an error will occur. If an error is detected, send the data again after a while.

■Channel

Each network module has a data area used for message communication, which is called 'channel.' By using multiple channels, simultaneous access from the host station to other stations, or simultaneous reading and writing to one module are allowed. The number of channels differ depending on network types.

■User program functions

The following shows the functions used for message communication.

Dedicated function library	Description
CCPU_DedicatedGInst	To execute dedicated instructions categorized as 'G' or 'GP.'
CCPU_DedicatedJInst	To execute dedicated instructions categorized as 'J' or 'JP.'
mdSendEx	To send messages.
mdReceiveEx	To receive messages.

Access to devices on another station

This method is used for direct access to devices on another station via a network from a user program. A network module on another station and its control CPU module can be accessed.

User program functions

Devices on another station can be accessed by using a user program with C Controller module dedicated functions.

Dedicated function library	Description
mdDevRstEx	To reset (turn OFF) bit devices.
mdDevSetEx	To set (turn ON) bit devices.
mdRandREx	To read devices randomly.
mdRandWEx	To write devices randomly.
mdReceiveEx	To read devices in a batch.
mdSendEx	To write devices in a batch.

■Accessible range of function

For the accessible range on each network, refer to the following:

📖MELSEC iQ-R C Controller Module Programming Manual

Devices specified with a function

■Access to link devices of another station

To directly access link devices, access them as link direct devices (J□\□) depending on network module specifications.

The following shows the specification method for accessing link direct devices (J□\□) with MELSEC data link functions.

In the dedicated function library, specify the device type with the device name defined for each function.

J□¥□
(1) (2)

Link device[*1]		Specification method for MELSEC data link function	
		(1), (2)	
		Device type (sDevType)[*2]	Start device number (ulDevNo)
Link input	Jn\X	DevLX (□)	Specify the start device number of an access target.
Link output	Jn\Y	DevLY(□)	
Link relay	Jn\B	DevLB (□)	
Link register	Jn\W[*3]	DevLW (□)	
Link special relay	Jn\SB	DevLSB (□)	
Link special register	Jn\SW	DevLSW (□)	

*1 Accessible link devices vary depending on network modules. For the accessible devices, refer to the manual for a network module.

*2 □: Specify the network number of an access target network module.

*3 For CC-Link IE TSN modules and CC-Link IE Field Network modules, access target devices (RWw, RWr, and LW) vary depending on a device number to be specified. Therefore, a device number which is actually accessed may be different from the specified device number.

Ex.

Accessing 'J1\W4000' (When a module of network number 1 is a CC-Link IE TSN module)

- Device type (sDevType): DevLW(1)
- Start device number (ulDevNo): 4000

LW0 in the CC-Link IE TSN module of the network number 1 is accessed.

■Access to the buffer memory of another station

To access the buffer memory of another station, specify a module access device or an intelligent function module device.

Link device		MELSEC data link function
Remote input	RX	DevSPG(0) to DevSPG(255)
Remote output	RY	
Remote register	RWw	
	RWr	
Link special relay	SB	
Link special register	SW	

■Access to a control CPU module of another station

For devices to be specified in a user program, refer to the following:

MELSEC iQ-R C Controller Module Programming Manual

16.4 Access Methods via a Network Module

This section explains the access methods using each network module.

CC-Link IE Controller Network module

The following shows the access methods via a CC-Link IE Controller Network module.

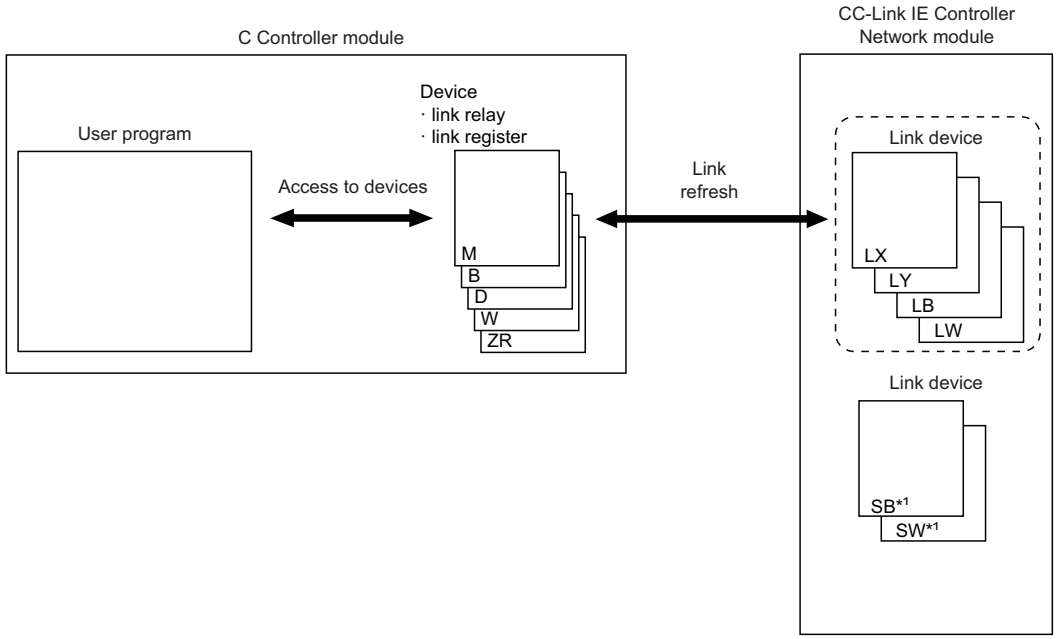
Available access methods

The following access methods are available.

Transmission type	Access method
Cyclic transmission	Access by link refresh
	Direct access
Transient transmission	Message communication
	Access to devices on another station

Access by link refresh

This method is used to access devices of a C Controller module from a user program using link refresh. Data in the device is transmitted cyclically to another station by reading from/writing to link devices of a network module by link refresh.



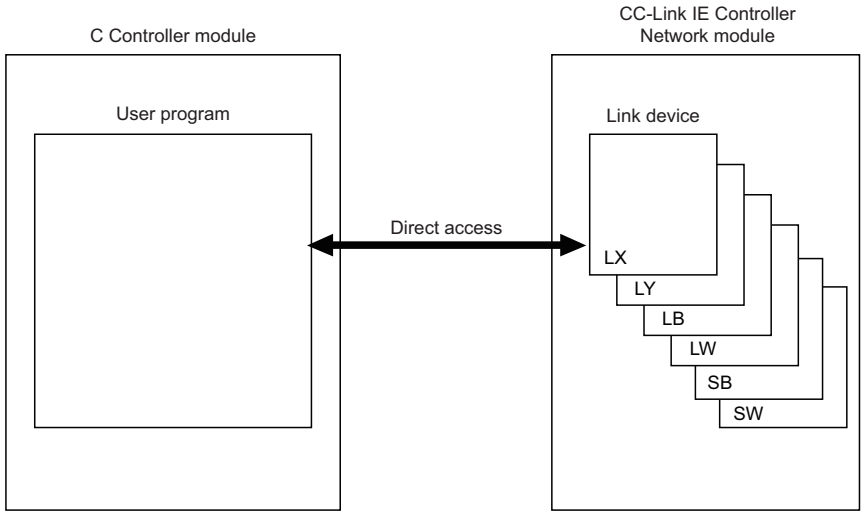
*1 Link refresh cannot be performed for SB/SW. Use the direct access.

Direct access

This method is used for direct access to devices in a network module from a user program. The link device data in the network module is transmitted cyclically to another station.

Point

To directly access link devices, access them as link direct devices (J□\□) depending on network module specifications. (☞ Page 202 Direct access)

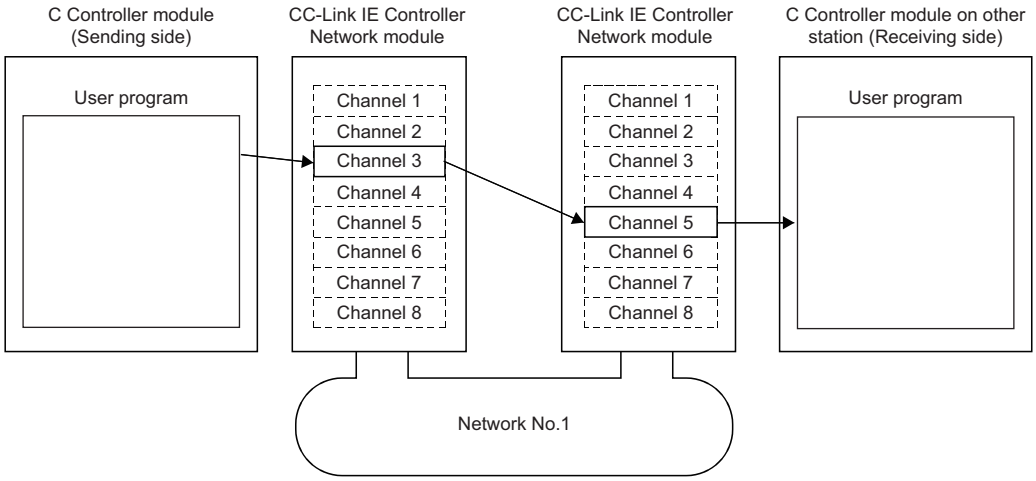


Message communication

A message communication can be performed via a CC-Link IE Controller Network module controlled by a C Controller module.

■Number of channels

For CC-Link IE Controller Network module, up to eight channels can be used for message communication.



CC-Link IE Field Network module

The following shows the access methods via a CC-Link IE Field Network module.

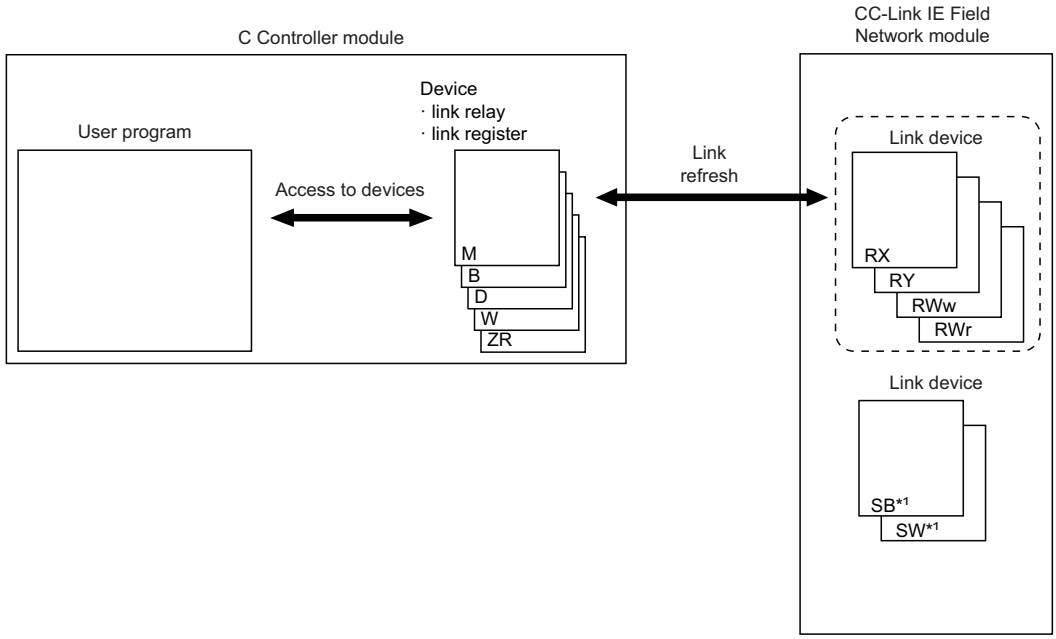
Available access methods

The following access methods are available.

Transmission type	Access method
Cyclic transmission	Access by link refresh
	Direct access
	Buffer memory access
Transient transmission	Message communication
	Access to devices on another station

Access by link refresh

This method is used to access devices of a C Controller module from a user program using link refresh. Data in the device is transmitted cyclically to another station by reading from/writing to link devices of a network module by link refresh.



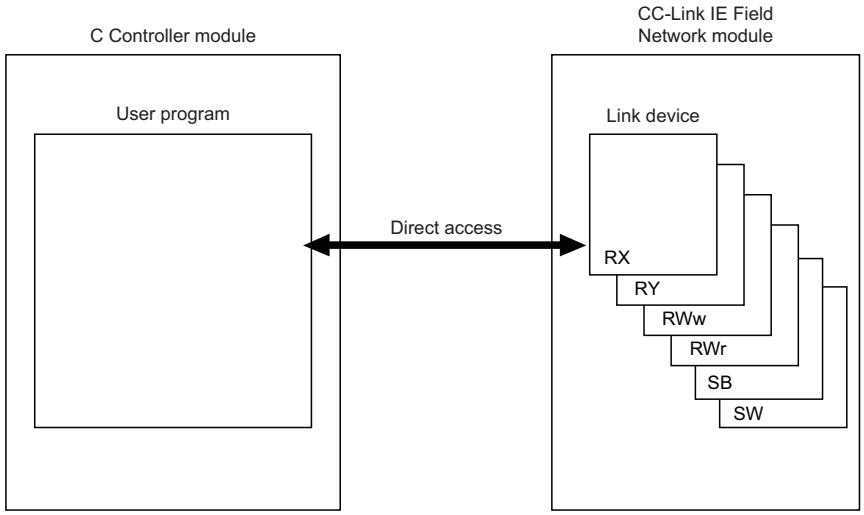
*1 Link refresh cannot be performed for SB/SW. Use the direct access or buffer memory access.

Direct access

This method is used for direct access to devices in a network module from a user program. The link device data in the network module is transmitted cyclically to another station.

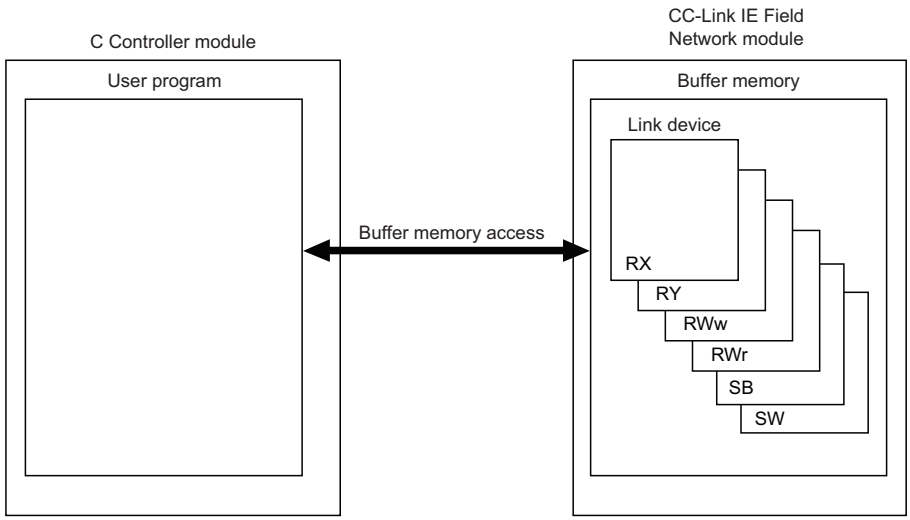
Point

To directly access link devices, access them as link direct devices (J□\□) depending on network module specifications. (☞ Page 202 Direct access)



Buffer memory access

This method is used to access the buffer memory of a network module from a user program. The data in the buffer memory is refreshed with link devices and transmitted cyclically to another station.

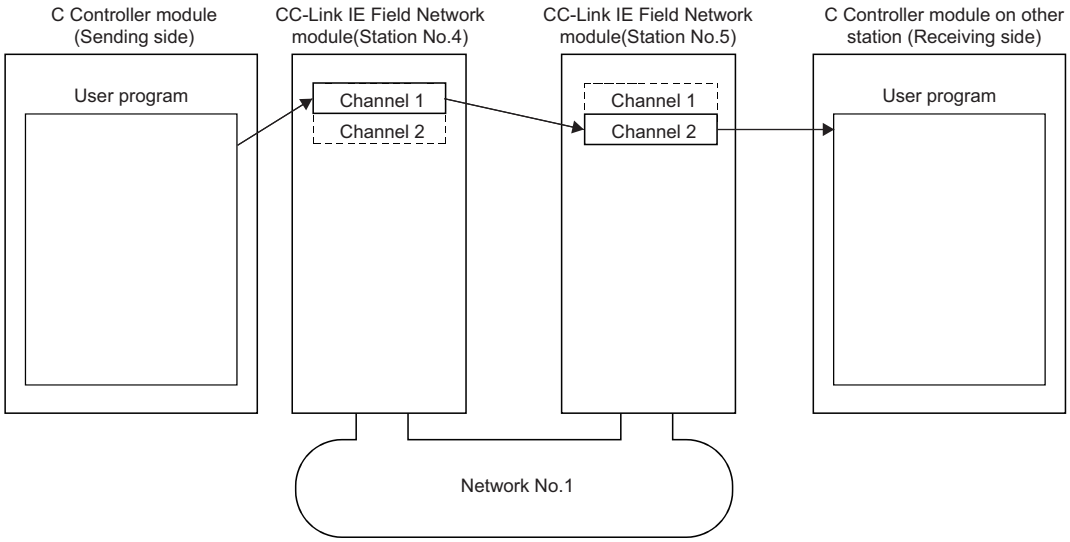


Message communication

A message communication can be performed via a CC-Link IE Field Network module controlled by a C Controller module.

■Number of channels

For CC-Link IE Field Network module, up to two channels can be used for message communication.



CC-Link IE TSN module

The following shows the access methods via a CC-Link IE TSN module.

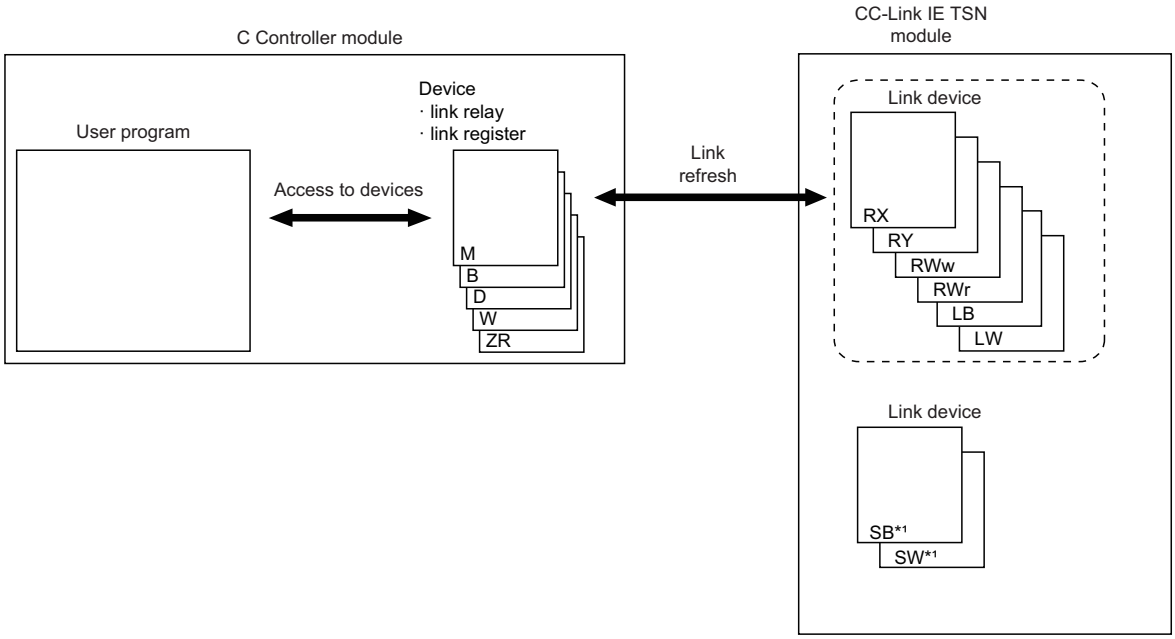
Available access methods

The following access methods are available.

Transmission type	Access method
Cyclic transmission	Access by link refresh
	Direct access
	Buffer memory access
Transient transmission	Message communication
	Access to devices on another station

Access by link refresh

This method is used to access devices of a C Controller module from a user program using link refresh. Data in the device is transmitted cyclically to another station by reading from/writing to link devices of a network module by link refresh.



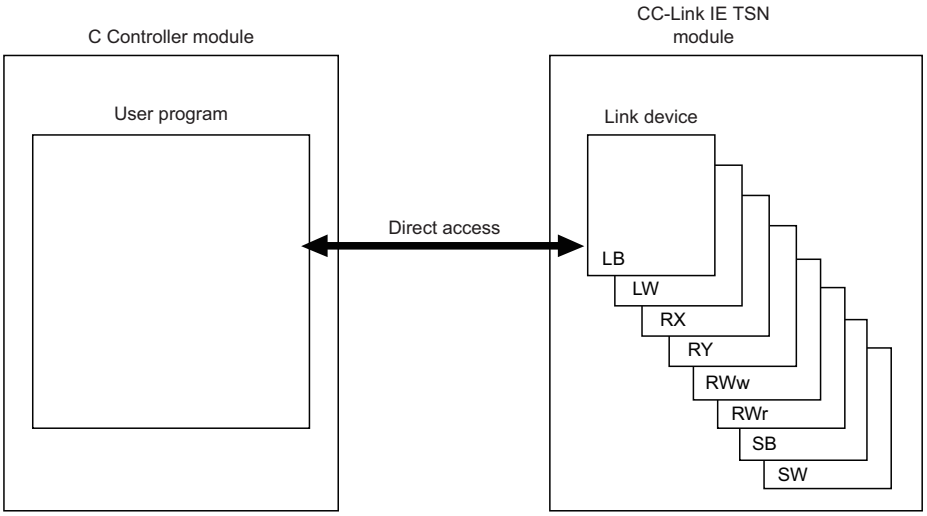
*1 Link refresh cannot be performed for SB/SW. Use the direct access or buffer memory access.

Direct access

This method is used for direct access to devices in a network module from a user program. The link device data in the network module is transmitted cyclically to another station.

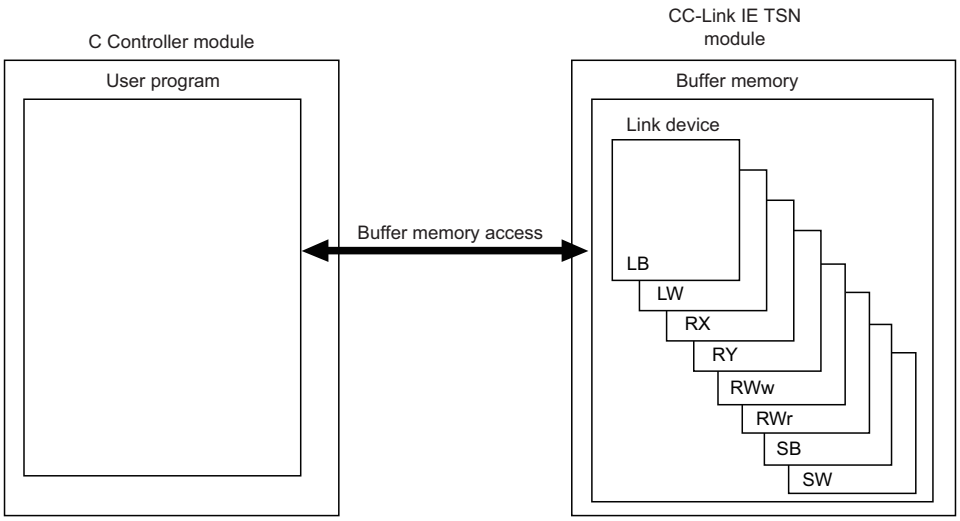
Point

To directly access link devices, access them as link direct devices (J□\□) depending on network module specifications. (☞ Page 202 Direct access)



Buffer memory access

This method is used to access the buffer memory of a network module from a user program. The data in the buffer memory is refreshed with link devices and transmitted cyclically to another station.



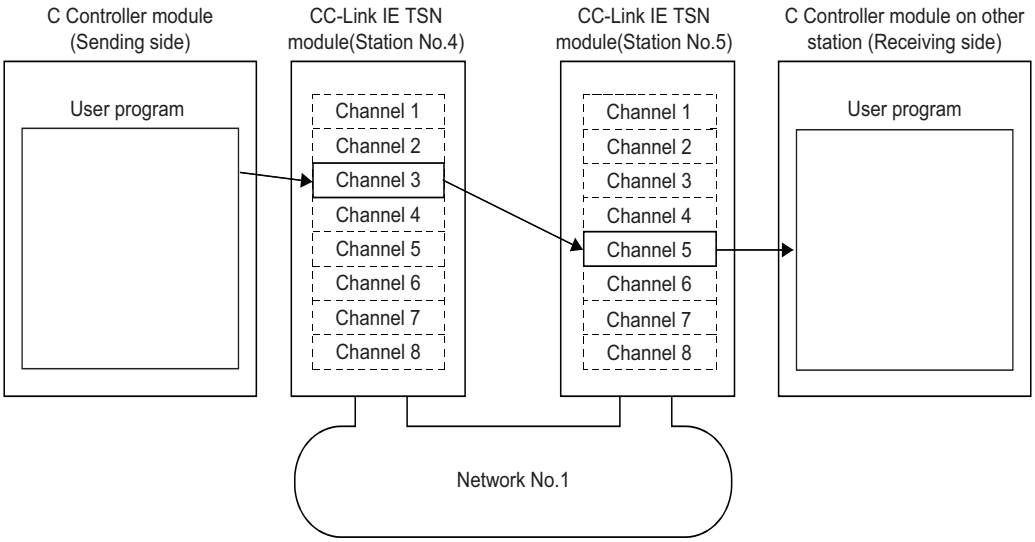
16

Message communication

A message communication can be performed via a CC-Link IE TSN module controlled by a C Controller module.

■Number of channels

For CC-Link IE TSN module, up to eight channels can be used for message communication.



MELSECNET/H network module

The following shows the access methods via a MELSECNET/H network module.

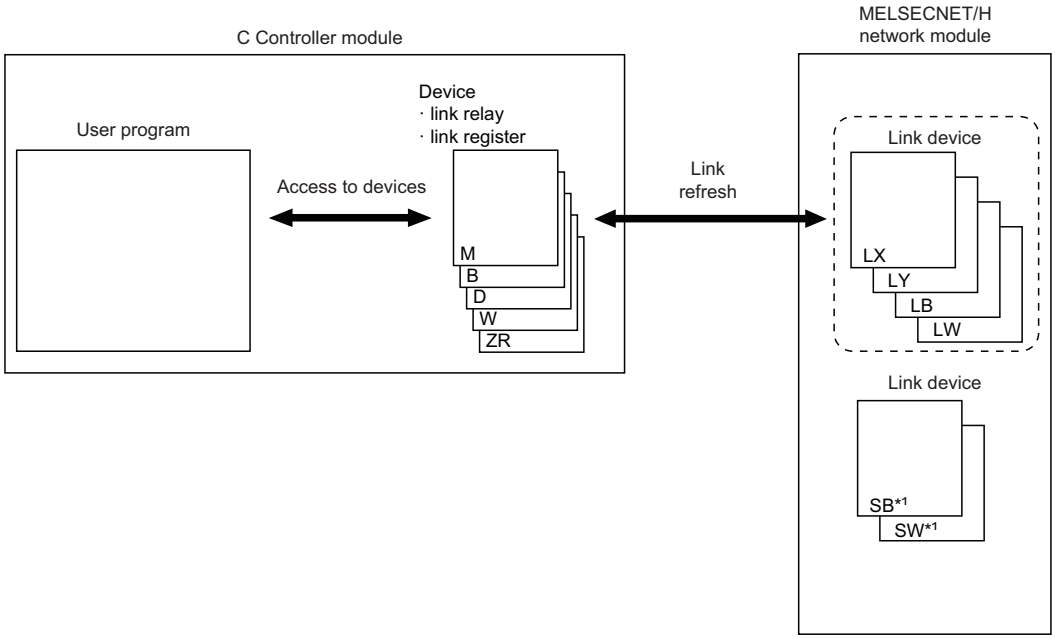
Available access methods

The following access methods are available.

Transmission type	Access method
Cyclic transmission	Access by link refresh
	Direct access
Transient transmission	Message communication
	Access to devices on another station

Access by link refresh

This method is used to access devices of a C Controller module from a user program using link refresh. Data in the device is transmitted cyclically to another station by reading from/writing to link devices of a network module by link refresh.



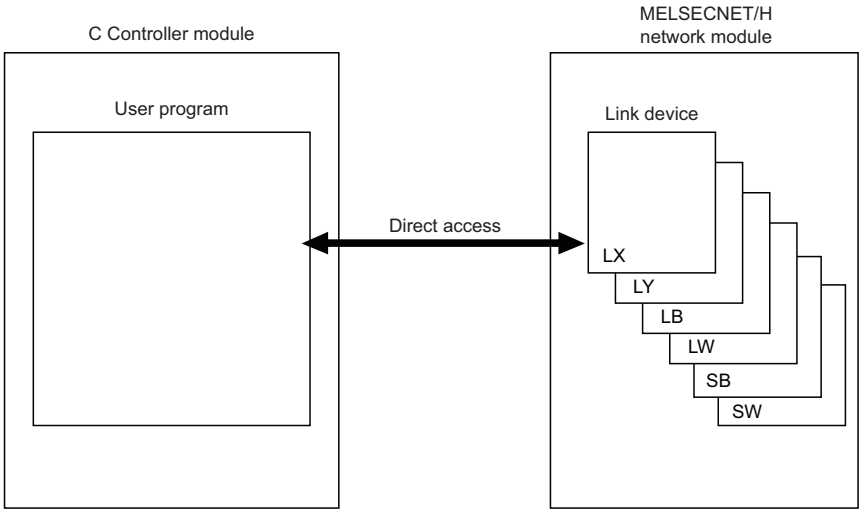
*1 Link refresh cannot be performed for SB/SW. Use the direct access.

Direct access

This method is used for direct access to devices in a network module from a user program. The link device data in the network module is transmitted cyclically to another station.

Point

To directly access link devices, access them as link direct devices (J□\□) depending on network module specifications. (☞ Page 202 Direct access)

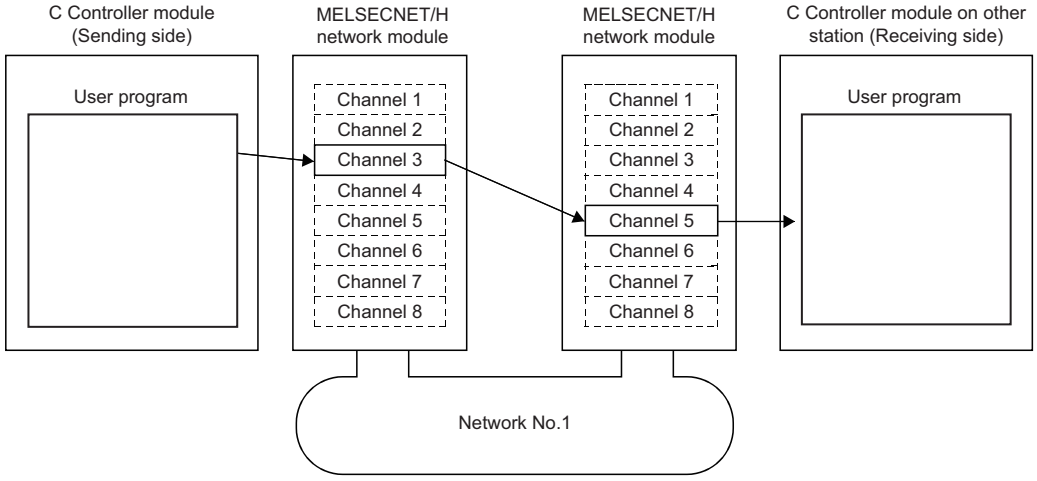


Message communication

A message communication can be performed via a MELSECNET/H network module controlled by a C Controller module.

■Number of channels

For MELSECNET/H network module, up to eight channels can be used for message communication.



CC-Link module

The following shows the access methods via a CC-Link module.

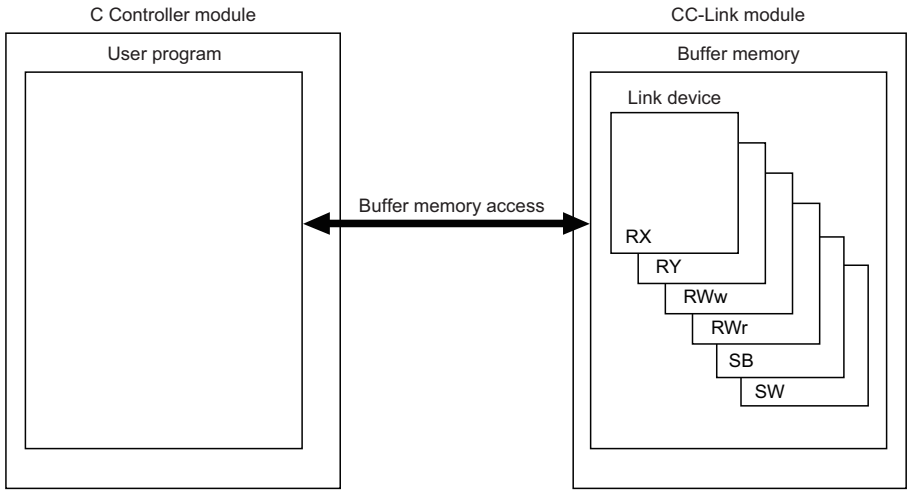
Available access methods

The following access methods are available.

Transmission type	Access method
Cyclic transmission	Buffer memory access
Transient transmission	Access to devices on another station

Buffer memory access

This method is used to access the buffer memory of a network module from a user program. The data in the buffer memory is refreshed with link devices and transmitted cyclically to another station.



16.5 Processing Time

This section shows the processing time of the cyclic transmission related to network module access.
For the processing time of the transient transmission, refer to the processing time of each function.

Link refresh time

The following shows the processing time of link refresh.

CC-Link IE Controller Network

The processing time when using CC-Link IE Controller Network is as follows.

- β_T: Total link refresh time (sending side)
- β_R: Total link refresh time (receiving side)
- KM1, KM2: Constant

Network module mounting position	KM1	KM2
Main base unit	1.0	0.36×10^{-3}
Extension base unit	1.0	0.97×10^{-3}

- LB: Total number of points of link relays to be refreshed*1
- LX: Total number of points of link inputs to be refreshed*1
- LY: Total number of points of link outputs to be refreshed*1
- LW: Total number of points of link registers to be refreshed*1

*1 This is the total number of points of link devices set for the setting range of "Refresh Setting" and "Network Range Assignment" ("Network Configuration Settings"). The number of points assigned to a reserved station is not included.

Link refresh time can be calculated with the following formula depending on the number of assigned points of link devices.

Time	Formula
Link refresh time (ms)	$\beta_T, \beta_R = KM1 + KM2 \times \{((LB + LX + LY) \div 16) + LW\}$

CC-Link IE Field Network

The processing time when using CC-Link IE Field Network is as follows.

- β_T: Total link refresh time (sending side)
- β_R: Total link refresh time (receiving side)
- KM1, KM2: Constant

Network module mounting position	KM1	KM2
Main base unit	1.0	0.41×10^{-3}
Extension base unit	1.0	0.99×10^{-3}

- RX: Total number of points of link inputs to be refreshed*1
- RY: Total number of points of link outputs to be refreshed*1
- RWw, RWr: Total number of points of link registers to be refreshed*1

*1 This is the total number of points of link devices set for the setting range of "Refresh Setting" and "Network Range Assignment" ("Network Configuration Settings"). The number of points assigned to a reserved station is not included.

Link refresh time can be calculated with the following formula depending on the number of assigned points of link devices.

Time	Formula
Link refresh time (ms)	$\beta_T, \beta_R = KM1 + KM2 \times \{((RX + RY) \div 16) + RWw + RWr\}$

CC-Link IE TSN

The processing time when using CC-Link IE TSN is as follows.

- β_T: Total link refresh time (sending side)
- β_R: Total link refresh time (receiving side)
- KM1, KM2: Constant

Network module mounting position	KM1	KM2
Main base unit	1.0	0.41×10^{-3}
Extension base unit	1.0	0.99×10^{-3}

- RX: Total number of points of link inputs to be refreshed*1
- RY: Total number of points of link outputs to be refreshed*1
- RWw, RWr: Total number of points of link registers to be refreshed*1
- LB: Total number of points of link relays to be refreshed*1
- LW: Total number of points of link registers to be refreshed*1

*1 This is the total number of points of link devices set for the setting range of "Refresh Setting" and "Network Range Assignment" ("Network Configuration Settings"). The number of points assigned to a reserved station is not included.

Link refresh time can be calculated with the following formula depending on the number of assigned points of link devices.

Time	Formula
Link refresh time (ms)	β_T, β_R = KM1 + KM2 × {((RX + RY + LB) ÷ 16) + RWw + RWr + LW}

MELSECNET/H network

The processing time when using MELSECNET/H network is as follows.

- β_T: Total link refresh time (sending side)
- β_R: Total link refresh time (receiving side)
- KM1, KM2: Constant

Network module mounting position	KM1	KM2
RQ extension base unit	65	0.41×10^{-3}
Q extension base unit	65	0.92×10^{-3}

- LB: Total number of points of link relays to be refreshed*1
- LX: Total number of points of link inputs to be refreshed*1
- LY: Total number of points of link outputs to be refreshed*1
- LW: Total number of points of link registers to be refreshed*1

*1 This is the total number of points of link devices set for the setting range of "Refresh Setting" and "Network Range Assignment" ("Network Configuration Settings"). The number of points assigned to a reserved station is not included.

Link refresh time can be calculated with the following formula depending on the number of assigned points of link devices.

Time	Formula
Link refresh time (ms)	β_T, β_R = KM1 + KM2 × {((LB + LX + LY) ÷ 16) + LW}

CC-Link Network

The processing time when using CC-Link Network is as follows.

- β_T: Total link refresh time (sending side)
- β_R: Total link refresh time (receiving side)
- KM1, KM2: Constant

Network module mounting position	KM1	KM2
Main/extension base unit	1.0	0.36×10^{-3}

- RX: Total number of points of link inputs to be refreshed*1
- RY: Total number of points of link outputs to be refreshed*1
- RWw, RWr: Total number of points of link registers to be refreshed*1

*1 This is the total number of points of link devices set for the setting range of "Refresh Setting" and "Network Range Assignment" ("Network Configuration Settings"). The number of points assigned to a reserved station is not included.

Link refresh time can be calculated with the following formula depending on the number of assigned points of link devices.

Time	Formula
Link refresh time (ms)	$\beta_T, \beta_R = KM1 + KM2 \times \{((RX + RY) \div 16) + RWw + RWr\}$

Refresh cycle

The following shows how to calculate the refresh cycle.

For the setting method of refresh cycle, refer to the following:

☞ Page 121 Fixed Cycle Processing Function

How to decide the refresh cycle

Follow the procedure below to decide the refresh cycle.

1. Calculation of the total link refresh time
2. Preliminary decision of the refresh cycle
3. Pre-operation of the C Controller system
4. Decision of the refresh cycle

■Calculation of the total link refresh time

The total link refresh time is equal to the sum of the link refresh time of all of the network modules controlled by a C Controller module. The total link refresh time of a C Controller system in operation can be checked with the special registers (SD526 to SD531).

For calculation method of link refresh time for each network module, refer to the following:

☞ Page 220 Link refresh time

■Preliminary decision of the refresh cycle

Set the value which satisfies the following relational expression to the refresh cycle and total link refresh time.

- Refresh cycle > Total link refresh time

Point

Pay attention to the processing of a user program so that the refresh cycle meets the system specifications of the C Controller system.

■Pre-operation of the C Controller system

Confirm whether the C Controller system operates correctly by the refresh cycle decided preliminarily.

Check the following at the time of pre-operation.

Check item	Description
Any error has been occurred on the C Controller module or not.	If the processing of the C Controller module has not been completed within the refresh cycle, an error, "Refresh cycle exceeded (1846H)" occurs.
The operation of the C Controller system meets the system specifications or not.	Check that the C Controller system operates correctly. If the operation of the user program is suspended more frequently than its normal operation time, the system specification may not be met.

- Take the corrective actions shown below in order for the system to operate correctly when the C Controller system operates abnormally or the processing performance of the user program is lowered.

Corrective action	Description
Increase the refresh cycle setting value.	Set the refresh cycle again so that the following relational expression is met, by using the link refresh time (maximum value) as an indication. • Measured value of link refresh time (maximum value) < Refresh cycle
Reduce the setting number of refresh points.	Review the number of refresh points in refresh parameter settings.
Review the user program.	Review the processing contents and task structure, and correct them to meet the system specifications and the processing performance.

Transmission delay time of cyclic transmission

The following shows the transmission delay time of cyclic transmission.

CC-Link IE Controller Network

For the calculation formulas of transmission delay time for a CC-Link IE Controller Network, refer to the following:

MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)

With reference to above mentioned manual, substitute the sequence scan time on sending side (ST) and the sequence scan time on receiving side (SR) with the following processing time.

ST = LT, SR = LR

- LT: Refresh cycle of a C Controller module (sending side)
- LR: Refresh cycle of a C Controller module (receiving side)

CC-Link IE Field Network

For information on the calculation formula for transmission delay time when using CC-Link IE Field Network, refer to the following:

MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

With reference to above mentioned manual, substitute the sequence scan time of the master station (SM) and the sequence scan time of the local station (SL) with the following processing time.

SM, SL = P = R + α

- P: Processing time on a C Controller system
- R: Refresh cycle
- α: Link refresh time

It does not take time for refresh when direct access or buffer memory access is used. For information on the function processing time, refer to the following:

Page 380 Processing Time of Functions

CC-Link IE TSN

For information on the calculation formula for transmission delay time when using CC-Link IE TSN, refer to the following:

MELSEC iQ-R CC-Link IE TSN User's Manual (Application)

With reference to above mentioned manual, substitute the sequence scan time of the master station (SM) and the sequence scan time of the local station (SL) with the following processing time.

SM, SL = P = R + α

- P: Processing time on a C Controller system
- R: Refresh cycle
- α: Link refresh time

It does not take time for refresh when direct access or buffer memory access is used. For information on the function processing time, refer to the following:

Page 380 Processing Time of Functions

MELSECNET/H network

For information on the calculation formula for transmission delay time when using MELSECNET/H network, refer to the following:

📖 Q Corresponding MELSECNET/H Network System Reference Manual (PLC to PLC network)

With reference to above mentioned manual, substitute the sequence scan time of the sending side (ST) and the sequence scan time of the receiving side (SR) with the following processing time.

ST = LT, SR = LR

- LT: Refresh cycle of a C Controller module (sending side)
- LR: Refresh cycle of a C Controller module (receiving side)

CC-Link Network

For information on the calculation formula for transmission delay time when using CC-Link Network, refer to the following:

📖MELSEC iQ-R CC-Link System Master/Local Module User's Manual (Application)

With reference to above mentioned manual, substitute the sequence scan time of the master station (SM) and the sequence scan time of the local station (SL) with the following processing time.

SM, SL = P = R + α

- P: Processing time on a C Controller system
- R: Refresh cycle
- α: Link refresh time

Transmission delay calculation example

The following shows a calculation example for the transmission delay time for single network system

The system configuration and its settings are as shown below.

Item		Description
CPU module on sending side, receiving side		C Controller module
Total station numbers per one network		Two stations (one control station and one normal station)
Total number of points of link devices	LB/LW	1024 points
	LX/LY	0 points
	SB/SW	0 points
Refresh cycle		100 ms
Transient transmission		None
Station-based block data assurance		Enabled
CC-Link IE Controller Network module		Main base unit
Error station		None

■Link refresh time (β_T, β_R)

The calculation formula is as follows:

Time	Formula
Link refresh time (ms)	β_T, β_R = KM1 + KM2 × {(LB + LX + LY) ÷ 16 + LW}

The values of each variable are as follows:

- KM1: Constant = 1.0
- KM2: Constant = 0.36×10^{-3}
- LB, LW: Total number of points of link devices = 1024
- LX, LY: Total number of points of link devices = 0

Link refresh time can be found by putting values of each variable in the following formula:

- $\beta_T, \beta_R = 1.0 + 0.36 \times 10^{-3} \times \{((1024 + 0 + 0) \div 16) + 1024\} \fallingdotseq 1.39$ (ms)

■Transmission delay time

When the station-based block data assurance setting is enabled and LT is longer than LS (LT > LS), the calculation formula is as follows:

Calculated value	Transmission delay time (ms)
Normal value	$(LT + \beta_T) \times 1.5 + LS \times 0.5 + (LR + \beta_R) \times 0.5$
Maximum value	$(LT + \beta_T) \times 2 + LS \times 1 + LR + \beta_R$

The values of each variable are as follows:

- LT, LR: Refresh cycle = 100 (ms)
- β_T, β_R: Link refresh time $\fallingdotseq$ 1.39 (ms)
- LS: Link scan time $\fallingdotseq$ 1.25 (ms)

Link refresh time can be found by putting values of each variable in the following formula:

- Normal value = (100+1.39) × 1.5 + 1.25 × 0.5 + (100+1.39) × 0.5 = 203.405 (ms)
- Maximum value = (100+1.39) × 2 + 1.25 × 1 + 100 + 1.39 = 305.420 (ms)

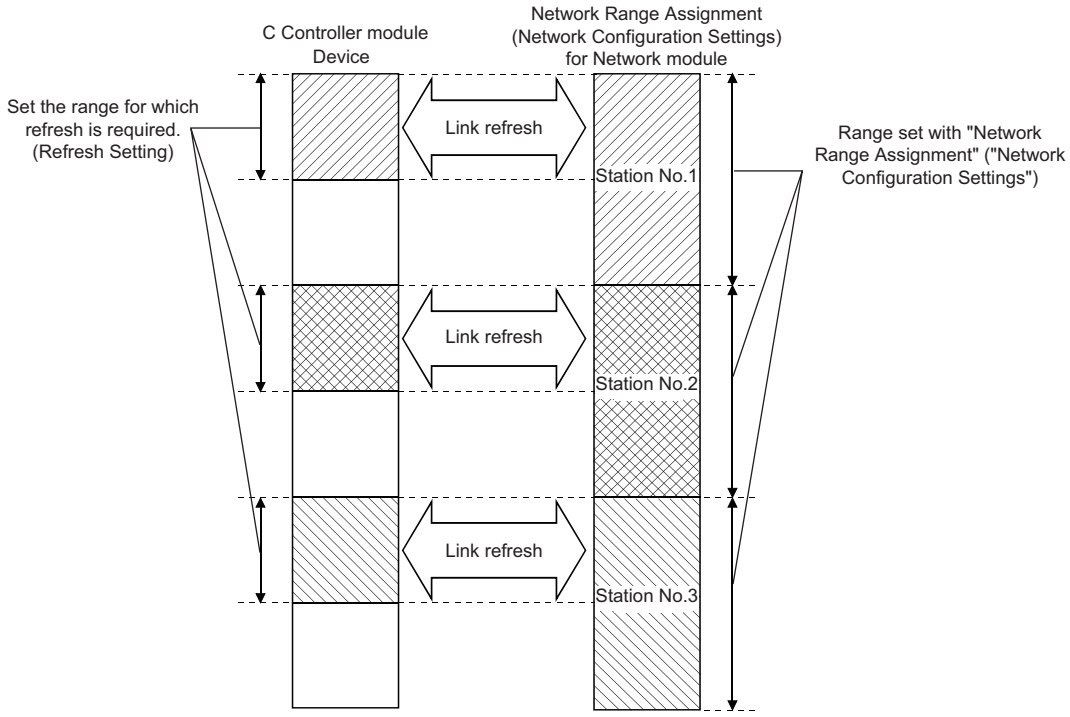
Reduction of link refresh time

The link refresh time can be shortened by reducing refresh points to a C Controller module by the settings for "Refresh Setting"/"Network Range Assignment" ("Network Configuration Settings") and with direct access of link devices.

How to reduce refresh points

■Reducing refresh points using parameters

In "Refresh Setting," set only the necessary range to be refreshed. The refresh of only necessary range can reduce the link refresh time.



■Reducing refresh points using direct access/buffer memory access

Excluding the rarely used link devices on own station from the link refresh range and using direct access or buffer access instead can reduce the link refresh time.

17 MULTIPLE CPU SYSTEM FUNCTION

An I/O module and intelligent function module can be controlled by each CPU module mounted on a base unit. In addition, communication between CPU modules can be performed.

Point

- For the concept of the multiple CPU system (system configuration specifications such as mounting position of CPU modules and assignment of CPU number and I/O number), refer to the following:

MELSEC iQ-R Module Configuration Manual

- For the start-up (settings and operating procedure) of a multiple CPU system, refer to the following:

CW Configurator Operating Manual

Restriction

- The startup time of the multiple CPU system may be long depending on the system configuration of the installed modules, boot operation, the configuration of the system parameters and CPU parameters. When the SD memory card diagnostics is performed due to the operation such as power OFF during the access to the SD memory card, the startup time of the multiple CPU system may be long as well.
- Create a program so that only one CPU module accesses a MELSEC-Q series module when possible. If multiple CPU modules access a MELSEC-Q series module simultaneously, the program execution processing (including the execution processing of interrupt programs) may be extended due to the access waiting time.

17.1 Out-of-Group I/O Fetch

In a multiple CPU system, the access range of the controlled module is different from that of the non-controlled module. The out-of-group I/O fetch function enables data which cannot be accessed to be imported. However, data cannot be imported from the module that is an inter-module synchronization target.

Accessing controlled module

The access range to the controlled module of the CPU module is the same as that of the single CPU system. I/O refresh for the controlled module or read from/write to the buffer memory of an intelligent function module is enabled.

Accessing non-controlled module

Access to the non-controlled module of the CPU module is limited to reading the buffer memory of the intelligent function module. However, the ON/OFF data of output (Y) of the non-controlled module can be imported by "I/O Setting Outside Group."

○: Accessible, ×: Not accessible

Access target		Accessibility	
		When "Import" is selected in "I/O Setting Outside Group"	When "Not Imported" is selected in "I/O Setting Outside Group"
Input (X)	Read data from another CPU module	○	×
Output (Y)	Write data to another CPU module	×	× (Non-processing)
	Read data from another CPU module	○	× (Non-processing)
Buffer memory of an intelligent function module	Read data	○	○
	Write data	× (Error)	× (Error)

I/O Setting Outside Group

Set the I/O state outside the group.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Other PLC Control Module Setting] ⇨ [I/O Setting Outside Group]

Window



Displayed items

Item	Description	Setting range	Default
Input Status Outside Group	Set the input state outside the group.	• Not Imported • Import	Not Imported
Output Status Outside Group	Set the output state outside the group.	• Not Imported • Import	Not Imported

Importing input (X) data

When "Import" is set for the input status in "I/O Setting Outside Group," the ON/OFF data from an input module or an intelligent function module controlled by another CPU can be imported.

■Modules from which input (X) data can be imported

Input (X) data can be imported from the following modules mounted on the main/extension base unit.

- Input module
- I/O combined module*1
- Intelligent function module

*1 For MELSEC iQ-R series, the areas assigned to the output portion (unused area) are not imported. The ON/OFF state is retained.

Point

Input (X) data can also be imported by using a user program with the C Controller module dedicated functions (CCPU_X_In_BitEx, CCPU_X_In_WordEx).

Importing output (Y) data

When "Import" is set for the output status in "I/O Setting Outside Group," the ON/OFF data from an output module or an intelligent function module controlled by another CPU can be imported.

■Modules from which output (Y) data can be imported*1

Input (X) data can be imported from the following modules mounted on the main/extension base unit.

- Output module
- I/O combined module
- Intelligent function module

*1 Data cannot be imported from MELSEC-Q series modules.

Point

Output (Y) data can also be imported by using a user program with the C Controller module dedicated functions (CCPU_Y_In_BitEx, CCPU_Y_In_WordEx).

Output to an output module/intelligent function module

ON/OFF data cannot be output to non-controlled modules.

If the output state of the non-controlled module is turned ON/OFF by a program, the output state in the CPU module is changed; however, the output state of the non-controlled module is not changed.

Accessing the buffer memory of an intelligent function module

■Reading data in buffer memory

Use the following functions to read data from the buffer memory of an intelligent function module.

- C Controller module dedicated function (CCPU_FromBuf)
- MELSEC data link function (mdReceiveEx, mdRandREx)

■Writing data to buffer memory

Use the following functions to write data to the buffer memory of an intelligent function module.

- C Controller module dedicated function (CCPU_ToBuf)
- MELSEC data link function (mdSendEx, mdRandWEx)

17.2 Operation Settings

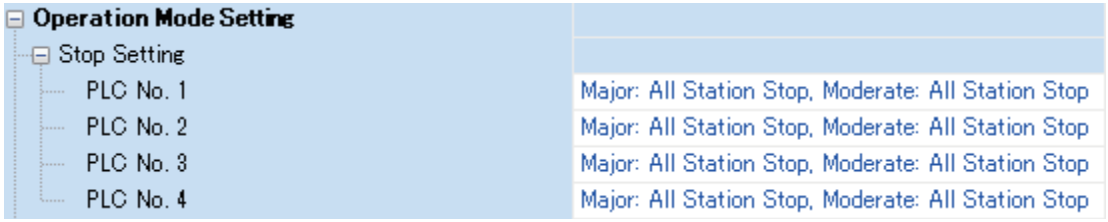
This section shows the operation settings for the multiple CPU system.

Stop setting

Set whether to stop or continue the operation in all CPUs if a major or moderate error occurred on each CPU.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Operation Mode Setting] ⇨ [Stop Setting]

Window



Displayed items

Item	Description	Setting range	Default
PLC No.1	Set whether to stop the operation of all CPUs if a major or moderate error occurred on CPU No.1.	• Major: All Station Stop Moderate: All Station Stop • Major: All Station Stop Moderate: All Station Continue • Major: All Station Continue Moderate: All Station Continue	Major: All Station Stop Moderate: All Station Stop
PLC No.2	Set whether to stop the operation of all CPUs if a major or moderate error occurred on CPU No.2.		
PLC No.3	Set whether to stop the operation of all CPUs if a major or moderate error occurred on CPU No.3.		
PLC No.4	Set whether to stop the operation of all CPUs if a major or moderate error occurred on CPU No.4.		

Synchronous startup setting

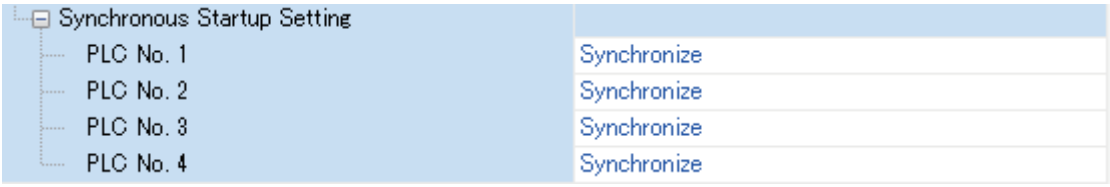
By synchronizing start-up time of each CPU module, operations can be started at the same time in the entire multiple CPU system. An interlock program that monitors the start-up time for each CPU module other than C Controller module is not required. However, this setting delays the system start-up.

An interlock program that monitors the execution completion of the script file is required for a C Controller module in a multiple CPU system configuration. The execution completion of the script file can be checked by the READY LED status indication. The status indication of the READY LED can be checked by following methods:

- Using the C Controller module dedicated function (CCPU_GetLEDStatus).
- Checking the special register (SD201).

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Operation Mode Setting] ⇨ [Synchronous Startup Setting]

Window



Displayed items

Item	Description	Setting range	Default
PLC No.1	Set the CPU No. of which start-up time is to be synchronized in a multiple CPU system.	• Synchronize • Do not Synchronize	Synchronize
PLC No.2			
PLC No.3			
PLC No.4			

Point

- Group setting for start-up synchronization is available. For example, a setting in which only CPU No.1 and No.2 start synchronously in a multiple CPU system with four CPU modules is possible.
- If a reserved (empty) CPU is specified to synchronize, the reserved CPU is ignored and the other CPU will be started.
- This setting is designed to access each CPU module with no interlock in the multiple CPU system. It is not intended to be used for starting operation at the same time among the CPU modules after the startup.

Clock data

The clock data in CPU No.2 to No.4 are synchronized with the clock data set to CPU No.1 automatically. (Even if the clock data is set to each CPU, they will be overwritten). Therefore, the system-unified clock data can be used only by setting the clock data for CPU No.1. (☞ Page 103 Clock Function)

Point

As with the clock data, the time zone setting for CPU No.2 to CPU No.4 follows the setting configured to CPU No.1. (☞ Page 104 Time zone setting)

17.3 Multiple CPU Parameter Check

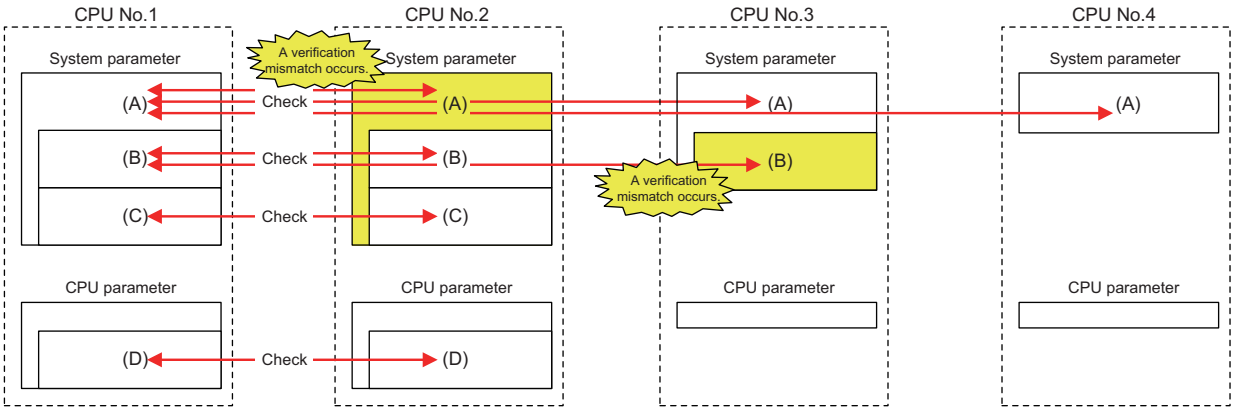
In a multiple CPU system, whether the same setting is configured for the multiple CPU refresh points between the system parameter and CPU parameter of each CPU is checked at the following timing. However, for the fixed cycle communication setting and the inter-module synchronization setting, the parameter check is performed for the only CPU which uses the respective functions.

- When the power is turned ON
- When the CPU No.1 is reset

Timing	Parameter to be checked	Check conditions for CPU No.1	Checking condition for CPU No.2 and later
Power ON or CPU No.1 is reset	System parameters (other than "Fixed Scan Communication Setting" and "Inter-module Synchronization Setting")	The check is not conducted.	Compares with the parameter of the smallest CPU number.
	Fixed Scan Communication Setting	The CPUs with no fixed cycle communication setting are not checked. The CPUs with a fixed cycle communication setting will be compared to the parameters with those of the CPU of the smallest number.	
	Inter-module Synchronization Setting	The CPUs with no inter-module synchronization setting are not checked. The CPUs with an inter-module synchronization setting will be compared to the parameters with those of the CPU of the smallest number.	
	CPU parameters (number of points of refresh setting)	The CPUs with no fixed cycle communication setting are not checked. The CPUs with a fixed cycle communication setting will be compared to the parameters with those of the CPU of the smallest number.	

Ex.

An error is detected in CPU No.2 and 3 by parameter check, and CPU No.1 and 4 starts up normally. (Operation in error)



○: Set (* if the next number is the same, it refers to the same parameter), ×: No setting

Item			Setting conditions for each CPU			
			CPU No.1	CPU No.2	CPU No.3	CPU No.4
Presence of parameters	System parameters (other than "Fixed Scan Communication Setting" and "Inter-module Synchronization Setting")	(A)	○(1)	○(5)	○(1)	○(1)
	Fixed Scan Communication Setting	(B)	○(2)	○(2)	○(6)	×
	Inter-module Synchronization Setting	(C)	○(3)	○(3)	×	×
	CPU parameters (number of points of refresh setting)	(D)	○(4)	○(4)	×	×

17.4 Data Communication between CPU Modules

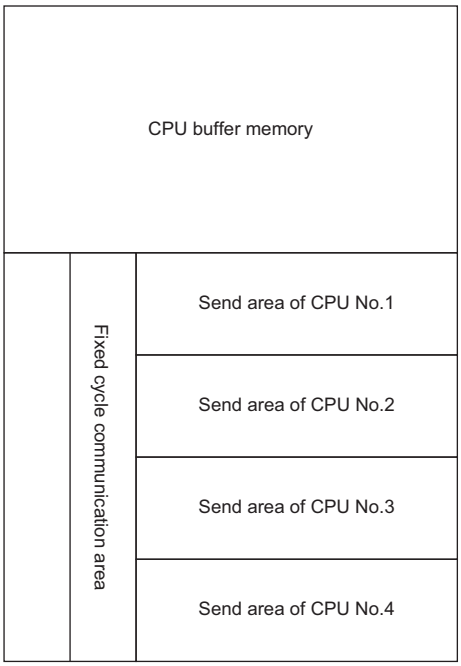
Data can be sent/received between CPU modules in a multiple CPU system. The direct access communication enables data writing or reading between CPU modules.

Communication method	Application	Description
Data communication with CPU buffer memory	Use this communication method when performing data send/receive at the timing of each CPU module.	The sending side CPU module writes data to the CPU buffer memory in the host CPU. The receiving side CPU module reads data from the CPU buffer memory of the send target CPU module (another CPU).
Data communication with fixed cycle communication area	Use this communication method when performing data send/receive with adjusting the timing between CPU modules.	The sending side CPU module writes data in the fixed cycle communication area (send area) in the host CPU. The receiving side CPU module reads data from the fixed cycle communication area (receive area) in the CPU module of the send source CPU module.

Used memory

CPU buffer memory is used for data communication among CPU modules.

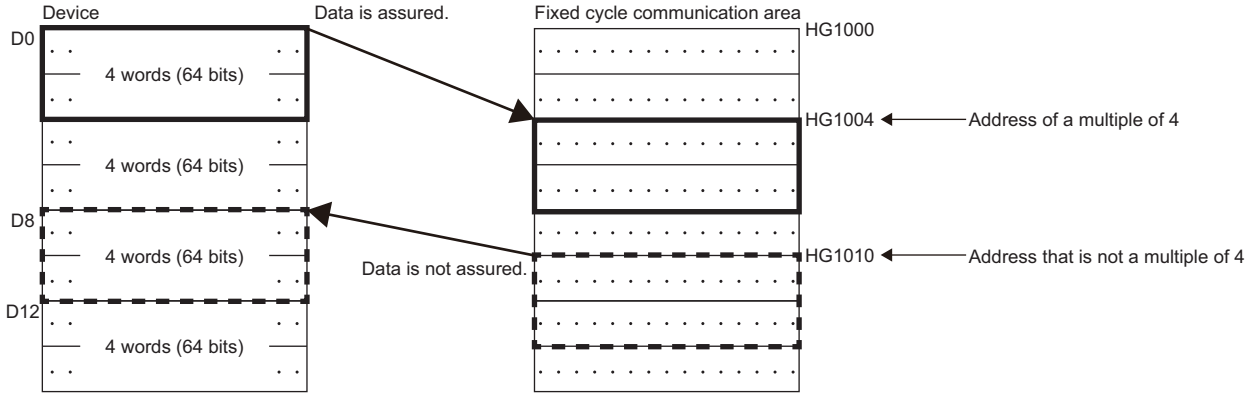
Memory configuration of CPU buffer memory



Area	Communication method	Description	Area size
CPU buffer memory	Communication through direct access	Reads/writes data in the area of the host CPU or other CPUs.	512K words fixed
Fixed cycle communication area	Communication through direct access	Performs data communication between the host CPU area and other CPU areas in accordance with the set interval.	Can be set within the range of 0 to 24K words in total. The send area per single CPU module can be set within the range of 0 to 12K words. (☞ Page 237 Setting fixed cycle communication area)

■Prevention of 64-bit data inconsistency

To avoid 64-bit data inconsistency, access the specified start address of the CPU buffer memory in multiples of four similarly to the device to be specified.

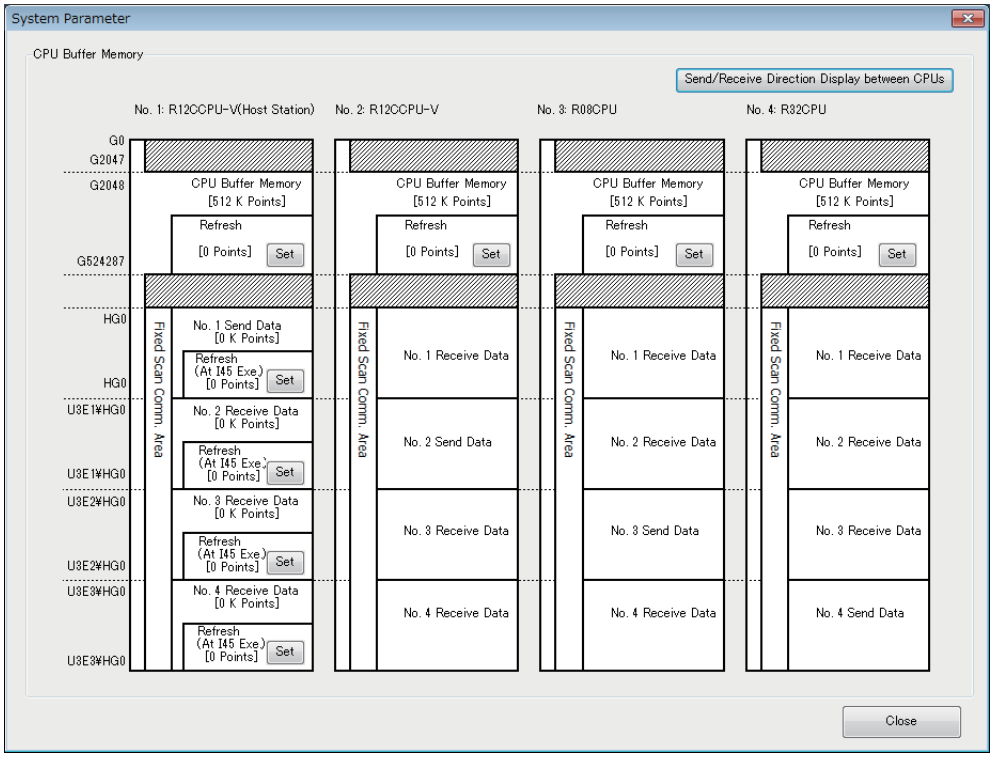


Checking memory configuration

Memory configuration can be checked with CW Configurator.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Communication Setting between CPU] ⇨ [CPU Buffer Memory Setting] ⇨ [<Detailed Setting>]

Window



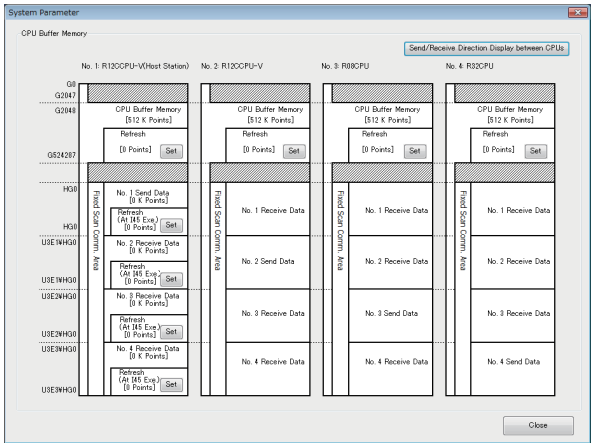
Displayed items

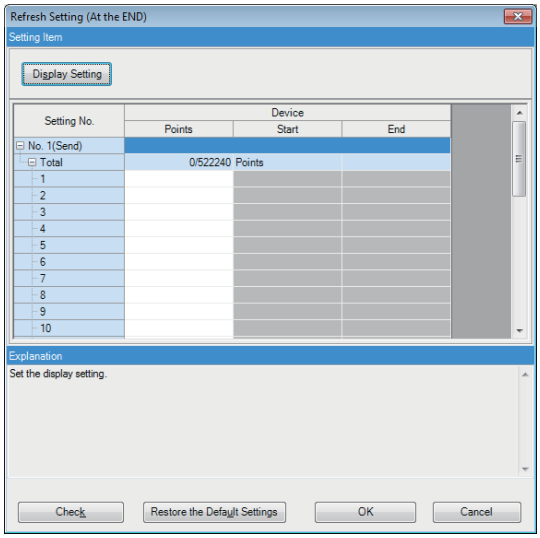
Item	Description	Setting range	Default
[Set] button in each refresh area	Click this to configure the refresh settings used for data communication between CPU modules.	—	0 points
[Send/Receive Direction Display between CPUs] button	Click this to display the arrow that indicates the send/receive direction.	—	—

■Setting refresh area

For data communication with a multiple CPU synchronous interrupt program (I45) using the fixed cycle communication area, setting the refresh area is required.

Set the refresh area in accordance with another CPU module with which data to be communicated.





1. Click the [Set] button in the refresh area to be set.
2. Set the number of points in accordance with the communication destination CPU module.
3. Click the [OK] button once the setting is completed.
4. Set the number of points for all of the refresh area used for data communication.

Another CPU module (communication destination)	Description
C Controller module	Set the number of points for refresh area as shown below: • Refresh area (at the END): 0 points • Refresh area (at the execution of I45): 0 points
Programmable controller CPU	Set the number of points for refresh area as shown below: • Refresh area (at the END): Same number of points with the communication destination programmable controller • Refresh area (at the execution of I45): Same number of points with the communication destination programmable controller.

Settings of data communication using fixed cycle communication area

The following shows the settings for performing data communication using the fixed cycle communication area.

■Setting method

To communicate data using the fixed cycle communication area, select "Enable" in "Fixed Scan Communication Function."

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Communication Setting between CPU] ⇨ [Fixed Scan Communication Function]

Window



Displayed items

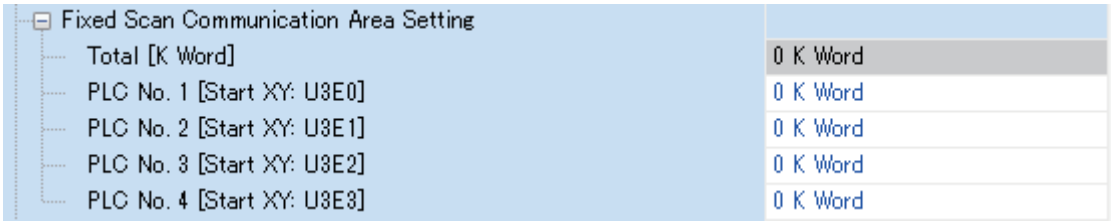
Item	Description	Setting range	Default
Fixed Scan Communication Function	Set whether to use the fixed scan communication function. Set the same settings only for the CPUs which are used by the fixed scan communication function.	• Not Use • Use	Not Use

■Setting fixed cycle communication area

Set the send area range (total of areas used for direct access communication) for each CPU in the fixed cycle communication area. The range of the fixed cycle communication area can only be changed with the parameter settings. The other areas cannot be changed.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Communication Setting between CPU] ⇨ [Fixed Scan Communication Area Setting]

Window



Displayed items

Item	Description	Setting range	Default
Total [K Word]	The total value is displayed.	Entire system: 0 to 24K words	—
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

Precautions

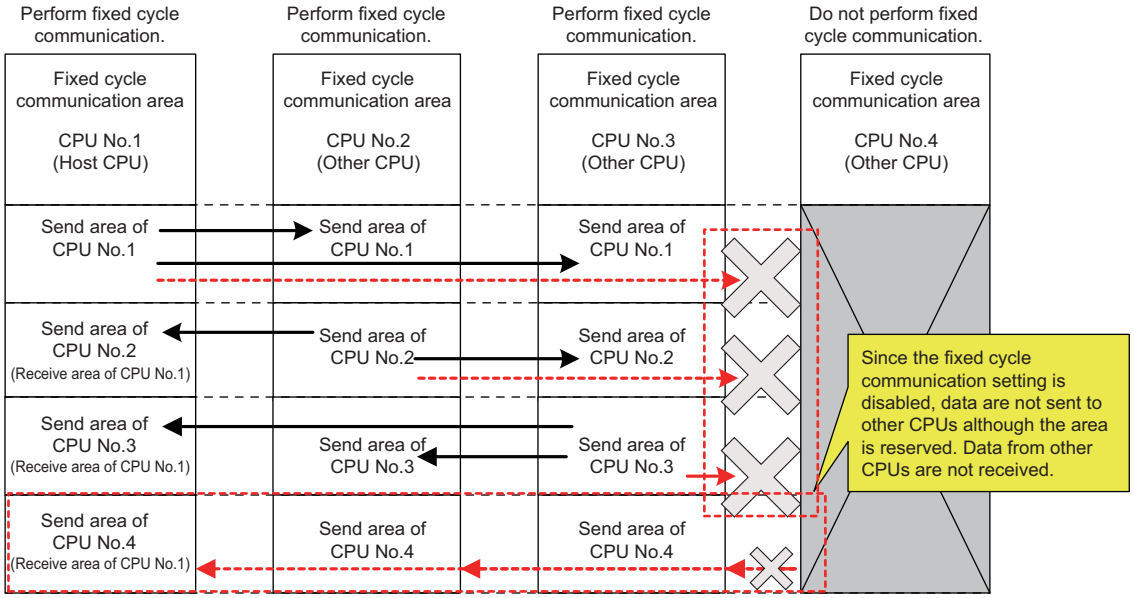
The maximum area size which can be set in the fixed cycle communication area varies depending on the fixed scan interval.

Fixed cycle interval	Maximum area size which can be set to fixed cycle communication area
0.10 ms	12K words
0.15 ms	20K words
Other than above	24K words

Point

When CPU of which fixed scan communication function is set to "Not Use" exists, setting the send area for the fixed cycle communication area to a CPU with the fixed scan communication function set to "Not Use" (that is, unset) with the host CPU parameter setting will result in no error since the unset CPU is regarded as a reserved CPU for future setting.

Example: When "Not Use" is set to "Fixed Scan Communication Function" for the CPU No.4

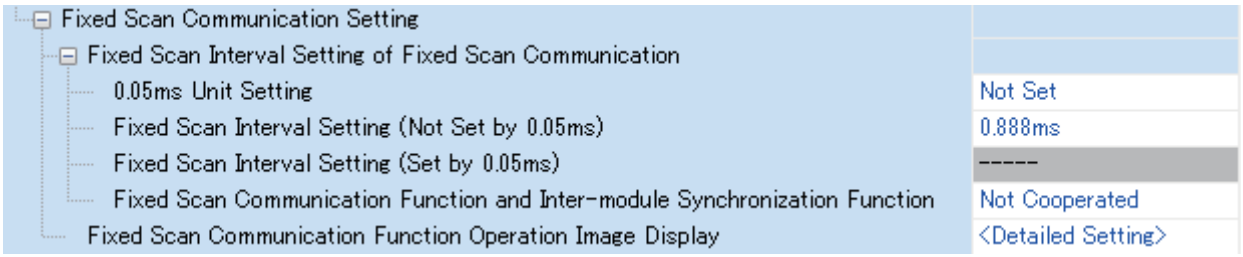


Fixed scan communication setting

Set the interval for data transfer between CPU modules. It also can be synchronized with the timing of the 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	Set whether to set the fixed scan interval setting in 0.05 ms unit.	• Not Set • Set	Not Set
Fixed Scan Interval Setting (Not Set by 0.05ms)*1	Set the cycle of fixed cycle communication interval by selecting from the items of the setting range. Set the same settings only for the CPUs which are used by the fixed cycle communication function.	• 0.222 ms • 0.444 ms • 0.888 ms • 1.777 ms • 3.555 ms • 7.111 ms	0.888 ms
Fixed Scan Interval Setting (Set by 0.05ms)*1	Set any value as a fixed cycle communication interval in 0.05 ms unit.	0.10 ms to 10.00 ms	0.10 ms
Fixed Scan Communication Function and Inter-module Synchronization Function	Set whether to operate the fixed cycle communication between CPUs matched with the cycle set in "Inter-module Synchronization Setting."	• Not Cooperated • Cooperate	Not Cooperated

*1 The maximum area size which can be set in the fixed cycle communication area varies depending on the fixed scan interval. (☞ Page 237 Setting fixed cycle communication area)

Point

- The send image for the fixed cycle communication can be checked by selecting "Fixed Scan Communication Function Operation Image Display."
- When "Fixed Scan Communication Function and Inter-module Synchronization Function" is set to "Not Cooperated," the timing will not be match even if the same value is set for the interval of both the fixed scan communication interval and the inter-module synchronization setting.

Error detection setting

During data transfer among the CPU modules in a multiple CPU system, data inconsistency may occur because of time required for writing all data within the fixed cycle. To detect or not to detect a continuation error can be set with this setting. (☞ Page 111 Error detection setting)

CPU number-based data assurance

Data communication is performed in 64-bit units between CPU modules. Therefore, when data larger than 64 bits are handled, data inconsistency in which old and new data overlap may occur for each CPU, and this depends on the timing between data reading by the host CPU and data writing by another CPU/data receiving from other CPU.

Prevention of data inconsistency using the CPU number-based data assurance

The table below shows whether or not to prevent data inconsistency by enabling/disabling the CPU number-based data assurance.

○: With data inconsistency control by system, ×: Without data inconsistency control by system[*1]

*1 The countermeasures by a program are required.

Communication method	CPU buffer memory		Fixed cycle communication area	
	CPU number-based data assurance enabled	CPU number-based data assurance disabled	CPU number-based data assurance enabled	CPU number-based data assurance disabled
Communication by refresh[*2]	×	×	×	×
Communication through direct access	×	×	○[*3]	×

*2 Communication by refresh can not be performed since the CPU buffer memory does not have the refresh area.

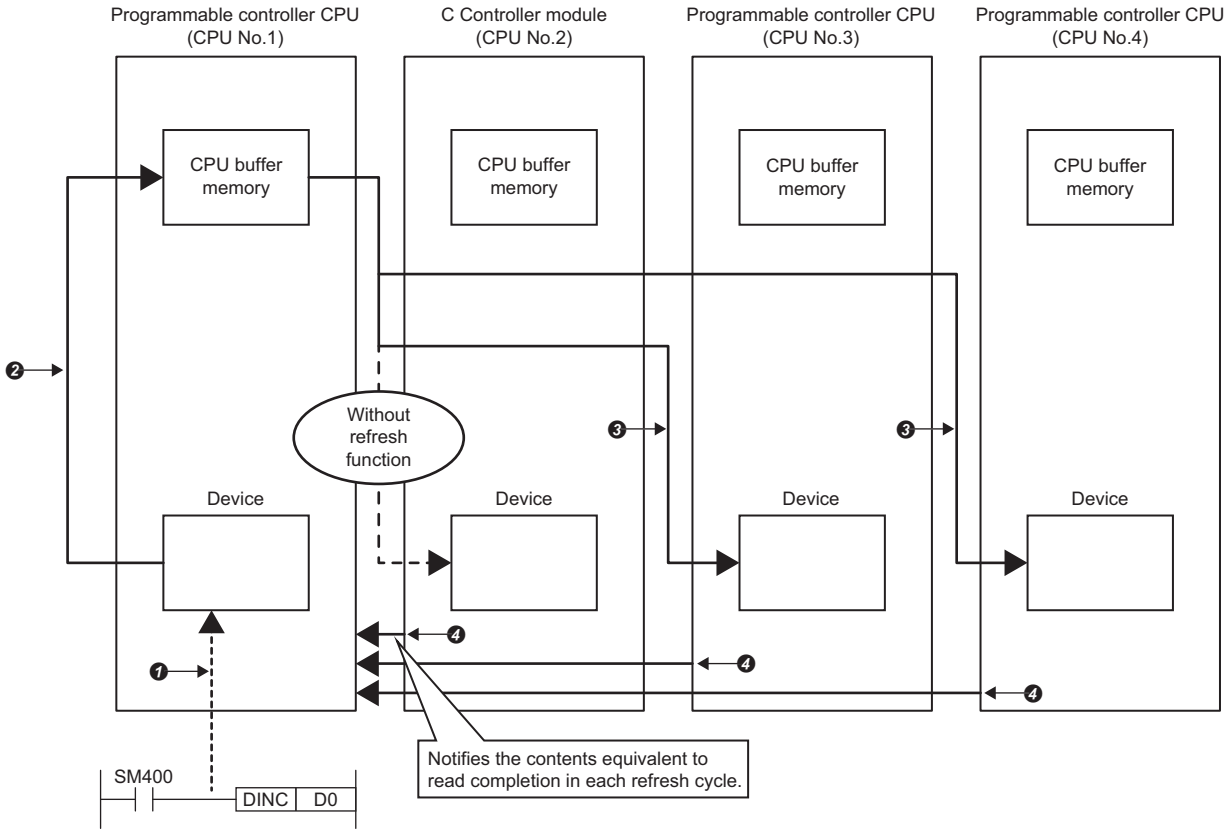
*3 The access in the multiple CPU synchronous interrupt program (I45) only.

Point

- CPU No.1 does not send/receive the following data until it receives the notification of the data read completion from another CPU. The timing of sending/receiving data is the update interval of the CPU module of which scan time of the programmable controller CPU or refresh cycle of the C Controller module is the latest.
- C Controller module notifies the contents equivalent to the read completion in each refresh cycle.
- In the data communication through direct access to the CPU buffer memory excluding the fixed cycle communication area, the data in the CPU buffer memory of another CPU is directly read after the execution of the read instruction. Therefore, it will not be subject to the CPU number-based data assurance.
- When the data in the CPU buffer memory of a C Controller module is read by a programmable controller CPU using the refresh, it will not be subject to the CPU number-based data assurance.

■Communication by refresh (when C Controller module is on the receiving side)

When the CPU number-based data assurance is enabled, the data is not assured.



❶The data is written from the program.

❷At END processing of the CPU No.1, the data is written.*1

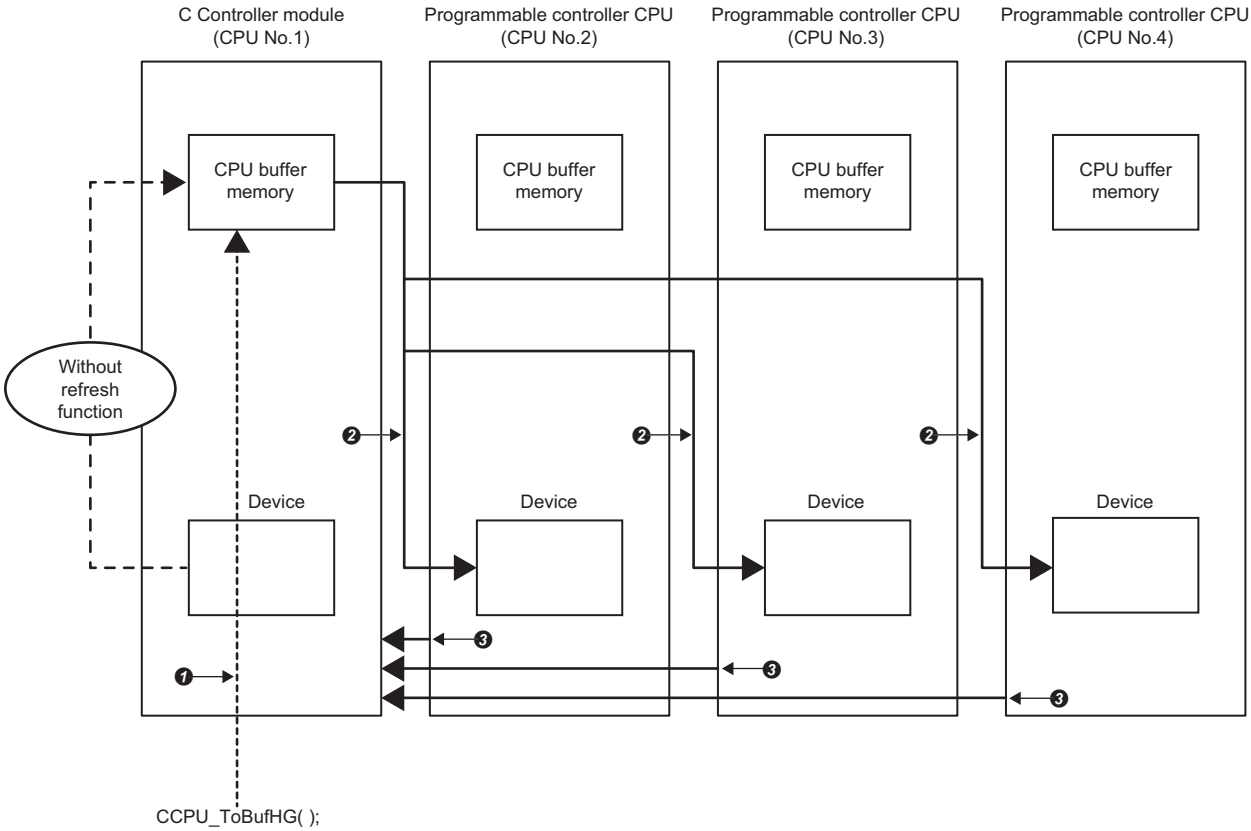
❸At END processing of each CPU, the data is read.

❹The data read completion from each CPU is notified to the CPU No.1.

*1 The update to the following data is not performed until the notification of the data read completion is received from other CPUs (No.2 to No.4).

■Communication by refresh (when C Controller module is on the sending side)

When the CPU number-based data assurance is enabled, the data is not assured.



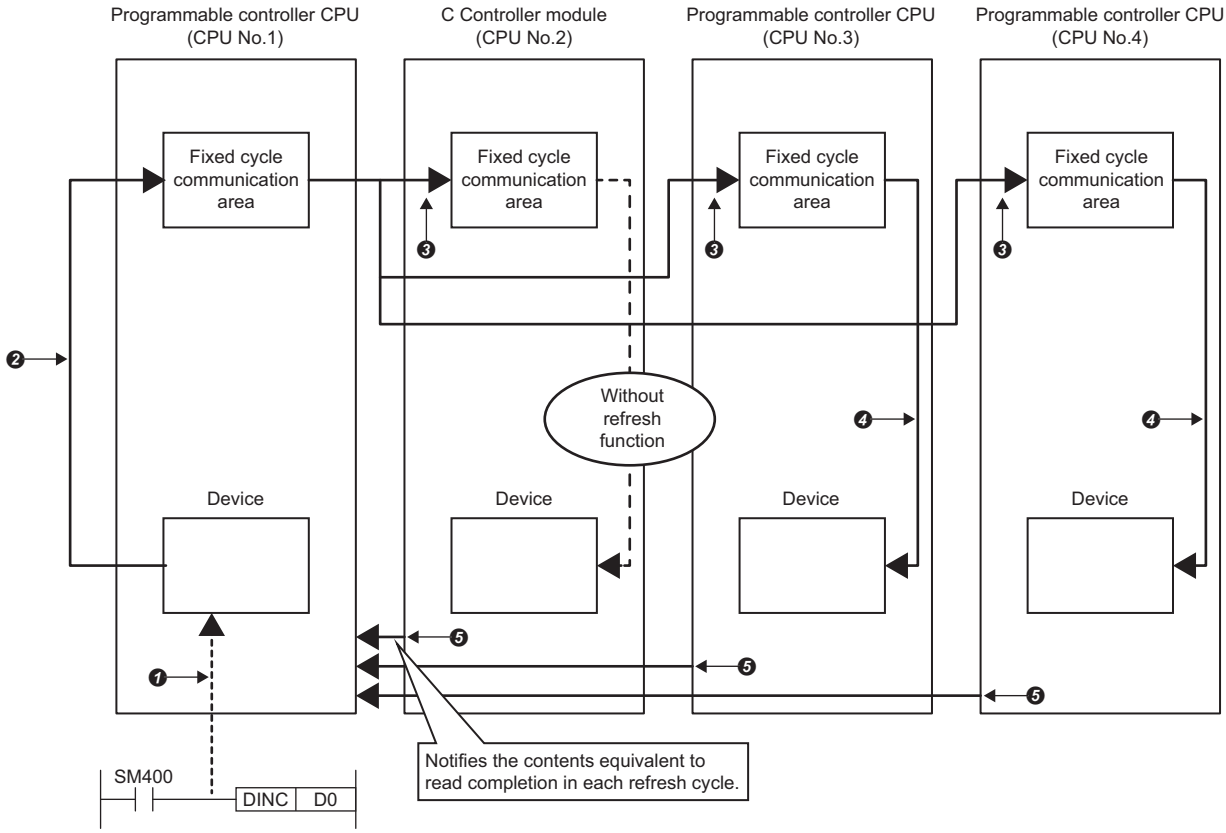
❶The data is written from the program.

❷At END processing of each CPU, the data is read.

❸The data read completion from each CPU is notified to the CPU No.1.

■Communication through direct access (when C Controller module is on the receiving side)

When the CPU number-based data assurance is enabled, the data is assured.



❶The data is written from the program.

❷At the multiple CPU synchronous interrupt program (I45) execution, the data is written.*1

❸At the multiple CPU synchronous interrupt program (I45) execution, the data is read.

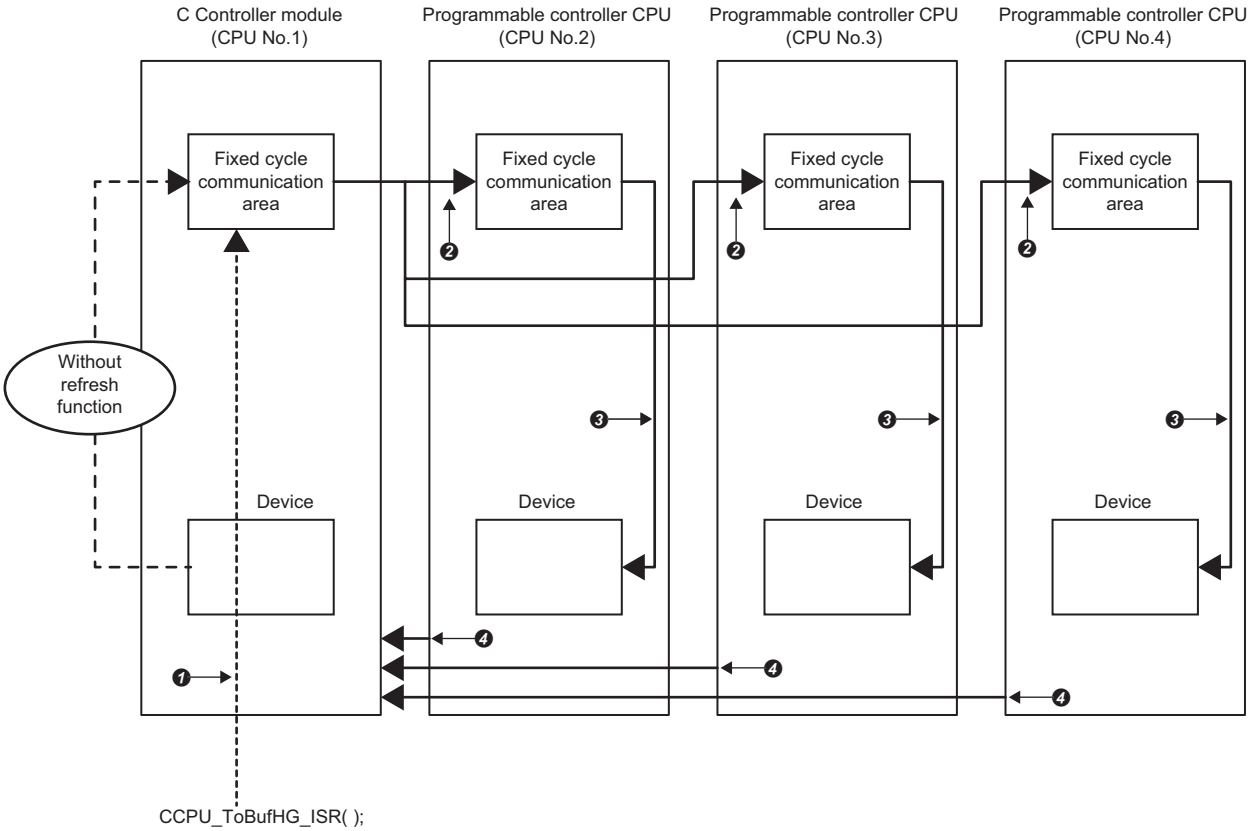
❹At the multiple CPU synchronous interrupt program (I45) execution, the data is refreshed.

❺The data read completion from each CPU is notified to the CPU No.1.

*1 The update to the following data is not performed until the notification of the data read completion is received from other CPUs (No.2 to No.4).

■Communication through direct access (when C Controller module is on the sending side)

When the CPU number-based data assurance is enabled, the data is not assured.



❶The data is written from the program.

❷At the multiple CPU synchronous interrupt program (I45) execution, the data is read.

❸At the multiple CPU synchronous interrupt program (I45) execution, the data is refreshed.

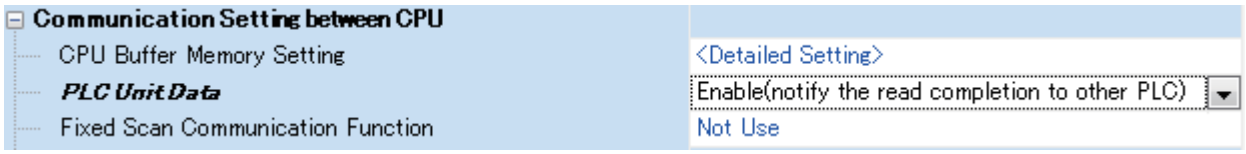
❹The data read completion from each CPU is notified to the CPU No.1.

CPU number-bases data assurance setting

Configure the CPU number-based data assurance setting.

[System Parameter] ⇨ [Multiple CPU Setting] ⇨ [Communication Setting between CPU] ⇨ [PLC Unit Data]

Window



Displayed items

Item	Description	Setting range	Default
PLC Unit Data	Set the data 'assure' (Notification of reading completed will be sent) or 'not assured' (Notification of reading completed will not be sent) in PLC module. Transmit read timing of PLC module to ensure operation of other module. C Controller module's refresh function will not be able to ensure operation of other module.When a configuration is multi-CPU with PLC, Read timing of other module is transmitted in case PLC does not transmit read timing and PLC refresh operation is stopped. Timing of C Controller module is sent to other module to match the timing of periodic execution(Refresh cycle).The C Controller module can not read and write, during the execution of matching periodic time(Refresh cycle).	• Disable (not notify the read completion to other CPUs) • Enable (notify the read completion to other CPUs)	Disable (not notify the read completion to other CPUs)

Data assurance by program

Prevent data inconsistency using a program when the data inconsistency control by system is not available.

■Accessing the CPU buffer memory

A program reads data in order from the start address of the CPU buffer memory excluding the refresh area, and writes send data in order from the end address to the start address excluding the refresh area by the write instruction. Therefore, data inconsistency can be prevented by setting a device for interlock at the head of data to be communicated.

■Accessing the fixed cycle communication area

When accessing within a multiple CPU synchronous interrupt program (I45), no interlock circuit is required with the CPU number-based data assurance setting is enabled. To access the fixed cycle communication area in a program other than the above, or when the CPU number-based data assurance setting is disabled, an interlock circuit is required as with the access to the CPU buffer memory.

Functions that can be used for communication

The following shows the functions which are used for communication using each memory area in a multiple CPU system.

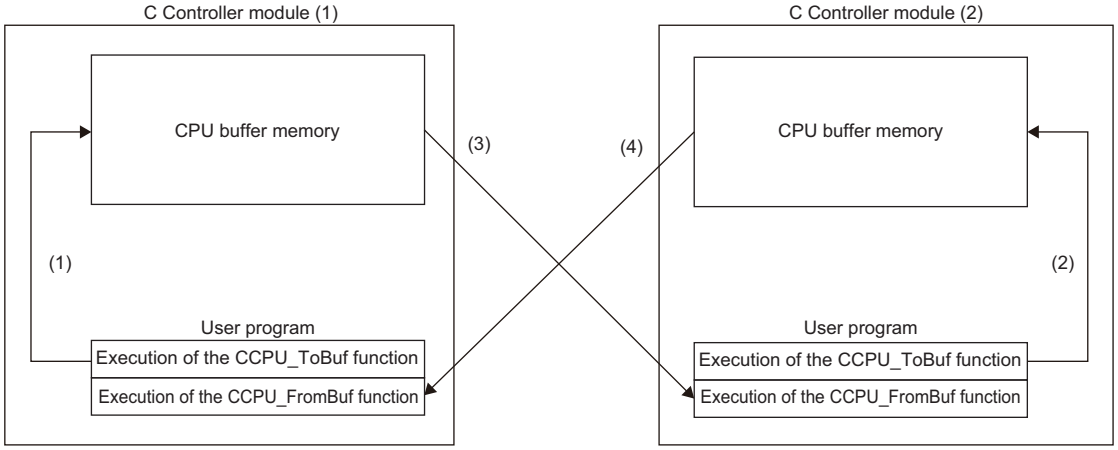
C Controller module dedicated function	Description
CCPU_FromBuf	To read data from the CPU buffer memory and intelligent function module buffer memory in the module on the specified module position.
CCPU_FromBuf_ISR	
CCPU_FromBufHG	To read data from the fixed cycle communication area of the module on the specified module position.
CCPU_FromBufHG_ISR	
CCPU_ToBuf	To write data to the CPU buffer memory and intelligent function module buffer memory in the module on the specified module position.
CCPU_ToBuf_ISR	
CCPU_ToBufHG	To write data to the fixed cycle communication area of the module on the specified module position.
CCPU_ToBufHG_ISR	

Communication using CPU buffer memory

Device data of the host CPU is written to the CPU buffer memory in the host CPU by execution of the C Controller module dedicated function (CCPU_ToBuf). The data written to the CPU buffer memory is transferred after the following processing is performed in another CPU.

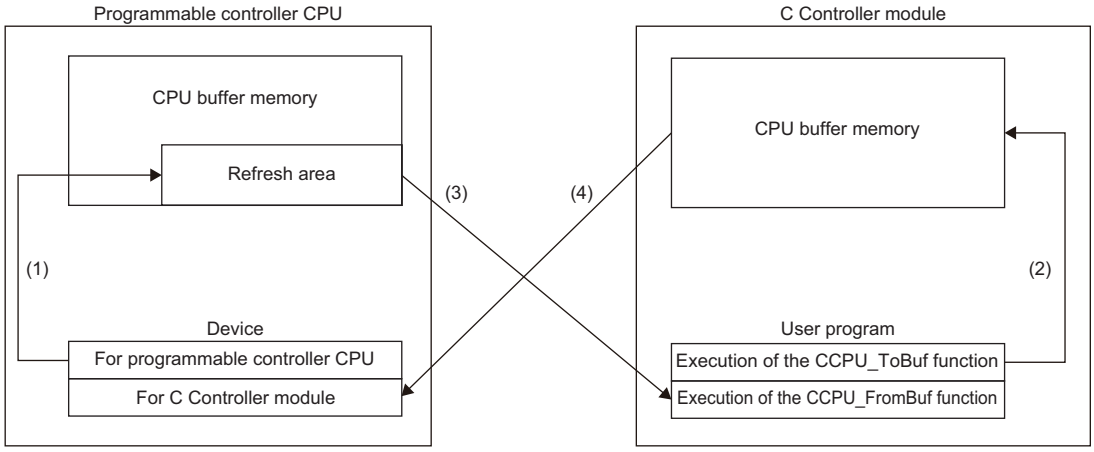
CPU module	Processing
C Controller module	At execution of the C Controller module dedicated function (CCPU_FromBuf)
Programmable controller CPU	At END processing

- C Controller module (1) ⇔ C Controller module (2)



Processing order	Processing at execution of the function in a C Controller module (1)	Processing at execution of the function in a C Controller module (2)
(1)	Device data of the user program is written to the CPU buffer memory with the C Controller module dedicated function (CCPU_ToBuf).	—
(2)	—	Device data of the user program is written to the CPU buffer memory with the C Controller module dedicated function (CCPU_ToBuf).
(3)	—	Device data of the CPU buffer memory in the C Controller module (1) is read to the user program with the C Controller module dedicated function (CCPU_FromBuf).
(4)	Device data of the CPU buffer memory in the C Controller module (2) is read to the user program with the C Controller module dedicated function (CCPU_FromBuf).	—

• Programmable controller CPU ⇔ C Controller module



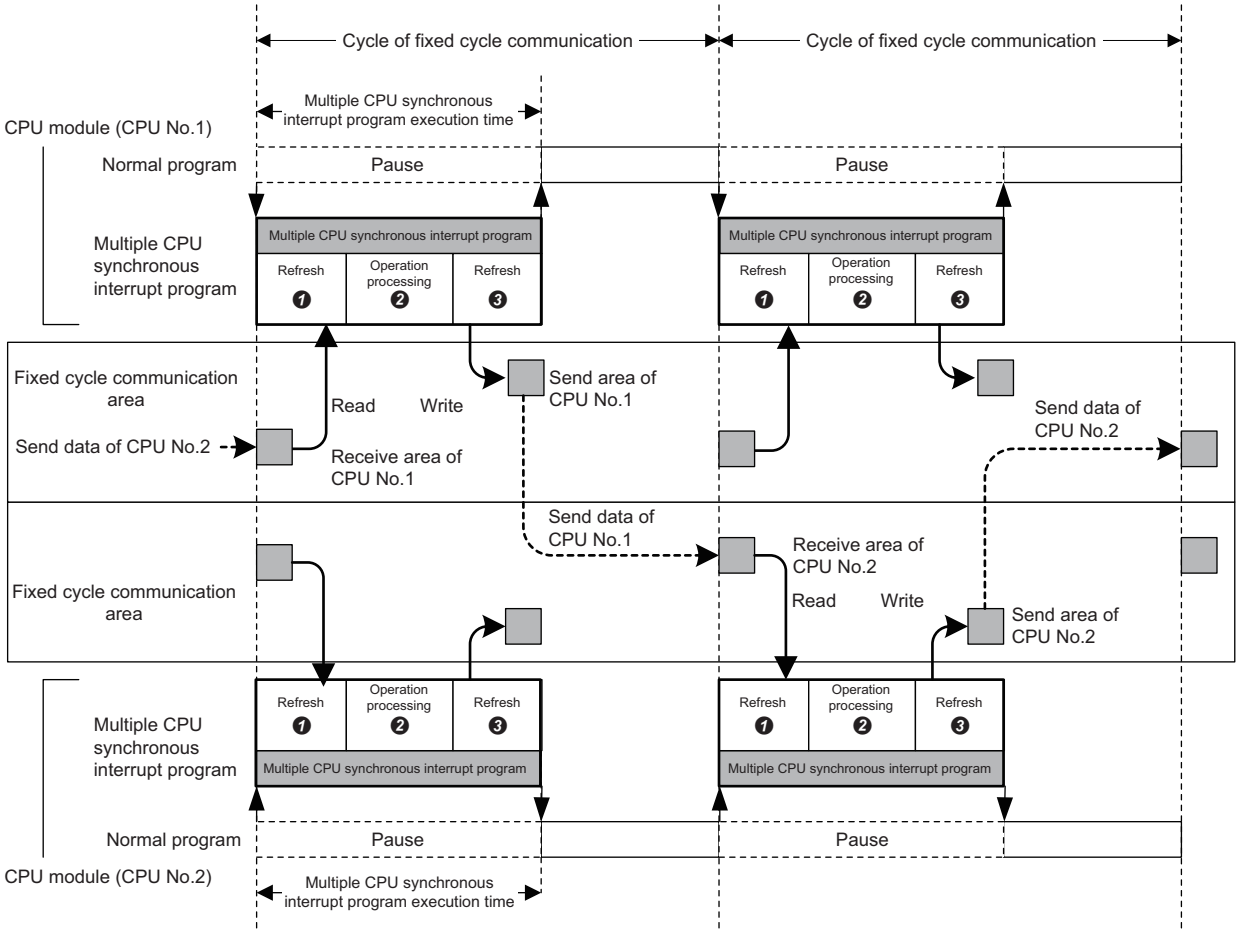
Processing order	Process at END processing in programmable controller CPU	Processing at execution of the function in a C Controller module
(1)	Device data for programmable controller CPU are transferred to the refresh area.	—
(2)	—	Device data of the user program is written to the CPU buffer memory with the C Controller module dedicated function (CCPU_ToBuf).
(3)	—	Device data in the refresh area of the programmable controller CPU is read to a user program with the C Controller module dedicated function (CCPU_FromBuf).
(4)	Device data of the CPU buffer memory in the C Controller module is transferred to a device for C Controller modules in the programmable controller CPU.	—

Communication using fixed cycle communication area

Data communication using the fixed cycle communication area is performed by using a multiple CPU synchronous interrupt program (I45).

Multiple CPU synchronous interrupt

Execute a multiple CPU synchronous interrupt program (I45) at the timing of the cycle of the fixed cycle communication set to the parameter. By using the multiple CPU synchronous interrupt function, data can be communicated between CPU modules synchronously with the cycle of the fixed cycle communication. (The data communication timing between CPU modules is synchronized.)



No.	Processing	Description
❶	Receiving data from another CPU (refresh)*1	The data received from another CPU module is read to a device. (The data in the host CPU receive area is read.)
❷	Operation processing	Multiple CPU synchronous interrupt program is executed.
❸	Sending data to another CPU (refresh)*1	The data to be sent to another CPU module is written from a device. (The data is written to the host CPU send area.)

*1 Performed with the C Controller module dedicated functions (CCPU_ToBufHG_ISR, CCPU_FromBufHG_ISR).

Point

Data communication will not be performed if the multiple CPU synchronous interrupt program (I45) is not registered.

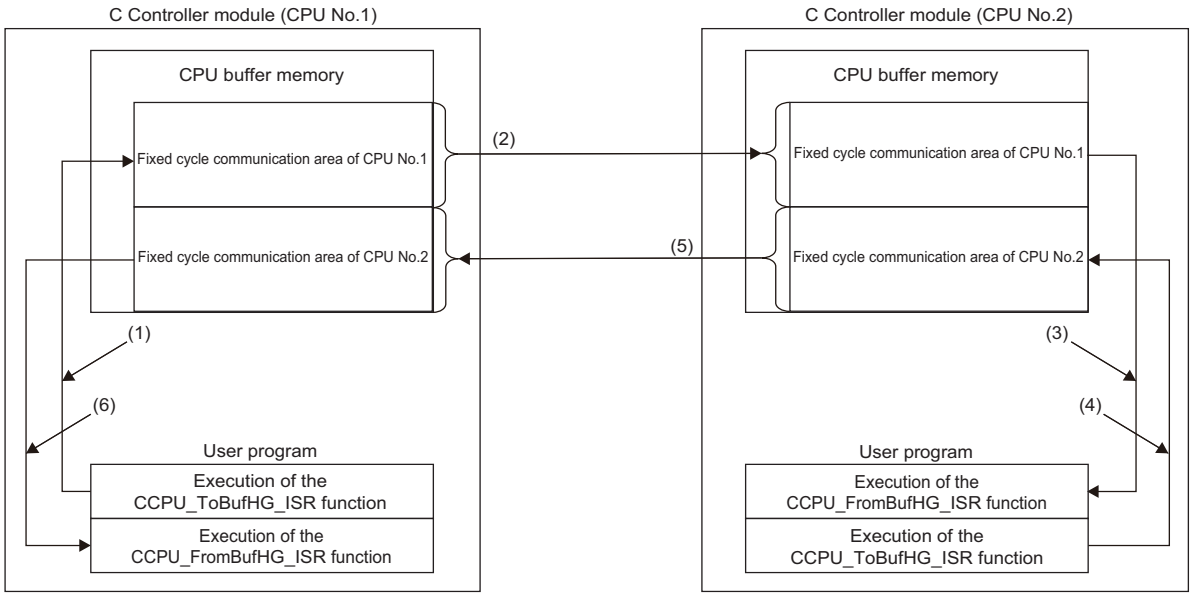
■Processing order for fixed cycle communication area

Device data of the host CPU is written to the fixed cycle communication area in the host CPU by execution of the C Controller module dedicated function (CCPU_ToBufHG_ISR). The data written to the fixed cycle communication area are read once sent to another CPU.

Point

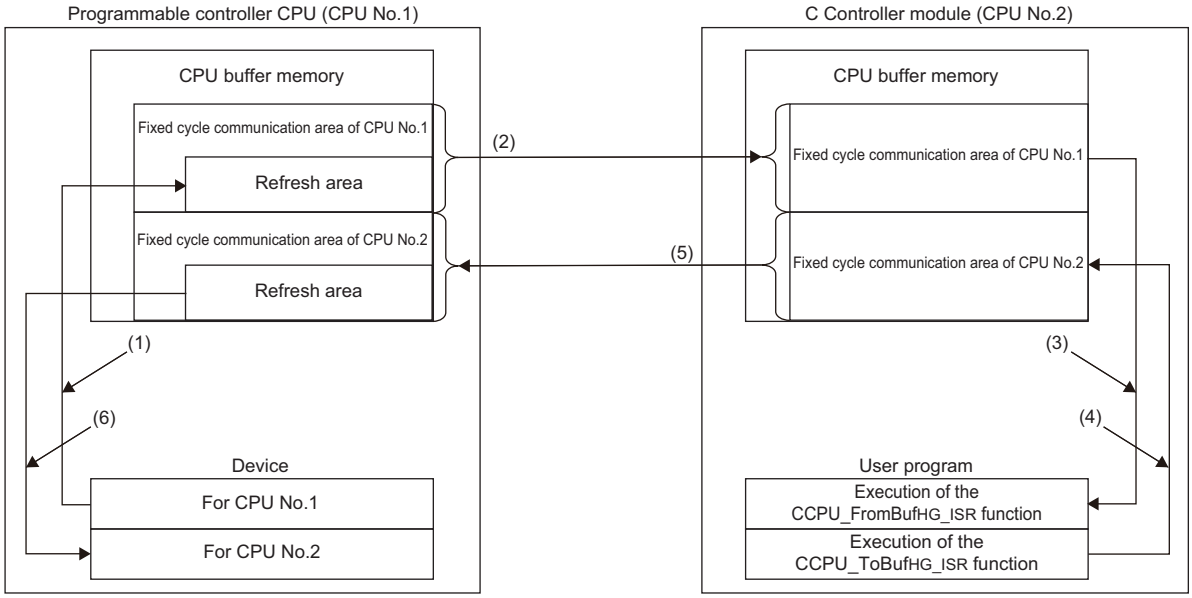
To write/read data to/from the fixed cycle communication area on a task, use the C Controller module dedicated functions (CCPU_ToBufHG, CCPU_FromBufHG).

- C Controller module (CPU No.1) ⇔ C Controller module (CPU No.2)



CPU number	Processing order	Process on C Controller module (CPU No.1)	Process on C Controller module (CPU No.2)
CPU No.1	(1)	Device data of the user program is written to the fixed cycle communication area of CPU No.1 with the C Controller module dedicated function (CCPU_ToBufHG_ISR).	—
	(2)	Device data in the fixed cycle communication area of CPU No.1 is sent to CPU No.2.	—
CPU No.2	(3)	—	Device data in the fixed cycle communication area of CPU No.1 is read to a user program with the C Controller module dedicated function (CCPU_FromBufHG_ISR).
	(4)	—	Device data of the user program is written to the fixed cycle communication area of CPU No.2 with the C Controller module dedicated function (CCPU_ToBufHG_ISR).
	(5)	—	Device data in the fixed cycle communication area of CPU No.2 is sent to CPU No.1.
CPU No.1	(6)	Device data in the fixed cycle communication area of CPU No.2 is read to a user program with the C Controller module dedicated function (CCPU_FromBufHG_ISR).	—

• Programmable controller CPU (CPU No.1) ⇔ C Controller module (CPU No.2)



CPU number	Processing order	Process on programmable controller CPU (CPU No.1)	Process on C Controller module (CPU No.2)
CPU No.1	(1)	The device data for CPU No.1 is transferred to the refresh area in CPU No.1.	—
	(2)	The device data in the refresh area of CPU No.1 is sent to CPU No.2.	—
CPU No.2	(3)	—	Device data in the refresh area of the CPU No.1 is read to a user program with the C Controller module dedicated function (CCPU_FromBufHG_ISR).
	(4)	—	Device data for CPU No.2 is written to the fixed cycle communication area of CPU No.2 with the C Controller module dedicated function (CCPU_ToBufHG_ISR).
	(5)	—	Device data in the fixed cycle communication area of CPU No.2 is sent to CPU No.1.
CPU No.1	(6)	The device data stored in the refresh area of CPU No.2 is transferred to the devices for CPU No.2.	—

■Refresh behavior of programmable controller CPU

When a programmable controller CPU and a multiple CPU system are configured, register a multiple CPU synchronous interrupt program (I45) on the interrupt routine, and enable the registered routine.

When a multiple CPU synchronous interrupt program (I45) is not registered on the interrupt routine or the registered routine is disabled on the C Controller module side, the refresh function of the programmable controller CPU does not perform.

• CPU number-based data assurance enabled

○: Data is updated, ×: Data is not updated

Processing order	Multiple CPU synchronous program (I45) is not registered on the interrupt routine.	Multiple CPU synchronous program (I45) is registered on the interrupt routine, but the registered routine is disabled.	Multiple CPU synchronous program (I45) is registered on the interrupt routine, and the registered routine is enabled.
(1)	×	×	○
(2)	○	○	○
(3)	○	○	○
(4)	○	○	○
(5)	○	○	○
(6)	×	×	○

• CPU number-based data assurance disabled

○: Data is updated, ×: Data is not updated

Processing order	Multiple CPU synchronous program (I45) is not registered on the interrupt routine.	Multiple CPU synchronous program (I45) is registered on the interrupt routine, but the registered routine is disabled.	Multiple CPU synchronous program (I45) is registered on the interrupt routine, and the registered routine is enabled.
(1)	○	○	○
(2)	○	○	○
(3)	○	○	○
(4)	○	○	○
(5)	○	○	○
(6)	×	×	○

Usage

1. Register a multiple CPU synchronous interrupt program (I45) on the interrupt routine. (☞ Page 88 Interrupt Function to a C Controller Module)
2. Enable the registered interrupt routine. (☞ Page 88 Interrupt Function to a C Controller Module)
3. Configure the refresh settings. (☞ Page 236 Setting refresh area)

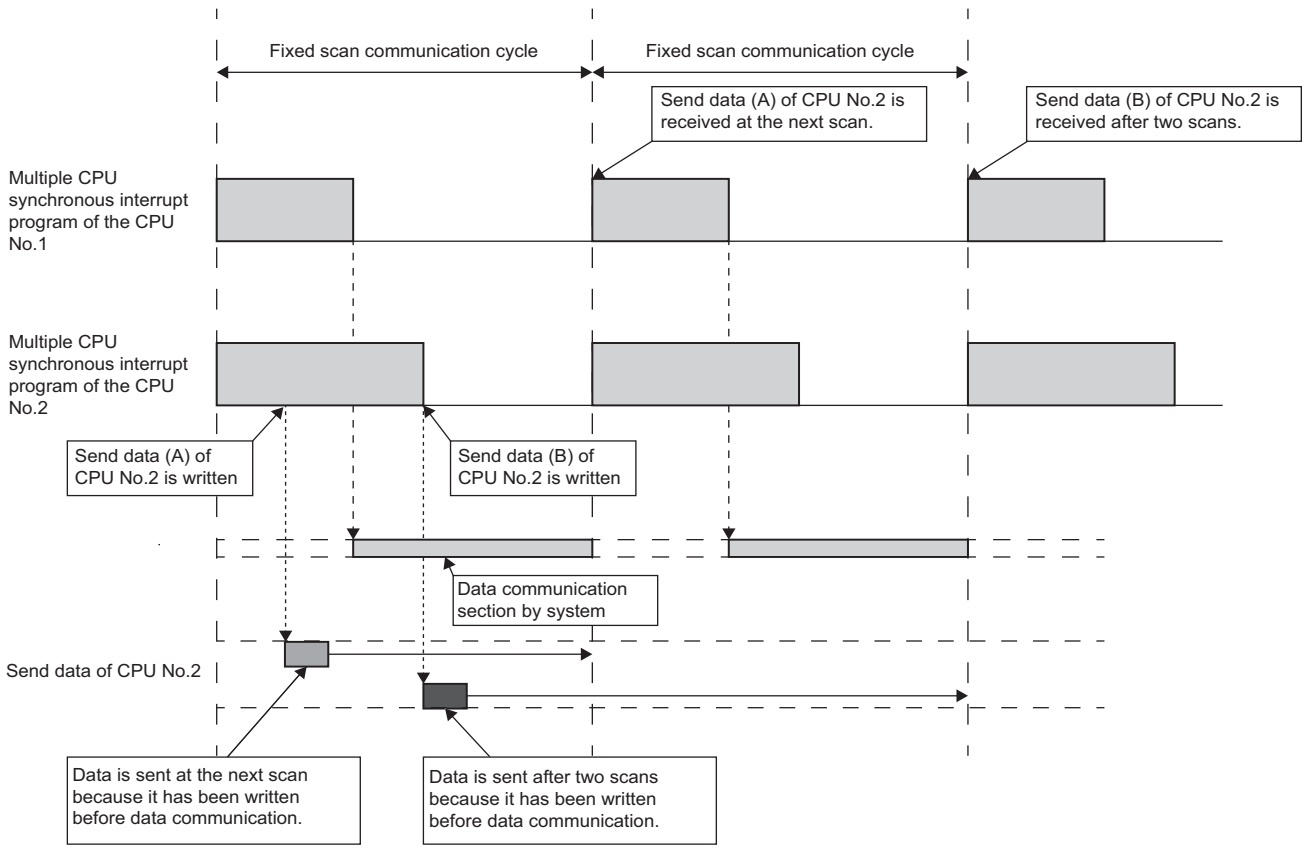
Interrupt timing

The multiple CPU synchronous interrupt program (I45) is executed at the timing for the cycle of the fixed cycle communication. The cycle of the fixed cycle communication can be changed in "Fixed Scan Communication Setting." (☞ Page 239 Fixed scan communication setting)

Considerations

The following shows the considerations for the multiple CPU synchronous interrupt program.

- Do not create a multiple CPU synchronous interrupt program of which execution processing time is longer than the cycle of the fixed cycle communication. If so, the multiple CPU synchronous interrupt interval cannot be guaranteed. For a multiple CPU synchronous interrupt program (I45), the settings for the execution time monitoring and the operation at the time of excess can be configured by "RAS Setting" of CPU parameter. (☞ Page 111 Error detection setting, Page 112 Operation setting when an error is detected)
- To ensure the data transmission at the following cycle of the fixed cycle communication, set "Program Execution Section Exceed (I45)" to "Detect" in the RAS setting of the CPU parameter. An error can be detected if data have been written at the timing when data cannot be sent in the cycle of the fixed cycle communication.



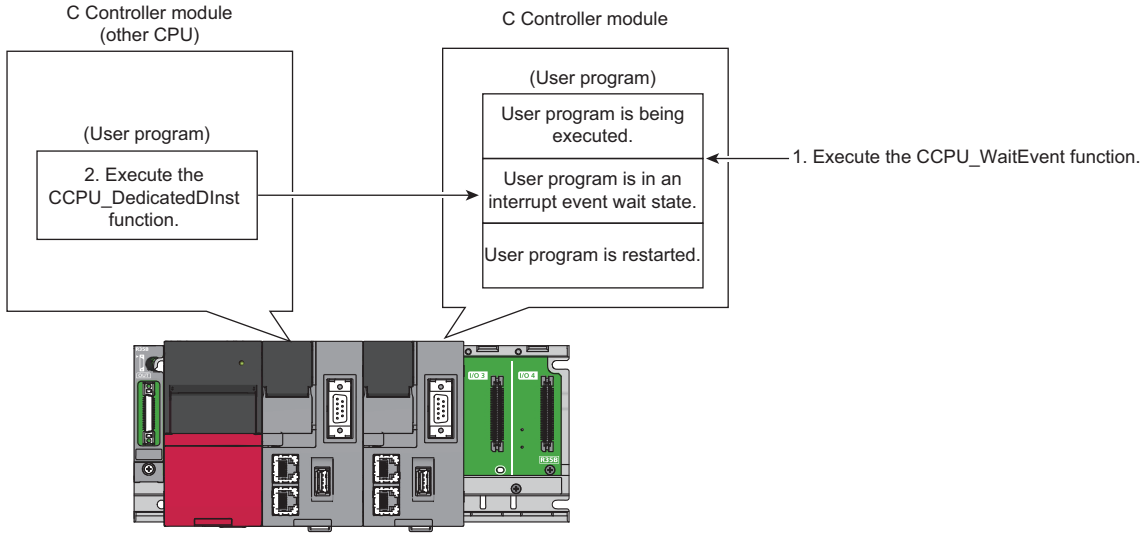
17.5 Interrupt from Another CPU

A user program waiting for an interrupt event can be restarted with an interrupt from another CPU module with the C Controller module dedicated function (CCPU_WaitEvent).

C Controller module dedicated function	Description
CCPU_WaitEvent	To wait for an interrupt event notification from another CPU.

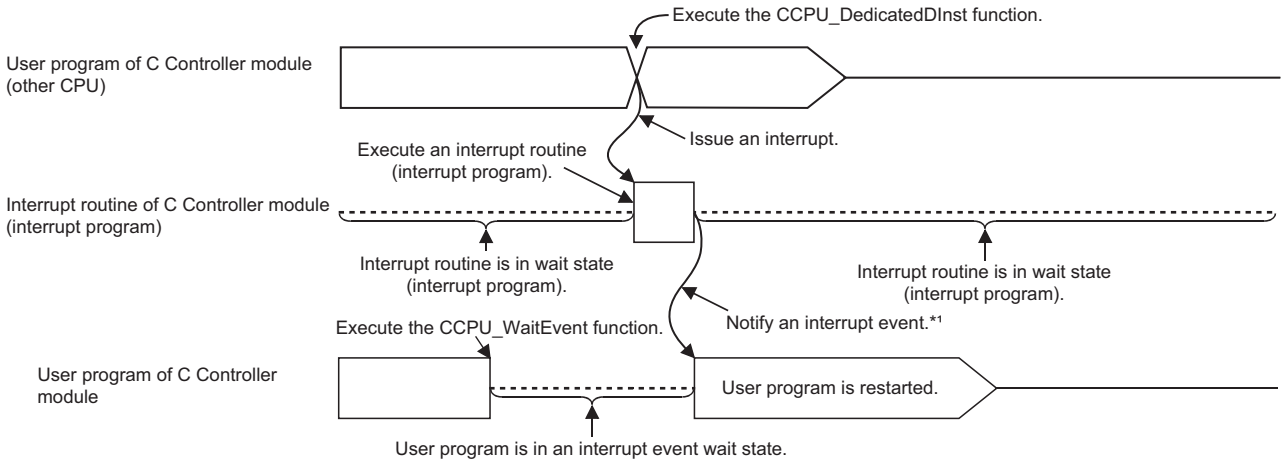
Interrupt from a C Controller module

The following figure shows the interrupt method from a C Controller module (another CPU).



1. With a user program, call the C Controller module dedicated function (CCPU_WaitEvent).
2. The user program is placed into the interrupt event wait state.
3. Execute the C Controller module dedicated function (CCPU_DedicatedDInst) with a user program of a C Controller module (another CPU).
4. The user program is restarted.

Operation timing



*1: The interrupt event is notified after the interrupt routine (interrupt program) is completed.

Function

The function used for an interrupt from a C Controller module (another CPU) is shown below.

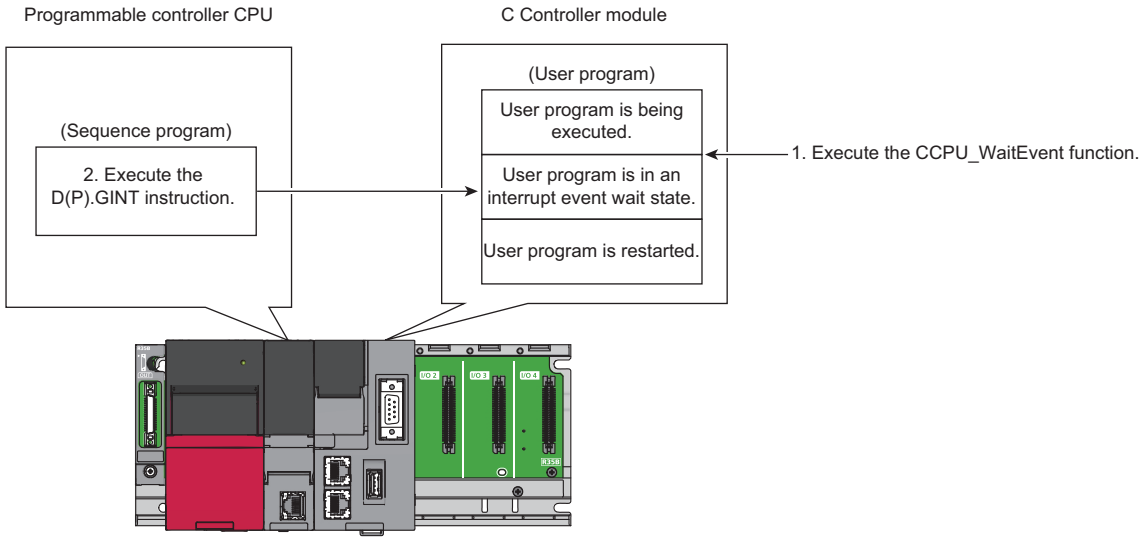
C Controller module dedicated function	Description
CCPU_DedicatedDInst	To execute dedicated instructions categorized as 'D' or 'DP.'

Point

The C Controller module dedicated function (CCPU_DedicatedDInst) can also execute an interrupt to a motion CPU.

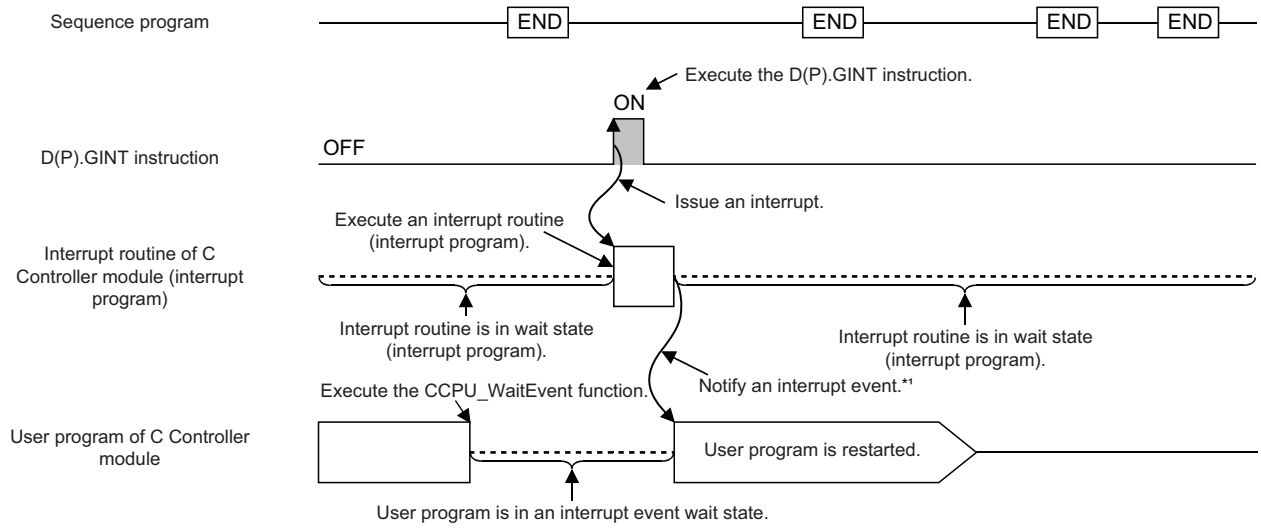
Interrupt from a programmable controller CPU

The following figure shows the interrupt method from a programmable controller CPU.



1. With a user program, call the C Controller module dedicated function (CCPU_WaitEvent).
2. The user program is placed into the interrupt event wait state.
3. Execute the D(P).GINT instruction with a sequence program of a programmable controller CPU (another CPU).
4. The user program is restarted.

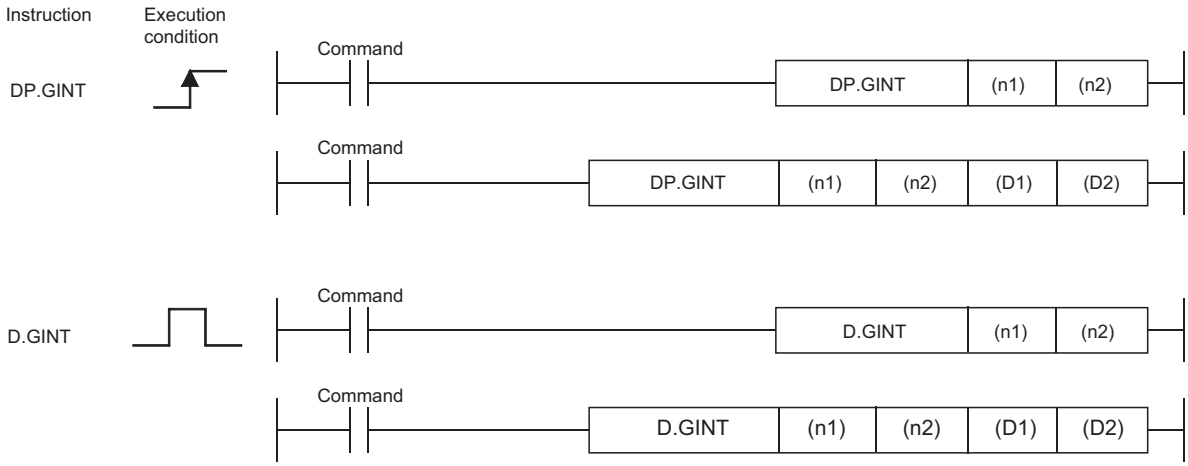
Operation timing



*1: The interrupt event is notified after the interrupt routine (interrupt program) is completed.

Dedicated instruction

The dedicated instructions used for an interrupt from a programmable controller CPU are shown below.



○: Applicable, △: Partly applicable

Setting data*1	Available devices										
	Internal device (System, user)		File register		Link direct device J□\□		Module access device J□\□		Index register Z□	Constant		Others
	Bit	Word	Bit	Word	Bit	Word	Bit	Word		Dec K, Hex H	Float, string	
(n1)	—	○	—	○	—		—		—	○	—	—
(n2)	—	○	—	○	—		—		—	○	—	—
(D1)*2	△*3	—	△*3	—	—		—		—	—		—
(D2)*2	—	△*3	—	△*3	—		—		—	—		—

*1 The setting data can be index-modified. (Constant is excluded.)
*2 Can be omitted only when both of (D1) and (D2) are omitted.
*3 Local devices cannot be used.

■Setting data

Setting data	Setting	Setting side	Data type
(n1)	Start input/output number of the target CPU divided by 16 The actual values to be specified are as follows: CPU No.1: 3E0H, CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H	User	BIN16 bits
(n2)	Interrupt pointer number (0 to 15)	User	BIN16 bits
(D1)*1	(D1+0): Device that is turned ON for one scan upon completion of the instruction processing (D1+1): Device that is turned ON for one scan upon abnormal completion of the instruction processing (For abnormal completion, D1+0 also turns ON.)	System	Bit
(D2)*1	Device where the completion status data is stored.	System	Word

*1 Can be omitted only when both of (D1) and (D2) are omitted.

■Control details

Make an interrupt occur to a C Controller module when the execution command of the D(P).GINT instruction rises (OFF→ON) using a sequence program.

When received an interrupt from a programmable controller CPU, C Controller module restarts a user program which is waiting for an interrupt event with the CCPU_WaitEvent function.

■Error details

In any of the following cases, an interrupt completes abnormally, and an error code is stored in the device specified with the completion status storage device (D2).

Error code*1	Description	Corrective action
0010H	The instruction request from the programmable controller CPU to the C Controller module exceeds the allowable value.	Check and correct the sequence program.
2282H	The interrupt pointer number set with the D(P).GINT instruction is out of the range of 0 to 15.	

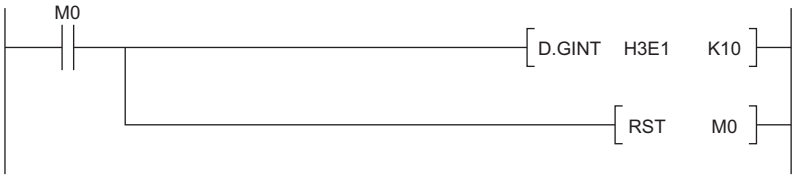
*1 "0000H" is stored when the processing is normally completed.

In any of the following cases, an operation error is caused, and the latest self-diagnostics error (SM0) turns ON. Then, an error code is stored in the latest self-diagnostics error code (SD0).

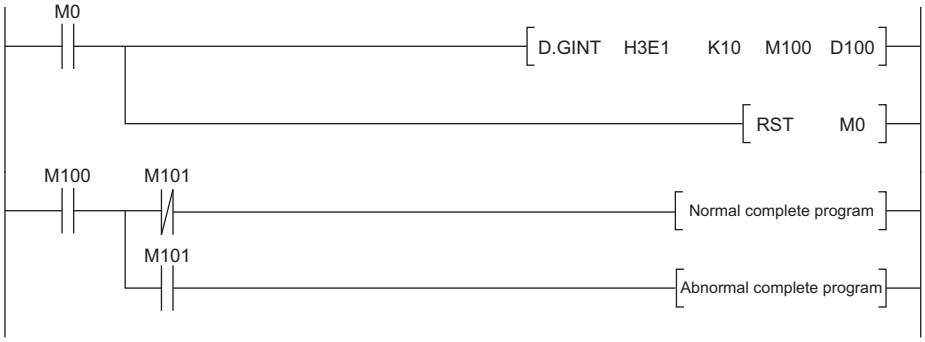
Error code	Description	Corrective action
4350	The specified target CPU module is wrong. • A CPU number set as reserved has been specified. • A CPU number of which module is not mounted has been specified. • The start I/O number of the target CPU module divided by 16 (n1) is out of the range of 3E0H to 3E3H.	Check and correct the sequence program.
4351	The instruction cannot be executed on the specified target CPU module. • The instruction name is wrong. • An instruction which is not supported by the target CPU was executed.	
4352	The number of devices of the specified instructions is incorrect.	
4353	An unavailable device is specified in the specified instruction.	

■Program example

- Program in which the completion device and completion status are omitted



- Program in which the completion device and completion status are used



Precautions

- When an interrupt event has already been notified from a programmable controller CPU and a C Controller module (another CPU) at the time of the CCPU_WaitEvent function execution, the user program is restored from the interrupt event wait state at the same time as the function execution.
  In addition, when multiple interrupt events have been notified to the same interrupt event number at the time of the CCPU_WaitEvent function execution, the user program processes them as a single interrupt event notification.
- If the same CPU number and the same interrupt event number are set in more than one user program, it will be undefined which one of the user programs receives the interrupt event.

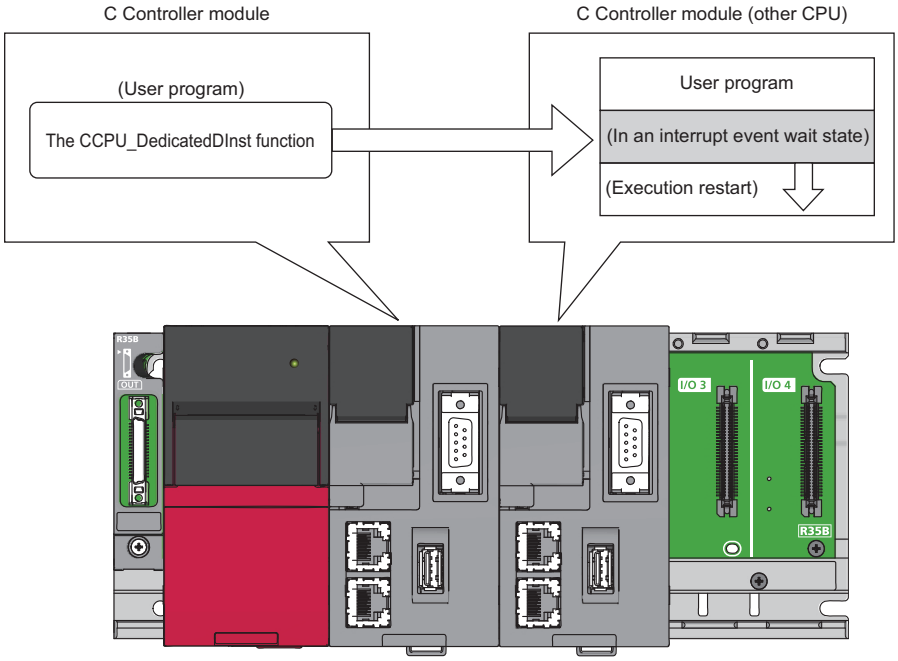
17.6 Issuing an Interrupt to Another CPU

An interrupt request to another CPU (C Controller module and motion CPU) from a C Controller module can be issued. An interrupt cannot be issued to programmable controller CPU.

C Controller module dedicated function	Description
CCPU_DedicatedDInst	To execute dedicated instructions categorized as 'D' or 'DP.'

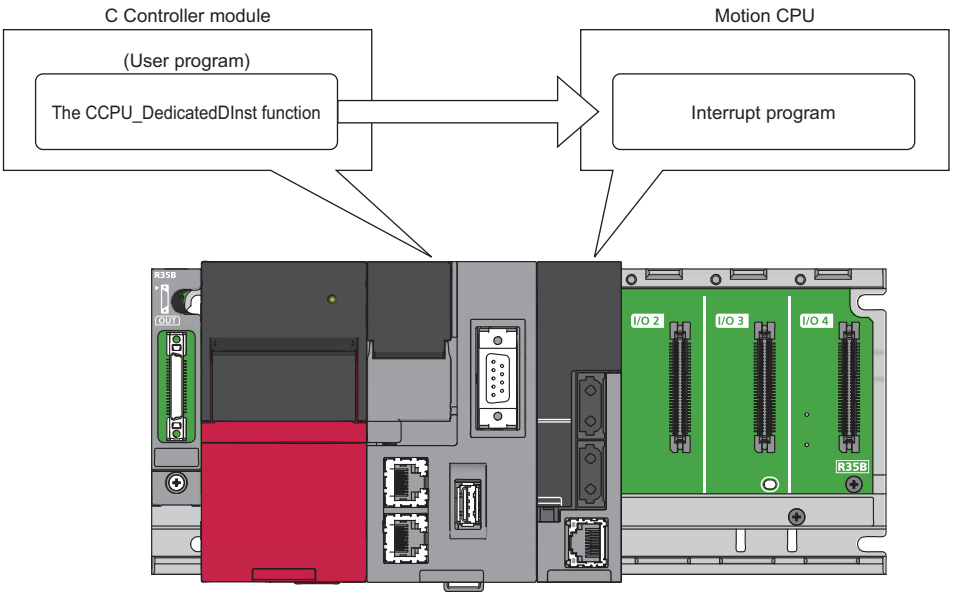
Interrupt to a C Controller module (another CPU)

When an interrupt is issued to a C Controller module (another CPU), the C Controller module restarts a user program which is waiting for an interrupt event with the C Controller module dedicated function (CCPU_WaitEvent).



Interrupt to motion CPU

When an interrupt is issued to a motion CPU, the motion CPU starts an interrupt program.



MEMO

18 FIRMWARE UPDATE FUNCTION

Acquire a firmware update file from the Mitsubishi Electric FA website to update the version of the firmware.
The following method for updating firmware is available.

Update method	Description
Update using an SD memory card	The firmware version of a module can be changed just using an SD memory card.

For the module models that support this function, precautions that are common to those modules, and update procedure, refer to the following:
MELSEC iQ-R Module Configuration Manual

Restriction

Before using the firmware update function, check the firmware version of a C Controller module. (Page 404 Added and Changed Functions)

Considerations

The following shows the considerations for a C Controller module when using the firmware update function.

■Startup time of a C Controller module

The startup time of a C Controller module may vary depending on the firmware version.
To make a system program run on the startup of a C Controller module, configure an interlock using special relays and special registers. (Page 360 Special Relay List, Page 362 Special Register List)

Special relay/special register	Description
CPU operating status (SD203)	Available to check the completion status of initial processing with operating status. If the operation status is in RUN, it indicates that the initial processing is completed.
LED status (SD201)	Available to check the activation status of a script file with the READY LED status. If the READY LED is flashing, it indicates that a script file is activated.
Special relays or special registers of each network module	Available to check the communication status with a target station.

MEMO

PART 7 DEVICES

This part consists of the following chapter.

19 DEVICES

19 DEVICES

This chapter explains the available devices.

19.1 Device List

This section shows the available device names and ranges.

Classification	Type	Device name	Default value			Number of points setting	Setting range
			Number of points	Range of use			
User device	Bit device	Input	4096 points	X0 to FFF	Hex	N/A	—
		Output	4096 points	Y0 to FFF	Hex		
	Bit device	Internal relay	61440 points	M0 to 61439	Dec	N/A	—
		Link relay	655360 points	B0 to 9FFFF	Hex		
	Word device	Data register	4184064 points	D0 to 4184063	Dec		
		Link register	1048576 points	W0 to FFFFF	Hex		
System device	Bit device	Special relay	4096 points	SM0 to 4095	Dec	N/A	—
	Word device	Special register	4096 points	SD0 to 4095	Dec		
Link direct device*1,*2	Bit device	Link input	163840 points	Jn\X0 to 27FFF	Hex	N/A	—
		Link output	163840 points	Jn\Y0 to 27FFF	Hex		
		Link relay	655360 points	Jn\B0 to 9FFFF	Hex		
		Link special relay	5120 points	Jn\SB0 to 13FF	Hex		
	Word device	Link register	2621440 points	Jn\W0 to 27FFFF	Hex		
		Link special register	5120 points	Jn\SW0 to 13FF	Hex		
Module access device	Word device	Module access device	268435456 points	Un\G0 to 268435455	Dec	N/A	—
CPU buffer memory access device	Word device	CPU buffer memory access device	268435456 points	U3En\G0 to 268435455	Dec	N/A	—
	Word device	Fixed cycle communication area access device	0 points	—	Dec	Available	U3En\HG0 to 12287
File register	Word device	File register	1835008 points	ZR0 to 1835007	Dec	N/A	—
Pointer	—	Interrupt pointer	1024 points	I0 to I15, I50 to I1023	Dec	N/A	—

*1 The number of points and the range to be used for the link direct device vary depending on network modules. For the number of points and the range to be used for the link direct device, refer to the manual for the network module used.

*2 The maximum number of points when setting "Extended Mode (iQ-R Series Mode)" for "Link Direct Device Setting."(☞ Page 203 Link direct device mode setting)
To use a link direct device mode setting, check the version of a C Controller module and CW Configurator. (☞ Page 404 Added and Changed Functions)

Point

Do not use any devices that are not listed in the device list.

19.2 Device Descriptions

This section shows the overview of available devices.

For details on devices, refer to the following:

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

Device name			Description
User device	Input	X	A device that provides CPU module with commands and/or data using an external device, such as pushbutton, transfer switch, limit switch, or digital switch.
	Output	Y	A device that outputs the operation results of the program to a device, such as external signal light/digital HMI/electromagnetic switch (contactor) /solenoid.
	Internal relay	M	A device that is used as an auxiliary relay within the CPU module.
	Data register	D	A device that stores numerical values.
	Link relay	B	A device that is used in a C Controller module when refreshing data between a network module and a C Controller module.
	Link register	W	
System device	Special relay	SM	A device that is used to store the status of a C Controller module. The specification of special relays is defined in a C Controller module. For details on the special relay, refer to the following: ☞ Page 360 Special Relay List For details on the special register, refer to the following: ☞ Page 362 Special Register List
	Special register	SD	
Link direct device	Link input	Jn\X	A device that directly accesses link relays and/or link registers in a network module on the CC-Link IE Controller Network and/or CC-Link IE Field Network.
	Link output	Jn\Y	
	Link relay	Jn\B	
	Link special relay	Jn\SB	
	Link register	Jn\W	
	Link special register	Jn\SW	
Module access device	Module access device	Un\G	A device that directly accesses from the CPU module to the buffer memory of an intelligent function module mounted on the main base unit and extension base unit.
CPU buffer memory access device	CPU buffer memory access device	U3En\G	A device that accesses memory used for reading/writing data between CPU modules in a multiple CPU system, or for the CPU module built-in function such as Ethernet function
	Fixed cycle communication area access device	U3En\HG	
File register	File register	ZR	A device that retains data while the power is OFF. It exists in the file storage area of the device/label memory.
Pointer	Interrupt pointer	I	A device that performs a corresponding routine when using the interrupt function

MEMO

PART 8 C CONTROLLER MODULE OPERATION EXAMPLE

This part consists of the following chapters.

20 SYSTEM CONFIGURATION EXAMPLE

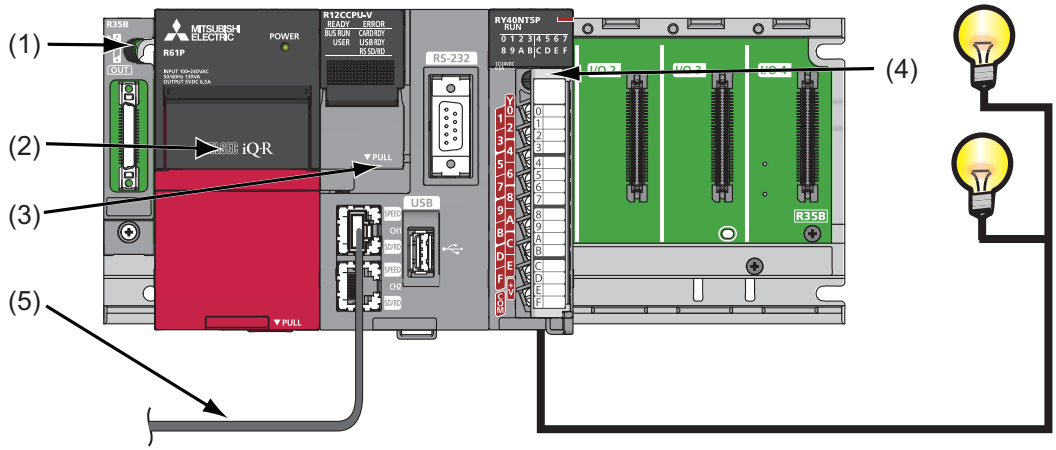
21 SETTING THE MODULE

22 PROGRAMMING

23 CHECKING OPERATIONS

20 SYSTEM CONFIGURATION EXAMPLE

The following system configuration is used in the operation example.



No.	Name	Model name	Description
(1)	Base unit	R35B	A unit to mount a power supply module, a C Controller module, and I/O modules.
(2)	Power supply module	R62P	A module which supplies power to modules, such as a C Controller module and I/O modules.
(3)	C Controller module	R12CCPU-V	A module which unifies control process of a C Controller system.
(4)	Output module	RY40NT5P	—
(5)	Cable (Ethernet cable)	An Ethernet cable that meets 10BASE-T/100BASE-TX/ 1000BASE-T standards	A cable which connects a personal computer in which CW Configurator and CW Workbench are installed and a C Controller module.

MEMO

21 SETTING THE MODULE

Set a C Controller module.

Point

The operation example in this chapter explains the settings in Windows® 7.

21.1 Initializing a C Controller Module

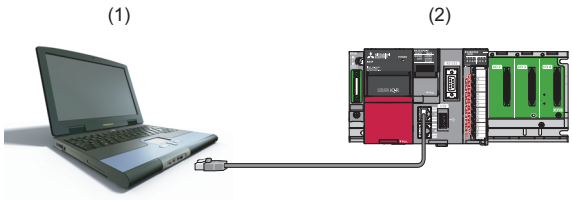
A C Controller module is required to be initialized before the first operation.
For the method for initializing a module, refer to the following:
☞ Page 28 Initialization

21.2 Setting Parameters

Set parameters for the C Controller module.

Connecting the C Controller module to the personal computer

Connect the personal computer (1) to the CH1 on the C Controller module (2) using an Ethernet cable.



Precautions

The IP address of the C Controller module and that of the personal computer are required to be set in the same segment. In this operation example, the default IP address is set for the C Controller module (192.168.3.3); therefore, set '192.168.3.* (*: except for 0, 3, and 55)' for the personal computer.
In addition, set the subnet mask for the personal computer to '255.255.255.0'.

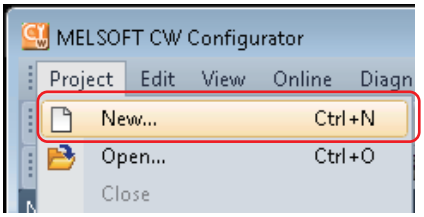
Point

For changing an IP address, refer to the following:
☞ Page 159 Parameter Settings

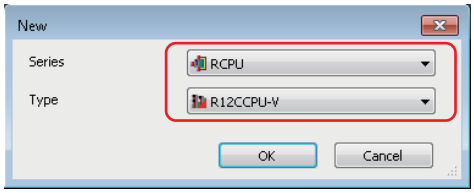
Starting CW Configurator to create a project

Operating procedure

1. Start CW Configurator from "MELSOFT" in Windows Start.
2. Select [Project] ⇨ [New].



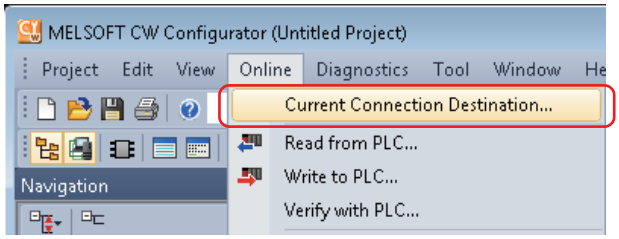
3. Check that "RCPU" and "R12CCPU-V" are selected, and click the [OK] button.



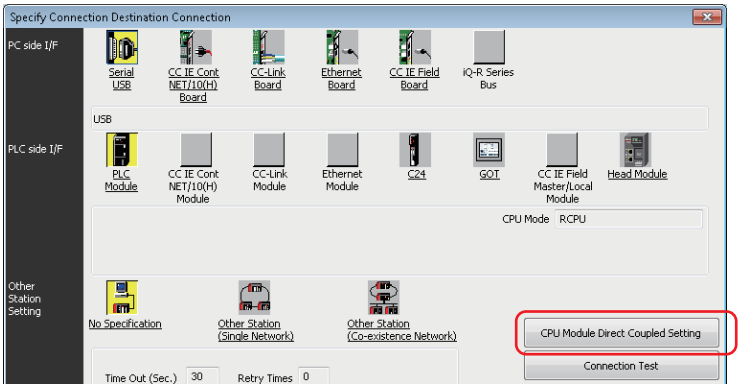
Establishing communication with the C Controller module

Operating procedure

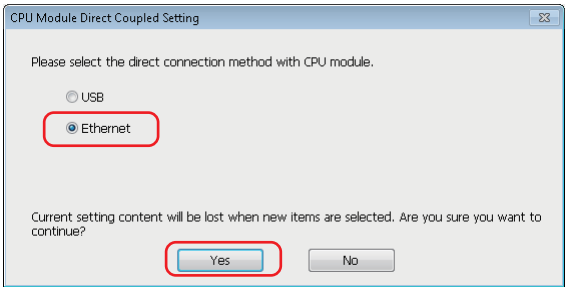
1. Select [Online] ⇨ [Current Connection Destination] from the menu of CW Configurator.



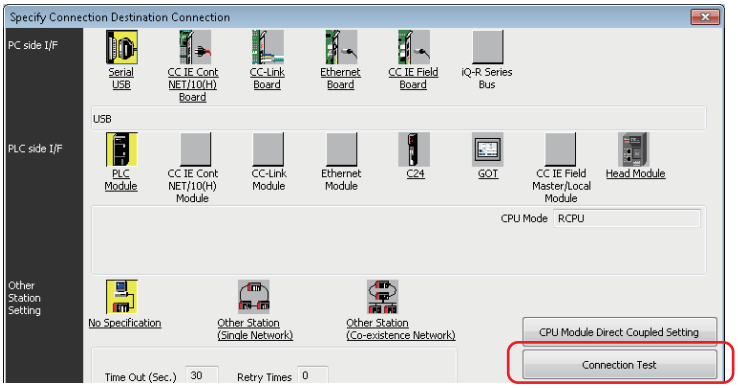
2. Click the [CPU Module Direct Coupled Setting] button in the "Specify Connection Destination Connection" screen.

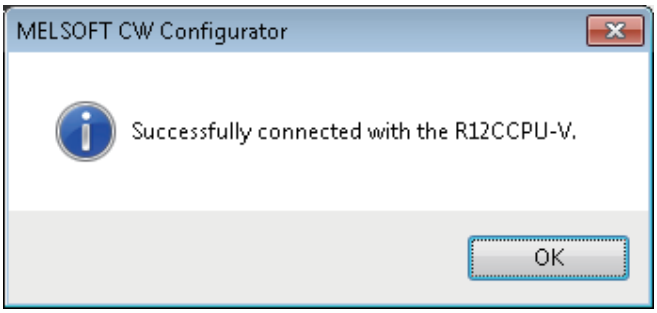


3. Select "Ethernet," and click the [Yes] button.



4. Click the [Connection Test] button, and check that the message "Successfully connected with the R12CCPU-V." appears.



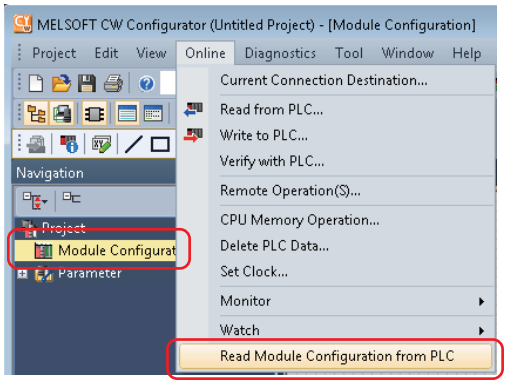


Setting parameters

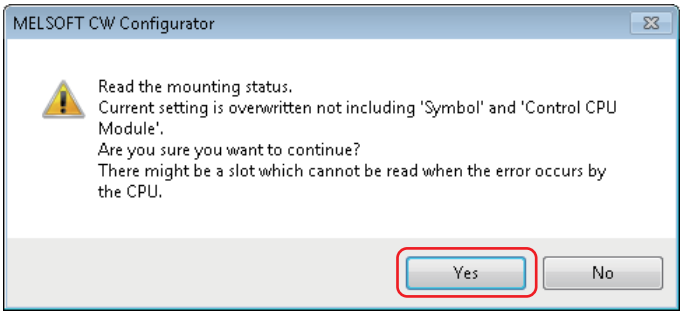
Set parameters for the system and modules.

Operating procedure

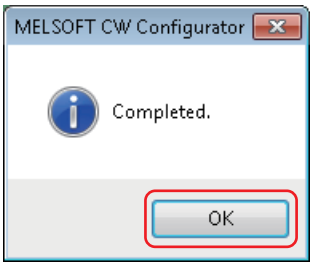
1. Double-click "Module Configuration" in the "Navigation" window to open, and select [Online] ⇨ [Read Module Configuration from PLC].



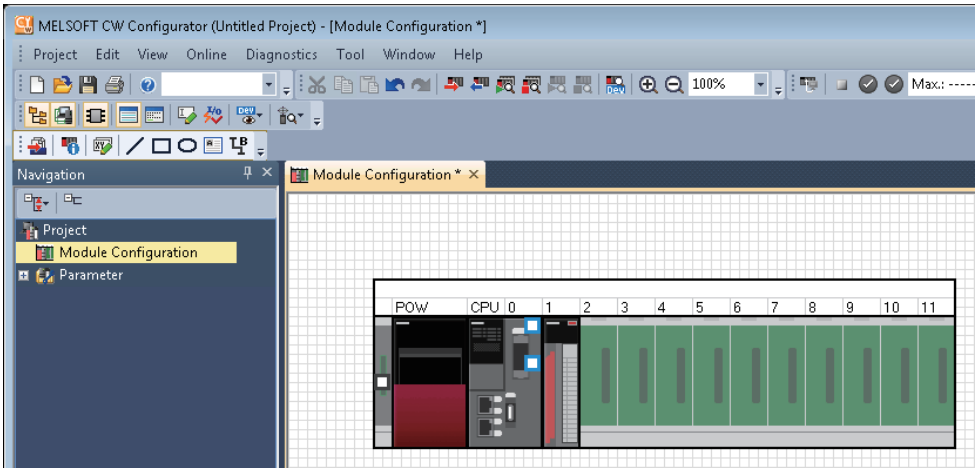
2. When the following message appears, click the [Yes] button.



3. Click the [OK] button when the following message appears after reading is completed.



4. The system parameters are automatically set, and the actual system configuration is displayed in the "Module Configuration" window.

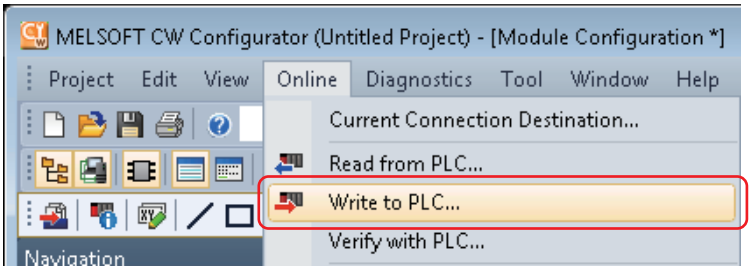


Writing parameters to the C Controller module

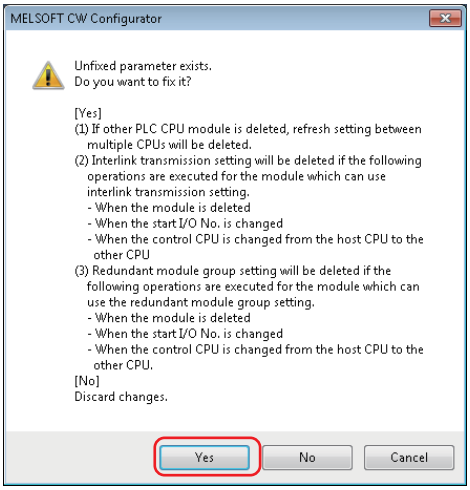
Write the parameters to the C Controller module using CW Configurator.

Operating procedure

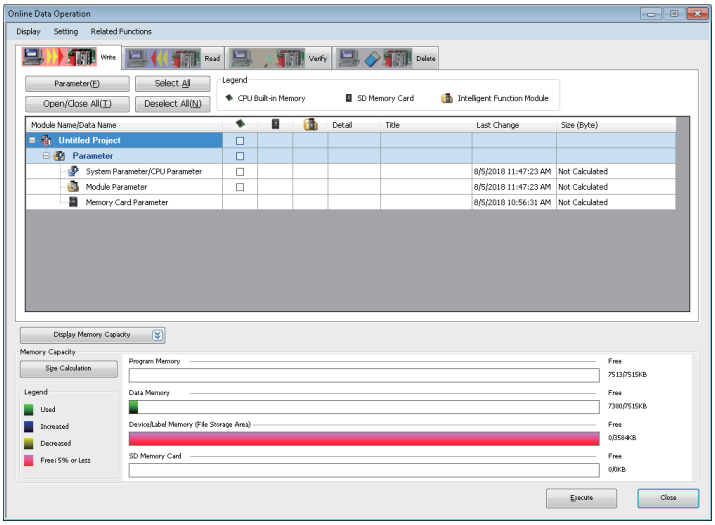
1. Select [Online] ⇨ [Write to PLC].



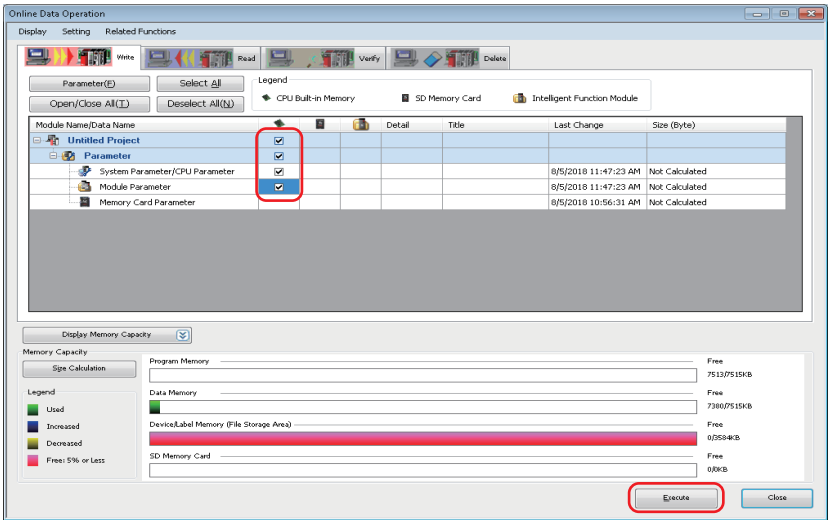
2. Click the [Yes] button.



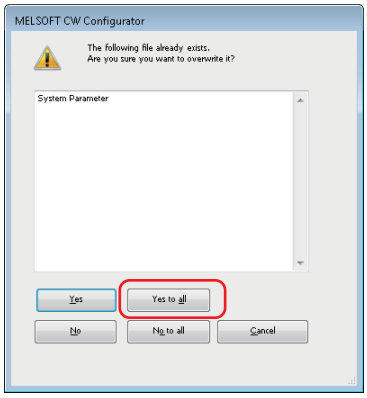
3. Check that the "Online Data Operation" screen appears.



4. Select "System Parameter/CPU Parameter" and "Module Parameter," and click the [Execute] button.

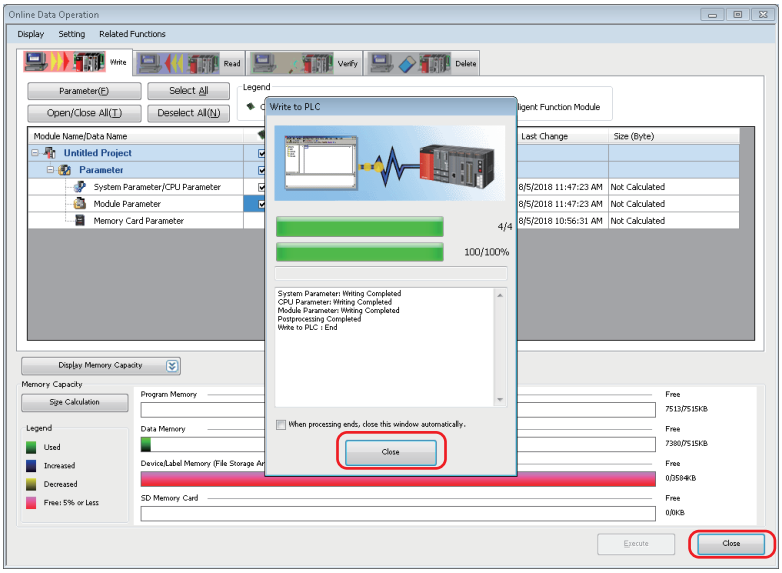


5. Click the [Yes to all] button.



Writing the parameters starts.

6. Click the [Close] button when writing to the C Controller module is completed.

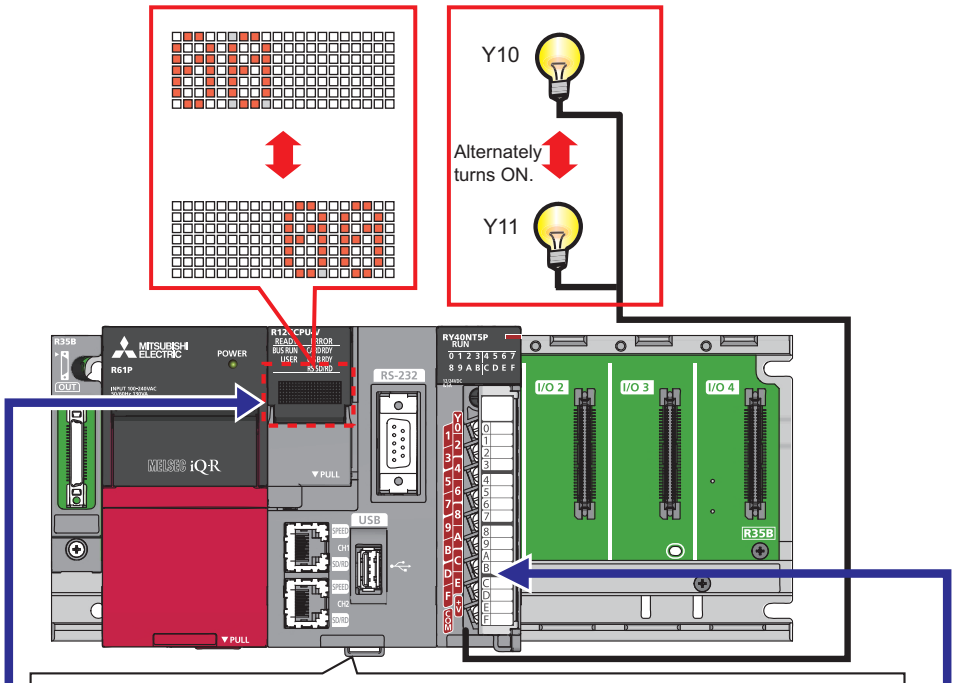


22 PROGRAMMING

Create a program in which lamps connected to an output module and the dot matrix LED on the front of the C Controller module flash.

22.1 Program Example and Control Description

When the C Controller module is set to RUN, the output lamps Y10 and Y11 alternately turn ON.
Synchronizing with the output lamps, the dot matrix LED on the front of the C Controller module switches alternately between "00__" and "__00."




```
/* Output control */
sRet = CCPU_Y_Out_WordEx(NORMAL_ACCESS, UNIT_XY, WORD, &usDataBuf, DUMMY);
if(sRet != 0){
    printf("ERROR : CCPU_Y_Out_WordEx_1 [%d(%04hxH)]\n", sRet, sRet);
    return;
}

/* Dot matrix LED control */
sRet = CCPU_SetDotMatrixLED(DISPMODE_ASCII, &pcdata[0]);
if(sRet != 0){
    printf("ERROR : CCPU_SetDotMatrixLED_1 [%d(%04hxH)]\n", sRet, sRet);
    return;
}
```

Source code

The following shows the source code used in the operation example. (☞ Page 289 Preparing a User Program)

```
/*****************************************************************************/
/* Function header */
/*****************************************************************************/
#include        <vxworks.h>     /* VxWorks function header */
#include        <taskLib.h>     /* VxWorks function header */
#include        <stdio.h>       /* Standard function header */
#include        <string.h>      /* Standard function header */
#include        "CCPUFunc.h"  /* C Controller module dedicated function header */
/*****************************************************************************/
/* Definition */
/*****************************************************************************/
/* For debugging */
#define UNIT_XY         0x0010 /* Start I/O number of the module */
#define RY_LED          0x5555 /* Initial output value of Y signals (even bit: on) */
#define LED_0           0x30    /* Initial output value on the dot matrix LED (LED1,2) */
#define LED_SPACE       0x20    /* Initial output value on the dot matrix LED (LED3,4) */

/* For C Controller module dedicated function */
#define WORD                    1 /* 1-word specification */
#define NORMAL_ACCESS           0 /* General access specification */
#define DUMMY                   0 /* Dummy */
#define DISPMODE_ASCII          1 /* Dot matrix LED output mode */
/*****************************************************************************/
/* Processing to output Y signals and to control dot matrix LED */
/*****************************************************************************/
void R12_SampleTask()
{
        /* Declare local variables. */
        short           sRet;           /* Return value of the C Controller module dedicated function */
        unsigned short  usDataBuf;              /* For accessing Y signals */
        unsigned short  usEmptyDataBuf;         /* For resetting Y signals */
        char            pcdata[4];              /* Value on the dot matrix LED */
        short           i;                      /* For loop */

        /* Set the output value of Y signals (turn on the even bit). */
        usDataBuf = RY_LED;

        /* Set the output value of the dot matrix LED (LED1,2: on). */
        pcdata[0] = LED_0;
        pcdata[1] = LED_0;
        pcdata[2] = LED_SPACE;
        pcdata[3] = LED_SPACE;

        /* Perform an output control and dot matrix LED control in turns for 20 times. */
        for(i = 0; i < 20; i++){
                /* Output control */
                sRet = CCPU_Y_Out_WordEx(NORMAL_ACCESS, UNIT_XY, WORD, &usDataBuf, DUMMY);
                if(sRet != 0){
                        printf("ERROR : CCPU_Y_Out_WordEx_1 [%d(%04hxH)]\n", sRet, sRet);
                        return;
                }
```

```
		/* Dot matrix LED control */
		sRet = CCPU_SetDotMatrixLED(DISPMODE_ASCII, &pcdata[0]);
		if(sRet != 0){
			printf("ERROR : CCPU_SetDotMatrixLED_1 [%d(%04hxH)]\n", sRet, sRet);
			return;
		}

		/* Invert the output value of Y signals (turn on the bits in order of odd bit -> even bit -> ...). */
		usDataBuf = ~usDataBuf;

		/* Switch the output of the dot matrix LED (LED1,2: on -> LED3,4: on). */
		if(i%2 ==0){
			pcdata[0] = LED_SPACE;
			pcdata[1] = LED_SPACE;
			pcdata[2] = LED_0;
			pcdata[3] = LED_0;
		}else{
			pcdata[0] = LED_0;
			pcdata[1] = LED_0;
			pcdata[2] = LED_SPACE;
			pcdata[3] = LED_SPACE;
		}
		/* Wait */
		taskDelay(60);
	}

	/* Reset Y signals. */
	usEmptyDataBuf = 0x00;
	sRet = CCPU_Y_Out_WordEx(NORMAL_ACCESS, UNIT_XY, WORD,&usEmptyDataBuf, DUMMY);
	if(sRet != 0){
		printf("ERROR : CCPU_Y_Out_WordEx_2 [%d(%04hxH)]\n", sRet, sRet);
		return;
	}

	/* Reset the dot matrix LED. */
	pcdata[0] = LED_SPACE;
	pcdata[1] = LED_SPACE;
	pcdata[2] = LED_SPACE;
	pcdata[3] = LED_SPACE;
	sRet = CCPU_SetDotMatrixLED(DISPMODE_ASCII, &pcdata[0]);
	if(sRet != 0){
		printf("ERROR : CCPU_SetDotMatrixLED_2 [%d(%04hxH)]\n", sRet, sRet);
		return;
	}
}
```

22.2 Creating a Project

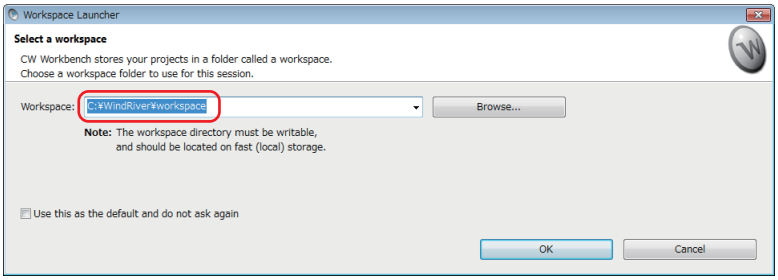
Starting CW Workbench

Operating procedure

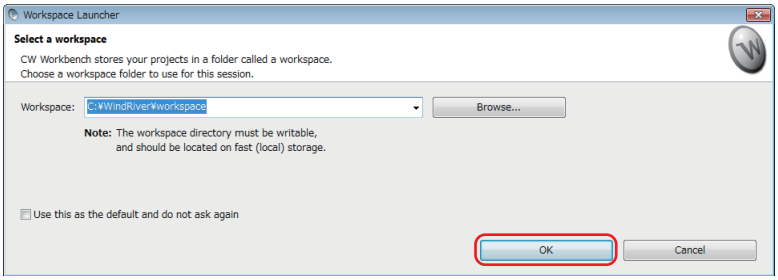
1. Start CW Workbench 3.3 from "Wind River" in Windows Start.

2. After CW Workbench is started, enter the save destination folder for a project.

In this procedure, enter 'C:\WindRiver\workspace'.



3. Click the [OK] button.



The main screen of CW Workbench appears.

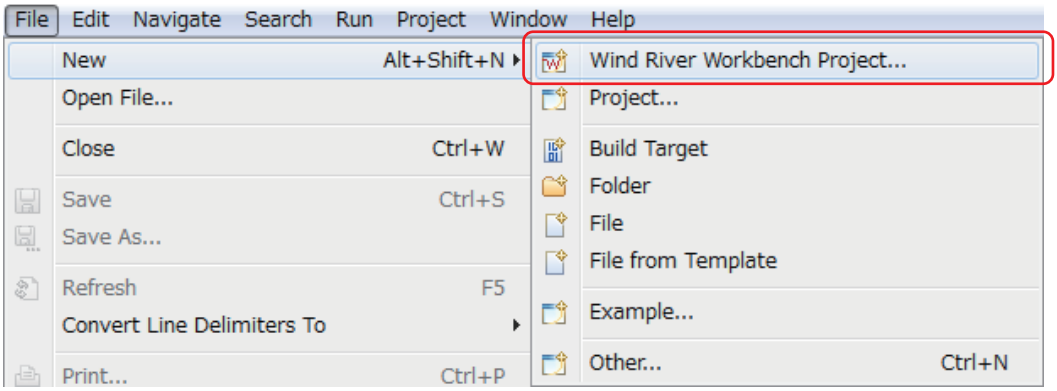
Point

- The default window sizes and icon positions on CW Workbench depend on a personal computer. If a window size differs from that shown in this operation example, adjust the size.
- To default an enlarged/deleted window, select [Window] ⇨ [New Window].

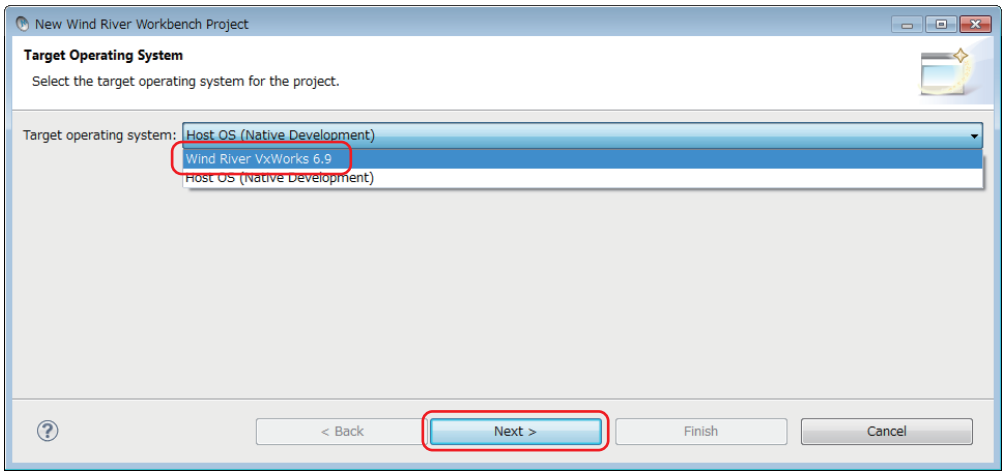
Creating a new project

Operating procedure

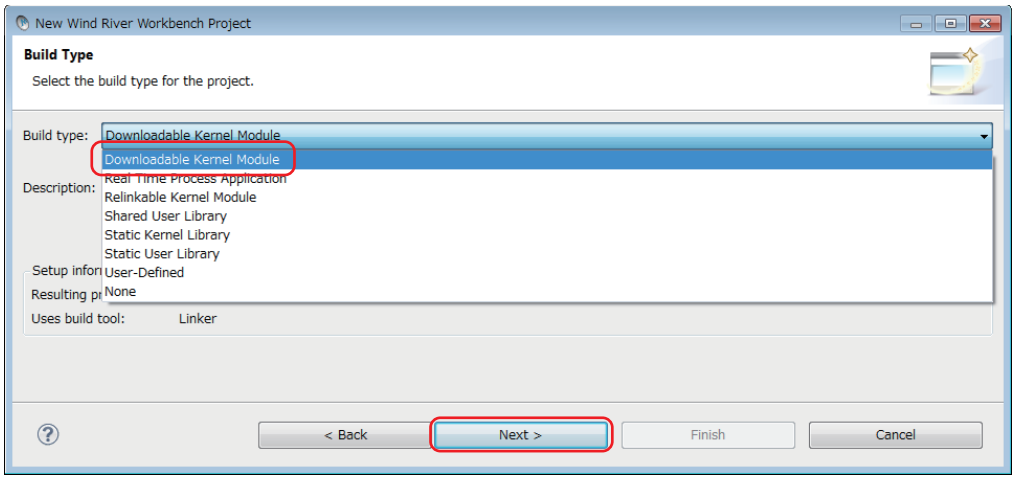
1. Select [File] ⇨ [New] ⇨ [Wind River Workbench Project].



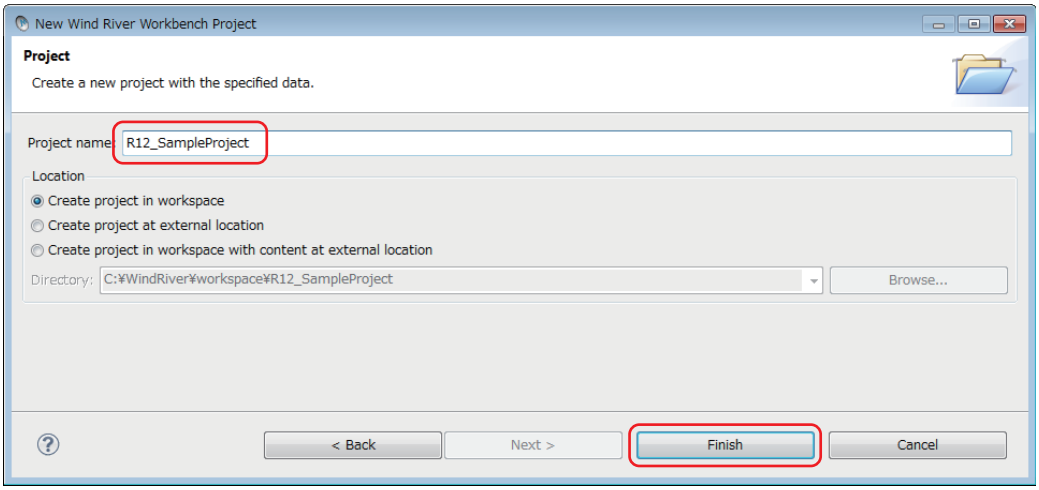
2. Select "Wind River VxWorks 6.9," and click the [Next] button.



3. Select "Downloadable Kernel Module," and click the [Next] button.



4. Enter a project name, and click the [Finish] button.
(Project name: 'R12_SampleProject' in this chapter)



Setting properties of the project

Configure the settings to convert (build) the created project into a module that can be executed on the C Controller module.

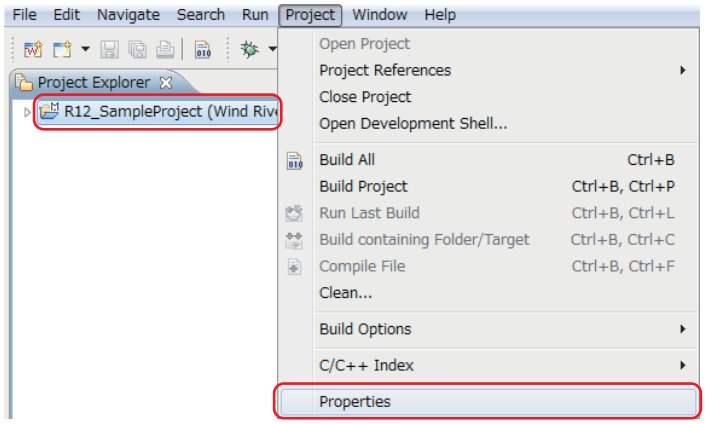
Point

Build: An operation that compiles source code according to a processor and links the code to the include file.

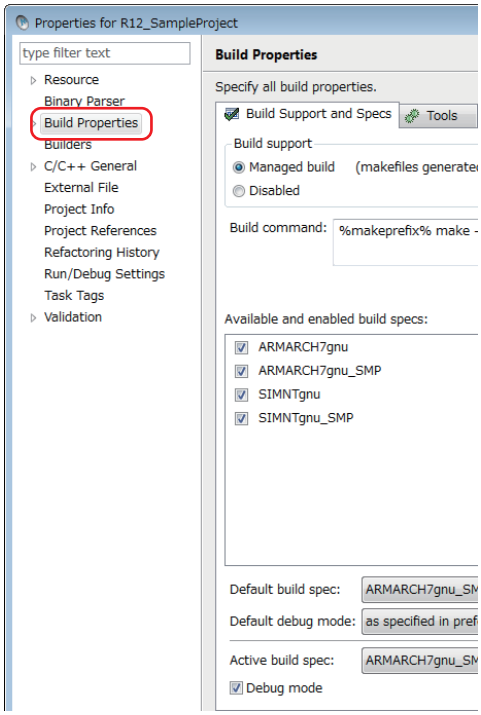
■Setting a processor

Operating procedure

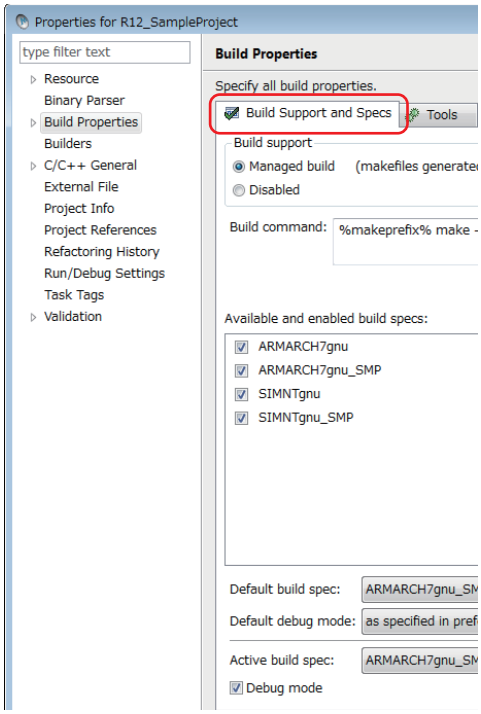
1. Select the created project in the "Project Explorer" window, and select [Project] ⇨ [Properties].



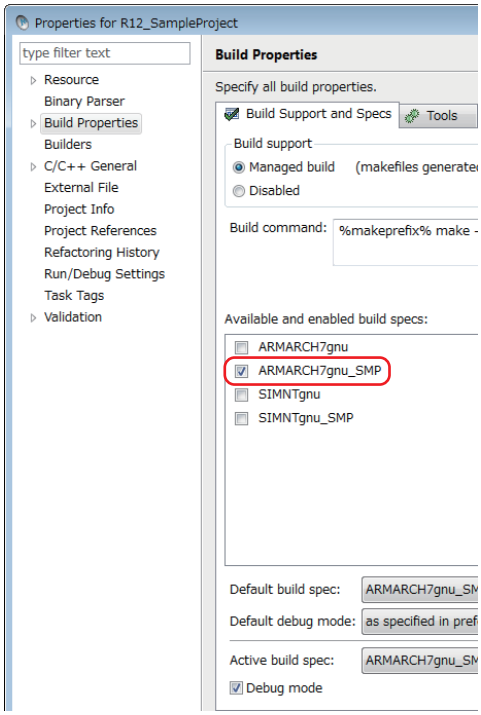
2. Select "Build Properties" from the tree to the left in the screen.



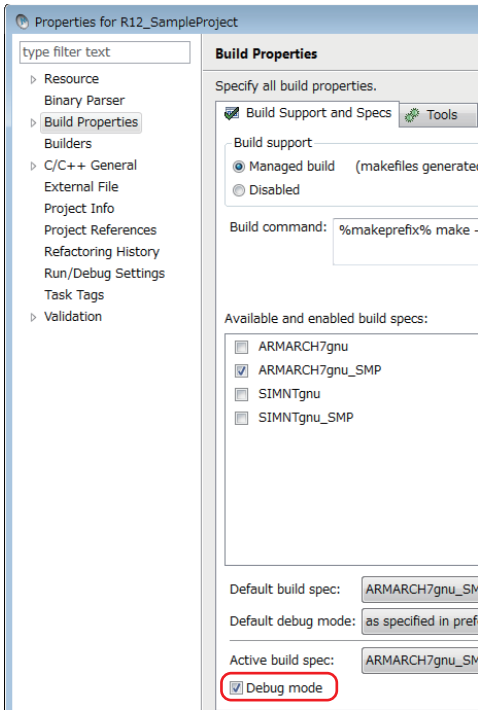
3. Select the [Build Support and Specs] tab.



4. Select only the checkbox of "ARMARCH7gnu_SMP" in "Available and enabled build specs."



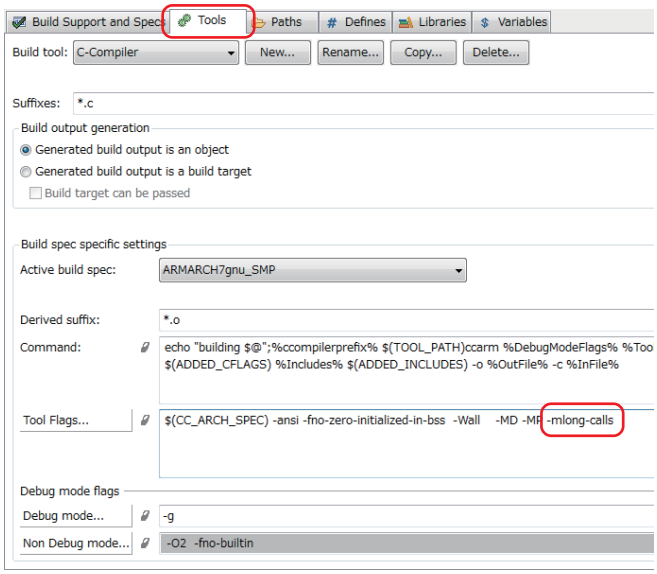
5. Select the checkbox of "Debug mode."



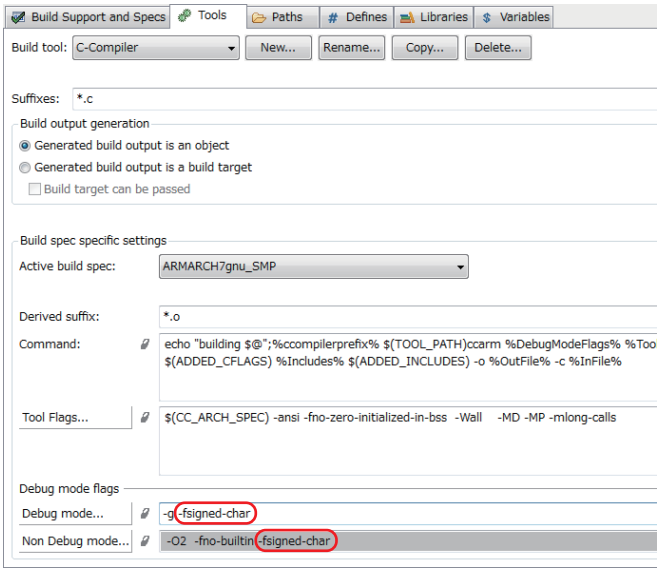
Point

For the actual system operation, unselect the checkbox of "Debug mode."

6. Select the [Tools] tab, enter "-mlong-calls" in the field next to the [Tool Flags] button.



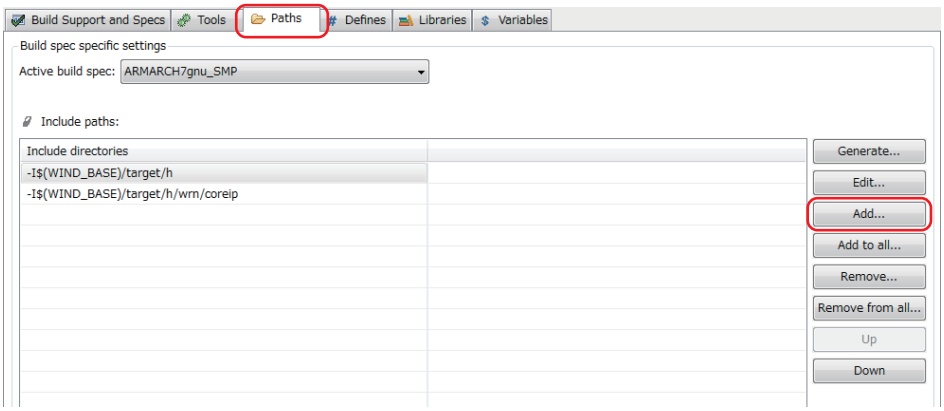
7. Enter "-fsigned-char" in the fields next to the [Debug mode] button and [Non Debug mode] button under "Debug mode flags."



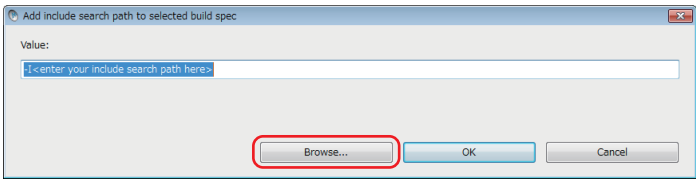
■Setting an include file

Operating procedure

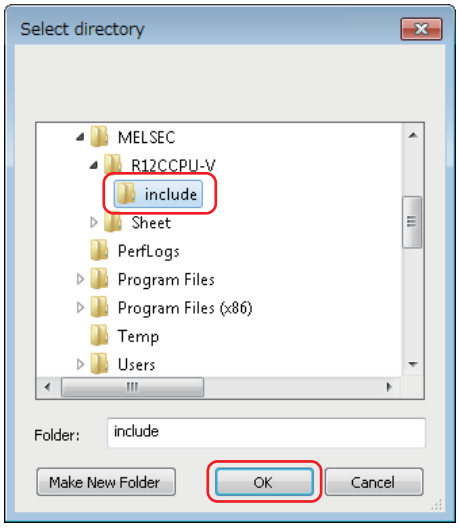
1. Select the [Paths] tab, and click the [Add] button.



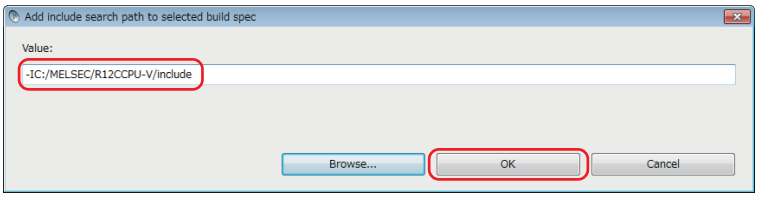
2. Click the [Browse] button.



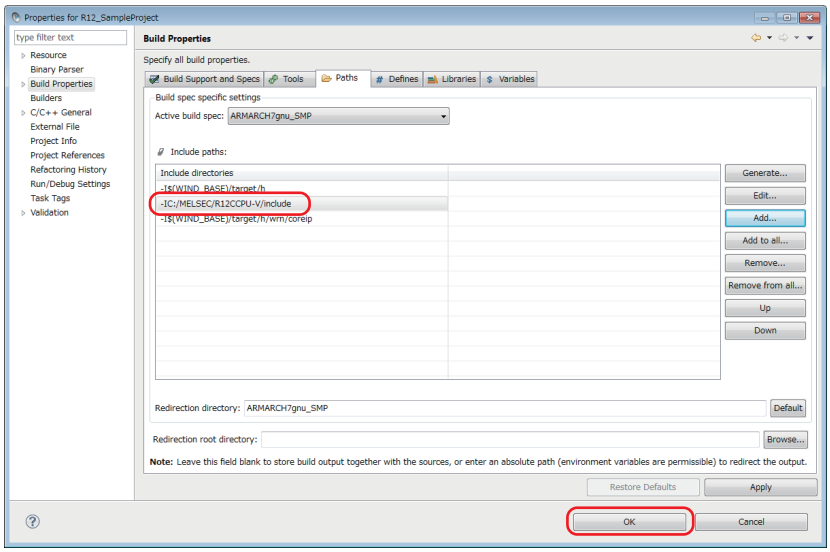
3. Select the include folder for the C Controller module in the "Select directory" screen, and click the [OK] button.
In this procedure, the include file is stored in "C:\MELSEC\R12CCPU-V."



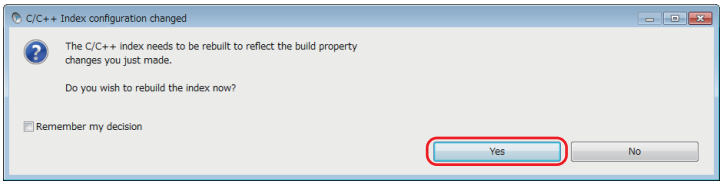
4. Check that the selected folder is specified in the "Add include search path to selected build spec" screen, and click the [OK] button.



5. Check that the added include path is displayed in "Include paths," and click the [OK] button.



6. If the following message appears after clicking the [OK] button, click the [Yes] button.



7. Add the include file to the include folder added in step 1 to 5.

To acquire an include file stored in the C Controller module, start Explorer and enter the following address in the address bar.

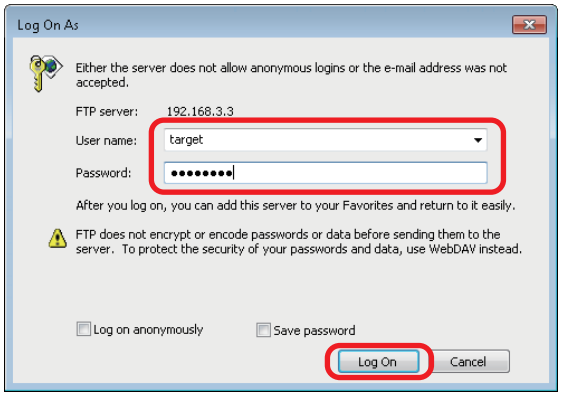
"ftp://192.168.3.3/SYSTEMROM/INCLUDE/"

The "Log On As" screen appears.

8. Enter the following user name and password in the "Log On As" screen.

- User name: target
- Password: password

9. Click the [Log On] button.



10. Copy the header file to the include folder added in step 1 to 5.

22.3 Preparing a User Program

Prepare a user program that controls the C Controller system.
A sample program for the quick start guide is used in this operation example.

Point

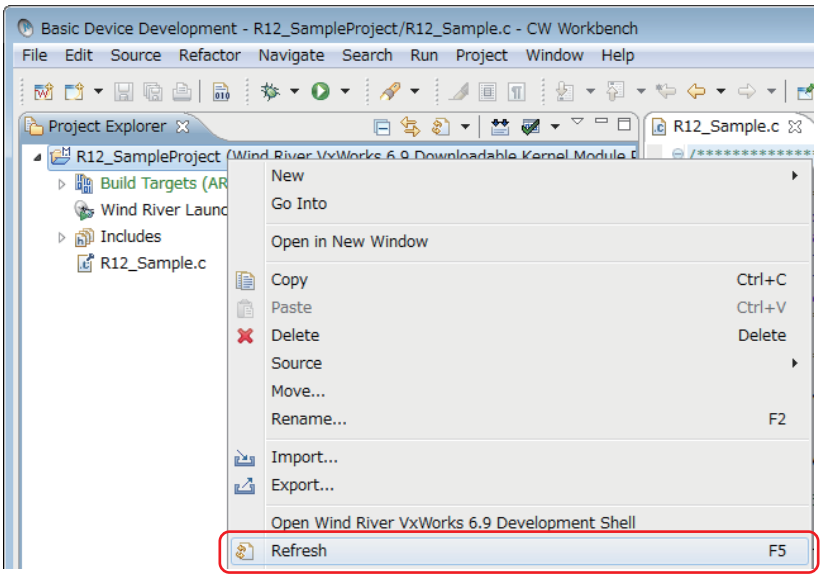
For information on how to obtain sample programs for the quick start guide, please contact your local Mitsubishi Electric sales office or representative.

Adding a sample program

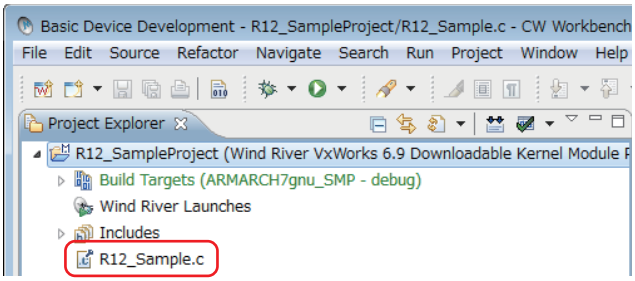
Operating procedure

1. Store a sample program immediately under the created project folder.
"C:\WindRiver\workspace\R12_SampleProject"

2. Select and right-click the created project in the "Project Explorer" window, and select [Refresh].



3. The sample program stored in step 1 is added to the project.

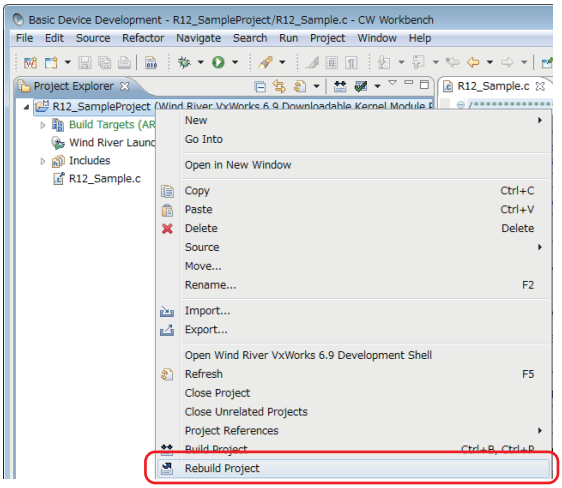


Generating an execution module from the user program

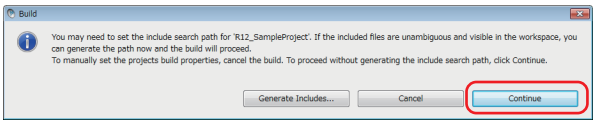
Convert (build) the created program into a module that can be executed on the C Controller module.

Operating procedure

1. Select and right-click the created project in the "Project Explorer" window, and click [Rebuild Project].

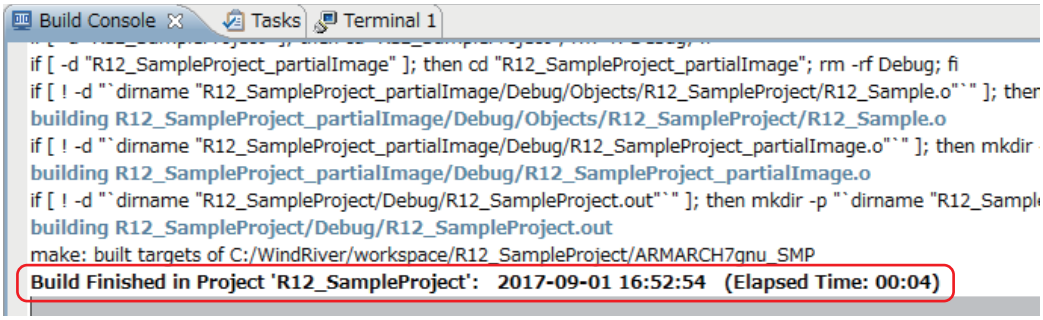


2. If the following message appears after selecting [Rebuild Project], click the [Continue] button.



The project starts to be built. The progress is displayed in the "Build Console" window.

3. Check that "Build Finished..." is displayed in the "Build Console" window.



Point

If "Build Finished..." is not displayed and an error occurs, check the error and modify the program.
After modifying the program, perform the operation again from 'Page 290 Generating an execution module from the user program.'

Connecting the C Controller module with CW Workbench

Connect the user Ethernet port CH1 of the C Controller module with CW Workbench to perform debugging in CW Workbench.

Operating procedure

1. To acquire a VxWorks image file from the C Controller module, start Explorer and enter the following address in the address bar.

"ftp://192.168.3.3/SYSTEMROM/OS_IMAGEFILE/"

The "Log On As" screen appears.

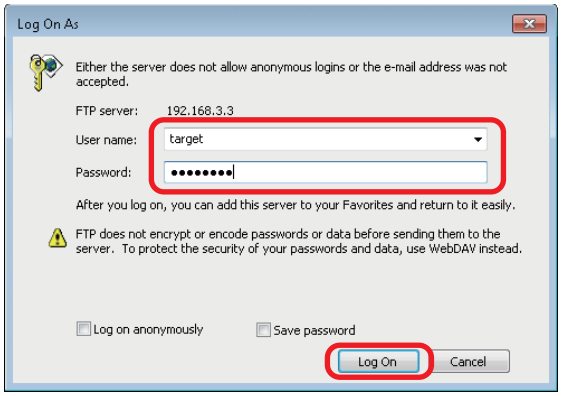
Point

To communicate between the C Controller module and the personal computer, specify the same VxWorks image file for both.

2. Enter the following user name and password in the "Log On As" screen.

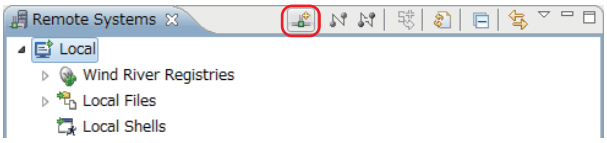
- User name: target
- Password: password

3. Click the [Log On] button.



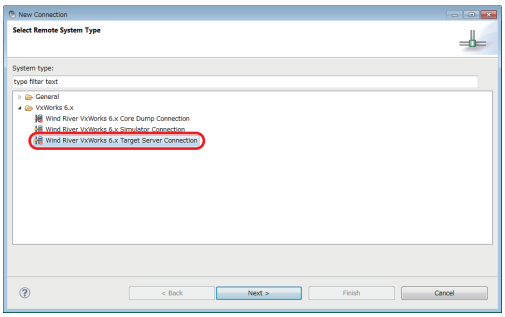
4. Create the "C:\MELSEC\R12CCPU-V\CCPUTool" folder, and copy the VxWorks image file stored in the C Controller module to the folder.

5. Click in the "Remote Systems" window.

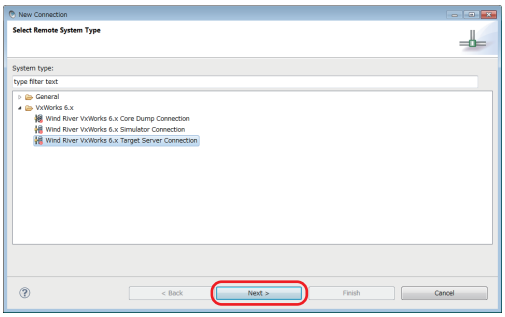


The "New Connection" screen appears.

6. Select "Wind River VxWorks 6.x Target Server Connection" in the "New Connection" screen.

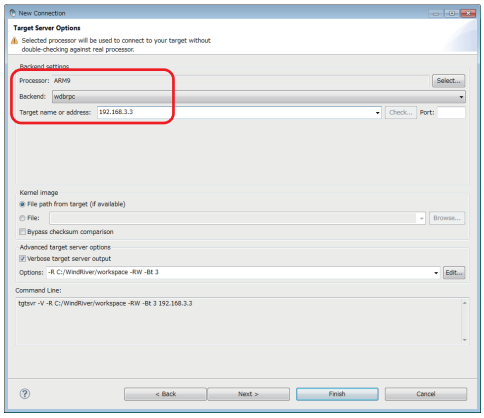


7. Click the [Next] button.

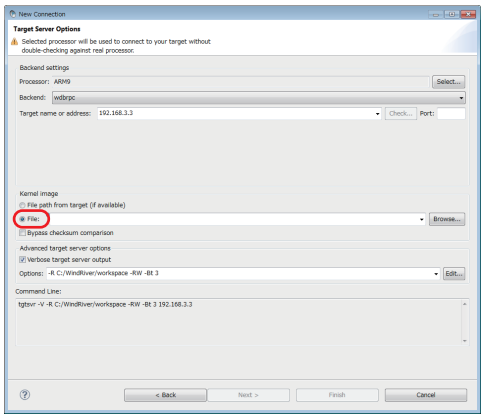


8. Set the following items in "Backend settings."

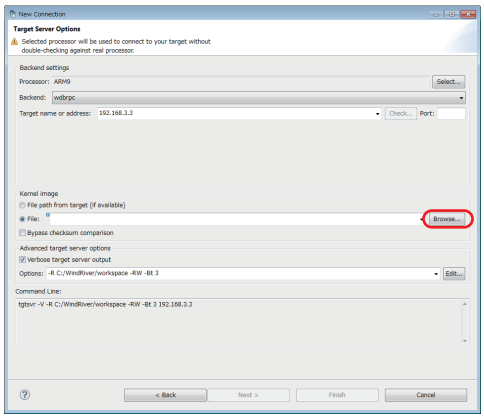
- Processor : ARM9 (Click the [Select] button and select the processor.)
- Backend: wdbrpc
- IP Address: 192.168.3.3 (default)
- Port: blank



9. Select "File" in "Kernel image."

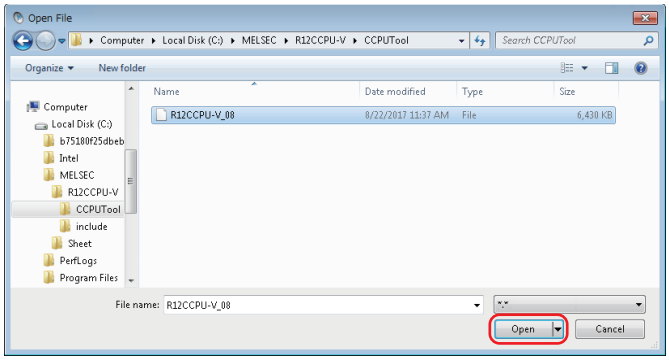


10. Click the [Browse] button.

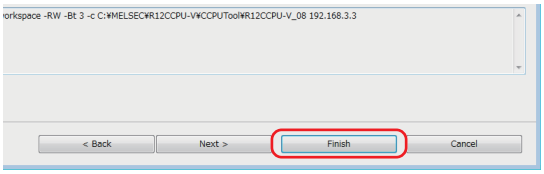


The "Open File" screen appears.

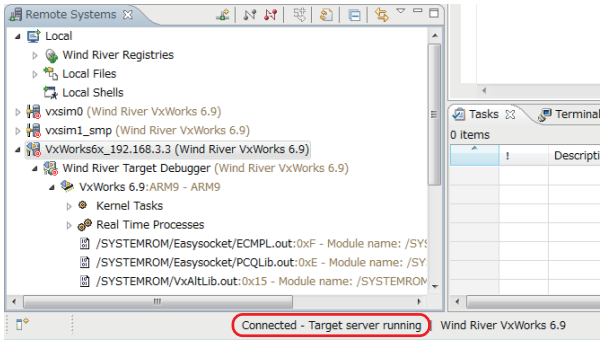
11. Select the VxWorks image file copied to the "C:\MELSEC\R12CCPU-V\CCPUTool" folder in step 4, and click the [Open] button.



12. Click the [Finish] button.



13. The connection is completed when "Connected - Target server running" is displayed at the bottom of the "Remote Systems" window.



Point

If "Connected - Target server running" is not displayed, check that the C Controller module is normally powered ON, and perform the operation again from 'Page 291 Connecting the C Controller module with CW Workbench.'

Debugging the user program

Check if the created program runs properly.

■Downloading the user program on the C Controller module

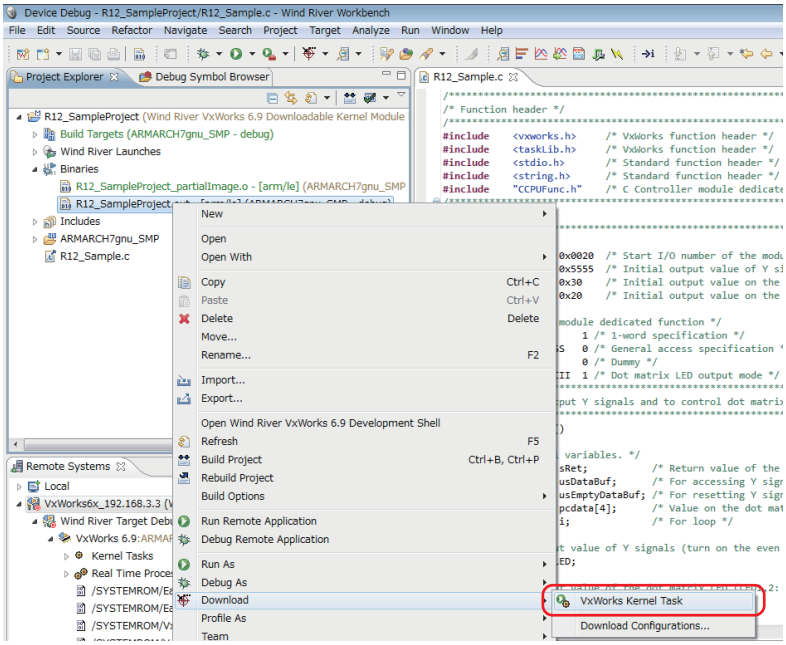
To debug the user program, download the execution module on the memory in the C Controller module.
By downloading a user program, the program can be executed with no script file.

Point

Script file: A file to describe a loading location of a user program that starts at the start of a C Controller module and a startup order of the user program.

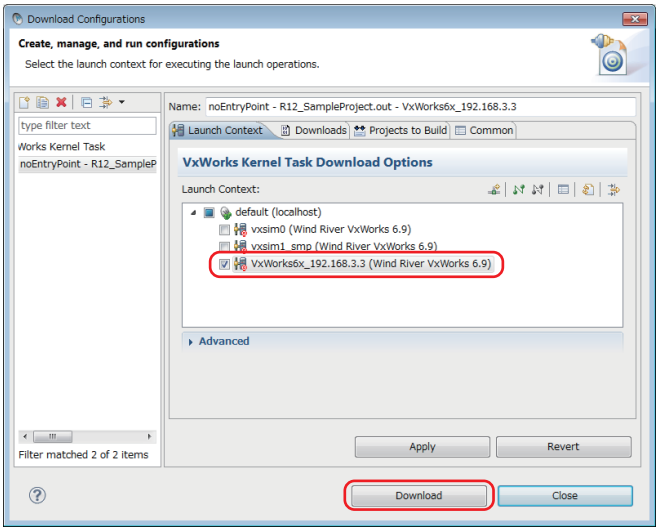
Operating procedure

1. Select and right-click the created module file "R12_SampleProject.out" in the "Project Explorer" window, and select [Download] ⇨ [VxWorks Kernel Task].



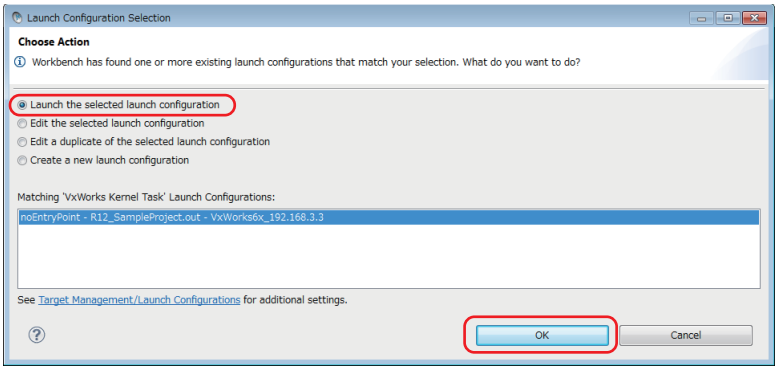
The "Download Configurations" screen appears.

2. Select only the checkbox of "VxWorks6x_192.168.3.3 (Wind River VxWorks 6.9)" in the [Launch Context] tab, and click the [Download] button.



Point

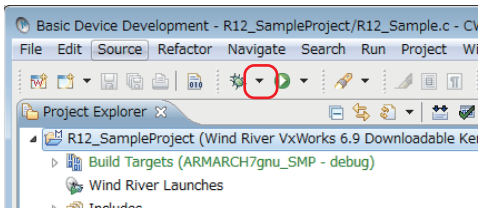
The "Launch Configuration Selection" screen appears for the second operation or later of step 2 .
Select "Launch the selected launch configuration," and click the [OK] button.



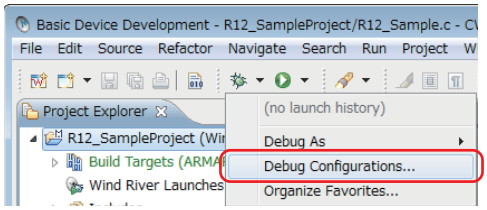
■Debugging the user program

Operating procedure

1. Select the created project in the "Project Explorer" window, and click the [▼] button on the right side of on the toolbar.

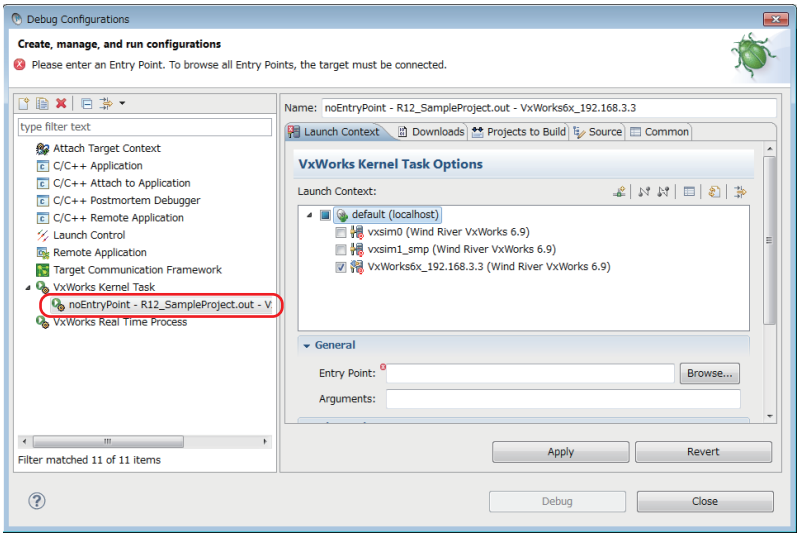


2. Select [Debug Configurations].

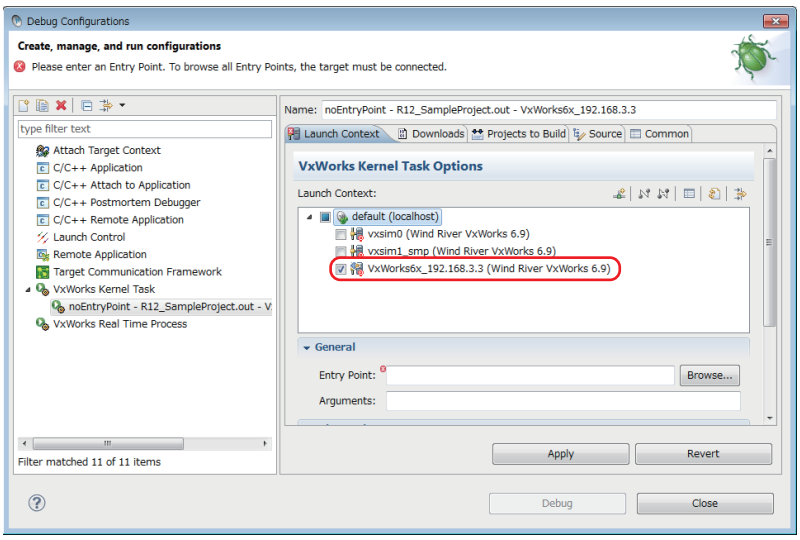


The "Debug Configurations" screen appears.

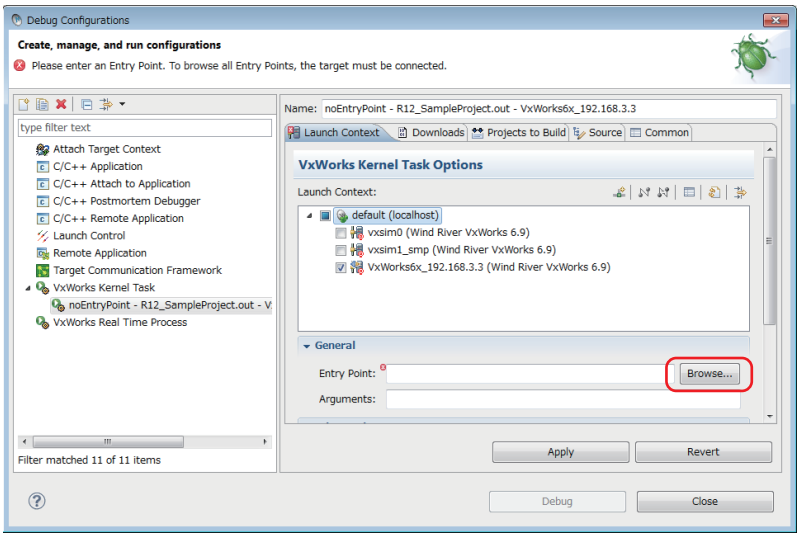
3. Select the downloaded module "R12_SampleProject.out" under "VxWorks Kernel Task."



4. Select a target server indicating connection to the C Controller module.

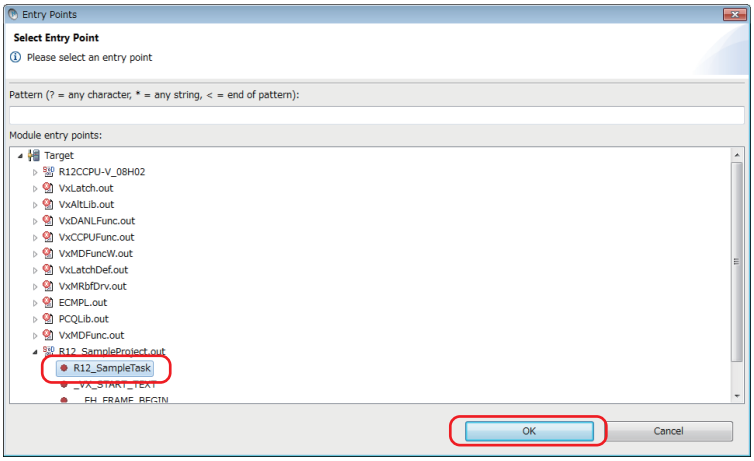


5. Click the [Browse] button.

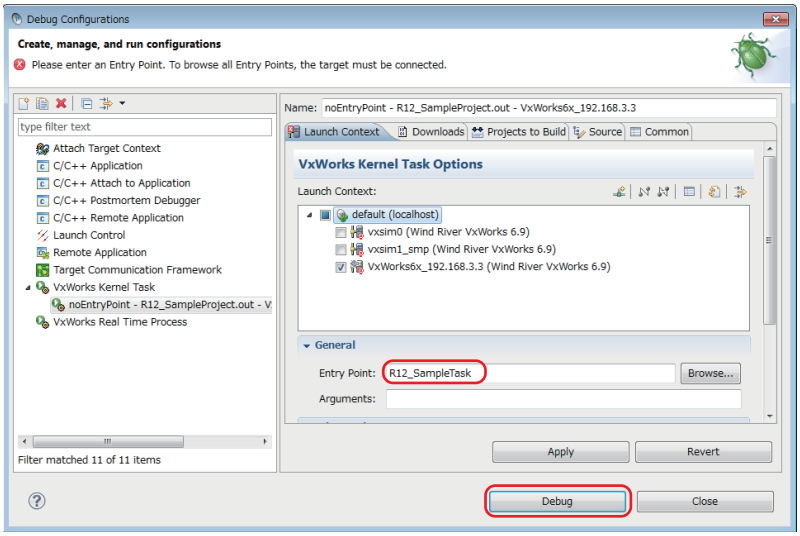


The "Entry Points" screen appears.

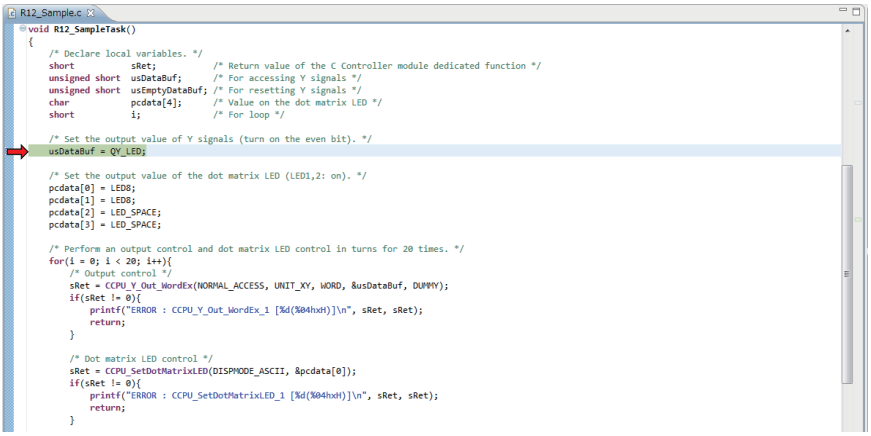
6. Select the function "R12_SampleTask" that starts debugging, and click the [OK] button.



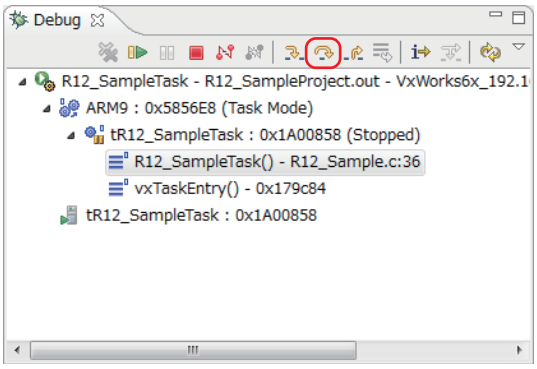
7. Check that the function name selected in step 6 is set for "Entry Point," and click the [Debug] button.



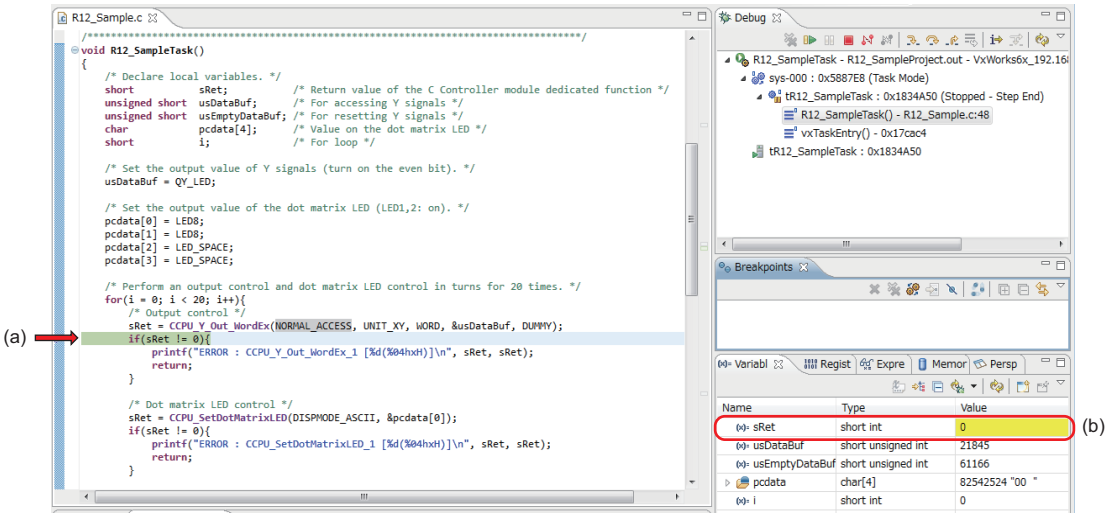
8. Debugging starts. Program execution stops at the start of the function specified for "Entry Point."



9. Click in the "Debug" window to debug a program by one step.



10. Variable values can be checked and changed in the "Variables" window on the bottom right of the screen.
Check that 'sRet,' the return value of the CCPU function, is '0' (normal value) in this step.
(a) Run the programs to the line indicated with the red arrow '→' by the step execution in step 9.
(b) Check that the value of sRet is '0' (normal value) in the [Variables] tab.



Repeat steps 9 and 10 to debug the whole created program.

Point

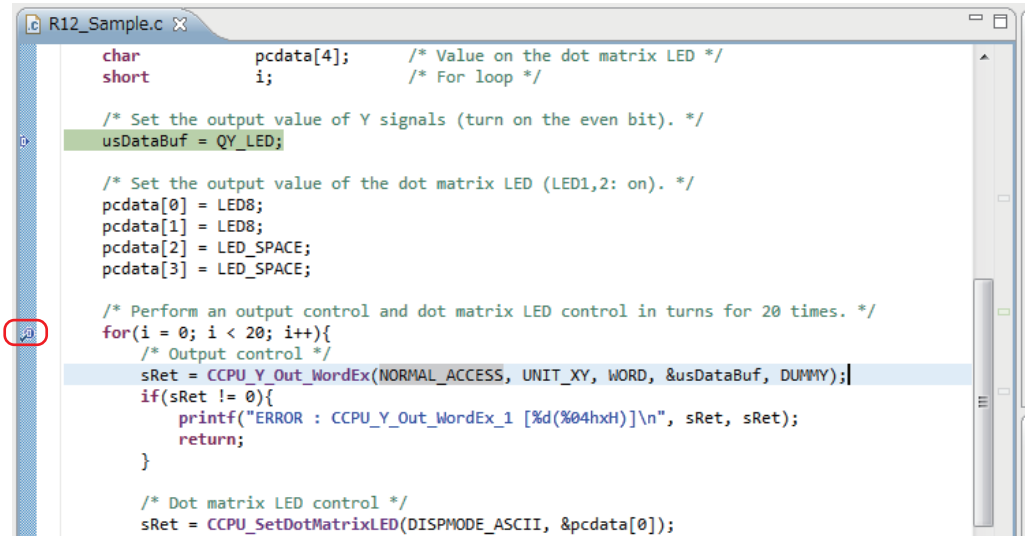
When the return value of the C Controller module dedicated function is other than '0,' refer to the following manual and troubleshoot the symptom.
MELSEC iQ-R C Controller Module Programming Manual

■Debugging using a breakpoint

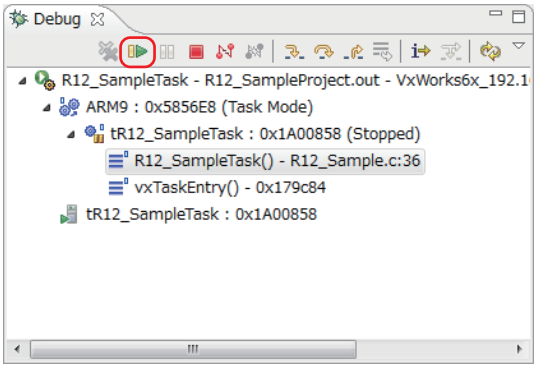
As well as debugging in units of one step described in step 9, debugging using a breakpoint specified in a program is available.

Operating procedure

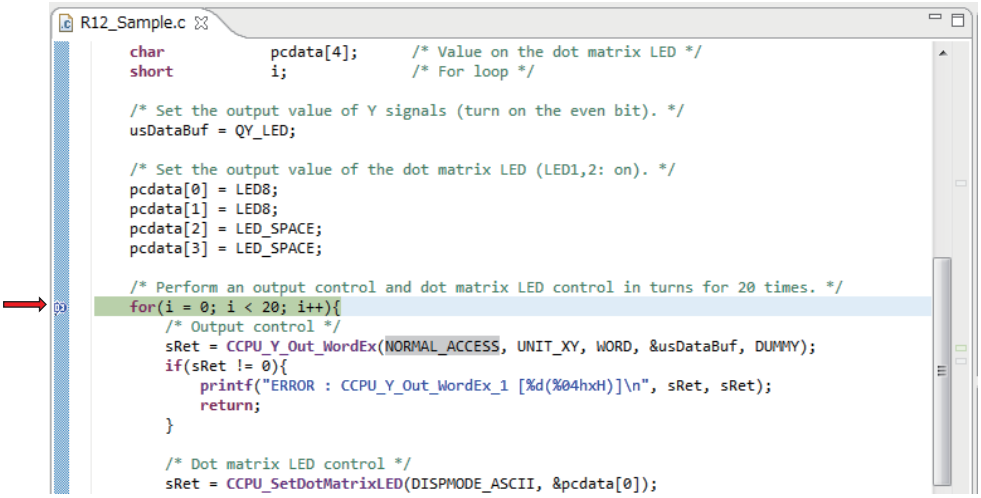
1. Double-click the left edge of the source file window and insert a breakpoint.



2. Click .



The program runs to the position where the breakpoint is specified.

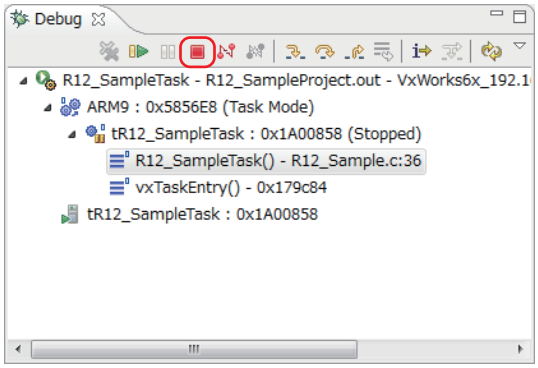


Point

The descriptions of icons are as follows:

- : Step Into (Steps into the called function and stops at the first line of the function.)
- : Step Over (Executes the current line of the function and then stops at the next line of the function.)
- : Continues execution of the current function until it returns to its caller.
- : Executes a program.
- : Stops program execution.
- : Ends debugging.

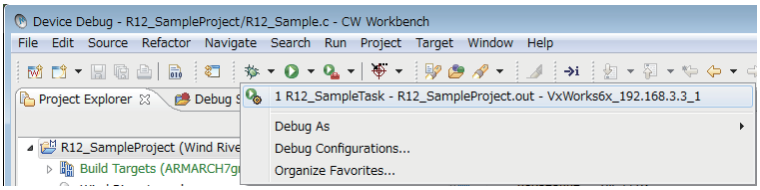
3. Click in the "Debug" window to end the debugging.



Point

To start debugging again, click the [▼] button on the right side of on the toolbar, and select the created debug configuration at the top of the pop-up menu.

By the operations above, the steps 1 to 8 in 'Page 296 Debugging the user program' can be skipped.



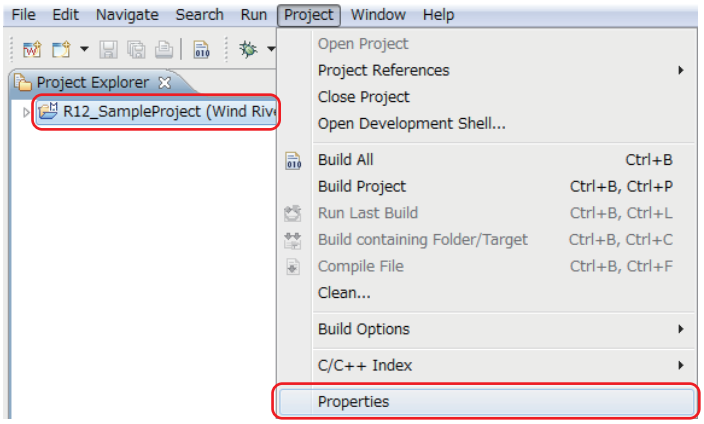
Registering the execution module

Build the created program for operation, and store the program on the C Controller module.

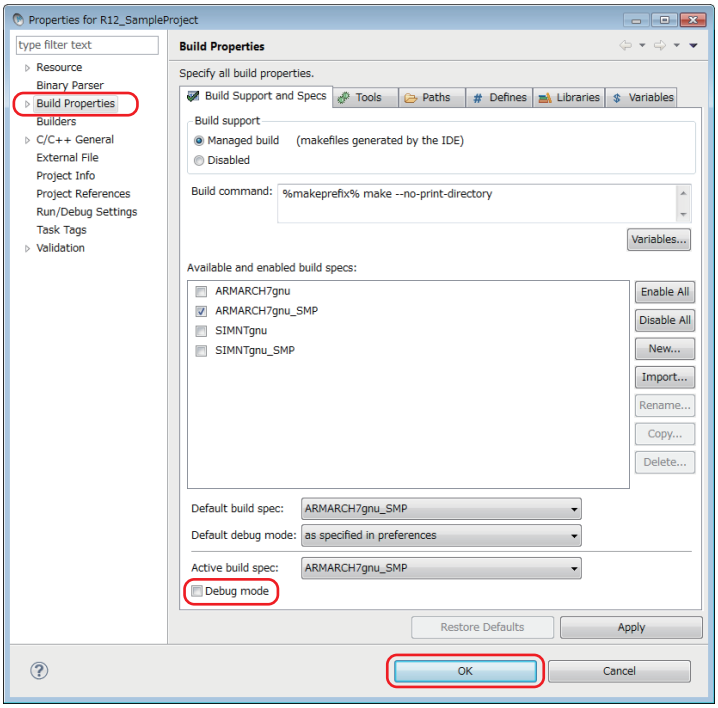
■Building the user program

Operating procedure

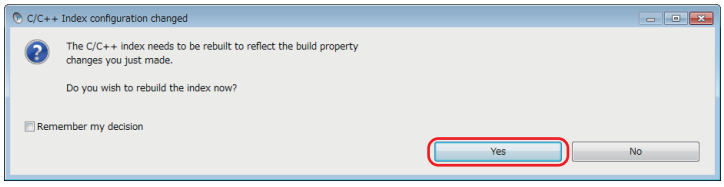
1. Select the created project in the "Project Explorer" window, and select [Project] ⇨ [Properties].



2. Select "Build Properties" from the tree to the left in the screen, unselect the "Debug mode" checkbox, and click the [OK] button.



3. If the following message appears after clicking the [OK] button, click the [Yes] button.



4. Build the program by following the procedure in 'Page 290 Generating an execution module from the user program.'

■Storing the user program

Operating procedure

1. Start Explorer, and enter the following address in the address bar.

"ftp://192.168.3.3/0"

2. Copy the created user program "R12_SampleProject.out" to the program memory '0' in the C Controller module by a drag and drop operation.

The user program created in this example is stored in the following:

C:\WindRiver\workspace\R12_SampleProject\ARMARCH7gnu_SMP\R12_SampleProject\Debug

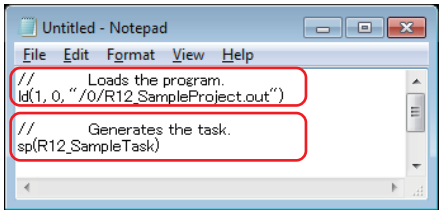
■Creating and storing a script file

Create a script file that automatically downloads the execution module at the start of the C Controller module.

Operating procedure

1. Open a text file and describe a script file that downloads a user program and generates a task, as shown below.

(1)
(2)



(1) The "R12_SampleProject.out" file is loaded from the program memory '0.'

(2) The "R12_SampleTask" function is generated under the default task name (t1).

2. Name the file as "STARTUP.CMD" and save the file.

3. Copy the created script file to the program memory in the C Controller module.

"ftp://192.168.3.3/0"

The script file has been created and stored.

Point

A user program and a script file can be stored on the SD memory card as well.
When a script file is stored in both the program memory and the SD memory card, one in the SD memory card is started by priority.

23 CHECKING OPERATIONS

Run the program registered in a C Controller module to check its operations.
The status of the C Controller module can be changed by the RESET/STOP/RUN switch on the front of the module.
Change the status of the RESET/STOP/RUN switch as follows, depending on the purpose.

- RUN: State where output (Y) from a user program and writing to the buffer memory are permitted
- STOP: State where output (Y) from a user program and writing to the buffer memory are prohibited
- RESET: Resets the module.

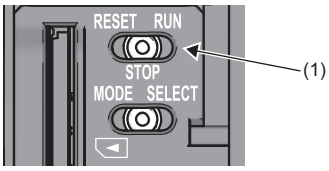
Point

- The C Controller module executes program operation regardless of the switch status (RUN/STOP).
- For details on the RUN/STOP/RESET switch, refer to the following:
  (☞ Page 37 RESET/STOP/RUN switch)

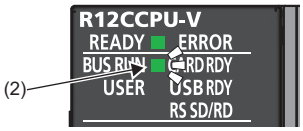
23.1 Enabling Outputs (Y) from a User Program

Operating procedure

1. Set the RESET/STOP/RUN switch (1) on the front of a C Controller module to the RUN position.



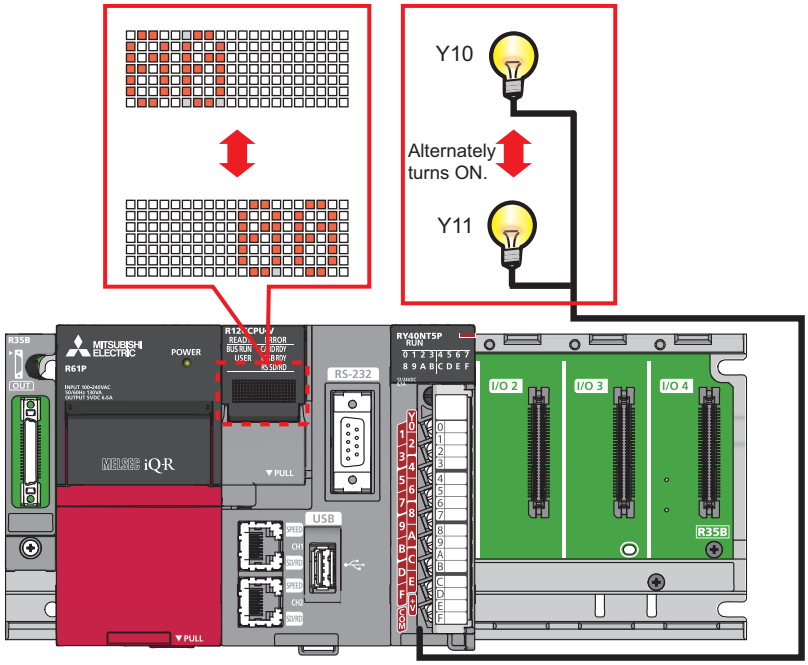
2. When the BUS RUN LED (2) lights in green, the program is running normally.



23.2 Checking Operations with the Dot Matrix LED and Lamps

The dot matrix LED on the front of a C Controller module and output lamps operate as follows:

1. The display of the dot matrix LED on the front of the C Controller module switches alternately ten times.
2. Synchronizing with the dot matrix LED, the output lamps Y10 and Y11 alternately turn ON.



3. To check the operations again, reset the C Controller module.

MEMO

PART 9 MAINTENANCE AND INSPECTION

This part consists of the following chapter.

24 MAINTENANCE AND INSPECTION

24 MAINTENANCE AND INSPECTION

This chapter explains the maintenance and inspection.

24.1 Daily Inspection

This section shows items that should be inspected daily.

Item	Inspection item		Inspection method	Acceptance criteria	Corrective action
1	Mounting status of a base unit	Looseness	Check that mounting screws are not loose and the cover is not dislocated.	The base unit is securely fixed.	Retighten the screws.
2	Mounting status of a module	Looseness	Check that the module is mounted and the module fixing hook is fixed securely on the base unit.	The base unit is securely fixed.	Fix the module fixing hook securely on the base unit.
3	Connection status	Terminal screw looseness	Check for the terminal screw looseness.	The terminal screws are not loose.	Retighten the terminal screws.
		Clearance between the solderless terminals	Check the clearance between the solderless terminals.	The proper clearance has been provided between the solderless terminals.	Provide the proper clearance between the solderless terminals.
		Connector looseness	Check for the cable connector looseness.	The cable connector is not loose.	Connect the connector without looseness.
4	LED status	POWER LED (Power supply module)	Check the lighting status.	The LED is ON. (Failure if it is OFF)	☞ Page 316 TROUBLESHOOTING BY SYMPTOM
		READY LED	Check the lighting status in the BUS RUN state.	The LED is ON. (Failure if it is OFF)	
		ERROR LED	Check the lighting status.	The LED is OFF. (Failure if it is ON or flashing)	
		Dot matrix LED	Check the lighting status.	The LED turns ON and OFF.	
		I/O indicator LED (I/O module)	Check the lighting status.	I/O signal is ON: The LED is ON. I/O signal is OFF: The LED is OFF.	🕮 User's manual for each module

24.2 Periodic Inspection

This section shows items that should be inspected once or twice every six months to a year.
Also, check these items when the equipment has been relocated or modified, or wiring layout has been changed.

Item	Inspection item		Inspection method	Acceptance criteria	Corrective action
1	Ambient environment	Ambient temperature	Measure the temperature by using a thermometer.	0 to 55℃	Create the environment that satisfies the acceptance criteria.
		Ambient humidity	Measure the humidity by using a hygrometer.	5 to 95 %RH	
		Atmosphere	Measure corrosive gases.	No corrosive gases.	
2	Power supply voltage check		Measure a voltage between the 100-240 VAC terminals and/or between the 24 VDC terminals.	85 to 264 VAC	Change the power supply.
				15.6 to 31.2 VDC	
3	Mounting status	Looseness and backlash	Touch the module to check for the looseness and rattling.	The module must be mounted securely.	Fix the module with screws. If the module is loose, retighten the screws.
		Attachment of dirt and foreign material	Check visually.	No dirt or foreign material is attached.	Remove any dirt or foreign material.
4	Connection status	Terminal screw looseness	Check for the terminal screw looseness.	The terminal screws are not loose.	Retighten the terminal screws.
		Clearance between the solderless terminals	Check the clearance between the solderless terminals.	The proper clearance has been provided between the solderless terminals.	Provide the proper clearance between the solderless terminals.
		Connector looseness	Check for the cable connector looseness.	The cable connector is not loose.	Connect the connector without looseness.

MEMO

PART 10 TROUBLESHOOTING

This part consists of the following chapters.

25 TROUBLESHOOTING PROCEDURE

26 TROUBLESHOOTING BY SYMPTOM

27 ERROR CODE LIST

28 EVENT LIST

25 TROUBLESHOOTING PROCEDURE

1. Check the LED on the power supply module. (☞ Page 316 POWER LED on a Power Supply Module Turns OFF)
2. Check the LED on the C Controller module. (☞ Page 317 READY LED on a C Controller Module Does not Turn ON (Green), Page 318 READY LED on a C Controller Module Keeps Flashing)
3. Check the LED on each I/O module or intelligent function module. (📖 User's manual for respective modules)
4. Connect CW Configurator, and start the system monitor. The module which causes error can be checked. (☞ Page 313 System monitor)
5. Select the module causing the error, and start the module diagnostics. The cause and corrective actions can be checked. (☞ Page 314 Module diagnostics)
6. If the cause cannot be identified by performing the module diagnostics, check the operation or error logs with CW Configurator to identify the causes. (☞ Page 315 Event history)
7. If the cause still cannot be identified by the step 1 to 6, perform the troubleshooting by symptom. (📖 User's manual for respective modules)

25.1 Checking with CW Configurator

Use CW Configurator to check the errors occurred logs and to identify the cause of the error. Detailed information, cause of error, and corrective actions can be checked.

CW Configurator has the following functions which support troubleshooting.

Function	Description
System monitor	To display the module configuration, detailed information for each module, and an error status
Module diagnostics	To diagnose the operating status of each module (module information, existence of an error, or error logs, etc.)
Event history	To display the event information including error occurred on each module, performed operations, and error on the network

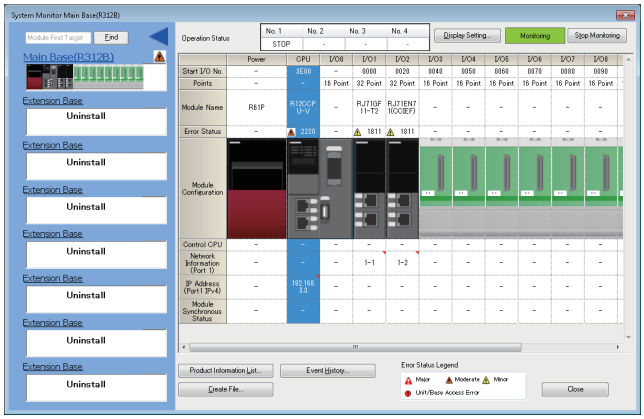
For details on each function of CW Configurator, refer to the following:

CW Configurator Operating Manual

System monitor

The system monitor displays the module configuration, detailed information of each module, and error status.

[Diagnostics] ⇨ [System Monitor]



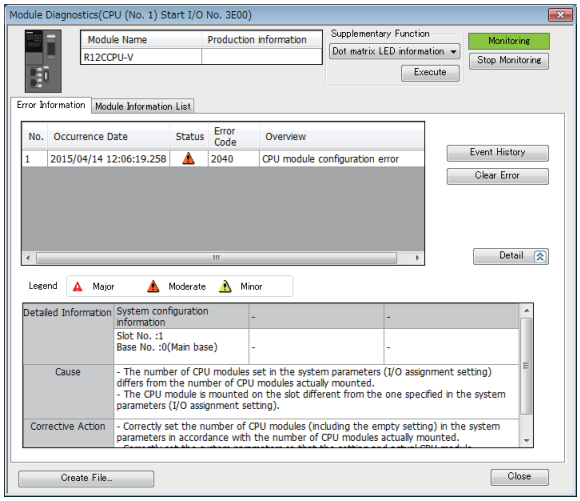
Module diagnostics

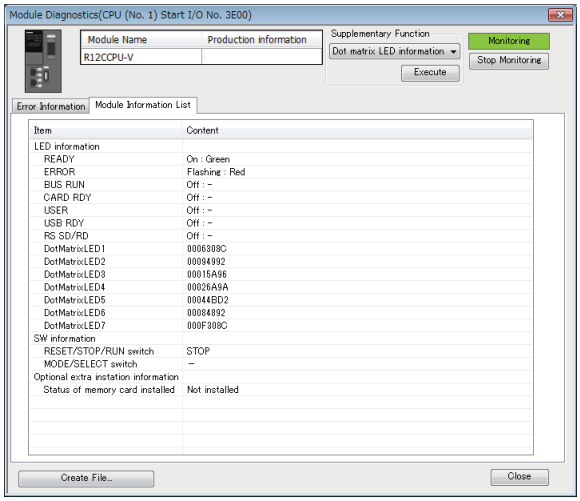
The module diagnostics diagnoses the operating status of each module (module status, existence of an error, or error logs, etc.).

It displays an error occurred, detailed status, causes, and corrective actions, so information necessary for troubleshooting can be checked. In addition, the error part on the parameter can be identified by selecting the error and click the [Error Jump] button.

In the [Module Information List] tab, the current LED information or switch information of the corresponding module can be checked.

[Diagnostics] ⇨ [System Monitor] ⇨ double-click an arbitrary module



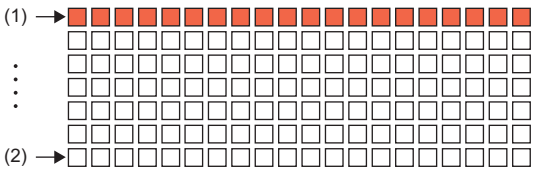


■Dot matrix LED information

"DotMatrixLED1" to "DotMatrixLED7" under "LED information" indicates the dot matrix LED display status.

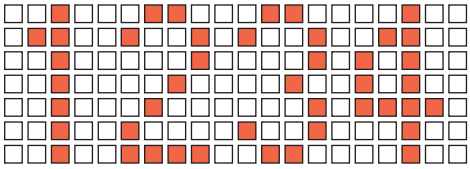
The contents are displayed in hexadecimal notation, and the lower 20 bits of each data are valid.

Item	Description
DotMatrixLED1	Lighting bit data of the 1st row from the top
DotMatrixLED2	Lighting bit data of the 2nd row from the top
DotMatrixLED3	Lighting bit data of the 3rd row from the top
DotMatrixLED4	Lighting bit data of the 4th row from the top
DotMatrixLED5	Lighting bit data of the 5th row from the top
DotMatrixLED6	Lighting bit data of the 6th row from the top
DotMatrixLED7	Lighting bit data of the 7th row from the top



(1) 1st row
(2) 7th row

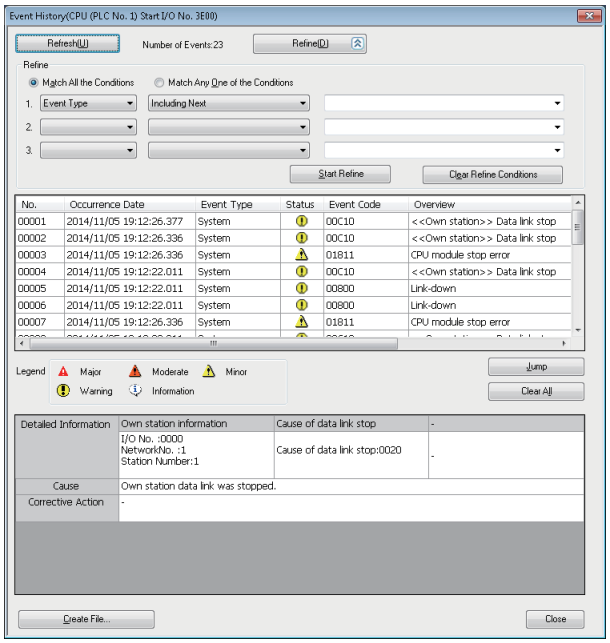
• Display example

Dot matrix LED display	LED information		Lower 20 bits (Binary)				
	DotMatrixLED1	00023184	0010	0011	0001	1000	0100
	DotMatrixLED2	00064A4C	0110	0100	1010	0100	1100
	DotMatrixLED3	00020854	0010	0000	1000	0101	0100
	DotMatrixLED4	00021094	0010	0001	0000	1001	0100
	DotMatrixLED5	0002205E	0010	0010	0000	0101	1110
	DotMatrixLED6	00024244	0010	0100	0010	0100	0100
	DotMatrixLED7	00027984	0010	0111	1001	1000	0100

Event history

The event history displays the event information including error occurred on each module, performed operations, and error on the network. Since the information collected before turning the power OFF or resetting can be saved, it is used to identify the cause of abnormalities from past operations or error occurrence tendency. The displayed information can be saved in CSV file format.

[Diagnostics] ⇨ [System Monitor] ⇨ [Event History]



Point

Use the event history function in the following cases.

- For checking error occurrence status for all modules, and clarifying the causes of malfunction which occurred on the facilities/equipment
- For checking when and from where a parameter of a user program has been changed
- For checking if any unauthorized access from a third party has been tried

For details on the function or information collected with the event history function, refer to the following:

☞ Page 116 Event history function

26 TROUBLESHOOTING BY SYMPTOM

If any of the functions of a C Controller module does not operate properly, perform troubleshooting by checking the following items. If the ERROR LED is ON or flashing, clear the error using an engineering tool.

26.1 POWER LED on a Power Supply Module Turns OFF

Check the following items.

Check item	Corrective action
The power supply module is not mounted on the base unit properly.	Re-mount the power supply module, and power it ON again.
The READY LED on the C Controller module is ON.	An error occurred in the power supply module. Replace the power supply module.
Power supply voltage is not appropriate.	Supply the appropriate power voltage. (MELSEC iQ-R Module Configuration Manual)
The rated output of the power supply module exceeds the internal current consumption within the entire system.	Review the system configuration so that the internal current consumption does not exceed the rated output current of the module. (MELSEC iQ-R Module Configuration Manual)
The POWER LED turns ON when the power is restored to the system after all the modules, except the power supply module, have been removed.	An error occurred in a module other than power supply module. Cycle the power, adding modules to the system one by one. An error occurred in the last module mounted immediately before the POWER LED turns OFF. Replace the corresponding module.

If the POWER LED does not turn ON even after the corrective actions listed above are taken, the possible cause is a hardware failure of the power supply module. Please contact your local Mitsubishi Electric sales office or representative.

26.2 READY LED on a C Controller Module Does not Turn ON (Green)

Check the following items.

Check item	Corrective action
The C Controller module is not mounted on the main base unit properly.	Re-mount the C Controller 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.)	If the READY LED turns ON, an error has occurred in the power supply module(s) before the replacement. 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.)	If the READY LED does not turn ON, an error has occurred in a module other than the power supply module. Cycle the power, adding modules to the system one by one. An error occurred in the last module mounted immediately before the READY LED turns OFF. Replace the corresponding module.
The ERROR LED is ON.	System watchdog timer error occurred. • Check if user tasks with higher priority occupy the system. • Take corrective actions against the occurrence of any system watchdog timer error (CPU error code: 3E41H).
The ERROR LED is flashing.	The hardware failure has occurred. Take corrective actions according to the event registered in the event history.
The module is started up in the hardware diagnostics mode.	Power OFF the module to terminate the hardware diagnostics mode and power ON the module again.
The power is turned OFF or the module is reset while accessing files.	Turn the power OFF and ON, or reset the module again to start the C Controller module.
The firmware update is being executed.	The firmware update is not complete. Turn the power OFF and ON, or reset the C Controller module while an SD memory card in which a firmware update file is stored is inserted, and complete the firmware update. For details on the firmware update function, refer to the following: MELSEC iQ-R Module Configuration Manual

If the READY LED does not turn ON even after the corrective actions listed above are taken, the possible cause is a hardware failure of the power supply module. Please contact your local Mitsubishi Electric sales office or representative.

26.3 READY LED on a C Controller Module Keeps Flashing

The READY LED flashing status indicates that the script file (STARTUP.CMD) is being executed.
After taking corrective actions corresponding to the "Check item" below, correct the script file and the user program that is activated from the script file.

Check item	Corrective action
The script file is stored in the program memory.	• Store the unprocessed script file in an SD memory card, and turn the power ON again. • Initialize the C Controller module.
The script file in the program memory cannot be overwritten.	• Store the unprocessed script file in an SD memory card, and turn the power ON again. • Secure a free space in the program memory. • Initialize the C Controller module.

26.4 Ethernet Communication between a C Controller Module and a Personal Computer cannot be Established

Issue PING command from the personal computer to the C Controller module and check the response.

When the PING command response is incorrect

Check the following items.

Check item	Corrective action
The IP address segments of the personal computer and the C Controller module are different.	Set the same segment to both the personal computer and C Controller module. If it is relayed on another segment LAN via the gateway, contact the network administrator of the connected LAN.
The duplicate IP addresses with the personal computer and the C Controller module exist in the connected LAN.	Contact the LAN network administrator to eliminate the IP address duplication.
As a result of the C Controller module replacement, PING does not respond normally.	Reset all devices on the network to which the C Controller module is connected.
The IP address outside range is specified.	Check the following items and specify a right IP address. • The IP address starts with a number from 1 to 233, excluding 127. • No space is included in the IP address.
The system in a multiple CPU system is faulty. (A self-diagnostics error such as CPU module configuration error and parameter error occurred.)	• Reset the multiple CPU system and restart it. • Connect CW Configurator after the READY LED turned ON, and write the correct parameters.
The network is in the overloaded conditions.	Disconnect the other Ethernet devices, and establish the connection only with the C Controller module.

When the PING command is correct

Perform the troubleshooting in accordance with the following symptoms.

■CW Configurator connection fails

Check item	Corrective action
The Ethernet cable is not connected to the Ethernet port.	Connect the Ethernet cable.
The connection destination of CW Configurator is set to a different route.	Set the connection destination of CW Configurator, via Ethernet.
Multiple target devices with different IP addresses are connected.	Add connection in "External Device Configuration."
A target device (with the same IP address) is connected to a C Controller module simultaneously using multiple MELSOFT connections (TCP/IP).	Specify the same Ethernet board adapter for the connection destination. If there are any MELSOFT products for which Ethernet board adapter cannot be specified as a connection destination, set "Not Specified" for the adapter of all the MELSOFT products.

■CW Workbench connection fails

Check item	Corrective action
The Ethernet cable is not connected to the Ethernet port.	Connect the Ethernet cable.
"Target Server Options" of CW Workbench is not set properly.	Set "Target Server Options" of CW Workbench properly.

■Telnet connection fails

Check item	Corrective action
The Ethernet cable is not connected to the Ethernet port.	Connect the Ethernet cable.
User name and password is not sure.	Initialize the C Controller module.
A message "Sorry, session limit reached." appears.	Terminate the Telnet connection from another personal computer and take any of the following actions: • Reconnect after the Telnet connection timeout time has elapsed. • Reset the C Controller system.

■FTP connection fails

Check item	Corrective action
The Ethernet cable is not connected to the Ethernet port.	Connect the Ethernet cable.
User name and password is not sure.	Initialize the C Controller module.
There are 11 or more FTP connections to one C Controller module.	Make an adjustment so that the number of FTP connections becomes 10 or less.
An FTP connection can be established normally from the Windows® command prompt.	Change the FTP client tool to be used when an FTP connection can be established from the command prompt.

26.5 File Access Fails

Check the following items.

Check item	Corrective action
The CARD RDY LED is OFF when the read/write target is an SD memory card.	Insert an SD memory card or re-insert an SD memory card.
The USB RDY LED is OFF when the read/write target is a USB Mass Storage Class-compliant device.	Connect a USB Mass Storage Class-compliant device or reconnect a USB Mass Storage Class-compliant device.
The SD memory card is write-protected when write target is the SD memory card.	Cancel the write protection of the SD memory card. (Refer to the manual for the SD memory card used.)
There is no free space in the write target drive.	• Secure a free space in the write target drive. • Change the write target to another drive.
The user program that uses the write target file is running.	Stop the user program that uses the write target file.
A file system error occurred in an SD memory card or a USB Mass Storage Class-compliant device.	• Restore the file system in the SD memory card or the USB Mass Storage Class-compliant device. • Format the SD memory card with CW Configurator. Use an applicable SD memory card. • Format the USB Mass Storage Class-compliant device with the supported file system by using a personal computer. (☞ Page 77 Drive names and file systems) • Do not partition the drive of a USB Mass Storage Class-compliant device.
The network is in the overloaded conditions.	Disconnect the other Ethernet devices, and establish the connection only with the C Controller module.
Writing a large amount of data to an SD memory card fails.	• Reduce data size to be written to the SD memory card. • Retry after a while.

26.6 Connection with Peripherals Fails

Check the following items.

Check item	Corrective action
The system in a multiple CPU system is faulty.	Reset the multiple CPU system and restart it.
The network is in the overloaded conditions.	Disconnect the other Ethernet devices, and establish the connection only with the C Controller module.

26.7 File Read from a C Controller Module Fails

Check the following items.

Check item	Corrective action
Check the transfer mode of FTP.	Change the transfer mode of FTP to an appropriate mode.

26.8 An Error Occurs during User Program Execution

Check the following items.

Check item	Corrective action
An error occurs in executable file (*.out) loading.	• Set "Build Spec" in compiling to "ARMARCH7gnu_SMP." • Download the files with all symbols required for files to load first. • Add "-mlong-calls" to the build option, For details on how to add the option, refer to "Considerations for creating a user program" in the following manual. (☞ Page 44 Considerations for creating a user program)
Event(s) is registered in the event history.	Take appropriate actions in accordance with details of the registered event.
An error occurs in a C Controller module dedicated function, MELSEC data link function, data analysis function, statistical analysis function, or motion module dedicated class.	Take appropriate actions in accordance with the error code at the time of function execution.
An error occurs in VxWorks standard API functions.	Refer to the manual for VxWorks. If the error persists, consult Wind River Systems, Inc.
Stack size of the task that runs the user program is insufficient.	Increase the task stack size.
The pointer used in the user program refers to an invalid address.	Make correction to make the pointer refer to a valid address.
The memory area specified to the size is not reserved.	Secure the memory area.
The VX_FP_TASK option is not specified for the task that performs floating-point operations.	Specify the VX_FP_TASK option for the task that performs floating-point operations.
A VxWorks message is displayed when an error occurs.	Consult the Wind River Systems, Inc.

26.9 The Serial Communication cannot be Established

Check the following items.

Check item	Corrective action
The option settings already configured have been changed.	Acquire the serial communication option currently set, and change the option settings. For details on how to acquire and change the serial communication option, refer to the manuals for VxWorks of the following version. • VxWorks Version 6.9

26.10 Communication with an Ethernet Device cannot be Established

Check the following item:

Check item	Corrective action
The Ethernet device is communicating with the C Controller module by specifying the following port number: • 1 to 1023, or 61440 to 65534	Since the port number 1 to 1023 is the number for reserved in general (WELL KNOWN PORT NUMBERS), and the port number 61440 to 65534 is the number to be used for other communication devices, these numbers cannot be used. Use the following number for the port number: • 1024 to 4999 or 5010 to 61439
The Ethernet port (CH1 or CH2) is in the overloaded conditions.	Reduce the load on the Ethernet port (CH1 or CH2) of a C Controller module.

26.11 Drive Name of an SD Memory Card is not Displayed Properly

Check the following item:

Check item	Corrective action
Drive name of the SD memory card is not displayed properly.	Format the SD memory card with CW Configurator. Use an SD memory card supported by C Controller module. (☞ Page 57 Available memory cards)

26.12 Drive Name of a USB Mass Storage Class-compliant Device is not Displayed Properly

Check the following item:

Check item	Corrective action
Drive name of the USB Mass Storage Class-compliant device is not displayed properly.	• Format the USB Mass Storage Class-compliant device with the supported file system by using a personal computer. (☞ Page 77 Drive names and file systems) • Do not partition the drive of a USB Mass Storage Class-compliant device.

26.13 CC-Link IE Field Network Basic Functions cannot be Used

Check the following items:

Check item	Corrective action
Disconnection occurs in CC-Link IE Field Network Basic.	Reduce the load on the Ethernet port (CH1 or CH2) of a C Controller module.
The Ethernet port (CH1 or CH2) is in the overloaded conditions.	

If the problem persists, refer to the following:

📖CC-Link IE Field Network Basic Reference Manual

26.14 Module Labels and Public Labels are not Refreshed Properly

Check the following items:

Check item	Corrective action
A major error occurs in the C Controller module.	Clear the cause of the major error and reset the C Controller module. After the reset, check that the module is in the normal status.
A major error occurs in the motion module.	Clear the cause of the major error and reset the motion module. After the reset, check that the module is in the normal status.
The label is not set to be published in parameters of the motion module.	Publish the label using the motion control setting function.
"At the Execution Time of END Instruction" is not set for "Refresh Timing" in "Refresh Setting."	Set "At the Execution Time of END Instruction" for "Refresh Timing."
When accessing labels with a C Controller module dedicated function, a header file, which is output before the latest public labels are applied, is used.	Apply the latest public labels first, and then output a header file. Access the labels using the output header file.
When accessing labels with a C Controller module dedicated function, the module label ID of an argument is executed with '0' specified.	Apply the latest public labels first, and then output a header file. Access the labels using the output header file. For the operation when '0' is specified for the module label ID, refer to the following: 📖MELSEC iQ-R C Controller Module Programming Manual
There is no consistency between parameters written in the C Controller module and the motion module.	Apply the latest public labels first, and then output a header file. After that, write parameters to the C Controller module and motion module.

26.15 Consistency with a Written Project is Unknown

Check the following item:

Check item	Corrective action
Module label ID mismatch error occurs even though project verification matches.	Check if a project that has a consistency between the module label ID written in a C Controller module and the module label ID of a header file output by CW Configurator is saved in a personal computer.*1 ■When a project with matching the module label IDs exist Use the project. In addition, use a header file that is output from the project for user programs. ■When a project with matching the module label IDs does not exist After reading parameters from a motion module, perform the procedure in 'Applying public labels.' (☞ Page 133 Usage) In addition, save the written project.

*1 The module label ID written to a C Controller module can be checked with 'Module label ID' (SD1596 to 1597). (☞ Page 362 Special Register List)

26.16 A Delay Occurs in the Link Scan Time of CC-Link IE Field Network Basic

Check the following items:

Check item	Corrective action
A delay occurs in the link scan time of CC-Link IE Field Network Basic.	Change the system timer mode from the normal mode to the high-precision mode with the C Controller module dedicated function (CCPU_SetSysTimerMode).
The network is in the overloaded conditions.	Disconnect the other Ethernet devices, and establish the connection only with the C Controller module.

27 ERROR CODE LIST

A C Controller module stores the error code in the special register (SD) when an error is detected by the self-diagnostics function. If an error occurs when the data communication is requested from an engineering tool, intelligent function module, or network system, the C Controller module returns the corresponding error code to the request source. The error details and cause can be identified by checking the error code.

- Module diagnostics of CW Configurator (☞ Page 313 Checking with CW Configurator)
- Check with special registers (SD0 (latest self-diagnostics error code), SD10 to SD25 (self-diagnostics error code))
- Check with the C Controller module dedicated function (CCPU_GetErrInfo) (MELSEC iQ-R C Controller Module Programming Manual)
- Check with the dot matrix LED*1(☞ Page 314 Dot matrix LED information)

*1 Select "ERROR" in the operation selection mode.

27.1 Error Code System

All error codes are given in hexadecimal format (4 digits) (16-bit unsigned integer). The type of error includes the error, which is detected through the self-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 CPU module
	5000H to 5FFFH	Error in slice I/O module
	7000H to 7FFFH	Error in serial communication module
	9000H to 9FFFH	Error in FA controller module
	B000H to BFFFH	Error in CC-Link module
	C000H to CFFFH	Error in Ethernet module
	D000H to DFFFH	Error in CC-Link IE Field Network module
	E000H to EFFFH	Error in CC-Link IE Controller Network module
	F000H to FFFFH	Error in MELSECNET/H network modules or MELSECNET/10 network module

Detailed information

When an error is detected by the self-diagnostics function, the detailed information of the error cause is stored all together. The detailed information of each error code can be checked with CW Configurator. 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 the error code.) Information of the latest error code can be checked with the special register (SD) as well. (☞ Page 362 Special Register List)

Detailed information	Item	Description
Detailed information 1	Drive and file information	Information on drive names and file names
	Parameter information	Information on parameters such as the parameter storage destination and the parameter type
	System configuration information	Information on system configurations such as the I/O number and the power supply number
	Frequency information	Information on the number of times such as the number of writes to memory
	Time information	Information on the time
	Failure information	Information on failures
	Detailed event code information	Detailed code registered with the C Controller module dedicated function (CCPU_RegistEventLog)
	Script position information	Script step number
Detailed information 2	Drive and file information	Information on drive names and file names
	Annunciator information	Information on annunciators
	Parameter information	Information on parameters such as the parameter storage destination and the parameter type
	System configuration information	Information on system configurations such as the I/O number and the power supply number
	Detailed event log information	Detailed information registered with the C Controller module dedicated function (CCPU_RegistEventLog)

27.2 Operation When an Error Occurs

There are two types of errors: stop errors and continuation errors.

Stop error

In a C Controller module, if a stop error occurs, the output (Y) from a user program and writing to buffer memory are disabled. The communication with a CPU module can be performed 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 112 Operation setting when an error is detected) Besides, if the stop error occurs in a multiple CPU system configuration, the stop error target CPU module (all CPU modules or only the relevant CPU module) can be set in the parameter. (☞ Page 231 Stop setting)

Continuation error

If a continuation error occurs, the CPU module retains the operation and continues processing.

27.3 Error Clear

Only continuation errors can be cleared. (☞ Page 114 Error clear function)

27.4 Error Code List

The following table shows the error codes detected by the self-diagnostics function.

Self-diagnostics error codes for CPU module (1000H to 3FFFH)

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
1000H	Power interruption	• A momentary power failure has occurred. • The power supply has been shut OFF.	Check the power supply status.	—	Always
1080H	ROM write count error	The number of writes to the FlashROM (data memory, program memory, and system memory) exceeded 100000 times. (Number of writes > 100000)	Replace the CPU module.	Frequency information	At power ON, RESET, or write
1100H	Memory card access error	Data cannot be written to the memory card because the write protect switch is in the locked position.	Set the write protect switch of the memory card in the unlocked position.	—	Always
1120H	SNTP clock setting error	Time setting has failed when the programmable controller is powered ON or reset.	• Check if the setting of time setting function is correct. • Check if the specified SNTP server is operating normally and there is no failure on the network accessing the SNTP server computer.	—	At power ON or RESET
112EH	Connection establishment failed	• A connection could not be established in the open processing. • Connection requests more than the number of set connections are sent to the same port from multiple target devices with different IP addresses.	• Check the operation of the target device. • Check if the open processing has been performed in the target device. • Review the port number of the module, IP address/port number of the target device, opening method, and the number of connections. • Check the number of connections in the external device configuration setting of module parameters. • When the firewall is set in the target device, check if the access is permitted. • Check if the Ethernet cable is disconnected.	—	Always

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
1165H	UDP/IP send failed	Data was not sent correctly with UDP/IP.	• Check the settings for connection with the target device. • Check the operation of the target device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub.	—	Always
1166H	TCP/IP send failed	Data was not sent correctly with TCP/IP.	• Check the settings for connection with the target device. • Check the operation of the target device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub.	—	Always
1200H	Module moderate error	A moderate error has been notified from an intelligent function module.	Check the detailed information (system configuration information) of the error by performing module diagnostics using CW Configurator and remove the error.	System configuration information	Always
1210H	Module moderate error	An inter-module synchronization signal error has been notified from the intelligent function module.	Check the detailed information (system configuration information) of the error by performing module diagnostics using CW Configurator and remove the error.	System configuration information	Always
1220H	Another CPU module moderate error	A moderate error has been notified from another CPU module.	• Check the detailed information (system configuration information) of the error by performing module diagnostics using CW Configurator, identify the error module, and eliminate the error cause. • Check the mounting status and reset status of another CPU module.	System configuration information	Always
1240H	Inter-module synchronization processing error	• The execution interval of the inter-module synchronous interrupt program exceeded the setting value. • The inter-module synchronous interrupt program (I44) has not been completed within the inter-module synchronization cycle.	Check the detailed information (time information) of the error by performing module diagnostics using CW Configurator, and take any of the following corrective actions. • Review the process content of the inter-module synchronous interrupt program to complete the process within the interval specified in "Fixed Scan Interval Setting." • Modify the value specified to "Fixed Scan Interval Setting" to an appropriate value.	Time information	At interrupt occurrence
1241H	Inter-module synchronization processing error	• The execution interval of the inter-module synchronous interrupt program exceeded the setting value. • A cycle, which was not executed, has been detected by the inter-module synchronous interrupt program (I44).	Review the interrupt prohibited section and program of which interrupt priority is high in order that the inter-module synchronous interrupt program can be executed.	—	At interrupt occurrence
1260H	Multiple CPU synchronization processing error	• The execution interval of the multiple CPU synchronous interrupt program exceeded the setting value. • The multiple CPU synchronous interrupt program (I45) has not been completed within the cycle of the multiple CPU fixed cycle communication.	Check the detailed information (time information) of the error by performing module diagnostics using CW Configurator, and take any of the following corrective actions. • Review the process content of the multiple CPU synchronous interrupt program in order to complete the process within the interval specified in "Fixed Scan Interval Setting." • Modify the value specified to "Fixed Scan Interval Setting" to an appropriate value.	Time information	At interrupt occurrence

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
1262H	Multiple CPU synchronization processing error	• The execution interval of the multiple CPU synchronous interrupt program exceeded the setting value. • The multiple CPU synchronous interrupt program (I45) has not been completed within an execution section of the program.	Check the detailed information (time information) of the error by performing module diagnostics using CW Configurator, and take any of the following corrective actions. • Review the process content of the multiple CPU synchronous interrupt program in order to complete the process within the program execution section . • Modify the value specified to "Fixed Scan Interval Setting" to an appropriate value.	Time information	At interrupt occurrence
1830H	Receive queue full	Number of reception requests of transient transmission exceeded upper limit of simultaneously processable requests.	Lower the transient transmission usage frequency, and then perform again.	—	Always
1831H	Receive processing error	Transient reception failed.	Lower the transient transmission usage frequency, and then perform again.	System configuration information	Always
1832H	Transient data error	Transient transmission cannot be performed because the number of processing are too large.	Review the number of transient transmission execution.	—	Always
1840H	Memory Card Error	An error was detected in the memory card.	• Replace the memory card. • Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	—	Always
1843H	Internal battery failure	The battery voltage inside the CPU module decreased to the defined value or lower.	Replace the battery. Please consult your local Mitsubishi representative to replace the battery.	—	Always
1846H	Refresh cycle exceeded	The refresh time exceeded the set refresh cycle. • The time set for the refresh cycle is too short. • Too many refresh points have been set. • Tasks with the higher CPU usage rate is in operation in such case as overloaded Ethernet communication. • Debugging or restarting of the C Controller module has been performed with CW Workbench connected online. • Command was executed from Shell for debugging.	• Lengthen the time set for the refresh cycle. • Reduce the set number of refresh points. • Lower the CPU usage rate of tasks with the higher rate, or make them inactivated. • Restart the C Controller module with CW Workbench not connected online. • Review the command executed from Shell.	Time information	Always
2000H	Module configuration error	The module type set in the I/O Assignment Setting of the System Parameter differs from that of the module actually mounted.	Reconfigure the I/O Assignment Setting in the System Parameter in accordance with the intelligent function module or CPU module actually mounted.	System configuration information	At power ON or RESET
2001H	Module configuration error	The I/O numbers set to I/O Assignment Setting in the System Parameter are overlapping between other modules.	Reconfigure the I/O Assignment Setting in the System Parameter in accordance with the intelligent function module or I/O module actually mounted.	System configuration information	At power ON or RESET
2002H	Module configuration error	The number of points assigned to the intelligent function module in the I/O Assignment Setting of the System Parameter is smaller than that of the module actually mounted.	Reconfigure the I/O Assignment Setting in the System Parameter in accordance with the intelligent function module actually mounted.	System configuration information	At power ON or RESET

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
2004H	Module configuration error	• A total of nine or more CC-Link IE Controller Network modules* and MELSECNET/H network modules are mounted in the entire system. • Five or more MELSECNET/H network modules are mounted in the entire system. * The CC-Link IE built-in Ethernet interface module is included if the module is used as a CC-Link IE Controller Network module.	• Reduce the total number of CC-Link IE Controller Network modules* and MELSECNET/H network modules to eight or less in the entire system. • Reduce the number of MELSECNET/H network modules to four or less in the entire system. * The CC-Link IE built-in Ethernet interface module is included if the module is used as a CC-Link IE Controller Network module.	System configuration information	At power ON or RESET
2005H	Module configuration error	• Two or more interrupt modules, QI60 of which interrupt pointer setting has not been configured are mounted. • The interrupt pointer number is duplicated between an interrupt module, QI60 of which interrupt pointer setting has not been configured and an module with the interrupt pointer setting configured.	• Mount only one QI60. • Configure the interrupt pointer setting for QI60. • Review the interrupt pointer setting.	System configuration information	At power ON or RESET
2006H	Module configuration error	A module is mounted on the 65th slot or later.	Remove the module mounted on the 65th slot or later.	System configuration information	At power ON or RESET
2007H	Module configuration error	A module is mounted on the slot of which number is later than that specified in the I/O assignment setting.	Remove the module mounted on the slot of which number is later than that specified slot in the I/O Assignment Setting.	System configuration information	At power ON or RESET
2008H	Module configuration error	• A module is mounted on the I/O points 4096 points or later. • The module is mounted over the boundary of I/O points (4096 points).	• Remove the module mounted on 4096 points or later. • Replace the module mounted on the last slot to the one that does not exceed 4096 points .	System configuration information	At power ON or RESET
2009H	Module configuration error	There is no response from the I/O module or intelligent function module accessed.	• Review the I/O Assignment Setting in the System Parameter. • Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the I/O module or intelligent function module. Please consult your local Mitsubishi representative.	System configuration information	Always
2020H	Module configuration error	An unsupported module is mounted, or a module which does not support the network type (module name) set to I/O Assignment Setting in the System Parameter is mounted.	• If an unsupported module is mounted, remove the module. • Check if the network type (module name) set to I/O Assignment Setting in the System Parameter is supported. • If all the modules or the network type (module name) are supported, the possible cause is a hardware failure of the CPU module, base unit, I/O module, or intelligent function module. Please consult your local Mitsubishi representative.	System configuration information	At power ON or RESET
2021H	Module configuration error	In a multiple CPU system, the control CPU of the MELSEC-Q series intelligent function module which does not support a multiple CPU system is set to other than CPU No.1.	• Replace the MELSEC-Q series intelligent function module with the one (function version B) supporting a multiple CPU system. • Change the control CPU of the MELSEC-Q series intelligent function module which does not support a multiple CPU system to CPU No.1.	System configuration information	At power ON or RESET
2040H	CPU module configuration error	• The number of CPU modules set in the I/O Assignment Setting of the System Parameter differs from the number of CPU modules actually mounted. • The CPU module is mounted on the slot different from the one specified to the I/O Assignment Setting.	• Set the number of CPU modules (including the empty setting) in the I/O Assignment Setting of the System Parameter correctly in accordance with the number of CPU modules actually mounted. • Set the I/O Assignment Setting in the parameter and actual CPU module mounting status correctly.	System configuration information	At power ON or RESET

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
2041H	CPU module configuration error	• The CPU module is not mounted on the slot that is set for the CPU module in the I/O Assignment Setting of the System Parameter. • The CPU module is mounted on the slot which was set as the empty setting in the I/O Assignment Setting of the System Parameter. • An I/O module or intelligent function module is mounted between the CPU modules.	• Set the number of CPU modules (including the empty setting) in the I/O Assignment Setting of the System Parameter correctly in accordance with the number of CPU modules actually mounted. • Remove the I/O module or intelligent function module mounted between the CPU modules.	System configuration information	At power ON or RESET
2043H	CPU module configuration error	The CPU module is mounted on the inapplicable slot.	• Mount the CPU module on the applicable slot (CPU slot or I/O slot 0 to 6). • Remove the CPU module from the inapplicable slot.	System configuration information	At power ON or RESET
2044H	CPU module configuration error	The host CPU number set in the I/O Assignment Setting differs from the one determined by the mounting position of the CPU module.	Re-set the host CPU number in the system parameters in accordance with the mounting position of the CPU module.	System configuration information	At power ON or RESET
2050H	CPU module configuration error	An unsupported CPU module is mounted.	Remove the unsupported CPU module. If all the CPU modules are supported, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power ON or RESET
2060H	Base unit configuration error	Eight or more extension base units are connected.	Reduce the number of extension base units to seven or less.	System configuration information	At power ON or RESET
2061H	Base unit configuration error	Any of the following base units is connected: QA1S3□B, QA1S5□B/QA1S6□B, QA6□B, QA6ADP+A5□B/A6□B, QA1S6ADP+A1S5□B/A1S6□B.	Remove QA1S3□B, QA1S5□B/QA1S6□B, QA6□B, QA6ADP+A5□B/A6□B□B, and QA1S6ADP+A1S5□B/A1S6□B.	System configuration information	At power ON or RESET
2063H	Base unit configuration error	Extension base unit level settings are duplicated.	Review the level setting of the extension base units.	System configuration information	At power ON or RESET
2070H	Base unit configuration error	• An unsupported base unit is connected. • A GOT is connected to the MELSEC-Q series extension base unit with a bus.	• Disconnect the unsupported base unit. If all base units are supported, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative. • Disconnect the GOT connected to the MELSEC-Q series extension base unit.	System configuration information	At power ON or RESET
2080H	Inter-module synchronization configuration error	An inter-module synchronization signal error was detected.	The possible cause is a hardware failure of the CPU module, base unit, I/O module, or intelligent function module. Please consult your local Mitsubishi representative.	System configuration information	At power ON or RESET
20E0H	Module unrecognized	• A module that cannot be recognized by the module is mounted. • In a multiple CPU system, the module cannot be recognized because the control CPU setting of the system parameter setting differs from that of another CPU module.	• Mount the applicable modules. • Review the system parameters in the CPU No.2 or later and match the number with those of the smallest numbered CPU module. • The possible cause is a hardware failure of the I/O module or intelligent function module. Please consult your local Mitsubishi representative.	System configuration information	Always
2120H	Memory Card Error	The memory card was removed without the card being disabled.	Disable the memory card, and then remove it.	Drive and file information	Always
2121H	Memory Card Error	An error was detected in the memory card.	Re-insert the memory card. If the same error code is displayed again, the possible cause is a hardware failure of the memory card. Replace the memory card.	Drive and file information	Always

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
2180H	Invalid file	An invalid file has been detected.	Check the detailed information (drive and file information) of the error by performing module diagnostics using CW Configurator, check the file name and write the specified file. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive and file information	At power ON or RESET
21A0H	File specification error	• The file specified in the CPU parameters doe not exist. • The file specified in the boot file setting of the memory card parameters does not exist in the memory card.	Check the detailed information (drive and file information) of the error by performing module diagnostics using CW Configurator, check the file name and write the specified file. If the same error code is displayed again, the possible cause is a hardware failure of the device/label memory in the CPU module or the memory card. Please consult your local Mitsubishi representative.	Drive and file information Parameter information	At instruction execution, interrupt occurrence, power ON, or RESET
21A1H	File specification error	The file specified in parameter cannot be created.	• Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and correct the name and size of the file corresponding to the displayed number (parameter number). • Check the detailed information (drive and file information) of the error by performing module diagnostics using CW Configurator, and take any of the following action. (1) Format the corresponding drive. (2) Delete unnecessary files on the corresponding drive to increase free space. (3) Unlock the corresponding drive if it is locked.	Drive and file information Parameter information	At write, power ON, or RESET
2200H	Parameter error	The system parameter file and CPU parameter file do not exist.	Write the system parameter and CPU parameter.	Parameter information	At power ON or RESET
2220H	Parameter error	• The parameter setting is corrupted. • Parameters not supported by the firmware version of the CPU module are written.	• Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and write the displayed parameter to the module. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, memory card, I/O module, or intelligent function module. Please consult your local Mitsubishi representative. • Check the firmware version of the CPU module, use a compatible product, and then write the parameters again.	Parameter information	At power ON or RESET
2221H	Parameter error	The set value is out of range.	Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and review the parameter setting corresponding to the displayed number (parameter number). If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, memory card, I/O module, or intelligent function module. Please consult your local Mitsubishi representative.	Parameter information	At power ON, RESET, fixed cycle processing execution, instruction execution, or module access

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
2222H	Parameter error	• Use of the function that is not supported by the module is enabled. • The module is non-operational. • Parameters that the firmware version of the module does not support were written. • The inter-module synchronous master settings are not set even though a local station is selected for the inter-module synchronization target module.	• Remove the unsupported module if any. • Use functions supported by the module. • Check the status of the module. • Check the firmware version of the module and use a supported product. • Check the inter-module synchronous master settings. • Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and review the parameter setting corresponding to the displayed number (parameter number). If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, memory card, I/O module, or intelligent function module. Please consult your local Mitsubishi representative.	Parameter information	At power ON or RESET
2224H	Parameter error	A memory area cannot be ensured.	Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, increase the capacity of the area corresponding to the displayed parameter with error jump. (If the capacity of the area cannot be increased, decrease the capacity of other areas.)	Parameter information	At write, power ON, or RESET
2225H	Parameter error	• The model type (CPU module name) set with CW Configurator differs from that of the CPU module actually mounted. • The operation set in the memory card parameters cannot be performed. (The boot function cannot be executed.)	• Correct the model type (CPU module name) set to the CW Configurator project in accordance with the CPU module actually mounted. • Delete the memory card parameters. • Remove the memory card so that the operation set in the memory card parameters will not be performed. (Do not execute the boot operation.)	Parameter information	At write, power ON, or RESET
2240H	Parameter error (module)	In a multiple CPU system, an I/O module or intelligent function module controlled by another CPU module is specified in the module parameter.	Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and review the parameter setting corresponding to the displayed value (parameter number). If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, I/O module, or intelligent function module. Please consult your local Mitsubishi representative.	Parameter information	At power ON or RESET
2241H	Parameter error (module)	• The I/O numbers set in the system parameters differ from those of the module actually mounted. • The target module is not mounted on the slot where the system parameters and module parameters are set. • The module type set in parameter differs from that of the module actually mounted.	• Check if the system configuration displayed on the System Monitor window of CW Configurator matches the actual system configuration. • Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and review the parameter setting corresponding to the displayed value (parameter number). If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module, I/O module, or intelligent function module. Please consult your local Mitsubishi representative.	Parameter information	At power ON, RESET, fixed cycle processing execution, instruction execution, or module access
2242H	Parameter error (module)	A module parameter error was detected in the intelligent function module.	Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and check the module corresponding to the displayed number (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. Please consult your local Mitsubishi representative.	System configuration information	At power ON or RESET

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
2260H	Parameter error (network)	Network numbers are duplicated.	Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and review the parameter setting corresponding to the displayed number (parameter 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. Please consult your local Mitsubishi representative.	Parameter information	At power ON or RESET
2261H	Parameter error (network)	Different network types (CC IE Control extended mode/normal mode) are set between the control station and the normal station.	Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and review the parameter setting corresponding to the displayed number (parameter 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. Please consult your local Mitsubishi representative.	Parameter information	At power ON or RESET
2262H	Parameter error (network)	• When the station number of the MELSECNET/H module is '0,' the parameters in the network between the programmable controllers are set. • The station type set in the module parameters differs that of the module actually mounted.	• Review the station number of the MELSECNET/H module. • Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and review the parameter setting corresponding to the displayed number (parameter 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. Please consult your local Mitsubishi representative.	Parameter information	At power ON or RESET
2263H	Parameter error (network)	Even though the CC-Link IE module or MELSECNET/H module is mounted, the different module is set in the I/O assignment setting of the system parameter, or module parameters of the CC-Link IE module or MELSECNET/H module are not set.	Set the system parameters and module parameters. If the same error code is displayed again, the possible cause is a hardware failure of the data memory in the CPU module or intelligent function module. Please consult your local Mitsubishi representative.	Parameter information	At power ON or RESET
2280H	Parameter error (refresh)	• The refresh setting is not set correctly. (Data was refreshed exceeding the file register capacity.) • The refresh settings (number of points) are different from those of another CPU module.	• Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and review the parameter setting corresponding to the number (parameter number) so that the data is refreshed within the specified device range. (Take any of the following actions: Increase the number of file register points (capacity), create a file register file having a capacity for all of the target data to be refreshed, or reduce the refresh device range.) • Rewrite the refresh settings (number of points) in the CPU parameters for all the CPU modules. (Use the same number of points in the refresh settings for all the CPU modules.)	Parameter information	At power ON, RESET, fixed cycle processing execution, instruction execution, or module access
2281H	Parameter error (refresh)	A device that cannot be used as a refresh device is specified.	• Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and review the parameter setting corresponding to the displayed number (parameter number). • Rewrite the CPU parameters and module parameters simultaneously. (These parameters must be the same.)	Parameter information	At power ON or RESET
2282H	Parameter error (refresh)	The number of specified refresh points is invalid.	Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and review the parameter setting corresponding to the displayed number (parameter number).	Parameter information	At power ON or RESET

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
2283H	Parameter error (refresh)	The total number of refresh points exceeded the maximum limit.	Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and review the parameter setting corresponding to the displayed number (parameter number).	Parameter information	At power ON or RESET
22E0H	Parameter verification error	• In a multiple CPU system, the system parameter settings differ from those of another CPU module. • In a multiple CPU system, the system parameters in the host CPU module are overwritten, and the settings differ from those of another CPU module.	• Check the detailed information (parameter information) of the error by performing module diagnostics using CW Configurator, and correct the system parameters corresponding to the displayed number (parameter number) in the CPU No.2 or later. (Set the same system parameters between the CPU modules used for the module synchronization setting and fixed scan communication setting.) • Rewrite the system parameters of all the modules. (The system parameter settings should be same in all the CPUs.)	Parameter information System configuration information	At write, power ON, or RESET
2400H	Module Verify Error	• The module information at powered ON differs from the information of modules actually mounted. • The I/O module or intelligent function module is not mounted properly or was removed during operation.	• Check the detailed information (system configuration information) of the error by performing module diagnostics using CW Configurator, and check the module corresponding to the displayed number (slot number). • Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the module where the error was detected. Please consult your local Mitsubishi representative.	System configuration information	Always
2401H	Module Verify Error	A CPU module, I/O module, or intelligent function module was mounted on the base unit during operation.	• Check the detailed information (system configuration information) of the error by performing module diagnostics using CW Configurator, and check the module corresponding to the displayed number (slot number). • Do not mount a CPU module, I/O module, nor intelligent function module on an empty slot during operation . • Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the module where the error was detected. Please consult your local Mitsubishi representative.	System configuration information	Always
2420H	Fuse blown error	The output module with a blown fuse has been detected.	• Check the FUSE LED on the output module, and replace the one with the LED ON. • Check the detailed information (system configuration information) of the error by performing module diagnostics using CW Configurator, and replace the output module corresponding to the displayed number (slot number).	System configuration information	Always
2440H	Module major error	• In a multiple CPU system, the control CPU setting in the system parameters is different from that of another CPU. • An error was detected in the I/O module or intelligent function module during the initial processing.	• Review the system parameters in the CPU No.2 or later and match the number with those of the smallest numbered CPU module. • Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the module where the error was detected. Please consult your local Mitsubishi representative.	System configuration information	At power ON or RESET
2441H	Module major error	An error was detected in the intelligent function module when a function was executed.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the module where the error was detected. Please consult your local Mitsubishi representative.	—	—

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
2442H	Module major error	An error was detected in the intelligent function module when a function was executed.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the module where the error was detected. Please consult your local Mitsubishi representative.	—	At module access
2443H	Module major error	An error was detected in the I/O module or intelligent function module.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the module where the error was detected. Please consult your local Mitsubishi representative.	System configuration information	At module access
2450H	Module major error	• A major error was notified from an intelligent function module. • The I/O module or intelligent function module is not mounted properly or was removed during operation.	• Take measures to reduce noise. • Check the connection status of the extension cable. • Check the detailed information (system configuration information) of the error by performing module diagnostics using CW Configurator, and check the module corresponding to the displayed number (slot number). • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the module where the error was detected. Please consult your local Mitsubishi representative.	System configuration information	Always
2460H	Another CPU module major error	An error was detected in another CPU module during the initial processing.	• Take measures to reduce noise. • Reset the CPU module. 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 was detected. Please consult your local Mitsubishi representative.	System configuration information	At power ON or RESET
2461H	Another CPU module major error	An error was detected in another CPU module when a function was executed.	• Take measures to reduce noise. • Reset the CPU module. 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 was detected. Please consult your local Mitsubishi representative.	—	—
2462H	Another CPU module major error	An error was detected in another CPU module when a function was executed.	• Take measures to reduce noise. • Reset the CPU module. 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 was detected. Please consult your local Mitsubishi representative.	—	At fixed cycle processing execution
2463H	Another CPU module major error	An error was detected in another CPU module.	• Take measures to reduce noise. • Reset the CPU module. 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 was detected. Please consult your local Mitsubishi representative.	System configuration information	At power ON or RESET
2470H	Another CPU module major error	A major error was notified from another CPU module.	• Take measures to reduce noise. • Reset the CPU module. 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 was detected. Please consult your local Mitsubishi representative.	System configuration information	Always
2480H	Multiple CPU error	• In a multiple CPU system, an error was detected in the CPU modules where "Stop" is set in the Operation Mode Setting. • Any CPU module other than CPU No.1 is mounted in the inapplicable slot. (An error occurs in the CPU module mounted in the inapplicable slot.)	• Check the detailed information (system configuration information) of the error by performing module diagnostics using CW Configurator and remove the error. • Remove the CPU module from the inapplicable slot.	System configuration information	Always

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
2481H	Multiple CPU error	In a multiple CPU system, any of the CPUs other than CPU No.1 was disconnected from the base unit during operation. Or, any CPU module other than CPU No.1 was reset.	Check the mounting status and reset status of the CPU modules other than CPU No.1.	System configuration information	Always
24C0H	System bus error	An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, I/O module, intelligent function module, base unit, or extension cable. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C1H	System bus error	An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, I/O module, intelligent function module, base unit, or extension cable. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C2H	System bus error	• The I/O module or intelligent function module is not mounted properly or was removed during operation. • An error was detected on the system bus.	• Check the detailed information (system configuration information) of the error by performing module diagnostics using CW Configurator, and check the module corresponding to the displayed number (slot number). • Check the connection status of the extension cable. • Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, I/O module, intelligent function module, base unit, or extension cable. Please consult your local Mitsubishi representative.	System configuration information	At module access, always
24C3H	System bus error	An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, I/O module, intelligent function module, base unit, or extension cable. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C4H	System bus error	An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the I/O module, intelligent function module, base unit, or extension cable. Please consult your local Mitsubishi representative.	System configuration information	At module access
24C5H	System bus error	An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the I/O module, intelligent function module, base unit, or extension cable. Please consult your local Mitsubishi representative.	—	At module access
24C6H	System bus error	An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module. 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.	—	At module access
24C8H	System bus error	An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the I/O module, intelligent function module, or extension cable. Please consult your local Mitsubishi representative.	—	At power ON or RESET

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
24D0H	System bus error	• In the extension level setting of the MELSEC-Q series extension base unit, the duplicated level setting with other extension base units is detected. • An error was detected on the system bus.	• Review the level setting of the MELSEC-Q series extension base unit. • Check the connection status of the extension cable. • Check if the 10 m mark is printed on the base unit connected when using the ten-meter extended cables (RC100B). (MELSEC iQ-R Module Configuration Manual) • Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, base unit, or extension cable. Please consult your local Mitsubishi representative.	System configuration information	At module access, always
24E0H	System bus error	An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	System configuration information	Always
2520H	Invalid interrupt	Even though an interrupt was requested, there is no interrupt factor.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, I/O module, intelligent function module, or base unit. Please consult your local Mitsubishi representative.	System configuration information	At interrupt occurrence
2521H	Invalid interrupt	Even though an interrupt was requested, there is no interrupt factor.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, I/O module, intelligent function module, or base unit. Please consult your local Mitsubishi representative.	—	At interrupt occurrence
2522H	Invalid interrupt	An interrupt request from the module with no interrupt setting has been detected.	• Review the interrupt setting in the module parameter. • Take measures so that no interrupt is requested from the module with no interrupt setting exists in the module parameter. • Review the interrupt setting in the buffer memory of the Intelligent function module.	System configuration information	At interrupt occurrence
2610H	Inter-module synchronization signal error	• An execution interval error of the synchronous interrupt program was detected. • An inter-module synchronization error was detected.	• Check the module set as the inter-module synchronization master. • Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, I/O module, intelligent function module, base unit, or extension cable. Please consult your local Mitsubishi representative.	—	Always
2611H	Inter-module synchronization signal error	• An inter-module synchronization error was detected. • An error of the module set as the inter-module synchronization master has been detected.	• Check the module set as the inter-module synchronization master. • Correct the inter-module synchronization master settings. • Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module, I/O module, intelligent function module, base unit, or extension cable. Please consult your local Mitsubishi representative.	System configuration information	At power ON, RESET, or fixed cycle processing execution
2630H	Multiple CPU synchronization signal error	• An execution interval error of the synchronous interrupt program was detected. • A multiple CPU synchronization error was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	—	Always

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
2631H	Multiple CPU synchronization signal error	A multiple CPU synchronization error was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module or base unit. Please consult your local Mitsubishi representative.	System configuration information	At power ON, RESET, or fixed cycle processing execution
3000H	Boot function execution error	The boot setting in the memory card parameters is incorrect.	Review the boot setting in the memory card parameters.	Drive and file information	At power ON or RESET
3001H	Boot function execution error	When the boot function was executed, the file format processing failed.	Reset the CPU module and perform the boot function again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Drive and file information	At power ON or RESET
3004H	Boot function execution error	When the boot function was executed, the CPU built-in memory capacity was exceeded.	• Review the boot setting. • Delete unnecessary files in the CPU built-in memory.	Drive and file information	At power ON or RESET
3042H	User WDT error	The user watchdog timer controlled by the system has detected an error because the C Controller module dedicated function (CCPU_ResetWDT) was not executed within the user watchdog timer setting time. Or, an error occurred in the user program. • The time set for the user watchdog timer is too short. • Tasks with the higher CPU usage rate is in operation. • A program causing an error in the memory or stack was executed. • Debugging has been performed with CW Workbench connected online. • Command was executed from Shell for debugging. • The following functions that increase the CPU usage rate of the system task are used. (1) Mounting/unmounting the memory card (2) Ethernet communication (3) NFS server communication	• Reset the CPU module. • Lower the CPU usage rate of tasks with the higher rate, or make them inactivated. • Review the user program. • Restart the C Controller module with CW Workbench not connected online. • Review the command executed from Shell. • Set the user watchdog timer setting time longer enough with consideration for the CPU usage rate of the system task. If an error still occurs, check the mounted modules, and replace a defective module.	—	Always
3044H	Program fault	The command in the script file cannot be executed. (The syntax is incorrect, no command exists, or the script file is corrupted.)	• Check that the syntax of the script file is not incorrect and a command exists. • Check if the power has been turned OFF, the CPU module has been reset, or the SD memory card has been removed while accessing the file. • If the script file is stored in the SD memory card, diagnose and recover the SD memory card. If the same error occurs again, the possible cause is a hardware failure of the SD memory card. Replace it with another one. • If the script file is stored in the program memory, initialize the program memory.	Script position information	At power ON or RESET
3C00H	Hardware failure	A hardware failure was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. • Check the event history, if major errors occur in another CPU module, I/O module, or intelligent function module, take measures according to the error codes.	Failure information	Always
3C01H	Hardware failure	A hardware failure was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
3C02H	Hardware failure	• A hardware failure was detected. • An invalid argument has been specified with the C Controller module dedicated functions for ISR.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative. • Review the argument of the C Controller module dedicated functions for ISR.	Failure information	At power ON, RESET, fixed cycle processing execution, or interrupt occurrence
3C03H	Hardware failure	A hardware failure was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always, at power-ON, RESET, or interrupt occurrence
3C0FH	Hardware failure	A hardware failure was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3C10H	Hardware failure	A hardware failure was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At power ON or RESET
3C11H	Hardware failure	A hardware failure was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At fixed cycle processing execution or instruction execution
3C12H	Hardware failure	• The wave error was detected in the power supply module. • A hardware failure of the power supply module, CPU module, base unit, or extension cable was detected.	• Check the wave of the voltage applied to the power supply module. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the power supply module, CPU module, base unit, or extension cable. Please consult your local Mitsubishi representative.	Failure information	Always
3C13H	Hardware failure	A hardware failure was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3C14H	Hardware failure	A hardware failure was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3C20H	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At power ON or RESET
3C21H	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At fixed cycle processing execution, power ON, or RESET
3C22H	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At fixed cycle processing execution, power ON, or RESET

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
3C2FH	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3C30H	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Format the memory. Write all files to the CPU module, and then reset it. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At instruction execution
3C31H	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Format the memory. Write all files to the CPU module, and then reset it. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3C32H	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3E40H	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3E41H	System WDT error	The system watch dog timer controlled by the system has detected an error. Or, an error occurred in the system software. • The time set for the system watchdog timer monitoring time is too short. • Tasks with the higher CPU usage rate is in operation. • A program causing an error in the memory or stack was executed. • The operation that increase the CPU usage rate of the system task (writing parameter) was performed. • The station on which the station-based block data assurance setting is enabled on the network has been accessed when the stop error occurred. • The CPU module is running out of control or is broken down. (Malfunction due to noise or hardware failure)	• Reset the CPU module. • Lower the CPU usage rate of tasks with the higher rate, or make them inactivated. • Review the user program. • Set the system watchdog timer monitoring time longer enough with consideration for the CPU usage rate of the system task. • Review the user program to prevent the station on which the station-based block data assurance setting is enabled from being accessed when the stop error occurred. • Take measures to reduce noise. • Check that the C Controller module is properly mounted on the base unit, and that the ambient environment is within the range of the general specifications. If the same error code is still displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3E48H to 3E4AH	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Format the memory. Write all files to the CPU module, and then reset it. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	Always
3E50H	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Format the memory. Write all files to the CPU module, and then reset it. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At power ON or RESET
3E51H	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At power ON or RESET

Error code	Error name	Error details and cause	Corrective action	Detailed information	Diagnostic timing
3E52H	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At power ON or RESET
3E53H	Hardware failure	A hardware failure was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At power ON or RESET
3E54H	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At power ON or RESET
3E55H	Hardware failure	A hardware failure was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At power ON or RESET
3E56H	Hardware failure	A hardware failure was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At power ON or RESET
3E57H	Memory error	An error was detected in the memory.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At power ON or RESET
3E58H	Hardware failure	A hardware failure was detected.	• Take measures to reduce noise. • Reset the CPU module. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.	Failure information	At power ON or RESET

Error code related to data communication (4000H to 4FFFH)

The table below lists the error codes detected by other than the self-diagnostics function of CPU module.

The error codes are not stored in SD0 since these errors are not detected by the self-diagnostics function.

Error code	Error name	Error details and cause	Corrective action
4001H	Common error	An unsupported request was executed. (The request was issued to CPU module which does not support the request.)	• Check the command data of SLMP/MC protocol. • Check the CPU module name selected with an engineering tool. • Check the target CPU module name.
4002H	Common error	An unsupported request was executed.	• Check the command data of SLMP/MC protocol. • Check the CPU module name selected with an engineering tool. • Execute the request again. If the same error code is displayed again, the possible cause is a hardware failure of the CPU module. Please consult your local Mitsubishi representative.
4003H	Common error	Command for which a global request cannot be performed was executed.	Check the command data of SLMP/MC protocol.
4004H	Common error	A request was issued to the system file.	Check the command data of SLMP/MC protocol.
4005H	Common error	The volume of data to be handled by the specified request is too large.	Check the command data of SLMP/MC protocol.
4008H	Common error	The CPU module is BUSY. (The buffer is not empty.)	Execute the request again after arbitrary period of time has elapsed.
4010H	CPU module operation related error	Because the CPU module is in the RUN state, the request cannot be executed.	Execute the request after placing the CPU module into the STOP state.
4013H	CPU module operation related error	Because the CPU module is not in the STOP state, the request cannot be executed.	Execute the request after placing the CPU module into the STOP state.
4021H	File related error	The specified drive (memory) does not exist, or has an error.	• Check the status of the specified drive (memory). • Back up the data in the CPU module, and then initialize the memory.
4022H	File related error	The file with the specified file name or file number does not exist.	Check the specified file name and file number.
4023H	File related error	The file name and file number of the specified file do not match.	Delete the file, and then create a file again.
4024H	File related error	The specified file cannot be handled.	Do not access the specified file.
4025H	File related error	The specified file is currently processing a request from another engineering tool.	Forcibly execute the request. Or, execute it again once the processing from another engineering tool has been completed.
4026H	File related error	Specifying the file password set to the target drive (memory) is required.	Specify the file password set to the target drive (memory), and access it.
4027H	File related error	The specified range exceeded the file size.	Check the specified range, and access within the range.
4028H	File related error	The same file already exists.	Forcibly execute the request. Or, change the file name and execute the request again.
4029H	File related error	The specified file capacity cannot be reserved.	Review the specified file capacity, and execute the request again.
402AH	File related error	The specified file has an error.	Back up the data in the CPU module, and then initialize the memory.
402BH	File related error	The request cannot be executed in the specified drive (memory).	Execute the request again after placing the CPU module into the STOP state.
402CH	File related error	The request cannot be executed currently.	Execute it again after a while.
4030H	Device specification error	The specified device name cannot be handled.	Check the specified device name.
4031H	Device specification error	• The specified device number is out of range. • The CPU module does not support the specified device name.	• Check the specified device number. • Check the device assignment of the CPU module. • Check the specified device name.

Error code	Error name	Error details and cause	Corrective action
4032H	Device specification error	The specified device modification is incorrect. Or, the device names (TS, TC, SS, SC, CS, or CC) which cannot be used for random read/random write (in word units)/monitor registration/ monitor command of SLMP/MC protocol have been specified.	• Check the specified device modification method. • Check the specified device name.
4033H	Device specification error	Data cannot be written to the specified device since it is for system use.	Do not write data to the specified device. Or, do not turn it ON/ OFF.
4040H	Intelligent function module specification error	The request cannot be executed to the specified intelligent function module.	Check whether the specified module is an intelligent function module which has the buffer memory.
4041H	Intelligent function module specification error	The access range exceeded the buffer memory range of the specified intelligent function module.	Check the start address and number of access points, and access within the range that exists in the intelligent function module.
4042H	Intelligent function module specification error	The specified intelligent function module cannot be accessed.	• Check whether the specified intelligent function module operates normally. • Check whether the specified module has hardware failure.
4043H	Intelligent function module specification error	The specified intelligent function module does not exist.	Check the I/O number of the specified intelligent function module.
4044H	Intelligent function module specification error	A bus error occurred during the access to an intelligent function module.	Check whether the specified intelligent function module, other modules, or base unit have hardware failure.
4050H	Protect error	The request cannot be executed because the write protect switch of the SD memory card is ON.	Turn OFF the write protect switch of the SD memory card.
4052H	Protect error	Data cannot be written since the specified file attribute is the read-only.	Do not write data to the specified file. Or, change the file attribute.
4053H	Protect error	An error occurred when writing data to the specified drive (memory).	Check the specified drive (memory). Or, write data again after changing the target drive (memory).
4054H	Protect error	An error occurred when deleting data from the specified drive (memory).	Check the specified drive (memory). Or, delete data again after changing the target drive (memory).
4080H	Other error	Request data error	Check the specified request data.
4082H	Other error	The specified command cannot be executed since it is being executed.	Execute the command again once the request from another engineering tool has been completed.
408BH	Other error	A remote request cannot be executed.	• Execute the request again after placing the CPU module in the state where a remote request can be executed. • For the remote reset operation, set "Remote Reset" to "Enable" with the parameter.
4121H	File related error	The specified drive (memory) or file does not exist.	Check the specified drive (memory) or file, and execute the request again.
4122H	File related error	The specified drive (memory) or file does not exist.	Check the specified drive (memory) or file, and execute the request again.
4123H	File related error	The specified drive (memory) has an error.	Initialize the memory, and restore the drive (memory) to its normal state.
4124H	File related error	The specified drive (memory) has an error.	Initialize the memory, and restore the drive (memory) to its normal state.
4125H	File related error	The specified drive (memory) or file is currently being processed.	Execute it again after a while.
4129H	File related error	The request cannot be executed since the specified drive (memory) is ROM.	Change the target drive (memory), and execute the request again.
412AH	File related error	The request cannot be executed since the specified drive (memory) is ROM.	Change the target drive (memory), and execute the request again.
412BH	File related error	The specified drive (memory) is write-prohibited.	Change the write-protect conditions or drive (memory), and execute the request again.

Error code	Error name	Error details and cause	Corrective action
412DH	File related error	The specified drive (memory) does not have enough free space.	Increase the free space of the drive (memory), and execute the request again.
412EH	File related error	The specified drive (memory) does not have enough free space.	Increase the free space of the drive (memory), and execute the request again.
412FH	File related error	The drive (memory) capacity differs between the copy destination and the copy source.	Check the copy destination and copy source drive (memory), and execute the request again.
4130H	File related error	The drive (memory) type differs between the copy destination and the copy source.	Check the copy destination and copy source drive (memory), and execute the request again.
4131H	File related error	The file name of the copy destination is the same as the one of the copy source.	Check the file name, and execute the request again.
4132H	File related error	The specified number of files does not exist.	Check the specified data, and execute the request again.
4133H	File related error	The specified device (memory) has no free space.	Increase the free space of the drive (memory), and execute the request again.
4134H	File related error	The attribute specification data for a file is incorrect.	Check the specified data, and execute the request again.
4135H	File related error	The date/time data of the engineering tool (personal computer) is out of range.	Check the clock setting of the engineering tool (personal computer), and execute the request again.
4136H	File related error	The specified file already exists.	Check the specified file name, and execute the request again.
4137H	File related error	The specified file is read-only.	Change the conditions for the specified file, and execute the request again.
4138H	File related error	Simultaneously accessible files exceeded the maximum.	Reduce the file operation, and execute the request again.
4139H	File related error	The specified file exceeded the file size range of the file already exists.	Check the size of the specified file, and execute the request again.
413AH	File related error	The specified file exceeded the file size of the file already exists.	Check the size of the specified file, and execute the request again.
413EH	File related error	Operation is disabled for the specified drive (memory).	Change the target drive (memory), and execute the request again.
413FH	File related error	Writing to the file storage area is prohibited for the file.	Change the specified drive (memory), and execute the request again.
414AH	Intelligent function module specification error	Operation was performed to the intelligent function module outside of the control group in a multiple CPU system.	Perform the operation from the control CPU module for the target module.
414CH	Intelligent function module specification error	Inaccessible buffer memory address was specified.	Check the buffer memory address, and execute the request again.
4150H	File related error	An attempt was made to initialize the drive protected by the system.	Do not initialize the target drive (memory) since it cannot be initialized.
4151H	File related error	An attempt was made to delete the file/folder protected by the system.	Do not delete the target file/folder since it cannot be deleted.
41C5H	File related error	• The specified file does not exist. • An attempt was made to write data to a read-only file.	Check the file, and execute the request again.
41DFH	File related error	The specified drive (memory) is write-protected.	Disable the write protection of the specified drive (memory), and execute the request again.
41E4H	File related error	Access to the SD memory card has failed.	• Check whether the SD memory card has been inserted, and access it again. • Replace the SD memory card, and access it again. • Back up the data, and initialize the memory.
41F8H	File related error	The same data is being accessed with another engineering tool.	Check the completion of the following functions, and execute the request again. • The function to write data to the program memory, or the function to transfer data to backup memory is being performed.
41FBH	File related error	The specified file has been operated with the same engineering tool.	Execute the request again once the currently performed operation has been completed.

Error code	Error name	Error details and cause	Corrective action
41FDH	File related error	Data is not written to the data memory.	Write file using the write to PLC function.
41FEH	File related error	• SD memory card is not inserted. • The SD memory card is in the disabled state.	• Insert an SD memory card. • Remove the SD memory card, and insert it again. • Cancel the SD memory card forced disable function.
41FFH	File related error	The type of SD memory card is different.	Check the type of the SD memory card.
4269H	Other error	The remote RUN operation cannot be performed.	Perform the remote RUN operation again after a while.
433CH	Maintenance related error	Clearing error failed. (The error clear function has been performed while an error is being cleared.)	Execute it again after a while. If the same error occurs even when the function is performed again, the possible cause is a hardware failure of the relevant module. Please consult your local Mitsubishi representative.
433DH		The relevant module does not support the error clear function.	Check the target module of the error clear function. (Check the module on which an error occurred.)
4A00H	Network related error	• The specified station cannot be accessed because the routing parameters have not been set to the start source CPU and the relay CPU module. • The control CPU module for the network module to which data is routed has not started for routing via a multiple CPU system. • The CPU module that relays IP packets is not the control CPU module for the CC-Link IE module on the path where IP packets travel.	• Set the routine parameters to the related stations for accessing the specified station. • Retry it after a while. Or, check the startup of the system that relays data, and start communication. • Set the CPU module that relays IP packets to the control CPU module for the CC-Link IE module on the path where IP packet travel.
4A01H	Network related error	• The network with the number set to the routing parameters does not exist. • The specified CPU module cannot be communicated via the network that is not supported by the specified CPU module.	• Check the routing parameters set to the related stations, and correct them. • Perform data communication using the communication route supported by the specified CPU module.
4A02H	Network related error	The specified station cannot be accessed.	• Check whether any error occurred on the network module, or it is offline. • Check whether the settings for the network number and station number are correct.
4A03H	Network related error	A request for network test was issued.	Check the request data of SLMP/MC protocol.
4B00H	Target module related error	• An error occurred on the access destination or the relay station. • The connection destination specification (the I/O number of the requested module) is invalid.	• Check the error occurred on the specified access destination or relay station to the station to be accessed, and take the corrective actions. • Check the connection destination specification (Request destination module I/O No. or PC No.) for the request data of SLMP/MC protocol. • Check the stop error, and take the corrective actions.
4B02H	Target module related error	The request is not the one addressed to the CPU module.	Perform the operation to the module that can perform the specified function.
4B03H	Target module related error	• The specified route is not supported by the version of the specified CPU module. • The communication target CPU module is not mounted.	• Check whether the specified route is supported. • Check the mounting status of the CPU module. • Check the stop error, and take the corrective actions.
4B04H	Target module related error	The connection destination specification (I/O number of the requested module) is not supported.	Invalid value has been set to the start I/O number for the module in "Target settings." Change the start I/O number to the one for the target module, and perform data communication again.

28 EVENT LIST

A C Controller module collects information from each module including errors detected by a module, operations performed for a module, and errors occurred on the network and saves them to the data memory or an SD memory card. (☞ Page 116 Event history function). When an event occurs, its event code and description can be checked using CW Configurator.

Point

For details on events occurred on each CPU module, refer to the manual for each module.

28.1 Guide for Reference of Event List

The event list contains the following information.

Item	Description
Event code	The ID number of each event. • System code: Event code for the event type "System" • Security code: Event code for the event type "Security" • Operation code: Event code for the event type "Operation" • Application code: Event code for the event type "Application"
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	The details of each detected event

Detailed information

The following table shows the contents of detailed information 1 to 3.

Detailed information	Item	Description
Detailed information 1	Operation initiator information	The following information on the operation source: • Connection port (connection information such as Ethernet and USB) • I/O number • CPU number (CPU number in a multiple CPU system) • Network number • Station number • IP address
	Event history file information	Information on the event history file
	Detailed code	The detailed code specified with the C Controller module dedicated function (CCPU_RegistEventLog)
	Daylight saving time status	The daylight saving time status (start/end)
	Firmware update information	Information on the firmware update
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
	Denied IP address information	Information on the denied IP address
	Drive and file information	Information on drive names and file names
	Copy source drive and file information	Information on drive names and file names
	Operation target information	The information on the operation target (I/O number)
	Clock information (before change)	Information on the clock before change
	Remote operation type information	Information on the remote operation type
	Device and label information	Information on devices and labels
	Detailed information	Detailed information specified with the C Controller module dedicated function (CCPU_RegistEventLog)
Detailed information 3	Clock information (after change)	Information on the clock after change
	Copy destination drive and file information	Information on drive names and file names

28.2 Event List

The following table shows the events for C Controller modules.

Event code	Event type	Event category	Detected event	Description	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
0400	System	Information	Power ON and RESET	The power has been turned ON or the C Controller module has been 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		
0450			Start/end of daylight saving time	Daylight saving time started or ended.	Daylight saving time status	—	—
1000 or higher		Error	When a self-diagnostics error occurs, the error information is stored as an event.				

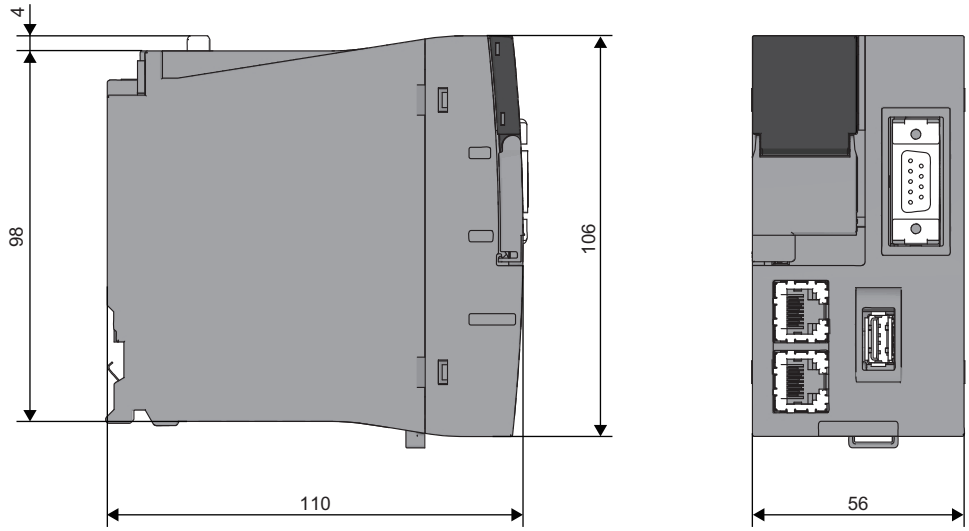
Event code	Event type	Event category	Detected event	Description	Detailed information		
					Detailed information 1	Detailed information 2	Detailed information 3
10300	Security	Information	Access acceptance from IP address prohibited with the IP Filter Settings	Access from an IP address for which access is prohibited with the IP Filter Settings was accepted.	Operation initiator information	Denied IP address information	—
20100	Operation		Error clear	An error was cleared.	Operation initiator information	Operation target information	—
20200			Event history clear	An event history was cleared.		—	—
20300			SD memory card available	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.	—		—
20400			Firmware update succeeded (CPU)	The firmware of the CPU module was successfully updated.	Firmware update information of a CPU module	—	—
20401			Firmware update failed (CPU)	The firmware update of the CPU module failed.			
24000			Clock setting	The clock setting was performed.	Operation initiator information	Clock information (before change)	Clock information (after change)
24001			Remote operation request acceptance	A remote operation request (RUN/STOP/PAUSE) was accepted.		Remote operation type information	—
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			New folder creation, data write to file/folder*1	• A new folder was created. • A new file was created or data was written to a file.	Operation initiator information	Drive and file information	
24201			File copy*1	A file was copied.	—	Copy source drive and file information	Copy destination drive and file information
24202			Folder/file rename*1	A folder or file was renamed.			
25000			Registration from a user program	An event history was registered with the C Controller module dedicated function.	Detailed code	Detailed information	—
2A200		Warning	Memory initialization*1	The memory was initialized.	Operation initiator information	Drive and file information	—
2A201			Device/label zero clear	Data in the device/label memory was cleared to zero.		Device and label information	
2A202			Folder/file deletion*1	A folder or file was deleted.		Drive and file information	

*1 For the file-related events such as write to and deletion of files, operations for the following files are logged in the event history:
· Program file
· Parameter file

APPENDIX

Appendix 1 External Dimensions

The following figure shows the external dimensions of a C Controller module.



(Unit: mm)

Appendix 2 Compliance with EMC and Low Voltage Directives

Method of ensuring compliance

To ensure that Mitsubishi Electric programmable controllers maintain the EMC and Low Voltage Directives or other regulations when incorporated into other machinery or equipment, certain measures may be necessary. Please refer to one of the following manuals.

- 📖MELSEC iQ-R Module Configuration Manual (SH-081262ENG)
- 📖Safety Guidelines (IB-0800525)

Certification marks on the side of the programmable controller indicate compliance with the relevant regulations.

Additional measures

To ensure that this product maintains the EMC and Low Voltage Directives, Ethernet cables, RS-232 cables, and USB cables must meet the noise immunity standards.

■Ethernet cable

Use a shielded twisted pair cable for connection to the 10BASE-T, 100BASE-TX, or 1000BASE-T connector.

■RS-232 cable

For RS-232 cable, be sure to ground the shield part of a shield cable.

Appendix 3 Handling of Batteries

The following table shows the battery used for a C Controller module.

Supplied status	Classification for transportation
Lithium battery built into the device	Non-hazardous material

Precautions

The battery replacement shall be charged. Please contact your local Mitsubishi Electric sales office or representative.

Transport guidelines

When transporting lithium-containing butteries, they should be handled in accordance with the transportation regulations. Products are packed in compliance with the transportation regulations prior to shipment. When repacking any of the unpacked products for transportation, make sure to observe the IATA Dangerous Goods Regulations, IMDG (International Maritime Dangerous Goods) Code, and other local transportation regulations. For details, please consult the shipping carrier used.

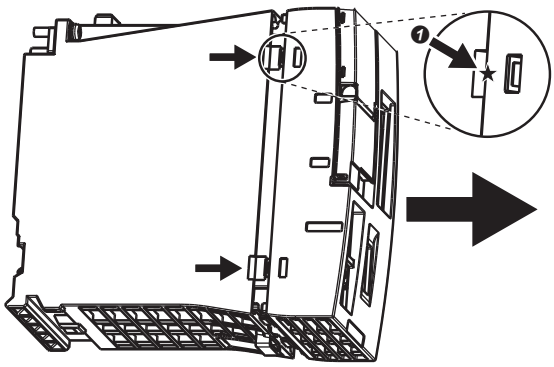
Handling of batteries and devices with built-in batteries in EU member states

For details on handling of batteries in EU member states, refer to the following:

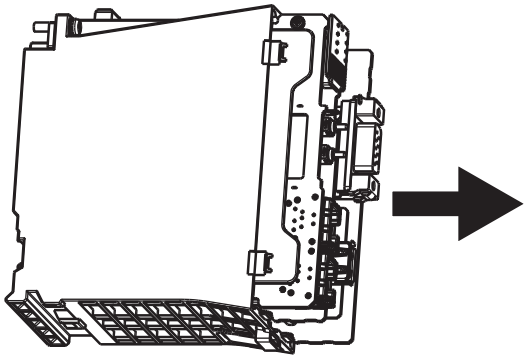
MELSEC iQ-R Module Configuration Manual

Disposal of a battery

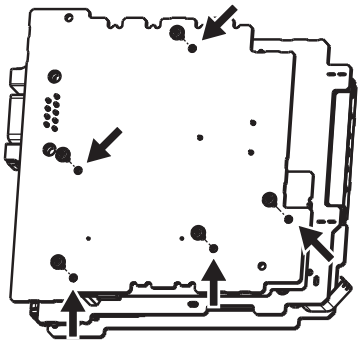
A C Controller module has a built-in lithium battery (primary battery). Prior to disposing of a C Controller module, follow the procedure below to dispose of the built-in battery.



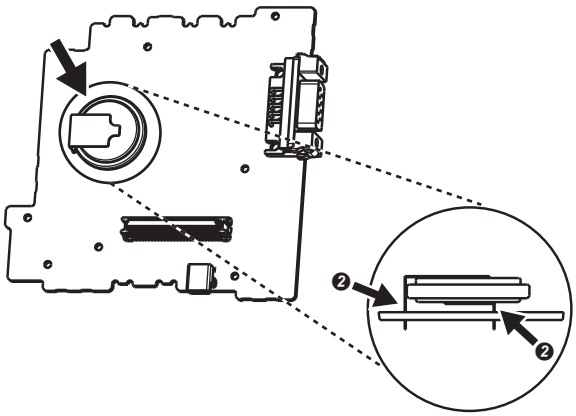
1. Lift the ❶ (★) portion on the side of the C Controller module, and remove the front cover.



2. Remove the printed circuit board from the case.



3. Remove five board fixing screws (M2) from the board.
4. Detach the printed circuit board from one another.



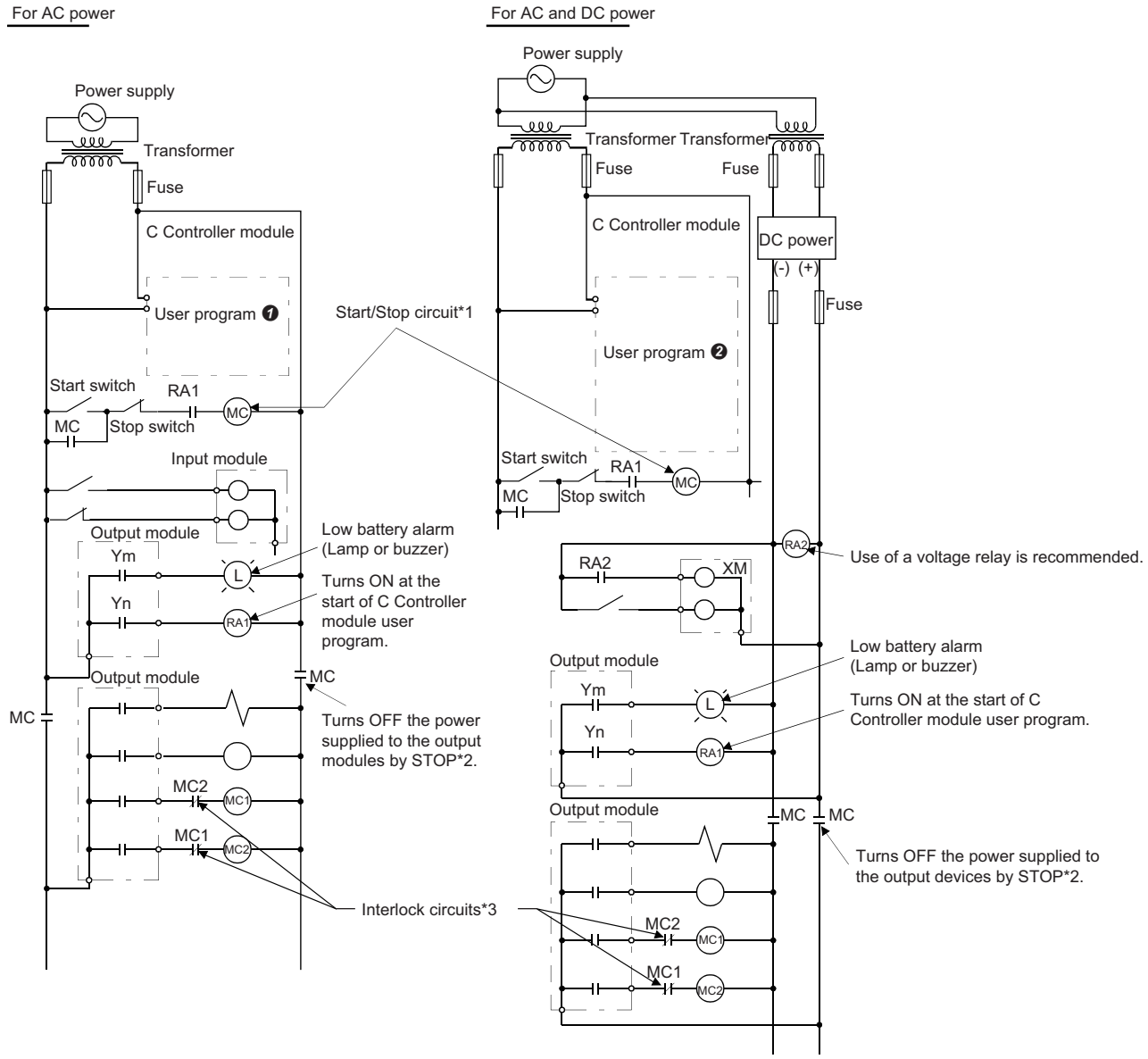
5. Cut off the ❷ (lead) portion of the battery, which is attached on the surface of the printed circuit board with RS-232 connector, using a tool such as nippers, and remove the battery.
6. Dispose of the battery in accordance with the disposal regulations of local governments.

A

Appendix 4 General Safety Requirements

When the power of a C Controller system is turned OFF and ON, the control output may not operate properly temporarily due to differences in the delay and startup time between the power supply for the C Controller module and the external power supply (especially, DC power) for the control module.
Signals also may not be output correctly when the external power supply or the C Controller module fails. In terms of fail-safe and to prevent any incorrect output signals from leading to the entire system failure, configure safety circuits (such as emergency stop circuits, protection circuits, and interlock circuits) external to the C Controller module for the parts where the incorrect output may cause damage to the machines or accidents.
This section shows system design circuit examples, considering the points described above.

When the $\overline{\text{ERR.}}$ contact of a power supply module is not used



*1 The C Controller module starts when RA1 (control starting output) turns ON.
*2 A stop caused by an emergency stop switch or a limit switch.
*3 Configure external interlock circuits for conflicting operations such as forward/reverse rotations and the parts where the incorrect output may cause damage to the machines or accidents.

AC power

1. Power the C Controller module ON.
2. Run the C Controller module.
3. Turn the start switch ON.
4. The output devices are activated by using a user program❶ when the relay (MC) turns ON.

AC and DC power

1. Power the C Controller module ON.
2. Run the C Controller module.
3. RA2 turns ON when DC power is established.
4. XM turns ON when RA2 turns ON, and the processing is started by a user program ❷ after the DC input signal is fully established.
5. Turn the start switch ON.
6. The output devices are activated by using a user program when the relay (MC) turns ON.

User program❶

Create a program to perform the following operations when a C Controller module starts.

■Turning Ym ON when the battery voltage drop is detected

Use the C Controller module dedicated functions (CCPU_GetErrInfo, CCPU_Y_Out_BitEx).

■Turning Yn ON when the user program is started

Use the C Controller module dedicated functions (CCPU_GetErrInfo, CCPU_Y_Out_BitEx).

User program❷

Create a program to perform the following operations when a C Controller module starts.

■Turning Ym ON when the battery voltage drop is detected

Use the C Controller module dedicated functions (CCPU_GetErrInfo, CCPU_Y_Out_BitEx).

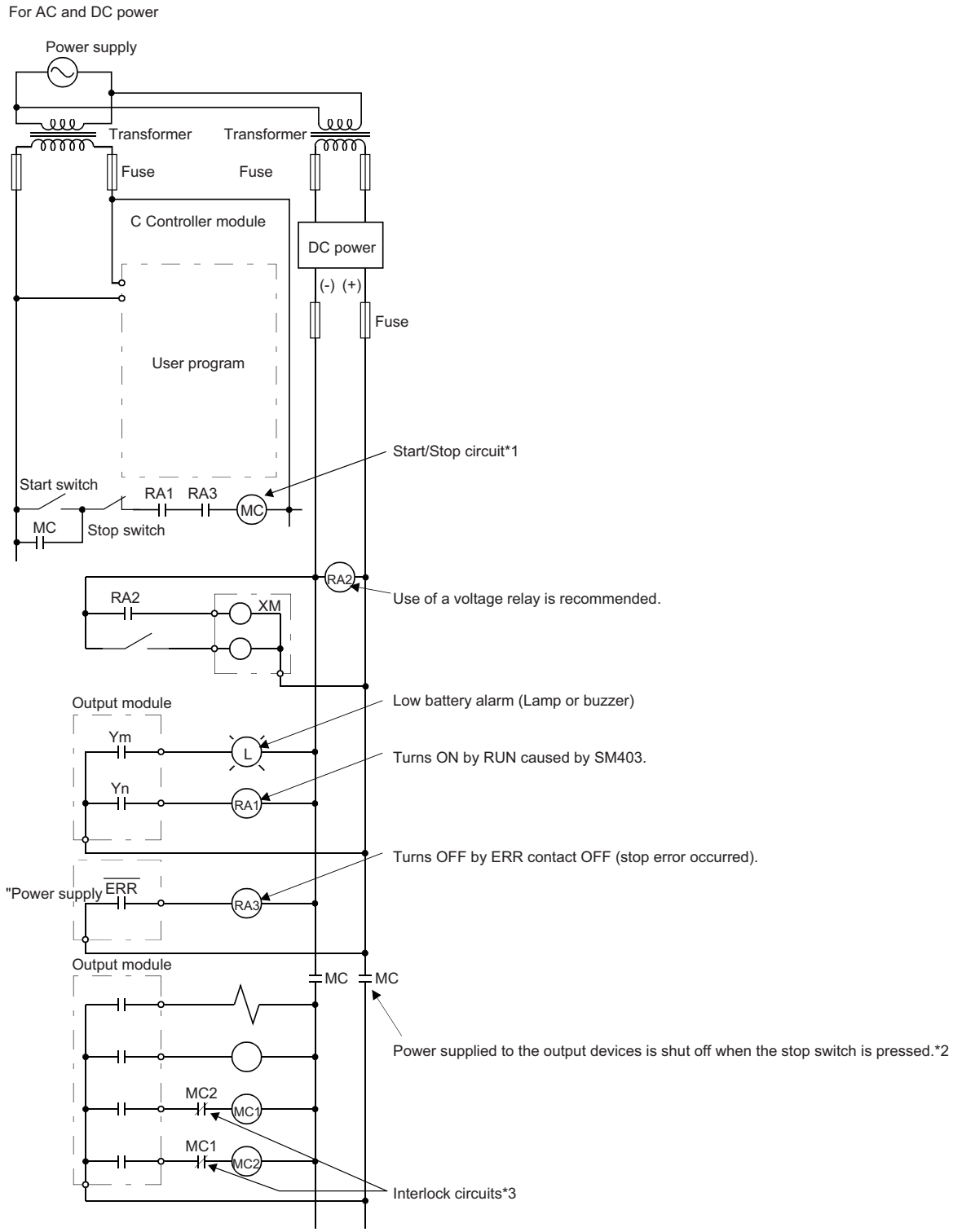
■Turning Yn ON when the user program is started

Use the C Controller module dedicated functions (CCPU_GetErrInfo, CCPU_Y_Out_BitEx).

■Starting the system processing once XM turns ON and the DC input signal is established (after DC input signal settling time)

Create a program to perform the system processing once the DC input signal setting time has passed after XM turns ON. The ON status of XM can be checked with the CCPU_X_In_BitEx function. (The DC input signal setting time is a time from when RA2 turns ON to when the DC input signal is fully established. The time must be set to 0.5 seconds.)
However, setting the DC input signal settling time in the user program is not required when a voltage relay is used for RA2.)

When the $\overline{\text{ERR.}}$ contact of a power supply module is used



*1 The C Controller module starts when RA1 (control starting output) turns ON.

*2 A stop caused by an emergency stop switch or a limit switch, or the $\overline{\text{ERR.}}$ contact is OFF.

*3 Configure external interlock circuits for conflicting operations such as forward/reverse rotations and the parts where the incorrect output may cause damage to the machines or accidents.

AC and DC power

1. Power the C Controller module ON.
2. Run the C Controller module.
3. RA2 turns ON when DC power is established.
4. XM turns ON when RA2 turns ON, and the processing is started by a user program after the DC input signal is fully established.
5. Turn the start switch ON.
6. The output devices are activated by using a user program when the relay (MC) turns ON.

User program

Create a program to perform the following operations when a C Controller module starts.

■Turning Ym ON when the battery voltage drop is detected

Use the C Controller module dedicated functions (CCPU_GetErrInfo, CCPU_Y_Out_BitEx).

■Turning Yn ON when the user program is started

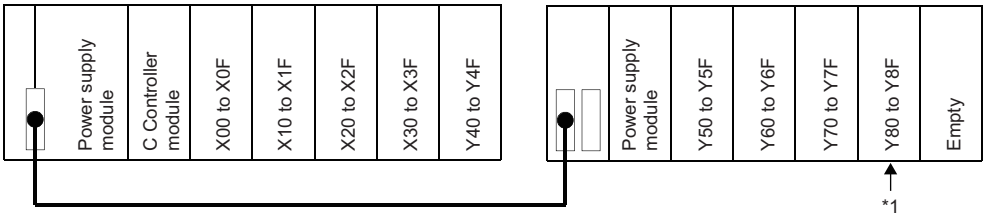
Use the C Controller module dedicated functions (CCPU_GetErrInfo, CCPU_Y_Out_BitEx).

■Starting the system processing once XM turns ON and the DC input signal is established (after DC input signal settling time)

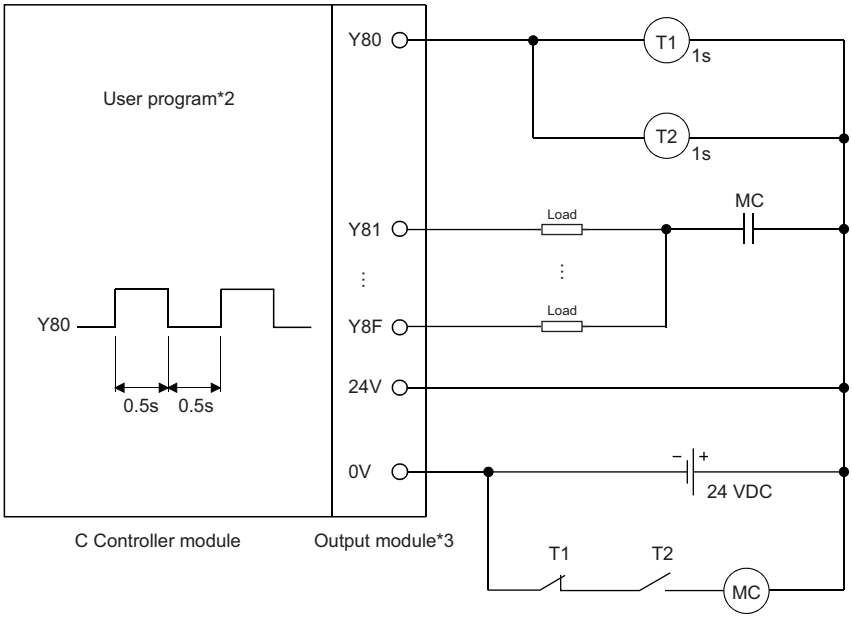
Create a program to perform the system processing once the DC input signal setting time has passed after XM turns ON. The ON status of XM can be checked with the CCPU_X_In_BitEx function. (The DC input signal setting time is a time from when RA2 turns ON to when the DC input signal is fully established. The time must be set to 0.5 seconds.)
However, setting the DC input signal settling time in the user program is not required when a voltage relay is used for RA2.)

Fail-safe measures when a C Controller system fails

Failures of a C Controller module can be detected by the self-diagnostics function; however, failures which occur in a part, such as an I/O control part, may not be detected.
In this case, all inputs or outputs may turn ON or OFF, or normal operation and safety of the control-target device may not be ensured depending on the failure.
Even though Mitsubishi programmable controllers are manufactured under strict quality control, they may fail due to some reasons. Configure fail-safe circuits external to the C Controller module so that no machine is damaged and no accident occurs.
The system example and its fail-safe circuit example are shown below.



*1 Mount an output module for fail-safe purpose on the last slot of the system. (In the example above, the output module is mounted on Y80 to Y8F.)



*2 Create a program to make Y80 repeats ON and OFF at the interval of 0.5 seconds.
*3 Since Y80 turns ON and OFF at the interval of 0.5 seconds, use an output module without contact. (In the example above, a transistor output module is used.)

Appendix 5 Calculation Method for Heat Generation of a C Controller Module

The temperature inside the control panel in which a C Controller module is installed must be 55℃ or lower, which is the operating ambient temperature of a C Controller module. Therefore, it is required to know the average power consumption (heat generation) of the equipment and devices installed in the control panel for the heat release design of the panel.
For details on the calculation method, refer to the following:
MELSEC iQ-R Module Configuration Manual
It should be noted that the terms in the left column shall be replaced with the one in the right column.

Term	Replaced term
Programmable controller	C Controller module
Programmable controller system	C Controller system

Appendix 6 Special Relay List

The following table shows how to read the list of special relay (SM).

Item	Description
No.	The number of the special relay
Name	The name of the special relay
Content	The content of the special relay
Details	The details of the special relay
Set by (when to set)	The timing to set each device by a system and/or user. (Set by) S: Set by system (When to set) • Initial: Set only when initial processing is performed (e.g. powering ON the system, changing the operating status from STOP to RUN). • Error occurrence: Set if an error occurs. • Status change: Set only when the status is changed. • At END processing: Set for every refresh cycle.

Point

Do not change the special relay set by system with the operations such as program execution or device test. Doing so may result in system down or disconnection of communication.

Diagnostic information

The special relays for diagnostics information are as follows.

No.	Name	Description	Details	Set by (when to set)
SM0	Latest diagnostics error	OFF: No error ON: Error	Turns ON when the diagnostics error occurred. The ON state is retained even after the error is cleared later.	S (at error occurrence)
SM1	Latest self-diagnostics error	OFF: No error ON: Error	Turns ON when the self-diagnostics error occurred. The ON state is retained even after the error is cleared later.	S (at error occurrence)
SM53	AC/DC DOWN	OFF: AC/DC DOWN not detected ON: AC/DC DOWN detected	• Turns ON when a momentary power failure within 20 ms is detected while the AC power supply module is in use. This relay is reset after the power is turned OFF and ON. • Turns ON when a momentary power failure within 10 ms is detected while the DC power supply module is in use. This relay is reset after the power is turned OFF and ON.	S (at error occurrence)
SM60	Fuse Blown	OFF: Normal ON: Module with blown fuse	• Turns ON when at least one output module is in fuse blown state. The ON state is retained even after the error is cleared later. • The fuse blown state check is also performed for output modules on the remote I/O station.	S (at error occurrence)
SM61	I/O module verify error	OFF: Normal ON: Error	• Turns ON when the state of the I/O module is different from the one registered during power ON. The ON state is retained even after the error is cleared later. • I/O module verification is also performed for modules on the remote I/O station.	S (at error occurrence)
SM80	Detailed information 1 in-use flag	OFF: Not used ON: In use	Turns ON if the detailed information 1 exists when SM0 turns ON.	S (status change)
SM112	Detailed information 2 in-use flag	OFF: Not used ON: In use	Turns ON if the detailed information 2 exists when SM0 turns ON.	S (status change)
SM600	Memory card usable flag	OFF: Unusable ON: Usable	Turns ON when the module is in SD memory card available state. (The flag turns ON after inserting an applicable SD memory card and it is in the available state.)	S (status change)
SM601	Memory card protect flag	OFF: Not protected ON: Protected	Turns ON when the write protect switch of the SD memory card is ON	S (status change)
SM603	Memory card (drive 2) flag	OFF: Not inserted ON: Inserted	Turns ON while an SD memory card is inserted. (Regardless of the availability of the SD memory card and its type.)	S (status change)

System information

The special relays for system information are as follows.

No.	Name	Description	Details	Set by (when to set)
SM220*1	CPU No.1 preparation completed	OFF: CPU No.1 preparation not completed ON: CPU No.1 preparation completed	Turns ON when the access from another CPU module to the CPU No.1 is enabled after turning the power ON or resetting the module.	S (status change)
SM221*1	CPU No.2 preparation completed	OFF: CPU No.2 preparation not completed ON: CPU No.2 preparation completed	Turns ON when the access from another CPU module to the CPU No.2 is enabled after turning the power ON or resetting the module.	S (status change)
SM222*1	CPU No.3 preparation completed	OFF: CPU No.3 preparation not completed ON: CPU No.3 preparation completed	Turns ON when the access from another CPU module to the CPU No.3 is enabled after turning the power ON or resetting the module.	S (status change)
SM223*1	CPU No.4 preparation completed	OFF: CPU No.4 preparation not completed ON: CPU No.4 preparation completed	Turns ON when the access from another CPU module to the CPU No.4 is enabled after turning the power ON or resetting the module.	S (status change)

*1 Used as an interlock to access the CPU No.1 when the multiple CPU synchronization setting is configured with asynchronous mode.

Firmware update function

The special relays for the firmware update function are as follows.

No.	Name	Description	Details	Set by (when to set)
SM922	Firmware update completion with/without an error	OFF: Update completed without an error (including successful completion) ON: Update completed with an error	Turns ON when the firmware update function is completed with an error. (SM922 turns ON when 'Latest firmware update result' (SD922) is in the range from '100H' to '300H.')	S (initial)

CC-Link IE Field Network Basic function

The special relays related to the CC-Link IE Field Network Basic function are as follows.

No.	Name	Description	Details	Set by (when to set)
SM1536	Cyclic transmission status	OFF: Transmission not executed ON: Transmitting	Turns ON when cyclic transmission is started, and turns OFF when the cyclic transmission is stopped.	S (END processing)
SM1540	Data link status	OFF: No error station ON: Error station found	Turns ON when an error occurs in one or more remote stations. The status of each remote station can be checked with 'Data link status of each station' (SD1540).	S (END processing)

Appendix 7 Special Register List

The following table shows how to read the list of special registers (SD).

Item	Description
No.	The number of the special register
Name	The name of the special register
Content	The content of the special register
Details	The details of the special register
Set by (when to set)	The timing to set each device by a system and/or user. (Set by) S: Set by system (When to set) • System: Set when a fixed cycle or the status is changed in the system. • Initial: Set only when performing an initial operation (such as power ON, STOP to RUN). • Status change: Set only when the status is changed. • Error occurrence: Set if an error occurs. • Switch change: Set when the switch is changed. • At END processing: Set for every refresh cycle.

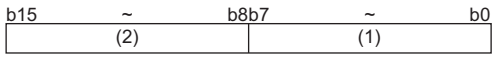
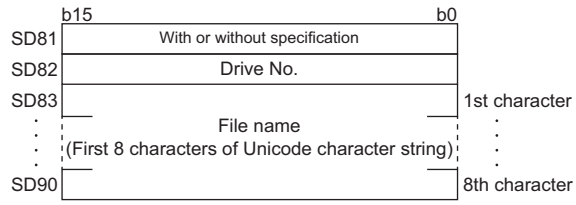
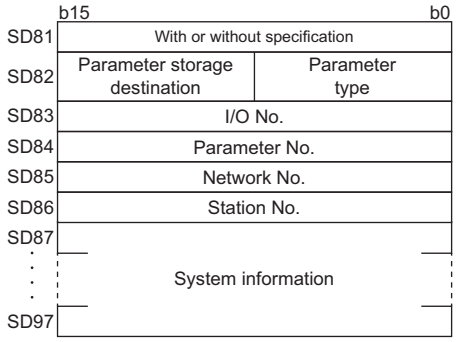
Point

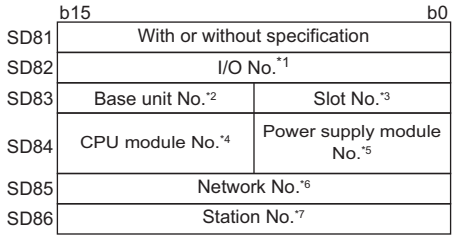
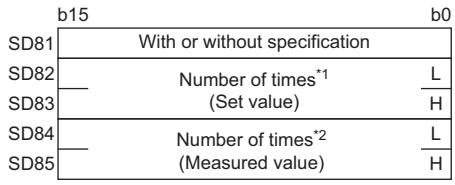
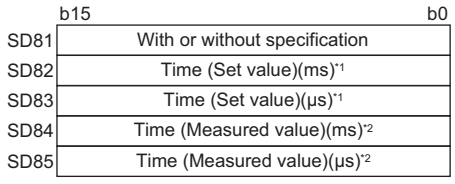
Do not change the special registers which are to be set by the system using operations such as program execution and device test. Doing so may result in system failure or disconnection of communication.

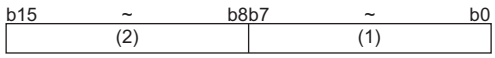
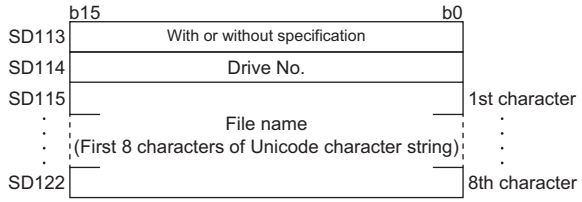
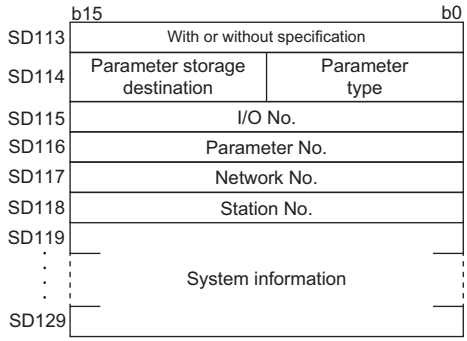
Diagnostic information

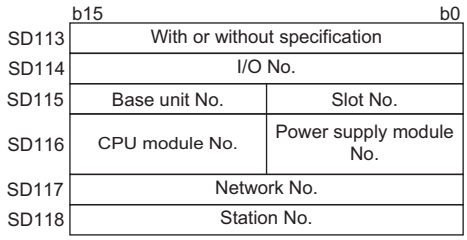
The special registers for diagnostics information are as follows.

No.	Name	Description	Details	Set by (when to set)
SD0	Latest self-diagnostics error code	Latest self-diagnostics error code	Error codes are stored in hexadecimal when an error is detected with the diagnostics. • The same information as the latest information displayed on the error history is displayed.	S (at error occurrence)
SD1	Self-diagnostics error occurrence time	Self-diagnostics error occurrence time	The year (four digits) when SD0 data was updated is stored as a BIN code.	S (at error occurrence)
SD2			The month when SD0 data was updated is stored as a BIN code .	
SD3			The date when SD0 data was updated is stored as a BIN code.	
SD4			The hour when SD0 data was updated is stored as a BIN code.	
SD5			The minute when SD0 data was updated is stored as a BIN code.	
SD6			The second when SD0 data was updated is stored as a BIN code.	
SD7			The day of the week when SD0 data was updated is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)	
SD10	Self-diagnostics error code	Self-diagnostics error code 1	Up to 16 types of error codes are stored to SD10 and higher when the diagnostics detects errors. (The same error code as the one already stored in SD10 and higher is not stored.) The 17th and succeeding error codes are not stored, in addition to the case that 16 types of error codes have already been stored to SD10 to SD25.	S (at error occurrence)
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	Number of times for AC/DC DOWN detection	A value is incremented by one each time when input voltage drops to 85% (AC power)/65% (DC power) or less of the rated value while the CPU module is in operation, and stores it as a BIN code. A counting cycle from 0 → 65535 → 0 is repeated.	S (at error occurrence)
SD60	Number of module with blown fuse	Number of module with blown fuse	The lowest I/O number of module in which a fuse blew is stored.	S (at error occurrence)
SD61	I/O module verify error module number	I/O module verify error module number	The lowest I/O number of the module in which an I/O module verification error has been detected is stored.	S (at error occurrence)

No.	Name	Description	Details	Set by (when to set)
SD80	Detailed information 1 information category	Detailed information 1 information category code	Detailed information 1 information category code is stored.  (1) Information category code (2) Not used (fixed to 0) The following codes are stored into the information category code. • 0: N/A • 1: N/A • 2: Drive number and file name • 4: Parameter information • 5: System configuration information • 6: Frequency information • 7: Time information • 24: Failure information	S (at error occurrence)
SD81 to SD111	Detailed information 1	Detailed information 1	• Detailed information 1 corresponding to the error code (SD0) is stored. • There are six types of information to be stored as shown in (2), (4) to (7), and (24). • The type of the detailed information 1 can be acquired from SD80. (The value of the "Detailed information 1 information category code" which is to be stored in SD80 corresponds to the number (2), (4) to (7), and (24) in the following figures.) ■(2) Drive number and file name  The following information is stored in SD81. • b0: Drive No. • b1: File name ■(4) Parameter information  The following information is stored in SD81. • b0: Parameter type • b1: Parameter storage destination • b2: I/O No. • b3: Parameter No. • b4: Network No. • b5: Station No. • b6: System information The parameter type is stored in SD82 (b0 to b7). • 1: System parameter • 2: CPU parameter • 3: Module parameter • 4: Module extended parameter • 5: Memory card parameter The parameter storage destination is stored in SD82 (b8 to b15). • 2: SD memory card • 4: Data memory The I/O No. is stored in SD83. (0xFFFFH if an I/O No. is not assigned.) The network No. (0 to 120) is stored in SD85. (0 for a master station)	S (at error occurrence)

No.	Name	Description	Details	Set by (when to set)
SD81 to SD111	Detailed information 1	Detailed information 1	■(5) System configuration information  Without specification: *1: 0xFFFF *2,*3,*4: 0xFF *5,*6,*7: 0 The following information is stored in SD81. • b0: I/O No. • b1: Slot No. • b2: Base unit No. • b3: Power supply module No. • b4: CPU module No. • b5: Network No. • b6: Station No. The slot No. (0 to 11) is stored in SD83 (b0 to b7). The base No. is stored in SD83 (b8 to b15). • 0: Main base unit • 1 to 7: Extension base unit, level 1 to 7 • 8: Extension base unit, level 8 (when the number of base is exceeded) The power No. (1 to 2) is stored in SD84 (b0 to b7). The CPU No. (1 to 4) is stored in SD84 (b8 to b15). The network No. (0 to 120) is stored in SD85. (0 for a master station) ■(6) Frequency information  *1: Set '0' when the number of times (Set value) is not specified. *2: Set '0' when the number of times (Measured value) is not specified. The following information is stored in SD81. • b0: Number of times (Set value) • b1: Number of times (Measured value) ■(7) Time information  *1: Set '0' when the time (Set value) is not specified. *2: Set '0' when the time (Measured value) is not specified. The following information is stored in SD81. • b0: Time (Set value) [ms] • b1: Time (Set value) [μs] • b2: Time (Measured value)) [ms] • b3: Time (Measured value) [μs] ■(24) Failure information The failure information is a part of system information.	S (at error occurrence)

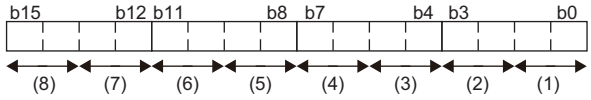
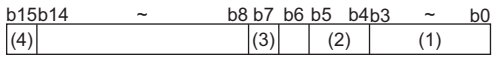
No.	Name	Description	Details	Set by (when to set)
SD112	Detailed information 2 information category	Detailed information 2 information category code	Detailed information 2 information category code is stored.  (1) Information category code (2) Not used (fixed to 0) The following codes are stored into the information category code. • 0: N/A • 2: Drive number and file name • 3: N/A • 4: Parameter information • 5: System configuration information	S (at error occurrence)
SD113 to SD143	Detailed information 2	Detailed information 2	• Detailed information 2 corresponding to the error code (SD0) is stored. • There are three types of information to be stored as shown in (2), (4), and (5). • The type of the detailed information 2 can be acquired from SD112. (The value of the "Detailed information 2 information category code" which is to be stored to SD112 corresponds to the number (2), (4), and (5) in the following figures.) ■(2) Drive number and file name  The following information is stored in SD113. • b0: Drive No. • b1: File name ■(4) Parameter information  The following information is stored in SD113. • b0: Parameter type • b1: Parameter storage destination • b2: I/O No. • b3: Parameter No. • b4: Network No. • b5: Station No. • b6: System information The parameter type is stored in SD114 (b0 to b7). • 1: System parameter • 2: CPU parameter • 3: Module parameter • 4: Module extended parameter • 5: Memory card parameter The parameter storage destination is stored in SD114 (b8 to b15). • 2: SD memory card • 4: Data memory The I/O No. is stored in SD115. (0xFFFFH if an I/O No. is not assigned.) The network No. (0 to 120) is stored in SD117. (0 for a master station)	S (at error occurrence)

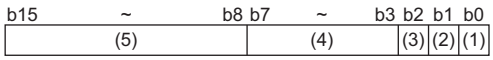
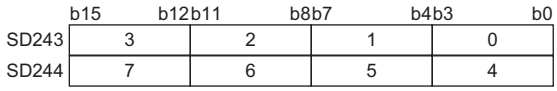
No.	Name	Description	Details	Set by (when to set)
SD113 to SD143	Detailed information 2	Detailed information 2	■(5) System configuration information  The following information is stored in SD113. • b0: I/O No. • b1: Slot No. • b2: Base unit No. • b3: Power supply module No. • b4: CPU module No. • b5: Network No. • b6: Station No. The slot No. (0 to 11) is stored in SD115 (b0 to b7). The base No. is stored in SD115 (b8 to b15). • 0: Main base unit • 1 to 7: Extension base unit, level 1 to 7 • 8: Extension base unit, level 8 (when the number of base is exceeded) The power No. (1 to 2) is stored in SD116 (b0 to b7). The CPU No. (1 to 4) is stored in SD116 (b8 to b15). The network No. (0 to 120) is stored in SD117. (0 for a master station)	S (at error occurrence)

A

System information

The special registers for system information are as follows.

No.	Name	Description	Details	Set by (when to set)
SD200	Switch status	CPU switch status (RESET/STOP/ RUN switch)	The switch status of the CPU module is stored as follows: 0: RUN, 1: STOP	S (when the switch status is changed)
SD201	LED status	Status of CPU-LED	This device stores the information that indicates LED status of the CPU module in the following bit patterns. 0: OFF, 1: ON, 2: Flashing (high speed/low speed)  (1) READY (2) ERROR (3) BUS RUN (4) USER (5) USB RDY (6) CARD RDY (7) Reserved (8) RS SD/RD	S (status change)
SD203	CPU operating status	CPU operating status	The operating status of the CPU module is stored as follows: 0: RUN, 1: Reserved, 2: STOP, 3: PAUSE	S (system)
SD228	Multiple CPU system information	Number of CPU modules	The number of CPU modules in the multiple CPU system is stored (one to four, including empty CPU).	S (initial)
SD229		CPU module number in multiple CPU system	The host station CPU number is stored when a multiple CPU system is configured.	S (initial)
SD230		CPU No.1 operating status	The operation information for each CPU number is stored. (Information for a number of multiple CPUs which is indicated in SD228 is stored.)  (1) The operating status is stored in b0 to b3. • 0: RUN • 2: STOP • 3: PAUSE • 4: Initial • FH: Reset (2) The classification is stored in b4 and b5. However, minor or moderate errors will be those set in the CPU parameter RAS settings, system parameter I/O assignment settings, and multiple CPU settings. • 0: Normal • 1: Minor error • 2: Moderate error • 3: Major error (3) The stop error flag is stored in b7. • 0: No stop error • 1: Stop error (4) The mounting status of the CPU module is stored in b15. • 0: Not mounted • 1: Mounted	S (at error occurrence)
SD231		CPU No.2 operating status		
SD232		CPU No.3 operating status		
SD233		CPU No.4 operating status		
SD241	Number of extension base unit	0: Main base unit only 1 to 7: Number of extension base units	The maximum number of the implemented extension base units is stored.	S (initial)

No.	Name	Description	Details	Set by (when to set)
SD242	Determination of mountability of MELSEC-Q series module	Identification of the base type 0: MELSEC-Q series module is not mountable. (There is no base unit on which MELSEC-Q series module can be mounted.) 1: MELSEC-Q series module is mountable. (There is a base unit on which MELSEC-Q series module can be mounted.)	Determines whether or not MELSEC-Q series module can be mounted. When no module is mounted, the value is fixed to 0.  (1) Main base unit (fixed to 0) (2) Extension base unit, level 1 (MELSEC-Q series module are mountable.) (3) Extension base unit, level 2 (MELSEC-Q series modules are mountable.) (4) Extension base unit, level 3 to 7 (MELSEC-Q series modules are mountable.) (5) 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 in the system parameters, is stored. When the number of slots of the base unit is not specified in the system parameter, that of the mounted base unit is stored.  • 0: Base unit • 1 to 7: Extension base unit 1 to 7	S (initial)
SD250	Latest I/O for implemented module	Latest I/O number for implemented module	The value of the last I/O number of the mounted module + 1 which is divided by 16 is stored. Example 1: Last input number: 010FH • SD250: 0011H Example 2: Last input number: 0FFFH • SD250: 0100H	S (initial)
SD260 SD261	Number of points assigned for bit devices	Number of points assigned for X (L) Number of points assigned for X (H)	The number of points of the device X currently set is stored as 32-bit data.	S (initial)
SD262 SD263		Number of points assigned for Y (L) Number of points assigned for Y (H)	The number of points of the device Y currently set is stored as 32-bit data.	S (initial)
SD264 SD265		Number of points assigned for M (L) Number of points assigned for M (H)	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)
SD266 SD267		Number of points assigned for B (L) Number of points assigned for B (H)	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)
SD280 SD281	Number of points assigned for word devices	Number of points assigned for D (L) Number of points assigned for D (H)	The number of points of the device D currently set is stored as 32-bit data. The number of points assigned is stored even when the number of points assigned to D is 32K points or less.	S (initial)
SD282 SD283		Number of points assigned for W (L) Number of points assigned for W (H)	The number of points of the device W currently set is stored as 32-bit data. The number of points assigned is stored even when the number of points assigned to W is 32K points or less.	S (initial)
SD306 SD307	Number of points assigned for file registers	Number of points assigned for ZR (L) Number of points assigned for ZR (H)	The number of points of the device ZR currently set is stored as 32-bit data. The number of points assigned is stored even when the number of points assigned to ZR is 32K points or less.	S (initial)

Fixed cycle function information

The special registers for fixed cycle function information are as follows.

No.	Name	Description	Details	Set by (when to set)
SD520	Current fixed cycle processing time*1	Current fixed cycle processing time (unit: ms)	The current fixed cycle processing time is stored in SD520 and SD521. (Measured in microseconds.) • SD520: Stores the millisecond portion of a value. (Storage range: 0 to 65535) • SD521: Stores the microsecond portion of a value. (Storage range: 0 to 999) When the current fixed cycle processing time is 23.6 ms; for example, it is stored as shown below: • SD520 = 23 (ms) • SD521 = 600 (μs)	S (system)
SD521		Current fixed cycle processing time (unit: μs)		
SD522	Minimum fixed cycle processing time*1	Minimum fixed cycle processing time (unit: ms)	The minimum fixed cycle processing time is stored in SD522 and SD523. (Measured in microseconds.) • SD522: Stores the millisecond portion of a value. (Storage range: 0 to 65535) • SD523: Stores the microsecond portion of a value. (Storage range: 0 to 999) When the minimum fixed cycle processing time is 23.6 ms; for example, it is stored as shown below: • SD522 = 23 (ms) • SD523 = 600 (μs)	S (system)
SD523		Minimum fixed cycle processing time (unit: μs)		
SD524	Maximum fixed cycle processing time*1	Maximum fixed cycle processing time (unit: ms)	The maximum fixed cycle processing time is stored in SD524 and SD525. (Measured in microseconds.) • SD524: Stores the millisecond portion of a value. (Storage range: 0 to 65535) • SD525: Stores the microsecond portion of a value. (Storage range: 0 to 999) When the maximum fixed cycle processing time is 23.6 ms; for example, it is stored as shown below. • SD524 = 23 (ms) • SD525 = 600 (μs)	S (system)
SD525		Maximum fixed cycle processing time (unit: μs)		
SD526	Current link refresh processing time	Current link refresh processing time (unit: ms)	The current link refresh processing time is stored in SD526 and SD527. (Measured in microseconds.) • SD526: Stores the millisecond portion of a value. (Storage range: 0 to 65535) • SD527: Stores the microsecond portion of a value. (Storage range: 0 to 999) When the current link refresh processing time is 23.6 ms; for example, it is stored as shown below. • SD526 = 23 (ms) • SD527 = 600 (μs)	S (system)
SD527		Current link refresh processing time (unit: μs)		
SD528	Minimum link refresh processing time	Minimum link refresh processing time (unit: ms)	The minimum link refresh processing time is stored in SD528 and SD529. (Measured in microseconds.) • SD528: Stores the millisecond portion of a value. (Storage range: 0 to 65535) • SD529: Stores the microsecond portion of a value. (Storage range: 0 to 999) When the minimum link refresh processing time is 23.6 ms; for example, it is stored as shown below: • SD528 = 23 (ms) • SD529 = 600 (μs)	S (system)
SD529		Minimum link refresh processing time (unit: μs)		
SD530	Maximum link refresh processing time	Maximum link refresh processing time (unit: ms)	The maximum link refresh processing time is stored in SD530 and SD531. (Measured in microseconds.) • SD530: Stores the millisecond portion of a value. (Storage range: 0 to 65535) • SD531: Stores the microsecond portion of a value. (Storage range: 0 to 999) When the maximum link refresh processing time is 23.6 ms; for example, it is stored as shown below: • SD530 = 23 (ms) • SD531 = 600 (μs)	S (system)
SD531		Maximum link refresh processing time (unit: μs)		

*1 The fixed cycle processing includes the refresh processing with network modules, the reset processing of watchdog timer, the self-diagnostics processing, and the completion processing of dedicated instructions.

Firmware update function

The special registers for the firmware update function are as follows.

No.	Name	Description		Details	Set by (when to set)
SD912	Information on latest firmware update	History information	Execution time (year)	The value of the year (four digits) when the firmware update was executed is stored as a BIN code.	S (initial)
SD913			Execution time (month)	The value of the month when the firmware update was executed is stored as a BIN code.	S (initial)
SD914			Execution time (day)	The value of the day when the firmware update was executed is stored as a BIN code.	S (initial)
SD915			Execution time (hour)	The value of the hour when the firmware update was executed is stored as a BIN code.	S (initial)
SD916			Execution time (minute)	The value of the minutes when the firmware update was executed is stored as a BIN code.	S (initial)
SD917			Execution time (second)	The value of the seconds when the firmware update was executed is stored as a BIN code.	S (initial)
SD918			Execution time (day of the week)	The value of the day of the week when the firmware update was executed is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)	S (initial)
SD919			Version after the update	The firmware version after the update execution is stored. (When the update is completed with an error, '0' is stored.)	S (initial)
SD920			Version before the update	The firmware version before the update is stored.	S (initial)
SD921	Latest firmware update result	Target		The start input/output number of the module where the firmware update was executed is stored. • C Controller module: 3FFH	S (initial)
SD922		Execution result		The execution result of the firmware update is stored. • 1H: Normal end • 100H: Flash ROM error • 200H: Model mismatched • 201H: File invalid • 203H: Firmware update prohibited state • 300H: Firmware data error	S (initial)
SD923	Information on previous firmware update	History information	Execution time (year)	The value of the year (four digits) when the firmware update was executed is stored as a BIN code.	S (initial)
SD924			Execution time (month)	The value of the month when the firmware update was executed is stored as a BIN code.	S (initial)
SD925			Execution time (day)	The value of the day when the firmware update was executed is stored as a BIN code.	S (initial)
SD926			Execution time (hour)	The value of the hour when the firmware update was executed is stored as a BIN code.	S (initial)
SD927			Execution time (minute)	The value of the minutes when the firmware update was executed is stored as a BIN code.	S (initial)
SD928			Execution time (second)	The value of the seconds when the firmware update was executed is stored as a BIN code.	S (initial)
SD929			Execution time (day of the week)	The value of the day of the week when the firmware update was executed is stored as a BIN code. (0: Sun, 1: Mon, 2: Tue, 3: Wed, 4: Thu, 5: Fri, 6: Sat)	S (initial)
SD930			Version after the update	The firmware version after the update execution is stored. (When the update is completed with an error, '0' is stored.)	S (initial)
SD931			Version before the update	The firmware version before the update is stored.	S (initial)
SD932	Previous firmware update result	Target		The start input/output number of the module where the firmware update was executed is stored. • C Controller module: 3FFH	S (initial)
SD933		Execution result		The execution result of the firmware update is stored. • 1H: Normal end • 100H: Flash ROM error • 200H: Model mismatched • 201H: File invalid • 203H: Firmware update prohibited state • 300H: Firmware data error	S (initial)

CC-Link IE Field Network Basic function

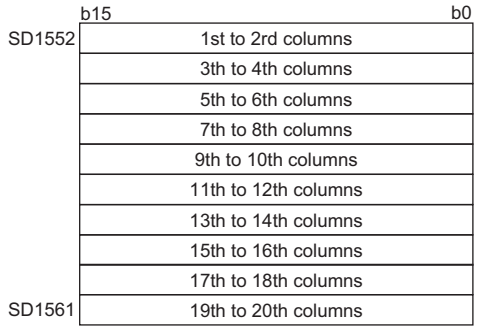
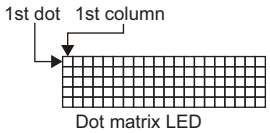
The special registers for the CC-Link IE Field Network Basic function are as follows.

No.	Name	Description	Details	Set by (when to set)
SD1536 to SD1539	Cyclic transmission status of each station	Cyclic transmission status of each station	The cyclic transmission status of each station is stored. • OFF: Cyclic transmission not executed • ON: Cyclic transmission executing b15 ... b0 SD1536: 16 15 14 13 12 ~ 5 4 3 2 1 SD1537: 32 31 30 29 28 ~ 21 20 19 18 17 SD1538: 48 47 46 45 44 ~ 37 36 35 34 33 SD1539: 64 63 62 61 60 ~ 53 52 51 50 49 The numbers in the figure indicate station numbers (1 to 64). (Conditions) • Only the bit of the start station number turns ON. • A reserved station and maximum station number and later are ignored. 'Cyclic transmission status of each station' (SD1536 to SD1539) can be used as an interlock for cyclic transmission. (☞ Page 176 Interlock program for cyclic transmission)	S (END processing)
SD1540 to SD1543	Data link status of each station	Data link status of each station	The data link transmission status of each station is stored. • OFF: Normal station*1 • ON: Error station b15 ... b0 SD1540: 16 15 14 13 12 ~ 5 4 3 2 1 SD1541: 32 31 30 29 28 ~ 21 20 19 18 17 SD1542: 48 47 46 45 44 ~ 37 36 35 34 33 SD1543: 64 63 62 61 60 ~ 53 52 51 50 49 The numbers in the figure indicate station numbers (1 to 64). (Conditions) • Only the bit of the start station number turns ON. • A reserved station and maximum station number and later are ignored. 'Data link status of each station' (SD1540 to SD1543) can be used for monitoring remote stations, connected cables, and connected hub.	S (END processing)

*1 This includes the situation where a remote station did not respond the first request from the master station with reasons such as the power of the remote station is turned OFF. (Since data link is regarded as undefined, the station is not regarded as an error station.)

Special registers for C Controller modules

The special registers for C Controller modules are as follows:

No.	Name	Description	Details	Set by (when to set)
SD672	CCIETSN module channel number (1st module)	CCIETSN module channel number (1st module)	Channel number (281 to 288)*1 of the CC-Link IE TSN module (1st module) controlled	S (initial)
SD673	CCIETSN module channel number (2nd module)	CCIETSN module channel number (2nd module)	Channel number (281 to 288)*1 of the CC-Link IE TSN module (2nd module) controlled	S (initial)
SD674	CCIETSN module channel number (3rd module)	CCIETSN module channel number (3rd module)	Channel number (281 to 288)*1 of the CC-Link IE TSN module (3rd module) controlled	S (initial)
SD675	CCIETSN module channel number (4th module)	CCIETSN module channel number (4th module)	Channel number (281 to 288)*1 of the CC-Link IE TSN module (4th module) controlled	S (initial)
SD676	CCIETSN module channel number (5th module)	CCIETSN module channel number (5th module)	Channel number (281 to 288)*1 of the CC-Link IE TSN module (5th module) controlled	S (initial)
SD677	CCIETSN module channel number (6th module)	CCIETSN module channel number (6th module)	Channel number (281 to 288)*1 of the CC-Link IE TSN module (6th module) controlled	S (initial)
SD678	CCIETSN module channel number (7th module)	CCIETSN module channel number (7th module)	Channel number (281 to 288)*1 of the CC-Link IE TSN module (7th module) controlled	S (initial)
SD679	CCIETSN module channel number (8th module)	CCIETSN module channel number (8th module)	Channel number (281 to 288)*1 of the CC-Link IE TSN module (8th module) controlled	S (initial)
SD1552 to SD1561	Dot matrix LED	Dot matrix LED	The dot matrix LED lighting status on the C Controller module is stored.   The following information is stored in SD1552. • b0 to b6: From the 7th dot to the 1st dot in the 1st column (the 7th dot is stored in b0.) • b7: Reserved • b8 to b14: From the 7th dot to the 1st dot in the 2nd column (the 7th dot is stored in b8.) • b15: Reserved The following information is stored in SD1561. • b0 to b6: From the 7th dot to the 1st dot in the 19th column (the 7th dot is stored in b0.) • b7: Reserved • b8 to b14: From the 7th dot to the 1st dot in the 20th column (the 7th dot is stored in b8.) • b15: Reserved	S (status change)
SD1566	MELSECNET/H network module channel number (1st module)	MELSECNET/H network module channel number (1st module)	Channel number (51 to 54)*1 of the MELSECNET/H network module (1st module) controlled	S (initial)

A

No.	Name	Description	Details	Set by (when to set)
SD1567	MELSECNET/H network module channel number (2nd module)	MELSECNET/H network module channel number (2nd module)	Channel number (51 to 54)*1 of the MELSECNET/H network module (2nd module) controlled	S (initial)
SD1568	MELSECNET/H network module channel number (3rd module)	MELSECNET/H network module channel number (3rd module)	Channel number (51 to 54)*1 of the MELSECNET/H network module (3rd module) controlled	S (initial)
SD1569	MELSECNET/H network module channel number (4th module)	MELSECNET/H network module channel number (4th module)	Channel number (51 to 54)*1 of the MELSECNET/H network module (4th module) controlled	S (initial)
SD1570	CC-Link module channel number (1st module)	CC-Link module channel number (1st module)	Channel number (81 to 88)*1 of the CC-Link module (1st module) controlled	S (initial)
SD1571	CC-Link module channel number (2nd module)	CC-Link module channel number (2nd module)	Channel number (81 to 88)*1 of the CC-Link module (2nd module) controlled	S (initial)
SD1572	CC-Link module channel number (3rd module)	CC-Link module channel number (3rd module)	Channel number (81 to 88)*1 of the CC-Link module (3rd module) controlled	S (initial)
SD1573	CC-Link module channel number (4th module)	CC-Link module channel number (4th module)	Channel number (81 to 88)*1 of the CC-Link module (4th module) controlled	S (initial)
SD1574	CC-Link module channel number (5th module)	CC-Link module channel number (5th module)	Channel number (81 to 88)*1 of the CC-Link module (5th module) controlled	S (initial)
SD1575	CC-Link module channel number (6th module)	CC-Link module channel number (6th module)	Channel number (81 to 88)*1 of the CC-Link module (6th module) controlled	S (initial)
SD1576	CC-Link module channel number (7th module)	CC-Link module channel number (7th module)	Channel number (81 to 88)*1 of the CC-Link module (7th module) controlled	S (initial)
SD1577	CC-Link module channel number (8th module)	CC-Link module channel number (8th module)	Channel number (81 to 88)*1 of the CC-Link module (8th module) controlled	S (initial)
SD1578	CC-Link IE Controller Network module channel number (1st module)	CC-Link IE Controller Network module channel number (1st module)	Channel number (151 to 158)*1 of the CC-Link IE Controller Network module (1st module) controlled	S (initial)
SD1579	CC-Link IE Controller Network module channel number (2nd module)	CC-Link IE Controller Network module channel number (2nd module)	Channel number (151 to 158)*1 of the CC-Link IE Controller Network module (2nd module) controlled	S (initial)
SD1580	CC-Link IE Controller Network module channel number (3rd module)	CC-Link IE Controller Network module channel number (3rd module)	Channel number (151 to 158)*1 of the CC-Link IE Controller Network module (3rd module) controlled	S (initial)
SD1581	CC-Link IE Controller Network module channel number (4th module)	CC-Link IE Controller Network module channel number (4th module)	Channel number (151 to 158)*1 of the CC-Link IE Controller Network module (4th module) controlled	S (initial)
SD1582	CC-Link IE Controller Network module channel number (5th module)	CC-Link IE Controller Network module channel number (5th module)	Channel number (151 to 158)*1 of the CC-Link IE Controller Network module (5th module) controlled	S (initial)
SD1583	CC-Link IE Controller Network module channel number (6th module)	CC-Link IE Controller Network module channel number (6th module)	Channel number (151 to 158)*1 of the CC-Link IE Controller Network module (6th module) controlled	S (initial)

No.	Name	Description	Details	Set by (when to set)
SD1584	CC-Link IE Controller Network module channel number (7th module)	CC-Link IE Controller Network module channel number (7th module)	Channel number (151 to 158)*1 of the CC-Link IE Controller Network module (7th module) controlled	S (initial)
SD1585	CC-Link IE Controller Network module channel number (8th module)	CC-Link IE Controller Network module channel number (8th module)	Channel number (151 to 158)*1 of the CC-Link IE Controller Network module (8th module) controlled	S (initial)
SD1586	CC-Link IE Field Network module channel number (1st module)	CC-Link IE Field Network module channel number (1st module)	Channel number (181 to 188)*1 of the CC-Link IE Field Network module (1st module) controlled	S (initial)
SD1587	CC-Link IE Field Network module channel number (2nd module)	CC-Link IE Field Network module channel number (2nd module)	Channel number (181 to 188)*1 of the CC-Link IE Field Network module (2nd module) controlled	S (initial)
SD1588	CC-Link IE Field Network module channel number (3rd module)	CC-Link IE Field Network module channel number (3rd module)	Channel number (181 to 188)*1 of the CC-Link IE Field Network module (3rd module) controlled	S (initial)
SD1589	CC-Link IE Field Network module channel number (4th module)	CC-Link IE Field Network module channel number (4th module)	Channel number (181 to 188)*1 of the CC-Link IE Field Network module (4th module) controlled	S (initial)
SD1590	CC-Link IE Field Network module channel number (5th module)	CC-Link IE Field Network module channel number (5th module)	Channel number (181 to 188)*1 of the CC-Link IE Field Network module (5th module) controlled	S (initial)
SD1591	CC-Link IE Field Network module channel number (6th module)	CC-Link IE Field Network module channel number (6th module)	Channel number (181 to 188)*1 of the CC-Link IE Field Network module (6th module) controlled	S (initial)
SD1592	CC-Link IE Field Network module channel number (7th module)	CC-Link IE Field Network module channel number (7th module)	Channel number (181 to 188)*1 of the CC-Link IE Field Network module (7th module) controlled	S (initial)
SD1593	CC-Link IE Field Network module channel number (8th module)	CC-Link IE Field Network module channel number (8th module)	Channel number (181 to 188)*1 of the CC-Link IE Field Network module (8th module) controlled	S (initial)
SD1594	Switch status	CPU switch status (MODE/SELECT switch)	The switch status (MODE/SELECT) of the CPU module is stored as follows: 0: Neutral 1: MODE 2: SELECT	S (when the switch status is changed)
SD1596 SD1597	Module label ID	Module label ID	The module label ID written to a C Controller module is stored. • SD1596: Module label ID (lower 1 word) • SD1597: Module label ID (upper 1 word)	S (initial)

*1 Only the channel number in use can be checked. A module cannot be specified.

Appendix 8 Buffer Memory

This section shows buffer memory.

Buffer memory values are reset to default (initial values) when the power is turned OFF or the C Controller module is reset.

Precautions

Do not write any data in the "system area" of the buffer memory.

Doing so may cause malfunction of this product.

Address assignment

Address	Name
Un\G0 to Un\G1023	System area
Un\G1024 to Un\G1151	CC-Link IE Field Network Basic function
Un\G1152 to Un\G2047	System area

Buffer memory list

The following shows the list of the buffer memory of a C Controller module.

CC-Link IE Field Network Basic function

The buffer memory list of the CC-Link IE Field Network Basic function (Un\G1024 to Un\G1151) is as follows.

Address	Name
1024	Connected Count
1025	Reserved station specification status
1026 to 1029	Reserved station specification status of each station
1030 to 1045	Link scan information
1046 to 1049	System area
1050	Diagnostic information display request
1051	Diagnostic request information
1052	Diagnostic information status flag
1053 to 1067	Diagnostic information 1
1068 to 1083	Diagnostic information 2
1084 to 1151	System area

Buffer memory details

The following shows how to read the list of buffer memory details.

Item	Description
Address	Buffer memory address of a C Controller module
Name	Buffer memory name of a C Controller module
Content	Description of the buffer memory of a C Controller module
Set by (when to set)	The timing to set each device by a system and/or user. (Set by) • S: Set by system. • U: Set by a user (by a program, engineering tool, GOT, or test function from an external device). • U/S: Set by both user and system. (When to set) • Initial: Set only when initial processing is performed (e.g. powering ON the system, changing the operating status from STOP to RUN). • Status change❶: Set when the status is changed for each refresh cycle. • Status change❷: Set only when the status is changed. • At END processing: Set for every refresh cycle.

Point

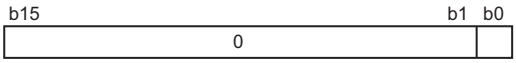
Do not change the buffer memory set by system by a program or device test. Otherwise, unintended operation may occur.

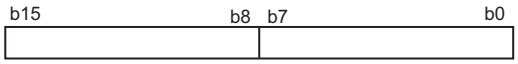
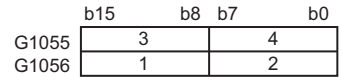
CC-Link IE Field Network Basic function

The details of buffer memory for the CC-Link IE Field Network Basic function are as follows.

Address	Name	Description	Set by (when to set)
Un to G1024	Connected Count	The total number of connected stations set in the parameter is stored. (Range: 1 to 64)	S (initial)
Un\G1025	Reserved station specification status	The reserved station specification status of the remote station specified in the parameter is stored. (0: Not specified, 1: Specified) b15 … b1 b0 0 • b0: Reserved station specification status • b1 to b15: Empty (fixed to 0)	S (initial)
Un\G1026 to Un\G1029	Reserved station specification status of each station	The reserved station specification status is stored with the following bit pattern. (OFF: Other than the reserved station, ON: Reserved station) b15 … b0 G1026: 16 15 14 13 12 ~ 5 4 3 2 1 G1027: 32 31 30 29 28 ~ 21 20 19 18 17 G1028: 48 47 46 45 44 ~ 37 36 35 34 33 G1029: 64 63 62 61 60 ~ 53 52 51 50 49 The numbers in the figure indicate station numbers (1 to 64). (Conditions) • Only the bit of the start station number turns ON. • The status is not stored for the station numbers after the maximum station number.	S (initial)

A

Address	Name		Description	Set by (when to set)
Un\G1030	Link scan information	Group No.1 maximum link scan	The maximum link scan time value during cyclic transmission is stored. (Unit: ms)	S (status change ❷)
Un\G1031		Group No.1 minimum link scan	The minimum link scan time value during cyclic transmission is stored. (Unit: ms)	S (status change ❷)
Un\G1032		Group No.1 current link scan	The current link scan time value during cyclic transmission is stored. (Unit: ms)	S (status change ❷)
Un\G1033		System area		
Un\G1034		Group No.2 maximum link scan	The maximum link scan time value during cyclic transmission is stored. (Unit: ms)	S (status change ❷)
Un\G1035		Group No.2 minimum link scan	The minimum link scan time value during cyclic transmission is stored. (Unit: ms)	S (status change ❷)
Un\G1036		Group No.2 current link scan	The current link scan time value during cyclic transmission is stored. (Unit: ms)	S (status change ❷)
Un\G1037		System area		
Un\G1038		Group No.3 maximum link scan	The maximum link scan time value during cyclic transmission is stored. (Unit: ms)	S (status change ❷)
Un\G1039		Group No.3 minimum link scan	The minimum link scan time value during cyclic transmission is stored. (Unit: ms)	S (status change ❷)
Un\G1040		Group No.3 current link scan	The current link scan time value during cyclic transmission is stored. (Unit: ms)	S (status change ❷)
Un\G1041		System area		
Un\G1042		Group No.4 maximum link scan	The maximum link scan time value during cyclic transmission is stored. (Unit: ms)	S (status change ❷)
Un\G1043		Group No.4 minimum link scan	The minimum link scan time value during cyclic transmission is stored. (Unit: ms)	S (status change ❷)
Un\G1044		Group No.4 current link scan	The current link scan time value during cyclic transmission is stored. (Unit: ms)	S (status change ❷)
Un\G1045		System area		
Un\G1046 to Un\G1049	System area			
Un\G1050	Diagnostic information display request		For every refresh cycle when a bit 0 is changed from OFF to ON, the diagnostic information of the remote station specified in 'Diagnostic request information' (Un\G1051) is read to 'Diagnostic information status flag' (Un\G1052) to 'Diagnostic information 2' (Un\G1083). When the setting of diagnostic information is completed at the completion of a link scan, b0 is turned OFF by the system.  • b0: Diagnostic information display request • b1 to b15: Empty (fixed to 0)	S (status change ❶)
Un\G1051	Diagnostic request information		Specify a remote station number whose diagnostic information is to be displayed. (Range: 1 to 64)	U

Address	Name		Description	Set by (when to set)
Un\G1052	Diagnostic information status flag		For every refresh cycle when the bit 0 of 'Diagnostic information display request' (Un\G1050) is changed from OFF to ON, the status of the diagnostic information (Diagnostic information 1, Diagnostic information 2) of the remote station specified in 'Diagnostic request information' (Un\G1051) is stored. (Valid: 1, Invalid: 0)  • b0 to b7: Diagnostic information 1 • b8 to b15: Diagnostic information 2 If the station number of the remote station specified in 'Diagnostic request information' (Un\G1051) is the start station number of occupied stations and the cyclic transmission is being performed for the remote station, '1' is stored in b0 to b7 and b8 to b15. (When the specified remote station is a reserved station, '0' is stored in b8 to b15.) If the station number of the remote station specified in 'Diagnostic request information' (Un\G1051) is other than the start station number of occupied stations or the cyclic transmission is not being performed for the remote station, '0' is stored in b0 to b7 and b8 to b15.	S (status change ❷)
Un\G1053	Diagnostic information 1*1	Number of occupied stations	When b0 to b7 of 'Diagnostic information status flag' (Un\G1052) are '1' (valid), number of occupied stations, group number, IP address, the accumulated number of timeouts, and the accumulated number of disconnection detection are stored. When invalid, '0' is stored. In addition, when a station number which does not exist in 'Diagnostic request information' (Un\G1051) is specified, the bits are cleared to '0.' ■G1053: Number of occupied stations ■G1054: Group number ■G1055, G1056: IP address (lower), IP address (upper)  • 1 to 4: First octet to fourth octet (When the IP address has not been set in the parameter, 0 is stored.) ■G1063: Accumulated number of timeouts • 0: No timeouts • 1 to 65535: Number of timeouts (accumulated number)*2 ■G1064: Accumulated number of disconnection detection • 0: No disconnections • 1 to 65535: Number of disconnection detections (accumulated number)*2	S (status change ❷)
Un\G1054		Group No.		
Un\G1055		IP address (lower)		
Un\G1056		IP address (upper)		
Un\G1057 to Un\G1062		System area		
Un\G1063		Accumulated number of timeouts		
Un\G1064		Accumulated number of disconnection detection		
Un\G1065 to Un\G1067		System area		
Un\G1068	Diagnostic information 2*1	Manufacturer code	When b8 to b15 of 'Diagnostic information status flag' (Un\G1052) are '1' (valid), the manufacturer code, model code, device version, module information, error code, and detailed module information are stored. When invalid, '0' is stored. In addition, when a station number which does not exist in 'Diagnostic request information' (Un\G1051) is specified, the bits are cleared to '0.' ■G1068: Manufacturer code ■G1070: Model code (lower) ■G1071: Model code (upper) ■G1071: Function version ■G1074: Module information ■G1075: Error code ■G1076: Detailed module information (lower) ■G1077: Detailed module information (upper)	S (status change ❷)
Un\G1069		System area		
Un\G1070		Model code (lower)		
Un\G1071		Model code (upper)		
Un\G1072		Function version		
Un\G1073		System area		
Un\G1074		Module information		
Un\G1075		Error code		
Un\G1076		Detailed module information (lower)		
Un\G1077		Detailed module information (upper)		
Un\G1078 to Un\G1083		System area		
Un\G1084 to Un\G1151	System area			

*1 If a specified remote station is being disconnected, the information immediately before the disconnection is stored.

*2 When the count exceeds 65535, counting is continued from 1 again.

Appendix 9 Processing Time of Functions

This section shows the time required for processing a function of dedicated function library used in user program, which is executed with the task priority set to 100.
For the specifications of the dedicated function library, refer to the following:
MELSEC iQ-R C Controller Module Programming Manual

Point

The processing time of each dedicated function library varies depending on the operation status of an operating system and system task, the execution status of a user program, and the module configuration of a C Controller module system.
Refer to the following processing time of each function as a guide.

C Controller module dedicated functions

I/O access time

When accessing an input module RX42C4 (input number: X0 and higher) in a single CPU system configuration

Function name		Access size	Access time
CCPU_X_In_WordEx	High speed	1 word	18 μs
		16 words	19 μs
		64 words	32 μs
	Normal	1 word	18 μs
		16 words	19 μs
		64 words	32 μs

Buffer memory access time

When accessing a CC-Link module RJ61BT11 (input number: X0 and higher) in a single CPU system configuration

Function name	Access size	Access time
CCPU_FromBuf	1 word	19 μs
	64 words	25 μs
	512 words	56 μs
CCPU_ToBuf	1 word	17 μs
	64 words	18 μs
	512 words	29 μs

CPU buffer memory access time

When accessing the buffer memory of the host CPU in a multiple CPU system configuration (CPU No.1: R120CPU, CPU No.2: C Controller module)

Function name	Access size	Access time
CCPU_FromBuf	1 word	10 μs
	64 words	15 μs
	512 words	44 μs
CCPU_ToBuf	1 word	10 μs
	64 words	14 μs
	512 words	21 μs

Appendix 10 Parameter List

This section shows the list of parameters.

System parameters

The list of system parameters is shown below.

Item			Parameter No.
I/O Assignment	Base/Power/Extension Cable Setting	Setting of base/power supply/extension cable model name	0203H
		Number of slots	0201H
	I/O Assignment	Setting of type/number of points/start XY/module status	0200H
		Module name	0203H
		Specification of control CPU	0202H
	Setting of Points Occupied by Empty Slot		0100H
Multiple CPU Setting	Setting for Number of CPU Modules		0301H
	Communication Setting between CPU	Refresh Setting	0303H
		CPU Buffer Memory Setting (when refresh END)	0304H
		CPU Buffer Memory Setting (when refresh I45 is executed)	0308H
		PLC Unit Data	0309H
		Fixed Scan Communication Function	—
		Fixed Scan Communication Area Setting	0307H
	Fixed Scan Communication Setting	Fixed Scan Interval Setting of Fixed Scan Communication	0306H
		Fixed Scan Communication Function and Inter-module Synchronization Function	0306H
	Operation Mode Setting	Stop Setting	0302H
		Synchronous Startup Setting	030AH
	Other PLC Control Module Setting	I/O Setting Outside Group	0305H
Inter-module Synchronization Setting	Use Inter-module Synchronization Function in System		—
	Select Inter-module Synchronization Target Module		0101H
	Fixed Scan Interval Setting of Inter-module Synchronization		0101H
	Inter-module Synchronization Master Setting		0102H

Point

In a multiple CPU system configuration, the system parameters in all CPUs should be the same.
The CPUs, in which "Not Use" is selected for "Fixed Scan Communication Function" or "Inter-module Synchronization Function," are not regarded as same system parameters. Set the same system parameter settings in all CPUs in the multiple CPU configuration.

A

CPU parameters

The list of CPU parameters is shown below.

Item			Parameter No.
Name Setting	Title Setting		3100H
	Comment Setting		3101H
Operation Related Setting	Remote Reset Setting		3202H
	Output Mode Setting at STOP to RUN		3203H
	Module Synchronous Setting		3207H
	Clock Related Setting		3209H
	Refresh Cycle Setting		6E02H
Memory/Device Setting	Link Direct Device Setting		340DH
RAS Setting	WDT Setting		3500H
	Error Detections Setting		3501H
	CPU Module Operation Setting at Error Detected		3501H
	LED Display Setting		3502H
	Event History Setting		3504H
Routing Setting	Routing Setting		3800H
Service Settings	Service Settings		6E00H
	Security password settings		6E01H
MELSEC data link function settings	Timeout value setting	MELSEC iQ-R series bus interface (Channel No.12)	6E03H
		CC-Link IE Controller Network (Channel No.151 to 158)	6E06H
		CC-Link IE Field Network(Channel No.181 to 188)	6E07H
		CC-Link(Channel No.81 to 88)	6E04H
		MELSECNET/H(Channel No.51 to 54)	6E05H
		CC-Link IE TSN(Channel No.281 to 288)	6E08H

Module parameters

The list of module parameters is shown below.

Item			Parameter No.
Basic Settings	Own Node Settings		A012H
	External Device Configuration		A031H
Application Settings	FTP Server Settings		A037H
	Time Setting		A039H
	Security		A034H
	Telnet Server Settings		A03BH
I/O Assignment	Base/Power/Extension Cable Setting	Base	7002H
	I/O Assignment	Module name	7000H
		Slot/Start XY	7002H
Input module setting	Input response time setting		7102H
	Interrupt setting		7800H
	Refresh Settings		—
Output module setting	Output Mode upon CPU Error		7101H
	Refresh Settings		—
Intelligent function module setting	Basic Settings		7100H
	Application Settings		7200H
	Interrupt setting		7800H
	Refresh Settings		7400H
Network module setting	Required Settings		7100H
	Station Type		7700H
	Basic Settings		7310H
	Refresh Settings		7401H
	Application Settings		7311H
	Interrupt setting		7800H
	Interlink Transmission Settings		7500H
CC-Link IEF Basic setting	Network Configuration Settings	—	7A00H
		Link Scan Setting	7A02H
	Activated ethernet port		7920H
	Link Scan Time Settings		
	Refresh Settings		7420H

Memory card parameters

This section shows the list of memory card parameters.

Item		Parameter No.
Boot Setting	Boot File Setting	2000H
Setting of File/Data Use or Not in Memory Card	Setting of File/Data Use or Not in Memory Card	2010H

A

Appendix 11 VxWorks Component List

This section shows the list of the VxWorks components in a C Controller module.

Description	Name (component list)
Altera SoC Gen 5 Fpga Manager support	DRV_ALT_SOC_GEN5_FPGA_MGR
__thread variables support	INCLUDE_TLS
Ability to restart/reset tasks	INCLUDE_TASK_RESTART
Address Space Allocator Show Routines	INCLUDE_ADR_SPACE_SHOW
address space shell commands	INCLUDE_ADR_SPACE_SHELL_CMD
AIM MMU Show Routines	INCLUDE_AIM_MMU_SHOW
Altera Dw EMAC Enhanced Network Driver	INCLUDE_ALT_SOC_GEN5_DW_END
Altera QSPI support	INCLUDE_ALT_SOC_GEN5_QSPI
Altera SoC Gen 5 DesignWare I2C support	INCLUDE_ALT_SOC_GEN5_DW_I2C
Altera SoC Gen 5 timer driver	INCLUDE_ALT_SOC_GEN5_TIMER
ANSI abort	INCLUDE_ANSI_ABORT
ANSI abs function	INCLUDE_ANSI_ABS
ANSI assert (default)	INCLUDE_ANSI_ASSERT
ANSI atof function	INCLUDE_ANSI_ATOF
ANSI ctype (default)	INCLUDE_ANSI_CTYPE
ANSI errno to error string conversion function (default)	INCLUDE_ANSI_STRERROR
ANSI extension - case insensitive string compare function	INCLUDE_ANSI_STRCASECMP
ANSI extension - case insensitive string compare function - first N characters	INCLUDE_ANSI_STRNCASECMP
ANSI locale	INCLUDE_ANSI_LOCALE
ANSI longjmp	INCLUDE_ANSI_LONGJMP
ANSI math (default)	INCLUDE_ANSI_MATH
ANSI memchr function	INCLUDE_ANSI_MEMCHR
ANSI memcmp function	INCLUDE_ANSI_MEMCMP
ANSI memcpy function	INCLUDE_ANSI_MEMCPY
ANSI memmove function	INCLUDE_ANSI_MEMMOVE
ANSI memset function	INCLUDE_ANSI_MEMSET
ANSI stdio (default)	INCLUDE_ANSI_STDIO
ANSI stdio extensions	INCLUDE_ANSI_STDIO_EXTRA
ANSI stdlib (default)	INCLUDE_ANSI_STDLIB
ANSI stdlib bsearch function	INCLUDE_ANSI_BSEARCH
ANSI stdlib string to number conversion	INCLUDE_ANSI_STDLIB_NUMBERS
ANSI string (default)	INCLUDE_ANSI_STRING
ANSI string cat function	INCLUDE_ANSI_STRCAT
ANSI string cat function	INCLUDE_ANSI_STRNCAT
ANSI string compare function	INCLUDE_ANSI_STRCMP
ANSI string compare function - first N characters	INCLUDE_ANSI_STRNCMP
ANSI string copy function	INCLUDE_ANSI_STRCPY
ANSI string copy function - first N characters	INCLUDE_ANSI_STRNCPY
ANSI string duplication function (default)	INCLUDE_ANSI_STRDUP
ANSI string length function	INCLUDE_ANSI_STRLEN
ANSI strlcpy function	INCLUDE_ANSI_STRLCPY
ANSI strtod function	INCLUDE_ANSI_STRTOD
ANSI time (default)	INCLUDE_ANSI_TIME
application initialization (default)	INCLUDE_USER_APPL
ARM Generic Interrupt Controller driver	DRV_ARM_GIC
arp utility wrapper	INCLUDE_IPWRAP_ARP
arpLib	INCLUDE_ARP_API
asynchronous IO show routine	INCLUDE_POSIX_AIO_SHOW
atomic operators support	INCLUDE_ATOMIC_OPERATORS

Description	Name (component list)
Attach END to IPv4	INCLUDE_IPATTACH
AUX clock	INCLUDE_AUX_CLK
Basic IO system	INCLUDE_IO_BASIC
basic memory allocator	INCLUDE_MEM_MGR_BASIC
basic MMU (default)	INCLUDE_MMU_BASIC
basic network support (default)	INCLUDE_NETWORK
binary semaphore creation routine (default)	INCLUDE_SEM_BINARY_CREATE
binary semaphores (default)	INCLUDE_SEM_BINARY
Boot parameter process (default)	INCLUDE_NET_BOOT
BSP Memory Configuration	INCLUDE_MEMORY_CONFIG
builti-in symbol table (default)	INCLUDE_STANDALONE_SYM_TBL
c line interpreter	INCLUDE_SHELL_INTERP_C
C++ compiler support routines (default)	INCLUDE_CPLUS_LANG
C++ core runtime (default)	INCLUDE_CPLUS
C++ iostreams and other standard library facilities (default)	INCLUDE_CPLUS_IOSTREAMS
C++ symbol demangler	INCLUDE_CPLUS_DEMANGLER
cache support	INCLUDE_CACHE_SUPPORT
cfiamdmtd	INCLUDE_MTD_CFIAMD
class show routine	INCLUDE_CLASS_SHOW
command line interpreter	INCLUDE_SHELL_INTERP_CMD
Common network infrastructure (default)	INCLUDE_COMMON_NET
Commonly used legacy mbuf routines	INCLUDE_MBUF_UTIL1
coprocessor	INCLUDE_COPROCESSOR
coprocessor show routine	INCLUDE_COPROCESSOR_SHOW
Core NFS client	INCLUDE_CORE_NFS_CLIENT
counting semaphore creation routine (default)	INCLUDE_SEM_COUNTING_CREATE
counting semaphores (default)	INCLUDE_SEM_COUNTING
CRYPTO (default)	INCLUDE_IPCRYPTO
debug shell commands	INCLUDE_DEBUG_SHELL_CMD
debugging facilities (default)	INCLUDE_DEBUG
Default SMP scheduler policy (default)	INCLUDE_SMP_SCHED_DEFAULT_POLICY
Device Manager	INCLUDE_DEVICE_MANAGER
DHCP Client	INCLUDE_IPDHCPC
DNS Client	INCLUDE_IPDNSC
DOS File System Consistency Checker	INCLUDE_DOSFS_CHKDSK
DOS File System FAT12/16/32 Handler (default)	INCLUDE_DOSFS_FAT
DOS File System Old Directory Format Handler	INCLUDE_DOSFS_DIR_FIXED
DOS File System VFAT Directory Handler (default)	INCLUDE_DOSFS_DIR_VFAT
DOS File System Volume Fomatter Module	INCLUDE_DOSFS_FMT
DOS filesystem backward-compatibility	INCLUDE_DOSFS
Dos FS BIO buffer size (default)	INCLUDE_DOSFS_VOL_BIO_BUFFER_SIZE
Dos FS Cache Handler	INCLUDE_DOSFS_CACHE
dosfs File System Main Module (dosFs2) (default)	INCLUDE_DOSFS_MAIN
Dos FS Show Routines (default)	INCLUDE_DOSFS_SHOW
doubly linked lists	INCLUDE_DLL
dynamic creation and deletion of tasks	INCLUDE_TASK_CREATE_DELETE
ED&R Policy Hooks (default)	INCLUDE_EDR_POLICY_HOOKS
ED&R shell commands	INCLUDE_EDR_SHELL_CMD
ED&R show routines	INCLUDE_EDR_SHOW
ED&R system debug flag	INCLUDE_EDR_SYSDBG_FLAG
EHCI	INCLUDE_EHCI
EHCI Init	INCLUDE_EHCI_INIT
enable caches	INCLUDE_CACHE_ENABLE

Description	Name (component list)
enable guard pages for kernel task stacks	INCLUDE_PROTECT_TASK_STACK
enable non-executable kernel task stacks	INCLUDE_TASK_STACK_NO_EXEC
END driver polled statistics support	INCLUDE_END_POLLED_STATS
END: common Enhanced Network Device support (default)	INCLUDE_END_COMMON
END: END-style interface support	INCLUDE_END
Error detection and reporting stub (always present).	INCLUDE_EDR_STUB
error status table	INCLUDE_STAT_SYM_TBL
Ethernet Interface support (default)	INCLUDE_IPCOM_USE_ETHERNET
Ethernet multicast library support	INCLUDE_ETHERNET
Event Reporting Framework	INCLUDE_ERF
eventpoint stub library	INCLUDE_EVENTPOINT_STUB
eventpoints library	INCLUDE_EVENTPOINT
exception handling (default)	INCLUDE_EXC_HANDLING
exception show routines	INCLUDE_EXC_SHOW
exception task (default)	INCLUDE_EXC_TASK
Extended Block Device	INCLUDE_XBD
extended object library	INCLUDE_OBJ_OPEN
Fast, non-deterministic ISR callable spin locks	INCLUDE_SPINLOCK_ISR_ND
File System and Disk Utilities	INCLUDE_DISK_UTIL
File System Event Utilities	INCLUDE_FS_EVENT_UTIL
File System IO	INCLUDE_IO_FILE_SYSTEM
File System Monitor	INCLUDE_FS_MONITOR
file system shell commands	INCLUDE_DISK_UTIL_SHELL_CMD
file upload path initialization	INCLUDE_WVUPLOAD_FILE
Firewall	INCLUDE_IPFIREWALL
floating point show routine	INCLUDE_HW_FP_SHOW
formatted IO (default)	INCLUDE_FORMATTED_IO
formatted IO output routines (default)	INCLUDE_FORMATTED_OUT_BASIC
fpp formatting for printf (default)	INCLUDE_FLOATING_POINT
ftllite	INCLUDE_TL_FTL
FTP Client Backend	INCLUDE_FTP
full featured memory allocator (default)	INCLUDE_MEM_MGR_FULL
General BSP macros (default)	INCLUDE_BSP_MACROS
Generic data collector library	INCLUDE_DATACOLLECTOR
Generic PHY driver	INCLUDE_GENERICPHY
get name info	INCLUDE_GETNAMEINFO
get servce by name	INCLUDE_GETSERVBYNAME
get servce by port	INCLUDE_GETSERVBYPORT
gethostbyaddr wrapper	INCLUDE_IPWRAP_GETHOSTBYADDR
getifaddrs wrapper	INCLUDE_IPWRAP_GETIFADDRS
getnameinfo wrapper	INCLUDE_IPWRAP_GETNAMEINFO
getopt function	INCLUDE_GETOPT
getservbyname wrapper	INCLUDE_IPWRAP_GETSERVBYNAME
getservbyport wrapper	INCLUDE_IPWRAP_GETSERVBYPORT
Global configurations	INCLUDE_IPAIP_GLOBAL_CONFIGS
GNU compiler support routines (default)	INCLUDE_GNU_INTRINSICS
GTF support	INCLUDE_GTF
gtf_timer_start	INCLUDE_GTF_TIMER_START
handle show routines	INCLUDE_HANDLE_SHOW
hardware fpp support	INCLUDE_HW_FP
hash library	INCLUDE_HASH
high resolution timestamping	INCLUDE_TIMESTAMP
Highly Reliable File System (default)	INCLUDE_HRFS

Description	Name (component list)
hook function table show support	INCLUDE_HOOK_SHOW
hook function table support	INCLUDE_HOOKS
host table (default)	INCLUDE_HOST_TBL
host table sysctl support	INCLUDE_HOST_TBL_SYSCTL
Host/target breakpoint synchronization	INCLUDE_WDB_BP_SYNC
host/target modules and symbols synchronization	INCLUDE_WDB_MDL_SYM_SYNC
HRFS Default Write Mode (default)	INCLUDE_HRFS_DEFAULT_WRITE_MODE
HRFS File System Consistency Checker	INCLUDE_HRFS_CHKDSK
HRFS Format	INCLUDE_HRFS_FORMAT
I2C generic device vxBus driver	DRV_I2C_GENERIC_DEV
ifconfig	INCLUDE_IFCONFIG
ifconfig wrapper	INCLUDE_IPWRAP_IFCONFIG
ifLib wrapper	INCLUDE_IPWRAP_IFLIB
ifShow wrapper	INCLUDE_IPWRAP_IFSHOW
INCLUDE_APOLLO_COMMON	INCLUDE_APOLLO_COMMON
INCLUDE_QSERIES_COMPATIBLE	INCLUDE_QSERIES_COMPATIBLE
INCLUDE_R12CCPU	INCLUDE_R12CCPU
inetLib	INCLUDE_INETLIB
inetLib wrapper	INCLUDE_IPWRAP_INETLIB
Init pre-kernel memory allocation globally (default)	INIT_HWMEMPOOL_GLOBAL
system symbol table initialization (default)	INCLUDE_SYM_TBL_INIT
Intel ICH SATA Controller	INCLUDE_DRV_STORAGE_INTEL_ICH
Intel ICH SATA Controller Show Routines	INCLUDE_DRV_STORAGE_INTEL_ICH_SHOW
Intel PRO/1000 VxBus Enhanced Network Driver (default)	INCLUDE_GEI825XX_VXB_END
Inter-Integrated Circuit Bus	INCLUDE_I2C_BUS
IO system (default)	INCLUDE_IO_SYSTEM
IP v4	INCLUDE_IPV4
IPCOM arp commands	INCLUDE_IPARP_CMD
IPCOM Firewall commands	INCLUDE_IPFIREWALL_CMD
IPCOM ifconfig commands	INCLUDE_IPIFCONFIG_CMD
IPCOM ipd commands	INCLUDE_IPD_CMD
IPCOM netstat commands	INCLUDE_IPNETSTAT_CMD
IPCOM ping commands	INCLUDE_IPPING_CMD
IPCOM radius client commands	INCLUDE_IPRADIUS_CMD
IPCOM RAM Disk Support	INCLUDE_IPCOM_USE_RAM_DISK
IPCOM route commands	INCLUDE_IPROUTE_CMD
IPCOM shell command interface	INCLUDE_IPCOM_SHELL_CMD
IPCOM sysctl commands	INCLUDE_IPSYSCTL_CMD
IPCOM sysvar commands	INCLUDE_IPCOM_SYSVAR_CMD
IPCOM uses native VxWorks file system (default)	INCLUDE_IPCOM_FS_NATIVE
IPNET (default)	INCLUDE_IPNET
IPNet loopback configuration	INCLUDE_IPNET_LOOPBACK_CONFIG
IPNet Stack	INCLUDE_IPNET_STACK
IPNet sysctl integration	INCLUDE_IPNET_SYSCTL
ipProto wrapper	INCLUDE_IPWRAP_IPPROTO
IPv4 (default)	INCLUDE_IPCOM_USE_INET
IPv4 AutoIP	INCLUDE_IPAIP
IPv4 Multicast routing	INCLUDE_IPNET_USE_MCAST_ROUTING
ISR deferral	INCLUDE_ISR_DEFER
Job Queue support (default)	INCLUDE_JOB_QUEUE
job task (default)	INCLUDE_JOB_TASK
kernel (default)	INCLUDE_KERNEL
kernel shell startup script	INCLUDE_STARTUP_SCRIPT

Description	Name (component list)
linkBufPool (default)	INCLUDE_LINKBUFPOOL
linked list library	INCLUDE_LSTLIB
Loopback Interface support (default)	INCLUDE_IPNET_USE_LOOPBACK
M_BLK ethernet/802.3 header build and parse	INCLUDE_END_ETHER_HDR
mapped files shell commands	INCLUDE_MAPPED_FILES_SHOW_SHELL_CMD
memory allocator info routines (default)	INCLUDE_MEM_MGR_INFO
Memory mapping	INCLUDE_MMAP
memory show routine	INCLUDE_MEM_SHOW
message queue info routines	INCLUDE_MSG_Q_INFO
message queue show routine	INCLUDE_MSG_Q_SHOW
message queues (default)	INCLUDE_MSG_Q
message queue creation and deletion library (default)	INCLUDE_MSG_Q_CREATE_DELETE
MIB2 ICMP Management APIs	INCLUDE_MIB2_ICMP
MIB2 IF Counter Instrumentation	INCLUDE_MIB2_IF
MIB2 TCP Management APIs	INCLUDE_MIB2_TCP
MIB2 UDP Management APIs	INCLUDE_MIB2_UDP
MII bus controller module	INCLUDE_MII_BUS
Miscellaneous IO	INCLUDE_IO_MISC
MMU global map (default)	INCLUDE_MMU_GLOBAL_MAP
module manager	INCLUDE_MODULE_MANAGER
mutex semaphore creation routine (default)	INCLUDE_SEM_MUTEX_CREATE
mutex semaphores (default)	INCLUDE_SEM_MUTEX
MUX common support (all service and device styles) (default)	INCLUDE_MUX_COMMON
MUX mux2Bind() service (defalt)	INCLUDE_MUX2
MUX mux2Bind() service / END-style device	INCLUDE_MUX2_OVER_END
MUX muxTkBind() service (default)	INCLUDE_MUXTK
MUX muxTkBind() service / END-style device	INCLUDE_MUXTK_OVER_END
MUX private support for M_BLK/Ipcom_pkt conversion (default)	INCLUDE_VXMUX_MBLK
MUX mux2Bind() service (default)	INCLUDE_MUX
NAT (default)	INCLUDE_IPNET_USE_NAT
message logging (default)	INCLUDE_LOGGING
netBufLib	INCLUDE_NETBUFLIB
netBufLib show routines (default)	INCLUDE_NETPOOLSHOW
netBufPool (default)	INCLUDE_NETBUFPOOL
NetDrv for remote IO (default)	INCLUDE_NET_DRV
Netlink socket	INCLUDE_IPNET_USE_NETLINKSOCK
netstat	INCLUDE_NETSTAT
netstat wrapper	INCLUDE_IPWRAP_NETSTAT
network boot device configuration	INCLUDE_NET_BOOT_CONFIG
Network Daemon Support (default)	INCLUDE_NET_DAEMON
network device netmask setup (default)	INCLUDE_NETMASK_GET
Network host show routines	INCLUDE_NET_HOST_SHOW
network init	INCLUDE_NET_INIT
network remote I/O access (default)	INCLUDE_NET_REM_IO
Network Stack Memory Pool Configuration	INCLUDE_NET_POOL
NETWORK SYSCTL (default)	INCLUDE_NET_SYSCTL
NFS client All	INCLUDE_NFS_CLIENT_ALL
NFS server	INCLUDE_CORE_NFS_SERVER
NFS server All	INCLUDE_NFS_SERVER_ALL
NFS server v2	INCLUDE_NFS2_SERVER
NFS server v3	INCLUDE_NFS3_SERVER
NFS v2 client	INCLUDE_NFS2_CLIENT
NFS v3 client	INCLUDE_NFS3_CLIENT

Description	Name (component list)
nullBufPool (default)	INCLUDE_VXMUX_NULLBUFPOOL
object information	INCLUDE_OBJ_INFO
object management (default)	INCLUDE_OBJ_LIB
object management ownership	INCLUDE_OBJ_OWNERSHIP
object show routines	INCLUDE_OBJECT_SHOW
oldRouteLib wrapper	INCLUDE_IPWRAP_OLDROUTELIB
PCI Bus legacy Auto Configuration Routines	INCLUDE_PCI_OLD_CONFIG_ROUTINES
PCI Bus Show Routines	INCLUDE_PCI_BUS_SHOW
Peripheral Component Interconnect Bus	INCLUDE_PCI_BUS
persistent error log	INCLUDE_EDR_ERRLOG
persistent memory	INCLUDE_EDR_PM
PING client	INCLUDE_PING
ping wrapper	INCLUDE_IPWRAP_PING
pipes (default)	INCLUDE_PIPES
pool allocation library	INCLUDE_POOL
POSIX advisory file locking (default)	INCLUDE_POSIX_ADVISORY_FILE_LOCKING
POSIX AIO driver (default)	INCLUDE_POSIX_AIO_SYSDRV
POSIX APIs for file systems.	INCLUDE_POSIX_FS
POSIX asynchoronous IO (default)	INCLUDE_POSIX_AIO
POSIX clocks (default)	INCLUDE_POSIX_CLOCKS
POSIX directory utilities (default)	INCLUDE_POSIX_DIRLIB
POSIX ftruncate (default)	INCLUDE_POSIX_FTRUNC
POSIX IO	INCLUDE_IO_POSIX
POSIX Memory Mapped Files	INCLUDE_POSIX_MAPPED_FILES
POSIX message queue show routine	INCLUDE_POSIX_MQ_SHOW
POSIX message queues (default)	INCLUDE_POSIX_MQ
POSIX mman	INCLUDE_POSIX_MEM
POSIX process scheduling	INCLUDE_POSIX_SCHED
POSIX scheduling policies SCHED_FIFO/SCHED_RR/SCHED_OTHER support in RTPs (default)	INCLUDE_PX_SCHED_DEF_POLICIES
POSIX semaphore show routine	INCLUDE_POSIX_SEM_SHOW
POSIX semaphores (default)	INCLUDE_POSIX_SEM
POSIX Shared Memory Objects	INCLUDE_POSIX_SHM
POSIX signals (default)	INCLUDE_POSIX_SIGNALS
POSIX thread CPU-time clock	INCLUDE_POSIX_THREAD_CPUTIME
POSIX thread scheduler in RTPs	INCLUDE_POSIX_PTHREAD_SCHEDULER
POSIX threads (default)	INCLUDE_POSIX_PTHREADS
Posix timer show component	INCLUDE_POSIX_TIMER_SHOW
POSIX timers (default)	INCLUDE_POSIX_TIMERS
POSIX TRACE	INCLUDE_POSIX_TRACE
POSIX trace timestamp	INCLUDE_POSIX_TRACE_TIMESTAMP
Pre-Kernel Memory Allocation	INCLUDE_HWMEM_ALLOC
process shell commands	INCLUDE_RTP_SHELL_CMD
Process shell commands.	INCLUDE_RTP_SHELL_C
process show shell commands	INCLUDE_RTP_SHOW_SHELL_CMD
Processor Local Bus (default)	INCLUDE_PLB_BUS
ProxyARP	INCLUDE_IPPROXYARP
Pseudo terminal driver	INCLUDE_PTYDRV
public hostname setup (default)	INCLUDE_NET_HOST_SETUP
Radius Authenication Support	INCLUDE_IPCOM_USE_AUTH_RADIUS
Radius client	INCLUDE_IPRADIUS
RAM Disk	INCLUDE_RAM_DISK
RAM disk driver	INCLUDE_RAMDRV

Description	Name (component list)
raw filesystem	INCLUDE_RAWFS
rBuff library (default)	INCLUDE_RBUFF
rBuff show routine	INCLUDE_RBUFF_SHOW
read the bootline	INCLUDE_BOOT_LINE_INIT
reader/writer semaphores	INCLUDE_SEM_READ_WRITE
reader/writer semaphore creation routine	INCLUDE_SEM_READ_WRITE_CREATE
Remote Command (default)	INCLUDE_REMLIB
Remote Command sysctl support	INCLUDE_REMLIB_SYSCTL
Removable IO	INCLUDE_IO_REMOVABLE
ring buffers	INCLUDE_RING_BUF
routec	INCLUDE_ROUTECMD
routec wrapper	INCLUDE_IPWRAP_ROUTECMD
Routing socket support (default)	INCLUDE_IPNET_USE_ROUTESOCK
RPC	INCLUDE_RPC
RTP (default)	INCLUDE_RTP
RTP getnameinfo() sysctl support	INCLUDE_GETNAMEINFO_SYSCTL
RTP Hook Support	INCLUDE_RTP_HOOKS
RTP IO	INCLUDE_IO_RTP
RTP Show	INCLUDE_RTP_SHOW
RTP Startup Facility: Command shell startup script	INCLUDE_RTP_APPL_INIT_CMD_SHELL_SCRIPT
RTP Startup Facility: User-defined code	INCLUDE_RTP_APPL_USER
run static initializers	INCLUDE_CTORS_DTORS
SD Host Controller driver	DRV_STORAGE_SD
select (default)	INCLUDE_SELECT
semaphore deletion routines (default)	INCLUDE_SEM_DELETE
semaphore exchenge routine	INCLUDE_SEM_EXCHANGE
semaphore info routines	INCLUDE_SEM_INFO
semaphore show routine	INCLUDE_SEM_SHOW
Serial line connection commands	INCLUDE_TIP_CMD
shared data region support in RTPs or kernel	INCLUDE_SHARED_DATA
Shared Data Show	INCLUDE_SHARED_DATA_SHOW
shared data show shell commands	INCLUDE_SHARED_DATA_SHOW_SHELL_CMD
shared library commands	INCLUDE_SHL_SHELL_CMD
Shared Library Show	INCLUDE_SHL_SHOW
shared library support in RTPs	INCLUDE_SHL
shell banner (default)	INCLUDE_SHELL_BANNER
show routine component (default)	INCLUDE_SHOW_ROUTINES
Show routines for memory mapped objects	INCLUDE_MAPPED_FILES_SHOW
sigevent notification library	INCLUDE_SIGEVENT
signals (default)	INCLUDE_SIGNALS
simple banner containing VxWorks version & creation date	INCLUDE_SIMPLE_BANNER
single linked lists	INCLUDE_SLL
SIO	INCLUDE_SIO
Sio Channel Utilities	INCLUDE_SIO_UTILS
SNTP Client (API)	INCLUDE_IPSNTPC_API
SNTP Client (daemon)[*1]	INCLUDE_IPSNTPC[*1]
SNTP common configurations	INCLUDE_IPSNTP_COMMON
sntpcTimeGet wrapper	INCLUDE_IPWRAP_SNTPCTIMEGET
Socket API (default)	INCLUDE_SOCKLIB
Socket API System Call support	INCLUDE_SC_SOCKLIB
Socket backend (default)	INCLUDE_IPNET_USE_SOCK_COMPAT
Socket support (default)	INCLUDE_IPNET_SOCKET
software fpp support	INCLUDE_SW_FP

Description	Name (component list)
spinLock (default)	INCLUDE_SPINLOCK
spy	INCLUDE_SPY
Spy CPU activity commands	INCLUDE_SPY_SHELL_CMD
Stack/Application Logging Utility	INCLUDE_APPL_LOG_UTIL
stdio (default)	INCLUDE_STDIO
stdio show routine	INCLUDE_STDIO_SHOW
Support for reboot hooks (default)	INCLUDE_REBOOT_HOOKS
symbol shell commands	INCLUDE_SYM_SHELL_CMD
symbol table show routine	INCLUDE_SYM_TBL_SHOW
SYNOPSYS HCI	INCLUDE_SYNOPSYSHCI
SYNOPSYS HCI Init	INCLUDE_SYNOPSYSHCI_INIT
SYSCTL (default)	INCLUDE_SYSCTL
SYSCTL CLI	INCLUDE_SYSCTL_CLI
SYSCTL H/W	INCLUDE_SYSCTL_HW
System Address Space Allocator	INCLUDE_ADR_SPACE_LIB
System Call Hook Support	INCLUDE_SYSCALL_HOOKS
System clock	INCLUDE_SYSCLK_INIT
system debug flag	INCLUDE_SYSDBG_FLAG
System Viewer class instrumentation (default)	INCLUDE_WINDVIEW_CLASS
System Viewer data collector library	INCLUDE_SV_DATACOLLECTOR
System Viewer library (default)	INCLUDE_WINDVIEW
system-defined timestamping	INCLUDE_SYS_TIMESTAMP
target loader (default)	INCLUDE_LOADER
target loader shell command	INCLUDE_MODULE_SHELL_CMD
target symbol table (default)	INCLUDE_SYM_TBL
target unloader	INCLUDE_UNLOADER
target-resident kernel shell (default)	INCLUDE_SHELL
task create hooks	INCLUDE_TASK_CREATE_HOOKS
task hook show routine	INCLUDE_TASK_HOOKS_SHOW
task hooks (default)	INCLUDE_TASK_HOOKS
task info routines	INCLUDE_TASK_INFO
task list management	INCLUDE_TASK_LIST
task shell commands	INCLUDE_TASK_SHELL_CMD
task show routine	INCLUDE_TASK_SHOW
task switch hooks	INCLUDE_TASK_SWITCH_HOOKS
task utility routines (default)	INCLUDE_TASK_UTIL
TCP (default)	INCLUDE_IPTCP
TELNET Server	INCLUDE_IPTELNETS
system-level password protection	INCLUDE_SECURITY
terminal driver (default)	INCLUDE_TTY_DEV
terminal driver support	INCLUDE_TYLIB
TFTP Client	INCLUDE_IPTFTPC
TFTP client APIs	INCLUDE_TFTP_CLIENT
TFTP common configurations	INCLUDE_IPTFTP_COMMON
timex	INCLUDE_TIMEX
tip serial line connection utility	INCLUDE_TIP
Transactional Block Layer	INCLUDE_XBD_TRANS
TSFS upload path initialization	INCLUDE_WVUPLOAD_TSFSSOCK
TrueFFS Flash File System	INCLUDE_TFFS
TrueFFS Show Routines	INCLUDE_TFFS_SHOW
UART support for ns16550-compatible devices	DRV_SIO_NS16550
unix compatible environment variables (default)	INCLUDE_ENV_VARS
unloader shell command	INCLUDE_UNLOADER_SHELL_CMD

Description	Name (component list)
USB Common Stack	INCLUDE_USB
USB Common Stack Init	INCLUDE_USB_INIT
USB GEN2 Helper Init	INCLUDE_USB_GEN2_HELPER
USB GEN2 Mass Storage	INCLUDE_USB_GEN2_STORAGE
USB GEN2 Mass Storage Init	INCLUDE_USB_GEN2_STORAGE_INIT
USB Host Class Driver Init	INCLUDE_USB_HOST_CLASS_INIT
USB Host Controller Start	INCLUDE_HCD_BUS
Use Authentication	INCLUDE_IPCOM_USE_AUTH
Uses native VxWorks shell	INCLUDE_USE_NATIVE_SHELL
Vector Floating Point	INCLUDE_VFP
vi-like editing mode	INCLUDE_SHELL_VI_MODE
VIO driver (default)	INCLUDE_WDB_VIO
virtual memory show shell commands	INCLUDE_VM_SHOW_SHELL_CMD
Virtual Root File Sytem	INCLUDE_VRFS
VLAN Pseudo Interface support	INCLUDE_IPNET_USE_VLAN
VM library show routine	INCLUDE_VM_SHOW
vxBus Aux Clk Support	INCLUDE_VXB_AUX_CLK
VxBus Device Table	VXBUS_TABLE_CONFIG
vxBus Driver DMA System	INCLUDE_DMA_SYS
vxBus Driver Parameter System	INCLUDE_PARAM_SYS
VxBus Interrupt Controller Library	INCLUDE_INTCTLR_LIB
VxBus Legacy Interrupt Support	INCLUDE_VXB_LEGACY_INTERRUPTS
vxBus subsystem (default)	INCLUDE_VXBUS
vxBus subsystem show routines	INCLUDE_VXBUS_SHOW
vxBus Sys Clk Support	INCLUDE_VXB_SYS_CLK
vxBus Timer Support	INCLUDE_TIMER_SYS
VxBus Timestamp Support	INCLUDE_VXB_TIMESTAMP
vxIpiLib	INCLUDE_VXIPI
vxMemProbe initializer for exception handler support (default)	INCLUDE_VXMEMPROBE_INIT
VxWorks debug library	INCLUDE_VXDBG
VxWorks events	INCLUDE_VXEVENTS
VxWorks IPCOM	INCLUDE_IPCOM
watchdog timer show routine	INCLUDE_WATCHDOGS_SHOW
watchdog timers (default)	INCLUDE_WATCHDOGS
watchdog timers creation and deletion library (default)	INCLUDE_WATCHDOGS_CREATE_DELETE
WDB agent (default)	INCLUDE_WDB
WDB banner (default)	INCLUDE_WDB_BANNER
WDB breakpoints (default)	INCLUDE_WDB_BP
WDB call functions (default)	INCLUDE_WDB_FUNC_CALL
WDB callouts (default)	INCLUDE_WDB_DIRECT_CALL
WDB dynamic printf	INCLUDE_WDB_DPRINTF
WDB eventpoints (default)	INCLUDE_WDB_EVENTPOINTS
WDB events (default)	INCLUDE_WDB_EVENTS
WDB exception notification (default)	INCLUDE_WDB_EXC_NOTIFY
WDB gopher (default)	INCLUDE_WDB_GOPHER
WDB is always enabled (default)	INCLUDE_WDB_ALWAYS_ENABLED
WDB memory access (default)	INCLUDE_WDB_MEM
WDB network connection	INCLUDE_WDB_COMM_NETWORK
WDB post kernel initialization (default)	INCLUDE_WDB_POST_KERNEL_INIT
WDB register access (default)	INCLUDE_WDB_REG
WDB RTP breakpoints	INCLUDE_WDB_RTP_BP
WDB RTP control support	INCLUDE_WDB_RTP_CONTROL
WDB RTP support	INCLUDE_WDB_RTP

Description	Name (component list)
WDB target server file system	INCLUDE_WDB_TSFS
WDB task breakpoints	INCLUDE_WDB_TASK_BP
WDB task creation (default)	INCLUDE_WDB_START_NOTIFY
WDB task debugging (default)	INCLUDE_WDB_TASK
WDB task exit notification (default)	INCLUDE_WDB_EXIT_NOTIFY
WDB task hooks	INCLUDE_WDB_TASK_HOOKS
WDB task registers	INCLUDE_WDB_TASK_REG
WDB tasks (default)	INCLUDE_WDB_CTXT
WDB user event (default)	INCLUDE_WDB_USER_EVENT
WDB virtual I/O library (default)	INCLUDE_WDB_VIO_LIB
write-project program text	INCLUDE_PROTECT_TEXT
XBD Block Device	INCLUDE_XBD_BLK_DEV
XBD Disk Partition Handler	INCLUDE_XBD_PART_LIB
XBD Ram Drive	INCLUDE_XBD_RAMDRV
XDR	INCLUDE_XDR

*1 Disabled for an R12CCPU-V with firmware version '15' or later.

Appendix 12 Communication Examples Using Serial Communication Module

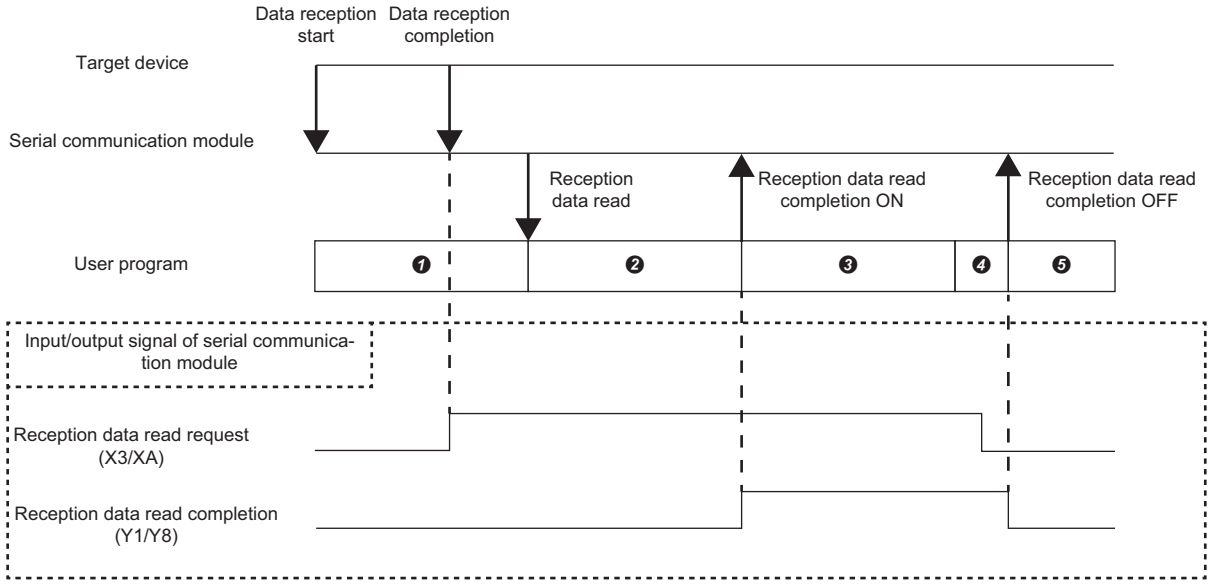
When communicating by using a serial communication module, data communication using nonprocedural protocol is required. This section shows the communication examples using a serial communication module in a C Controller module.

Point

For details on sample programs, please contact your local Mitsubishi Electric sales office or representative.

Receiving data from target devices

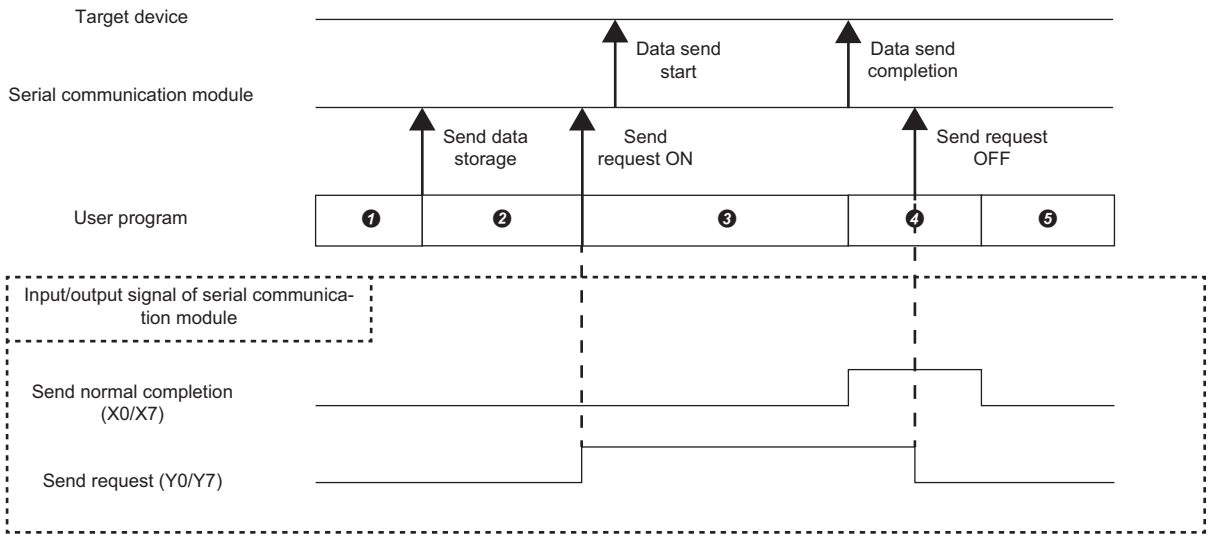
The following shows the communication example when receiving data.



❶: The reception data read request (X3/XA) is monitored with the C Controller module dedicated function (CCPU_X_In_BitEx). When receiving data from the target device, the reception data read request (X3/XA) turns ON.
❷: The reception data is read from the buffer memory for reception with the C Controller module dedicated function (CCPU_FromBuf) after the reception data read request (X3/XA) turns ON.
❸: The reception data read completion (Y1/Y8) turns ON with the C Controller module dedicated function (CCPU_Y_Out_BitEx).
❹: The reception data read request (X3/XA) is monitored with the C Controller module dedicated function (CCPU_X_In_BitEx). Wait until the signal turns OFF.
❺: The reception data read completion (Y1/Y8) turns OFF with the C Controller module dedicated function (CCPU_Y_Out_BitEx).

Sending data to target devices

The following shows the communication example when sending data.



❶: The send data is stored in the buffer memory for sending with the C Controller module dedicated function (CCPU_ToBuf).

❷: The send request (Y0/Y7) turns ON with the C Controller module dedicated function (CCPU_Y_Out_BitEx). The data is sent to the target device.

❸: The send normal completion (X0/X7) is monitored with the C Controller module dedicated function (CCPU_X_In_BitEx). Wait until the signal turns ON.

❹: The send request (Y0/Y7) turns OFF with the C Controller module dedicated function (CCPU_Y_Out_BitEx).

❺: The send normal completion (X0/X7) turns OFF.

A

Appendix 13 How to Use MELSEC-Q Series Modules

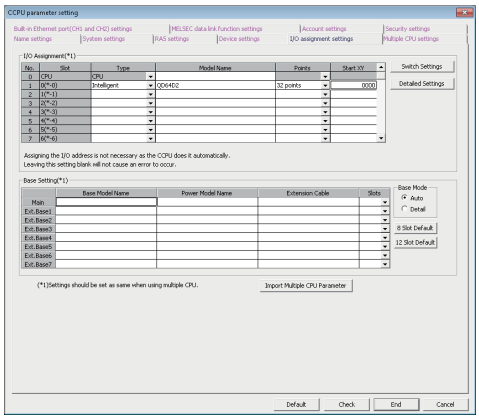
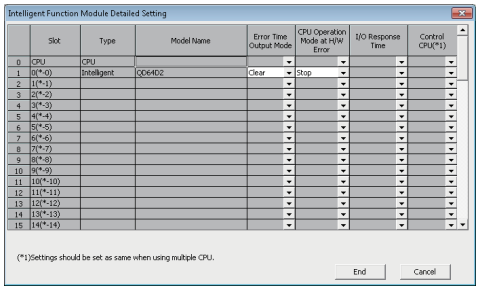
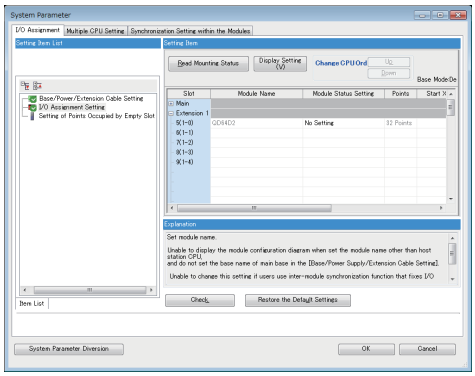
This section explains how to use MELSEC-Q series modules.

Screen change in CW Configurator

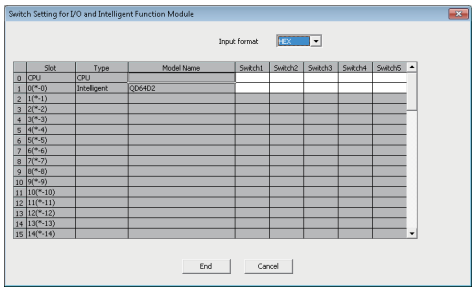
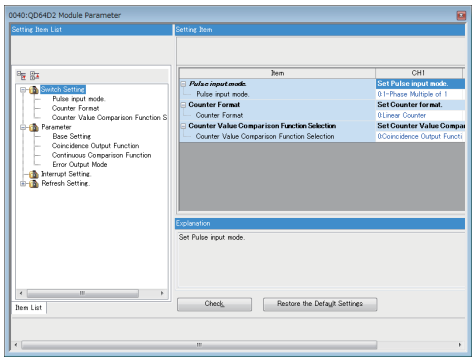
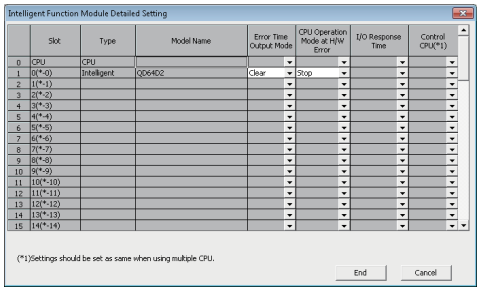
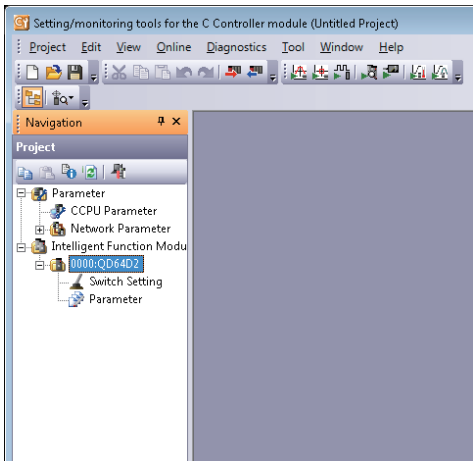
Setting items in Setting/monitoring tools for the C Controller module used for MELSEC-Q series are changed to the following screens in CW Configurator.

Parameter items

■"Points," "Start XY," and "PLC Operation Mode at H/W Error" in "I/O assignment settings"

Setting/monitoring tools for the C Controller module	CW Configurator
[Navigation window] ⇨ [CCPU Parameter] ⇨ [I/O assignment settings] tab  [Navigation window] ⇨ [CCPU Parameter] ⇨ [I/O assignment settings] tab ⇨ [Detailed Settings] button 	[Navigation window] ⇨ [Parameter] ⇨ [System Parameter] ⇨ [I/O Assignment Setting] tab 

■"Switch Setting" and "Error Time Output Mode" in "I/O assignment setting," and intelligent function module parameters

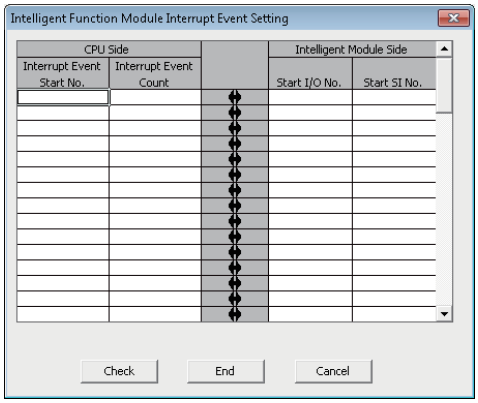
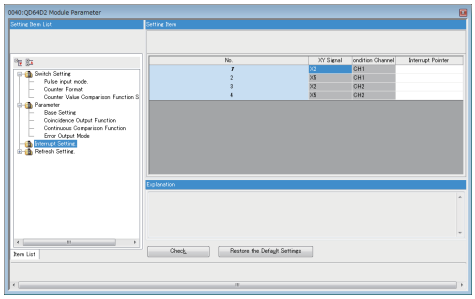
Setting/monitoring tools for the C Controller module	CW Configurator
[Navigation window] ⇨ [CCPU Parameter] ⇨ [I/O assignment settings] tab ⇨ [Switch Setting]	[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [(module name)] ⇨ [Module Parameter]
	
[Navigation window] ⇨ [CCPU Parameter] ⇨ [I/O assignment settings] tab ⇨ [Detailed Settings] button	
	
[Navigation window] ⇨ [Intelligent Function Module] ⇨ [(module name)]	
	

A

■Interrupt pointer setting

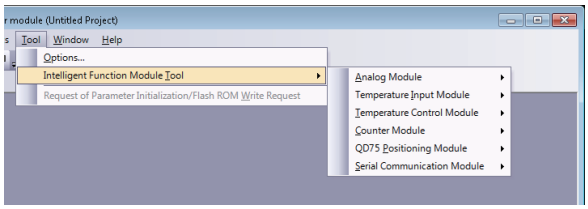
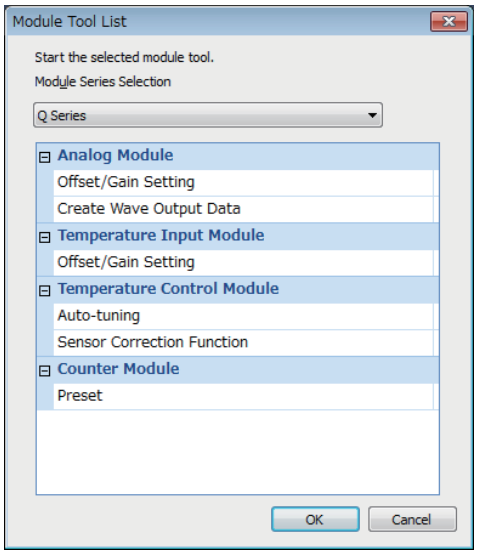
In CW Configurator, the setting for interrupt pointer is changed as follows.

- Setting the SI No. for an interrupt module is not required.
- "Interrupt Pointer" of an interrupt module must be sequential numbers. (Set I51 and I52 to No.2 and No.3 respectively when I50 is set to No.1, for example.)

Setting/monitoring tools for the C Controller module	CW Configurator
[Navigation window] ⇨ [Parameter] ⇨ [CCPU Parameter] ⇨ [System settings] tab ⇨ [Intelligent Function Module Setting] ⇨ [Interrupt event settings] button 	[Navigation window] ⇨ [Parameter] ⇨ [Module Information] ⇨ [(module name)] ⇨ [Module Parameter] 

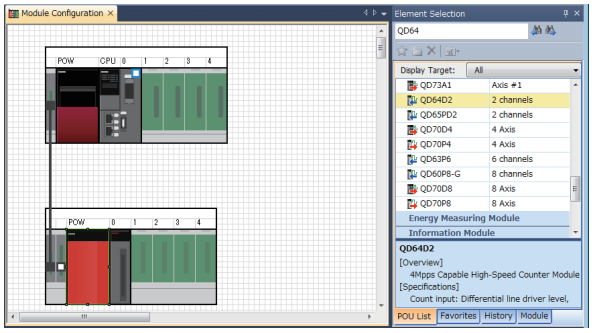
Intelligent function module tool

"Intelligent Function Module Tool" is changed to the following screen.

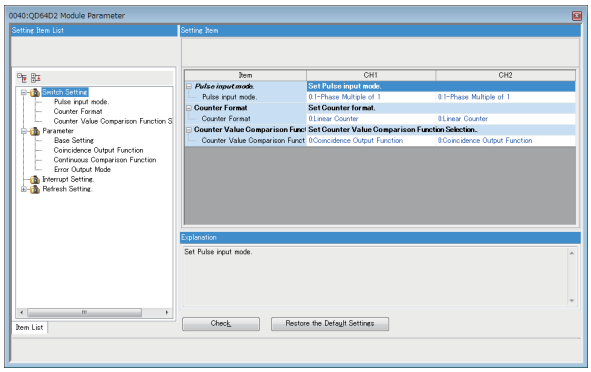
Setting/monitoring tools for the C Controller module	CW Configurator
[Tool] ⇨ [Intelligent Function Module Tool] ⇨ [(tool for each module)] 	[Tool] ⇨ [Module Tool List] ⇨ [Q Series] ⇨ [(tool for each module)] 

Setting procedure

The setting procedure for a MELSEC-Q series module in CW Configurator is shown below.



1. Create a module configuration in accordance with the actual configuration by dragging each MELSEC-Q series module from the element selection window and dropping it onto the module configuration diagram.



2. Set the parameters of each module.

Point

- When CW Configurator can be connected to the actual system configuration, the configuration can be read by selecting [Online] ⇨ [Read Module Configuration from PLC]. For modules of which configuration cannot be read, drag each module from the element selection window and drop it on the module configuration diagram.
- To change "Points," "Start XY," "Control PLC Settings," or "CPU Module Operation Setting at Error Detected," set the parameters in "System Parameter."

A

Considerations

The modules requiring particular attentions are shown below.
For details, refer to the manual for each module.

I/O module

■High-speed input module

- When the module is used as a high-speed input module, select the following model in accordance with the ON/OFF of a noise filter. The module operates with the default setting when the ON/OFF of the noise filter does not correspond to the ON/OFF of the model.

Module name	Model name displayed on CW Configurator	
	Noise filter: ON	Noise filter: OFF
QX40H	QX40H (NoiseF_ON)	QX40H (NoiseF_OFF)
QX70H	QX70H (NoiseF_ON)	QX70H (NoiseF_OFF)
QX80H	QX80H (NoiseF_ON)	QX80H (NoiseF_OFF)
QX90H	QX90H (NoiseF_ON)	QX90H (NoiseF_OFF)

- When the module is used as an interrupt module, select "QI60." The module operates with the default setting when it is used with the noise filter set to OFF.
- When using one or more modules as interrupt modules, set the number of modules to be used in "I/O Assignment Setting" of the "System Parameter" screen.

■Interrupt module

When using one or more modules, set the number of modules to be used in "I/O Assignment Setting" of the "System Parameter" screen.

High-speed counter module and pulse input module

Check the latest error code in the system monitor since error codes for QD63P6 and QD60P8-G cannot be checked by performing the module diagnostics. When more than one error occurred, refer to the buffer memory.

Positioning module

- Set parameters in a user program because the positioning data parameter cannot be written to the module.
- Check the latest error code in the system monitor since error codes for QD70P4, QD70P8, QD70D4, and QD70D8 cannot be checked by performing the module diagnostics. When more than one error occurred, refer to the buffer memory (error codes for each axis).

Loop control module

Check the latest error code in the system monitor since error codes for Q62HLC cannot be checked by performing the module diagnostics. When more than one error occurred, refer to the buffer memory.

FL-net interface module

- Select the following module name displayed on the module configuration diagram.

Module name	Model name displayed on CW Configurator
QJ71FL71	QJ71FL71(-T -B5)(-F01)
QJ71FL71-F01	
QJ71FL71-B5	
QJ71FL71-B5-F01	
QJ71FL71-T	
QJ71FL71-T-F01	
QJ71FL71-B2	QJ71FL71-B2(-F01)
QJ71FL71-B2-F01	

- Select "Online" for "Operation Mode Setting" under "Switch Setting" in the "Module Parameter" screen.
- The size setting of area 1 for the cyclic data set in bit units in Setting/monitoring tools for the C Controller module changes to the setting in word units in CW Configurator.

MODBUS interface module and MODBUS/TCP interface module

- Before setting "Start MODBUS Device No." of "MODBUS Device Assignment Parameter" in the "Module Parameter" screen, check the setting range in the manual in advance.
- For "Basic parameter starting method" and "MODBUS device assignment parameter starting method" of "Switch setting" in "Basic setting," only "Start with the user-set parameters" can be selected. Write a program to set the automatic communication parameters as well as module parameters to which only "Switch Setting" is set in order to start with the default assignment parameters.

AnyWireASLINK master module and AnyWire DB A20 master module

When AnyWire DB A20 master module is used, select the following model displayed on the module configuration diagram.

Module name	Model name displayed on CW Configurator
QJ51AW12D2	MELSEC_Partner

Q series large type extension base unit (AnS series size)

Select the module which can be mounted on the actual each extension base unit on the module configuration diagram. When a Q series large type blank cover is used, select the module mounted on the Q series large type blank cover.

Appendix 14 When the Parameters of Intelligent Function Modules are not Enabled

To enable some parameters of intelligent function modules, an initialization instruction (turn the power of a specific device OFF and ON, and OFF again) is required for the module every time a C Controller module[*1] starts.

*1 For a C Controller module with firmware version '06' or later, the parameters are enabled automatically.

For details on intelligent function modules, refer to the manual for each module.

Methods for enabling parameters

The following two methods are available for enabling parameters (turn the power of a device OFF and ON, and OFF again).

- Describe the processing to turn the power of a device OFF and ON, and OFF again in a user program, and write it to a C Controller module.
- Describe the processing to turn the power of a device OFF and ON, and OFF again in a script file (STARTUP.CMD), and write it to a C Controller module.

■Example of script file (STARTUP.CMD) description

When turning OFF and ON, and OFF again the device (Y9) of the module whose start XY is 00C0

```
CCPU_Y_Out_BitEx(0,0xC9,0)      /*(1)*/
CCPU_Y_Out_BitEx(0,0xC9,1)      /*(2)*/
CCPU_Y_Out_BitEx(0,0xC9,0)      /*(3)*/
```

(1) Turns the device (Y9) of the module OFF.
(2) Turns the device (Y9) of the module ON.
(3) Turns the device (Y9) of the module OFF.

Corresponding device list

The following tables show the devices corresponding to an initialization instruction to an intelligent function module.

MELSEC iQ-R series

Product name	Model name	Device	Signal name
Analog-digital converter module	R60AD4 R60ADI8 R60ADV8	Y9	Operating condition setting request
Channel isolated analog-digital converter module	R60AD8-G R60AD16-G	Y9	Operating condition setting request
Digital-analog converter module	R60DA4 R60DAI8 R60DAV8	Y9	Operating condition setting request
Channel isolated digital-analog converter module	R60DA8-G R60DA16-G	Y9	Operating condition setting request
Channel isolated RTD input module	R60RD8-G	Y9	Operating condition setting request
Channel isolated thermocouple input module	R60TD8-G	Y9	Operating condition setting request

MELSEC-Q series

Product name	Model name	Device	Signal name
Analog-digital converter module	Q64AD Q68ADV Q68ADI	Y9	Operating condition setting request
Channel isolated high resolution analog-digital converter module	Q64AD-GH	Y9	Operating condition setting request
Channel isolated high resolution analog-digital converter module (with signal conditioning function)	Q62AD-DGH	Y9	Operating condition setting request
Channel isolated analog-digital converter module	Q68AD-G	Y9	Operating condition setting request
Channel isolated analog-digital converter module (with signal conditioning function)	Q66AD-DG	Y9	Operating condition setting request
High speed analog-digital converter module	Q64ADH	Y9	Operating condition setting request
Digital-analog converter module	Q62DAN Q64DAN Q68DAVN Q68DAIN Q62DA Q64DA Q68DAV Q68DAI	Y9	Operating condition setting request
Channel isolated digital-analog converter module	Q62DA-FG Q66DA-G	Y9	Operating condition setting request
High speed digital-analog converter module	Q64DAH	Y9	Operating condition setting request
Analog input/output module	Q64AD2DA	Y9	Operating condition setting request
Load cell input module	Q61LD	Y9	Operating condition setting request
Current transformer input module	Q68CT	Y9	Operating condition setting request
RTD input module	Q64RD	Y9	Operating condition setting request
Channel isolated RTD input module	Q68RD3-G Q64RD-G	Y9	Operating condition setting request
Thermocouple input module	Q64TD	Y9	Operating condition setting request
Channel isolated thermocouple/micro voltage input module	Q64TDV-GH	Y9	Operating condition setting request
Channel isolated thermocouple input module	Q68TD-G-H01 Q68TD-G-H02	Y9	Operating condition setting request
Temperature control module	Q64TCTTN Q64TCRTN Q64TCTTBWN Q64TCRTBWN	YB	Setting change instruction

Appendix 15 Added and Changed Functions

This section shows the functions added and changed for a C Controller module and CW Configurator, the applicable firmware version for a C Controller module, and applicable software version for CW Configurator.

Point

- For the method for checking the versions, refer to the following:
  Firmware version: MELSEC iQ-R Module Configuration Manual
  Software version: CW Configurator Operating Manual
- By using the firmware update function, the firmware version of a CPU module can be updated. For details on the target CPU modules and firmware update, refer to the following:
  MELSEC iQ-R Module Configuration Manual

—: Not related to the firmware version or software version

Added and changed function	Firmware version	Software version	Reference
USB devices are supported.	'03' or later	—	☞ Page 55 Peripheral Configuration ☞ Page 74 Memory Configuration ☞ Page 77 Files
Multiple CPU system configuration with a programmable controller CPU controlling a MELSECNET/H network module is supported.	'03' or later	'1.001B' or later	MELSEC iQ-R Module Configuration Manual
The following modules are supported: • Process CPU • Channel isolated analog-digital converter module • Channel isolated digital-analog converter module • Channel isolated RTD input module • Channel isolated thermocouple input module	'04' or later	'1.002C' or later	MELSEC iQ-R Module Configuration Manual
MELSEC iQ-R series modules which occupy two slots are supported.			
CC-Link IE Controller Network using RJ71EN71 is supported.			MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup) MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)
The label communication function is supported.		—	☞ Page 85 Label Communication Function
The following modules are supported: • Extended temperature range main base unit • Extended temperature range extension base unit • MES interface module • High speed analog-digital converter module • Temperature control module • MELSECNET/H network module	'06' or later	'1.004E' or later	MELSEC iQ-R Module Configuration Manual
The daylight saving time function is supported.			☞ Page 105 Daylight saving time function
Label communication with R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, and R120ENCPU is supported.		—	☞ Page 85 Label Communication Function
Device access to R04ENCPU, R08ENCPU, R08SFCPU, R16ENCPU, R16SFCPU, R32ENCPU, R32SFCPU, R120ENCPU, and R120SFCPU is supported.			☞ Page 84 Device Access Function
The data analysis function is supported.	'07' or later	—	☞ Page 87 Data Analysis Function
The following modules are supported: • DC high-speed input module (RX61C6HS) • Transistor high-speed output module (RY41NT2H) • Flexible high-speed I/O control module	'07' or later	'1.006G' or later	MELSEC iQ-R Module Configuration Manual
Multiple CPU system using a safety CPU is supported.	'08' or later	'1.004E' or later	MELSEC iQ-R Module Configuration Manual
The CC-Link IE Field Network Basic function is supported.		—	☞ Page 165 CC-Link IE Field Network Basic FUNCTION
The firmware update function is supported.	'09' or later	—	MELSEC iQ-R Module Configuration Manual

Added and changed function	Firmware version	Software version	Reference
The following modules are supported: • Power supply modules (R62P and R64P) • DC high-speed input modules (RX40PC6H and RX40NC6H) • High speed data logger module (RD81DL96) • C intelligent function module (RD55UP06-V) • CANopen module (RJ71CN91) • Device net master/slave module (RJ71DN91) • Simple motion modules (RD77GF4, RD77GF8, RD77GF16, and RD77GF32)	'10' or later	'1.009K' or later	MELSEC iQ-R Module Configuration Manual
The following modules are supported: • CC-Link IE TSN module (RJ71GN11-T2) • Motion modules (RD78G4, RD78G8, RD78G16, RD78G32, RD78G64, RD78GHV, and RD78GHW) • MES interface module (RD81MES96N) • High speed data communication module (RD81DC96) • Laser displacement sensor control modules (R60MH112 and R60MH112NA) • CC-Link IE Controller Network module (RJ71GP21S-SX) • GP-IB interface module (RJ71GB91) • Channel isolated analog-digital converter modules (R60AD6-DG and R60AD6-DG(Q)) • C intelligent function module (RD55UP12-V) • Main base unit (R33B)	'15' or later	'1.012N' or later	MELSEC iQ-R Module Configuration Manual
The link direct device mode setting is supported.			☞ Page 203 Link direct device mode setting
The motion module control function is supported.			☞ Page 131 MOTION MODULE CONTROL FUNCTION
Extension of points for CC-Link IE Controller Network is supported.	'16' or later	'1.014Q' or later	☞ Page 203 Link direct device mode setting
The following module is supported: • CC-Link IE TSN module (RJ71GN11-SX)	'18' or later	'1.017T' or later	MELSEC iQ-R Module Configuration Manual
The following modules are supported: • Power supply module (R69P) • AC input module (RX10-TS) • DC input modules (RX40C7-TS, RX41C4-TS) • Contact output module (RY10R2-TS) • Transistor output modules (RY40NT5P-TS, RY40PT5P-TS, RY41NT2P-TS, RY41PT1P-TS) • Temperature control modules (R60TCTRT2TT2-TS, R60TCRT4-TS)	'19' or later	'1.018U or later	MELSEC iQ-R Module Configuration Manual
The system timer mode switching function is supported.	'20' or later	—	☞ Page 130 System Timer Mode Switching Function

MEMO

INDEX

B

Bus master CPU 99
BUS RUN LED 23

C

CARD RDY LED 23
Checking file attribute 188
Clearing error 115
Clearing value 76
Clock data 103
Connecting and disconnecting USB Mass Storage Class-compliant Device 36
CPU built-in memory 77

D

Data communication with CPU buffer memory . . 234
Data communication with fixed cycle communication area. 234
Default IP setting 29
Dot matrix LED 23,30

E

Endian format 44,60
ERROR LED 23,316
Ethernet cable 64
Ethernet port 23,60
Event history file 118
External dimensions 60

F

File attribute 188
FTP 319

G

GOT 68

H

Hardware diagnostics 30,31

I

Initialization 76
Inserting/removing an SD memory card 35
Inter-module synchronous interrupt program . . . 123
IP address 65

M

MODE/SELECT switch 23
Module extended parameter. 96
Module initialization setting 29
Multiple CPU synchronous interrupt program . . . 248

N

Number of occupied I/O points 60

O

Operating status 37
Operation processing in PAUSE state. 73
Operation processing in RUN state. 72
Operation processing in STOP state. 72
Operation selection mode 38
OS 60
Output mode setting of STOP to RUN. 93

P

Parameter No. 381
PING. 318
POWER LED 316
Product information marking 23

R

READY LED 23,317,318
Refresh cycle 121
RESET/STOP/RUN switch 23
RS SD/RD LED 23
RS-232 connector. 23,60,66

S

SD memory card 23,77
SD memory card slot. 24,60
SD/RD LED 23
Security password 29
Setting file attribute 188
Setting fixed cycle communication area. 237
Setting refresh area 236
SNTP 163
SNTP server 156
SPEED LED. 23
System watchdog timer 108

T

Telnet 319
Time zone 104

U

USB cable 67
USB connector 23,60
USB Mass Storage Class-compliant device . . . 23,77
USB RDY LED 23
USER LED. 23
User watchdog timer 108

I

MEMO

REVISIONS

*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
January 2023	SH(NA)-082558ENG-A	First edition
October 2023	SH(NA)-082558ENG-B	■Added or modified parts GENERIC TERMS AND ABBREVIATIONS, Section 2.1, Appendix 15
June 2024	SH(NA)-082558ENG-C	■Added or modified parts Section 9.4, Chapter 10, Section 11.7, Section 14.1, Section 14.2, Section 14.3, Section 14.4, Section 14.5, Appendix 3, Appendix 6, Appendix 7, Appendix 8, Appendix 15
January 2025	SH(NA)-082558ENG-D	■Added or modified parts Section 2.1, Chapter 10, Section 11.13, Section 26.10, Section 26.13, Section 26.16, Appendix 15

Japanese manual number: SH-082557-D

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2023 MITSUBISHI ELECTRIC CORPORATION

CONDITIONS OF USE FOR THE PRODUCT

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

(2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries.
MITSUBISHI ELECTRIC SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI ELECTRIC USER'S, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.
("Prohibited Application")
Prohibited Applications include, but not limited to, the use of the PRODUCT in;
- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
- Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
- Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.

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

(3) Mitsubishi Electric shall have no responsibility or liability for any problems involving programmable controller trouble and system trouble caused by DoS attacks, unauthorized access, computer viruses, and other cyberattacks.

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

(1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.

(2) Even within the gratis warranty term, repairs shall be charged for in the following cases.

1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
2. Failure caused by unapproved modifications, etc., to the product by the user.
3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

(1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.

(2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to:

(1) Damages caused by any cause found not to be the responsibility of Mitsubishi.

(2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.

(3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.

(4) Replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

INFORMATION AND SERVICES

For further information and services, please contact your local Mitsubishi Electric sales office or representative.
Visit our website to find our locations worldwide.

MITSUBISHI ELECTRIC Factory Automation Global Website
Locations Worldwide
www.MitsubishiElectric.com/fa/about-us/overseas/

TRADEMARKS

Microsoft and Windows are trademarks of the Microsoft group of companies.
Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
VxWorks, and Wind River are either registered trademarks or trademarks of Wind River Systems, Inc.
Unicode is either a registered trademark or a trademark of Unicode, Inc. in the United States and other countries.
The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.
In some cases, trademark symbols such as '™' or '®' are not specified in this manual.

COPYRIGHTS

The screens (screenshots) are used in accordance with the Microsoft Corporation guideline.

SH(NA)-082558ENG-D(2501)KWIX
MODEL: R-CCPU-U-E

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA 461-8670, JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.