MITSUBISHI
ELECTRIC

Mitsubishi Programmable Controller

MELSEC iQ-R series

MELSEC iQ-R C Controller Module
Programming Manual

SAFETY PRECAUTIONS

(Read these precautions before using this product.)

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

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

CONDITIONS OF USE FOR THE PRODUCT

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

i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and

ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.

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

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

("Prohibited Application")

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

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

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

CONSIDERATIONS FOR USE

Considerations for the Wind River Systems product

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

INTRODUCTION

Thank you for purchasing the Mitsubishi MELSEC iQ-R series programmable controllers.
This manual describes the functions required for programming.
Before using the product, please read this manual and relevant manuals carefully and develop familiarity with the performance of MELSEC iQ-R series programmable controller to handle the product correctly.
Please make sure that the end users read this manual.

CONTENTS

SAFETY PRECAUTIONS 1
CONDITIONS OF USE FOR THE PRODUCT 1
CONSIDERATIONS FOR USE 2
INTRODUCTION 2
RELEVANT MANUALS 5
TERMS 6

CHAPTER 1 COMMON ITEMS 7

1.1 Header Files 7
1.2 C Controller Module Dedicated Functions 8
Program processing 8
Considerations 8
Argument specification 9
1.3 MELSEC Data Link Functions 10
Program processing 10
Considerations 11
Accessible range and devices 13
Argument specification 37
1.4 Considerations on Interrupt Service Routine (ISR) 43

CHAPTER 2 FUNCTION LIST 44

2.1 C Controller Module Dedicated Functions 44
C Controller module dedicated functions 44
C Controller module dedicated functions for ISR 46
2.2 MELSEC Data Link Functions 47

CHAPTER 3 DETAILS OF FUNCTION 48

3.1 C Controller Module Dedicated Functions 48
C Controller module dedicated functions 48
C Controller module dedicated functions for ISR 119
3.2 MELSEC Data Link Functions 140

CHAPTER 4 ERROR CODE LIST 168

4.1 Common Error Codes 168
4.2 C Controller Module Dedicated Functions 172
4.3 MELSEC Data Link Functions 177
4.4 Error Codes Different From Conventional Functions 180

APPENDIX 181

Appendix 1 Example for Replacing Ladder by C Language 181
Program example 181
Appendix 2 How to Replace an Existing Product 183
Replacement of projects 183
Replacement of VxWorks standard API functions 183
Replacement of functions 183
Replacement of device type 183
Compilation of replaced project 186

Appendix 3 Correspondence Table to Conventional Functions 187
C Controller module dedicated functions. 187
C Controller module dedicated function for ISR 187
Bus interface functions 188
Bus interface functions for ISR 189
MELSEC data link functions 190

INDEX 192

FUNCTION INDEX 194

REVISIONS. 196
WARRANTY 197
TRADEMARKS 198

RELEVANT MANUALS

Manual name [manual number]	Description	Available form
MELSEC iQ-R C Controller Module Programming Manual [SH-081371ENG] (this manual)	Explains the programming specifications and dedicated function libraries of C Controller module.	e-Manual PDF
MELSEC iQ-R C Controller Module User's Manual (Startup) [SH-081367ENG]	Explains the performance specifications, module startup procedure, and troubleshooting of C Controller module.	Print book e-Manual PDF
MELSEC iQ-R C Controller Module User's Manual (Application) [SH-081369ENG]	Explains the functions, devices, and parameters of C Controller module.	Print book e-Manual PDF
CW Workbench/CW-Sim Operating Manual [SH-081373ENG]	Explains the system configuration, specifications, functions, and troubleshooting of CW Workbench/CW-Sim.	e-Manual PDF
CW Configurator Operating Manual [SH-081382ENG]	Explains the system configuration, parameter settings, and operation methods for the online function of CW Configurator.	e-Manual PDF

Point

e-Manual refers to the Mitsubishi FA electronic book manuals that can be browsed using a dedicated tool.
e-Manual has the following features:
- Required information can be cross-searched in multiple manuals.
- Other manuals can be accessed from the links in the manual.
- The hardware specifications of each part can be found from the product figures.
- Pages that users often browse can be bookmarked.

TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
Basic model QCPU	A generic term for Q00JCPU, Q00CPU, and Q01CPU
C Controller module	An abbreviation for MELSEC iQ-R series C Controller module
C Controller module dedicated function	A dedicated function library offered by C Controller module It is used to control the C Controller module.
CC-Link IE Controller Network module	An abbreviation for CC-Link IE Controller Network module, RJ71GP21-SX
CC-Link IE Field Network module	An abbreviation for CC-Link IE Field Network module, RJ71GF11-T2
CC-Link module	An abbreviation for CC-Link module, RJ61BT11
CW Configurator	A generic product name for model names, SWnDND-RCCPU ('n' indicates version.)
CW Workbench	An abbreviation for C Controller module and C intelligent function module engineering tool, CW Workbench.
CW-Sim	An abbreviation for VxWorks simulator that can operate and debug the C Controller module and C intelligent function module programs on a personal computer on which CW Workbench installed, without connecting to the actual machine (target).
Dedicated function library	A generic term for C Controller module dedicated functions and MELSEC data link functions
Existing product	A generic term for Q12DCCPU-V (Basic mode/Extended mode)
High Performance model QCPU	A generic term for Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, and Q25HCPU
Intelligent function module	A generic term for modules which has functions other than input and output, such as A/D converter module and D/A converter module
MELSEC data link function	A data link function library offered by C Controller module. It is used to access other CPU modules as a connection target via network or in a multiple CPU system.
Process CPU	A generic term for Q02PHCPU, Q06PHCPU, Q12PHCPU, and Q25PHCPU
R12CCPU-V	An abbreviation for R12CCPU-V C Controller module.
RCPU	A generic term for R04CPU, R04ENCPU, R08CPU, R08PCPU, R08ENCPU, R08SFCPU, R16CPU, R16PCPU, R16ENCPU, R16SFCPU, R32CPU, R32PCPU, R32ENCPU, R32SFCPU, R120CPU, R120PCPU, R120ENCPU, and R120SFCPU.
Redundant CPU	A generic term for Q12PRHCPU and Q25PRHCPU
Universal model QCPU	A generic term for Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q03UDVCPU, Q03UDECPU, Q04UDHCPU, Q04UDVCPU, Q04UDEHCPU, Q06UDHCPU, Q06UDVCPU, Q06UDEHCPU, Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDVCPU, Q13UDEHCPU, Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDVCPU, Q26UDEHCPU, Q50UDEHCPU, and Q100UDEHCPU
VxWorks	A product name for the real-time operating system manufactured by Wind River Systems, Inc.

1 COMMON ITEMS

A user program is created by using the VxWorks standard API functions*1 and dedicated function library offered by a C Controller module in accordance with the specification of VxWorks, the operating system of C Controller module.

*1 For details on the VxWorks standard API functions, refer to the following programmer's guide supported.
📖VxWorks "KERNEL PROGRAMMER'S GUIDE"

Dedicated function libraries offered by a C Controller module are as follows:

- C Controller module dedicated function
- MELSEC Data Link Functions

Point

For the execution procedure of user programs, refer to the following manual.
📖 MELSEC iQ-R C Controller Module User's Manual (Startup)

1.1 Header Files

Include the following header files in a user program to use the dedicated function library.

Dedicated function library	Header file
C Controller module dedicated function	CCPUFunc.h
MELSEC data link function	MDFunc.h

Point

A header file is stored in a C Controller module.
(📖 MELSEC iQ-R C Controller Module User's Manual (Application))

1.2 C Controller Module Dedicated Functions

C Controller dedicated functions of the dedicated function libraries are used to control C Controller module.
These functions can be used for reading status of C Controller module or accessing resources such as LED control and clock.

Program processing

The following shows the user program processing using the C Controller module dedicated function.

1. Start a task.
2. Read the status of C Controller module and access the resources such as LED control and clock by using the C Controller module dedicated function.
3. Complete the task.

Considerations

Considerations for user WDT (user watchdog timer)

■User WDT error occurrence

If the user WDT cannot be reset due to a user program runaway, a user WDT error occurs.
In this case, take the following corrective actions.

- Increase the user WDT period set with the CCPU_StartWDT function.
- Lower the number of tasks with high CPU usage rate or make them deactivated.
- Review the user program.

Reset the C Controller module once the corrective actions have been taken.

Point

In the user program, user WDT can be used to monitor the hardware and status of user program, and processing timeout for accessing and controlling respective modules.

■User WDT setting range

The user WDT period can be set within the range of 100 ms to 10,000 ms.

■Output when a user WDT error occurs

When a user WDT error occurs, the output turns OFF.

Considerations for the dedicated instructions

Considerations for argument are shown below:

- For the first argument, "pcInstName", categorizing the dedicated instructions (D/DP/J/JP/G/GP/M) is not required.
- Devices of the own station cannot be specified. Reserve the required area with a user program, and specify the start address of the area to the relevant argument.
- Specify devices of other stations by using a character string. Set the value (the number of elements to which one is added for the termination code) to the number of elements for the array.

(Example) To specify D4: char cDev[3] = {"D4"};
Set '3', the total number of '2' for D4 and '1' for termination code, to the number of elements for the character string.

- When the data type is device name (control data, input data, and output data), specify the argument using an array.
- The size is required to be set according to the number of elements for the array, which is specified to the argument. When data type is 16-bit binary, BCD 4-digit, or real number, set the size to '1'. When data type is 32-bit binary or BCD 8-digit, set the size to '2'.
- When data type is bit (completion device), specify the argument using an array. Specify '1' for completion or '0' for incompletion to the first array. In the second array, '0' or '1' is stored for the normal completion or abnormal completion, respectively.
- For arguments without the setting data, set the setting data to NULL, and set the size to '0'.
- Errors which occur in the dedicated instruction are not registered in the event history.

Argument specification

This section shows the argument specifications of the C Controller module dedicated functions.

Device type

The following tables show the device types specified to the C Controller module dedicated functions.
Devices are defined in the header file "CCPUFunc.h".

Point

Either a code or a device name can be specified as a device type.

■Device types for CC-Link IE Controller Network module access

Device type can be specified to the argument sDevType of the CCPU_WriteLinkDevice/CCPU_ReadLinkDevice functions.

Device (Device name)	Device type		
	Code		Device name
	Decimal	Hexadecimal	
Direct link input (LX)	1000	3E8H	Dev_LX
Direct link output (LY)	2000	7D0H	Dev_LY
Direct link relay (LB)	23000	59D8H	Dev_LB
Direct link register (LW)	24000	5DC0H	Dev_LW
Direct link special relay (SB)	25000	61A8H	Dev_LSB
Direct link special register (SW)	28000	6D60H	Dev_LSW

■Device types for CC-Link IE Field Network module access

Device type can be specified to the argument sDevType of the CCPU_WriteLinkDevice/CCPU_ReadLinkDevice functions.

Device (Device name)	Device type		
	Code		Device name
	Decimal	Hexadecimal	
Direct link input (RX)	1000	3E8H	Dev_LX
Direct link output (RY)	2000	7D0H	Dev_LY
Direct link register (RWr, RWw)[*1]	24000	5DC0H	Dev_LW
Direct link special relay (SB)	25000	61A8H	Dev_LSB
Direct link special register (SW)	28000	6D60H	Dev_LSW

*1 Using the following device range enables direct link registers (RWw, RWr) to be accessed.
RWw: LW0 to LW1FFF
RWr: LW2000 to LW3FFF

■Device types for an Internal user device and internal system device access

Device type can be specified to the argument sDevType of the CCPU_WriteDevice/CCPU_ReadDevice/CCPU_SetDevice/CCPU_ResetDevice/CCPU_WriteDevice_ISR/CCPU_ReadDevice_ISR/CCPU_SetDevice_ISR/CCPU_ResetDevice_ISR functions.

Device (Device name)	Device type		
	Code		Device name
	Decimal	Hexadecimal	
Internal relay (M)	4	4H	Dev_CCPU_M
Special relay (SM)	5	5H	Dev_CCPU_SM
Data register (D)	13	DH	Dev_CCPU_D
Special register (SD)	14	EH	Dev_CCPU_SD
Link relay (B)	23	17H	Dev_CCPU_B
Link register (W)	24	18H	Dev_CCPU_W
File register (ZR)	220	DCH	Dev_CCPU_ZR

1.3 MELSEC Data Link Functions

MELSEC data link functions are the integrated communication function libraries which are independent of the communication protocols.
A program to communicate with a CPU module can be created regardless of a target hardware or communication protocols by using the MELSEC data link functions.
The communication functions supported by the MELSEC data link functions are as follows:

Communication function	Description
Bus interface communication	Accesses a CPU module mounted on the same base unit.
CC-Link IE Controller Network communication	Accesses a CPU module on the CC-Link IE Controller Network via a CC-Link IE Controller Network module.
CC-Link IE Field Network communication	Accesses a CPU module on the CC-Link IE Field Network via a CC-Link IE Field Network module.
MELSECNET/H network communication	Accesses a CPU module on the MELSECNET/H network via a MELSECNET/H network module.
CC-Link communication	Accesses a CPU module on the CC-Link via a CC-Link module.

Program processing

The following shows the user program processing using the MELSEC data link function.

When accessing with a device name

1. Start a task.
2. Open a communication line. (mdOpen function)
3. Perform dummy access (such as device/model name reading) to an access target.
4. Access the target by using the MELSEC data link function.
5. To stop accessing the target, go to the procedure 6.
   To access the target again, go back to the procedure 4.
6. Close the communication line. (mdClose function)
7. Complete the task.

When accessing with a label name

1. Start a task.
2. Open a communication line. (mdOpen function)
3. Obtain device information (label assignment information) from a target CPU module. (mdGetLabelInfo function)
4. Access the target CPU module by using the obtained device information (label assignment information).
   (mdRandRLabelEx/mdRandWLabelEx function)
5. Check if there is no change in the device information (label assignment information) of the target CPU module.
   If it is changed, go back to the procedure 3.
6. To stop accessing the target, go to the procedure 7.
   To access the target again, go back to the procedure 4.
7. Close the communication line. (mdClose function)
8. Complete the task.

Considerations

The following shows the considerations when using the MELSEC data link functions.

Considerations for programming

■Open/close processing of a communication line (mdOpen/mdClose function)

Perform the open/close processing of communication line (the mdOpen/mdClose function) only once at the start of task (task activation) and at the end of task (task completion) respectively in each user program. Opening/closing the line every communication decreases the communication performance.

■Execution after using the mdOpen function

At the first execution of the function after using the mdOpen function, it takes longer to execute the function since the CPU module information needs to be obtained. The succeeding processing time can be shortened by performing dummy access at the first time.

■Access to other stations on the same task

Accessing 33 or more other stations simultaneously on the same task of C Controller module using a user program may decrease the communication performance. To access other stations simultaneously on the same task, limit it to 32 or less stations.

■mdGetLabelInfo function call

The mdGetLabelInfo function does not need to be called each time to access a target CPU module.
Only if the error occurs (Error code: -81) when accessing by using the mdRandRLabelEx/mdRandWLabelEx function , call the mdGetLabelInfo function again.

■taskDelete execution

Do not execute the taskDelete in a task using the MELSEC data link function. Also, do not delete a task using the MELSEC data link function with the taskDelete. Otherwise, the MELSEC data link function may not operate properly.

Accessing devices of the own station and of a CPU module on other stations

An interlock may be required to be provided depending on the link status of the own station and other stations.

■Access to devices on the own station

Create a user program that provides an interlock to enable reading/writing data when the following conditions are satisfied to access devices via each network module.

Module to be routed	Condition that requires interlock
CC-Link IE Controller Network module CC-Link IE Field Network master/local module	When the following conditions are met: • The bits of the own station data link error status (SB49) is OFF (data linking). • The cyclic transmission status (the bit corresponding to the communication target station which is stored from SWB0 to B7) is OFF (normal communication).
MELSECNET/H network module	When the following conditions are met: • The module status (SB20) is OFF (normal). • The bits of the own station baton pass status (SB47) is OFF (normal). • The bits of the own station data link status (SB49) is OFF (data linking).
CC-Link module	When the following conditions are met: • The module error (Xn0) is OFF (normal). • 'Module READY' (XnF) is ON (activated). • The bits of the own station data link status (Xn1) is ON (data linking).

Even if the above conditions are not satisfied; however, read/write processing to the own station is completed normally.

■Other station transient access (other station CPU module remote operation and device access)

Create a user program that provides an interlock to enable reading/writing data when the following conditions are satisfied to access devices via each network module.

Module to be routed	Condition that requires interlock
CC-Link IE Controller Network module CC-Link IE Field Network master/local module	When the following conditions are met: • The bits of the own station baton pass status (SB47) is OFF (normal). • The baton pass status of a station to be accessed (the bit corresponding to the communication target station which is stored from SWA0 to A7) is OFF (normal communication).
MELSECNET/H network module	When the following conditions are met: • The condition that turns the interlock ON is met when accessing devices on the own station. • The baton pass status of a station to be accessed (the bit corresponding to the communication target station which is stored from SW70 to 73) is OFF (normal communication). • The data link status (the bit corresponding to the communication target station which is stored from SW74 to 77) is OFF (normal communication).
CC-Link module	When the following conditions are met: • The condition that turns the interlock ON is met when accessing devices on the own station. • The data link status of a station to be accessed (the bit corresponding to the communication target station which is stored from SW80 to 83) is OFF (normal communication).

Accessible range and devices

This section explains the accessible range and accessible devices when using MELSEC data link functions.

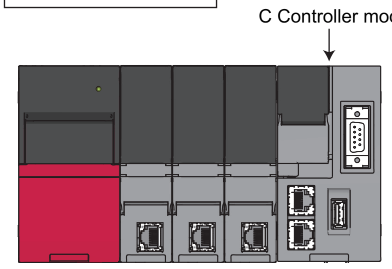
Bus interface communication

The accessible range and devices for bus interface communication are shown below:

■Accessible range

The accessible range for bus interface communication includes the own station (C Controller module), and the CPU module and C Controller module in a multiple CPU system.

System configuration



■Accessible devices

The following explains the devices accessible for communication via a bus.

Point

- 'Batch' and 'Random' in the following table indicate as follows:
  Batch: Batch write (mdSendEx function), batch read (mdReceiveEx function)
  Random: Random write (mdRandWEx function), random read (mdRandREx function), bit set (mdDevSetEx function), bit reset (mdDevRstEx function), random write by using a label name (mdRandWLabelEx function), random read by using a label name (mdRandRLabelEx function)
- Bit devices can be accessed by using 'bit set' (mdDevSetEx function) and 'bit reset' (mdDevRstEx function).
- The fixed cycle communication area can be accessed only when the multiple CPU setting is configured.
- Device extension specifications (digit specification, bit specification and index specification) cannot be used.

- Accessing the host CPU

The following table shows the accessible devices when accessing the host module.

○: Accessible, ×: Not accessible

Device		Access method	Access target CPU
			R12CCPU-V
Input relay	X	Batch/random	○
Output relay	Y	Batch/random	○
Internal relay	M	Batch/random	○
Special relay	SM	Batch/random	○
Data register	D	Batch/random	○
Special register	SD	Batch/random	○
Link relay	B	Batch/random	○
Link register	W	Batch/random	○
File register	ZR	Batch/random	○
Intelligent function module device, module access device	Un\G	Batch/random	○
CPU buffer memory	U3En\G	Batch	○
		Random	×
Fixed cycle communication area	U3En\HG	Batch	○
		Random	×

- Accessing the other CPU

The following table shows the accessible devices when accessing the other CPUs (that is a CPU module and a C Controller module in a multiple CPU system).

No.	Access target CPU
(1)	RCPU
(2)	R12CCPU-V

○: Accessible, ×: Not accessible

Device		Access method	Access target CPU	
			(1)	(2)
Input relay	X	Batch/random	○	○
Output relay	Y	Batch/random	○	○
Latch relay	L	Batch/random	○	×
Internal relay	M	Batch/random	○	○
Special relay	SM	Batch/random	○	○
Annunciator	F	Batch/random	○	×
Timer (Contact)	T	Batch/random	○	×
Long timer (Contact)	LT	Batch/random	○	×
Timer (Coil)	T	Batch/random	○	×
Long timer (Coil)	LT	Batch/random	○	×
Counter (Contact)	C	Batch/random	○	×
Long counter (Contact)	LC	Batch/random	○	×
Counter (Coil)	C	Batch/random	○	×
Long counter (Coil)	LC	Batch/random	○	×
Timer (Current value)	T	Batch/random	○	×
Long timer (Current value)	LT	Batch/random	○	×
Counter (Current value)	C	Batch/random	○	×
Long counter (Current value)	LC	Batch/random	○	×
Data register	D	Batch/random	○	○
Special register	SD	Batch/random	○	○
Index register	Z	Batch/random	○	×
Long index register	LZ	Batch/random	○	×
File register	R	Batch/random	○	×
	ZR	Batch/random	○	○
Refresh data register	RD	Batch/random	○	×
Link relay	B	Batch/random	○	○
Link register	W	Batch/random	○	○
Link special relay	SB	Batch/random	○	×
Retentive timer (Contact)	ST	Batch/random	○	×
Long retentive timer (Contact)	LST	Batch/random	○	×
Retentive timer (Coil)	ST	Batch/random	○	×
Long retentive timer (Coil)	LST	Batch/random	○	×
Link special register	SW	Batch/random	○	×
Edge relay	V	Batch/random	○	×
Own station random access buffer	—	Batch/random	×	×
Retentive timer (Current value)	ST	Batch/random	○	×
Long retentive timer (Current value)	LST	Batch/random	○	×
Remote register for sending	RWw	Batch/random	×	×
Remote register for receiving	RWr	Batch/random	×	×
Own station buffer memory	—	Batch/random	×	×
Link direct device (Link input)	Jn\X	Batch/random	○	○
Link direct device (Link output)	Jn\Y	Batch/random	○	○
Link direct device (Link relay)	Jn\B	Batch/random	○	○
Link direct device (Link register)	Jn\W	Batch/random	○	○

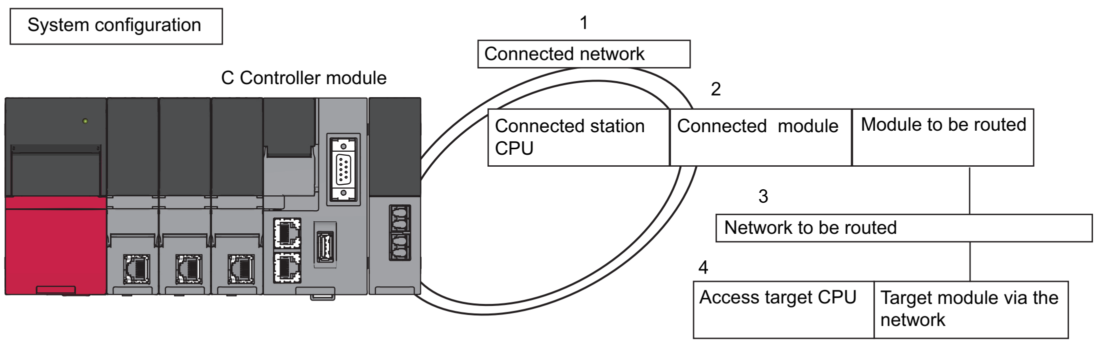
Device		Access method	Access target CPU	
			(1)	(2)
Link direct device (Link special relay)	Jn\SB	Batch/random	○	○
Link direct device (Link special register)	Jn\SW	Batch/random	○	○
Intelligent function module device, module access device	Un\G	Batch/random	○	○
Other station buffer memory	—	Batch/random	×	×
Other station random access buffer	—	Batch/random	×	×
Remote input	RX	Batch/random	×	×
Remote output	RY	Batch/random	×	×
Remote register	RW	Batch/random	×	×
Link special relay	SB	Batch/random	×	×
Link special register	SW	Batch/random	×	×
CPU buffer memory	U3En\G	Batch	○	○
		Random	×	×
Fixed cycle communication area	U3En\HG	Batch	○	○
		Random	×	×
Global label	GV	Batch	×	×
		Random	○	×
Safety input	SA\X	Batch/random	×	×
Safety output	SA\Y	Batch/random	×	×
Safety internal relay	SA\M	Batch/random	×	×
Safety link relay	SA\B	Batch/random	×	×
Safety timer	SA\T	Batch/random	×	×
Safety retentive timer	SA\ST	Batch/random	×	×
Safety counter	SA\C	Batch/random	×	×
Safety data register	SA\D	Batch/random	×	×
Safety link register	SA\W	Batch/random	×	×
Safety special relay	SA\SM	Batch/random	×	×
Safety special register	SA\SD	Batch/random	×	×

CC-Link IE Controller Network communication

The range and devices accessible for communication via a CC-Link IE Controller Network module are shown below:

■Accessible range

The following shows the system configuration in the accessible range and the accessibility of each access target CPU via a CC-Link IE Controller Network module.



■Applicable access

Accessibility is shown in the following table. The own station and the connected station CPU are accessible.

○: Accessible, ×: Not accessible

1. Connected network	2. Connected station CPU	3. Network to be routed	4. Access target CPU						
			Programmable controller			C Controller module		WinCPU module	Interface board for personal computer
			MELSEC iQ-R series	MELSEC -Q series	MELSEC -L series	MELSEC iQ-R series	MELSEC -Q series	MELSEC -Q series	
CC-Link IE Controller Network	MELSEC iQ-R series programmable controller	CC-Link IE Controller Network	○	○	×	○	○	×	×
		CC-Link IE Field Network	○	○	○	○	○[1]	×	×
		MELSECNET/H network	○	○	×	○	○	○	×
		MELSECNET/10 network	○	○	×	○	○	○	×
		Ethernet	○	○	○	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×
	MELSEC iQ-R series C Controller module	CC-Link IE Controller Network	×	×	×	×	×	×	×
		CC-Link IE Field Network	×	×	×	×	×	×	×
		MELSECNET/H network	×	×	×	×	×	×	×
		MELSECNET/10 network	×	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×
	MELSEC-Q series programmable controller (Q mode)	CC-Link IE Controller Network[2]	○	○[3]	×	○	○	×	×
		CC-Link IE Field Network[2]	○	○[3]	○	○	○[1]	×	×
		MELSECNET/H network	○	○[3]	×	○	○	○	×
		MELSECNET/10 network	○	○[3]	×	○	○	○	×
		MELSECNET(Ⅱ)	×	×	×	×	×	×	×
		Ethernet	○	○	○	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×
	MELSEC-Q series C Controller module	CC-Link IE Controller Network	×	×	×	×	×	×	×
		CC-Link IE Field Network	×	×	×	×	×	×	×
		MELSECNET/H network	×	×	×	×	×	×	×
		MELSECNET/10 network	×	×	×	×	×	×	×
		MELSECNET(Ⅱ)	×	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×

*1 The following CPUs are accessible:
Q12DCCPU-V (Extended mode)
Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS

*2 The station number 65 or later is accessible only when all control CPUs on the network to be routed are universal model QCPUs.

*3 It is not accessible when the connected station CPU is Q00J/Q00/Q01CPU.

■Accessible devices

The following explains the accessible devices in the communication via a CC-Link IE Controller Network module.

Point

- 'Batch' and 'Random' in the following table indicate as follows:
  Batch: Batch write (mdSendEx function), batch read (mdReceiveEx function)
  Random: Random write (mdRandWEx function), random read (mdRandREx function), bit set (mdDevSetEx function), bit reset (mdDevRstEx function), random write by using a label name (mdRandWLabelEx function), random read by using a label name (mdRandRLabelEx function)
- Bit devices can be accessed by using 'bit set' (mdDevSetEx function) and 'bit reset' (mdDevRstEx function).
- Device extension specifications (digit specification, bit specification and index specification) cannot be used.

- Accessing the own station

The accessible devices when accessing the CC-Link IE Controller Network module controlled by C Controller module are shown in the following table.

○: Accessible, ×: Not accessible

Device	**Access method**	**Access target CPU**
		R12CCPU-V
RECV function	Batch	○
	Random	×

For details on replacement from the device types specified with an existing product, refer to the following section.

☞ Page 183 Replacement of device type

<For other than the RECV function>

To access a CC-Link IE Controller Network module controlled by a C Controller module, use the method explained in the following section. Accessing the own station by using a CC-Link IE Controller Network communication will result in the station number/network number error.

☞ Page 183 Replacement of device type

- Accessing other stations

No.	Access target CPU
(1)	Basic model QCPU, high performance model QCPU, process CPU, redundant CPU, universal model QCPU
(2)	Q12DCCPU-V, Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS
(3)	Personal computer, WinCPU module
(4)	L02CPU, L26CPU-BT, L02CPU-P, L26CPU-PBT, LJ72GF15-T2, NZ2GF-ETB, L02SCPU, L26CPU, L06CPU
(5)	RCPU
(6)	R12CCPU-V

○: Accessible, ×: Not accessible

Device		Access method	Access target CPU					
			(1)	(2)	(3)	(4)	(5)	(6)
Input relay	X	Batch/random	○	○[*1]	×	○	○	○
Output relay	Y	Batch/random	○	○[*1]	×	○	○	○
Latch relay	L	Batch/random	○	×	×	○	○	×
Internal relay	M	Batch/random	○	○[*1]	×	○	○	○
Special relay	SM	Batch/random	○	○[*1]	×	○	○	○
Annunciator	F	Batch/random	○	×	×	○	○	×
Timer (Contact)	T	Batch/random	○	×	×	○	○	×
Long timer (Contact)	LT	Batch/random	×	×	×	×	○	×
Timer (Coil)	T	Batch/random	○	×	×	○	○	×
Long timer (Coil)	LT	Batch/random	×	×	×	×	○	×
Counter (Contact)	C	Batch/random	○	×	×	○	○	×
Long counter (Contact)	LC	Batch/random	×	×	×	×	○	×
Counter (Coil)	C	Batch/random	○	×	×	○	○	×
Long counter (Coil)	LC	Batch/random	×	×	×	×	○	×
Timer (Current value)	T	Batch/random	○	×	×	○	○	×
Long timer (Current value)	LT	Batch/random	×	×	×	×	○	×
Counter (Current value)	C	Batch/random	○	×	×	○	○	×
Long counter (Current value)	LC	Batch/random	×	×	×	×	○	×
Data register	D	Batch/random	○	○[*1]	×	○	○	○
Special register	SD	Batch/random	○	○[*1]	×	○	○	○
Index register	Z	Batch/random	○	×	×	○	○	×
Long index register	LZ	Batch/random	○	×	×	○	○	×
File register	R	Batch/random	○[*2]	×	×	○	○	○
	ZR	Batch/random	○[*2]	×	×	○	○	○
Refresh data register	RD	Batch/random	×	×	×	×	○	×
Link relay	B	Batch/random	○	○[*1]	×	○	○	○
Link register	W	Batch/random	○	○[*1]	×	○	○	○
Link special relay	SB	Batch/random	○	×	×	○	○	×
Retentive timer (Contact)	ST	Batch/random	○	×	×	○	○	×
Long retentive timer (Contact)	LST	Batch/random	×	×	×	×	○	×
Retentive timer (Coil)	ST	Batch/random	○	×	×	○	○	×
Long retentive timer (Coil)	LST	Batch/random	×	×	×	×	○	×
Link special register	SW	Batch/random	○	×	×	○	○	×
Edge relay	V	Batch/random	○	×	×	○	○	×
Own station random access buffer	—	Batch/random	×	×	×	×	×	×
Retentive timer (Current value)	ST	Batch/random	○	×	×	○	○	×
Long retentive timer (Current value)	LST	Batch/random	×	×	×	×	○	×
Own station link register (for sending)	—	Batch/random	×	×	×	×	×	×
Own station link register (for receiving)	—	Batch/random	×	×	×	×	×	×

Device		Access method	Access target CPU					
			(1)	(2)	(3)	(4)	(5)	(6)
Own station buffer memory	—	Batch/random	×	×	×	×	×	×
SEND function (with arrival confirmation)[*3]	—	Batch	○	○	×	○	○	○
		Random	×	×		×	×	×
SEND function (without arrival confirmation)[*3]	—	Batch	○	○	×	○	○	○
		Random	×	×		×	×	×
Direct link input (other station side)	—	Batch/random	○	○[*1]	×	×	○	○
Remote input for CC-Link IE Field	RX	Batch/random	○	○[*1]	×	×	○	○
Direct link output (other station side)	—	Batch/random	○	○[*1]	×	×	○	○
Remote output for CC-Link IE Field	RY	Batch/random	○	○[*1]	×	×	○	○
Direct link relay (other station side)	—	Batch/random	○	○[*1]	×	×	○	○
Direct link register (other station side)	—	Batch/random	○	○[*1]	×	×	○	○
Remote register for CC-Link IE Field (for sending)	RWw	Batch/random	○	○[*1]	×	×	○	○
Remote register for CC-Link IE Field (for receiving)	RWr	Batch/random	○	○[*1]	×	×	○	○
Direct link special relay (other station side)	—	Batch/random	○	○[*1]	×	×	○	○
Direct link special register (other station side)	—	Batch/random	○	○[*1]	×	×	○	○
Intelligent function module device, module access device	Un\G	Batch/random	○	○[*1]	×	○	○	○
CPU shared memory, CPU buffer memory (CPU No.1 area)	U3E0\G	Batch	○	○	×	×	○	○
		Random	×	×			×	×
CPU shared memory, CPU buffer memory (CPU No.2 area)	U3E1\G	Batch	○	○	×	×	○	○
		Random	×	×			×	×
CPU shared memory, CPU buffer memory (CPU No.3 area)	U3E2\G	Batch	○	○	×	×	○	○
		Random	×	×			×	×
CPU shared memory, CPU buffer memory (CPU No.4 area)	U3E3\G	Batch	○	○	×	×	○	○
		Random	×	×			×	×
Fixed cycle communication area (CPU No.1 area)	U3E0\HG	Batch	×	×	×	×	○	○
		Random					×	×
Fixed cycle communication area (CPU No.2 area)	U3E1\HG	Batch	×	×	×	×	○	○
		Random					×	×
Fixed cycle communication area (CPU No.3 area)	U3E2\HG	Batch	×	×	×	×	○	○
		Random					×	×
Fixed cycle communication area (CPU No.4 area)	U3E3\HG	Batch	×	×	×	×	○	○
		Random					×	×
Other station buffer memory	—	Batch/random	×	×	×	×	×	×
Other station random access buffer	—	Batch/random	×	×	×	×	×	×
Remote input for CC-Link	RX	Batch/random	×	×	×	×	×	×
Remote output for CC-Link	RY	Batch/random	×	×	×	×	×	×
Other station link register	—	Batch/random	×	×	×	×	×	×
Link special relay for CC-Link	SB	Batch/random	×	×	×	×	×	×
Link special register for CC-Link	SW	Batch/random	×	×	×	×	×	×
Global label	GV	Batch	×	×	×	×	×	×
		Random	×	×	×	×	○	×
Safety input	SA\X	Batch/random	×	×	×	×	×	×
Safety output	SA\Y	Batch/random	×	×	×	×	×	×
Safety internal relay	SA\M	Batch/random	×	×	×	×	×	×

Device		Access method	Access target CPU					
			(1)	(2)	(3)	(4)	(5)	(6)
Safety link relay	SA\B	Batch/random	×	×	×	×	×	×
Safety timer	SA\T	Batch/random	×	×	×	×	×	×
Safety retentive timer	SA\ST	Batch/random	×	×	×	×	×	×
Safety counter	SA\C	Batch/random	×	×	×	×	×	×
Safety data register	SA\D	Batch/random	×	×	×	×	×	×
Safety link register	SA\W	Batch/random	×	×	×	×	×	×
Safety special relay	SA\SM	Batch/random	×	×	×	×	×	×
Safety special register	SA\SD	Batch/random	×	×	×	×	×	×

*1 The following CPUs are accessible:
Q12DCCPU-V (Extended mode)
Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS

*2 Q00JCPU is not accessible.

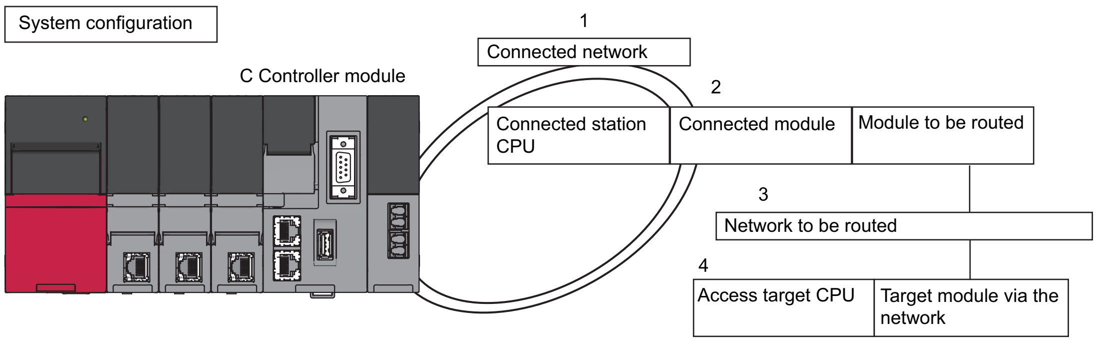
*3 This is a function to send a message to a network module on other stations via a CC-Link IE Controller Network module. Access to a multiple CPU system (when a logical station number is specified) is not available.

CC-Link IE Field Network communication

The range and devices accessible for communication via a CC-Link IE Field Network module are shown below:

■Accessible range

The following shows the system configuration in the accessible range and the accessibility of each access target CPU via a CC-Link IE Field Network module.



■Applicable access

Accessibility is shown in the following table. The own station and the connected station CPU are accessible.

○: Accessible, ×: Not accessible

1. Connected network	2. Connected station CPU	3. Network to be routed	4. Access target CPU						
			Programmable controller			C Controller module		WinCPU module	Interface board for personal computer
			MELSEC iQ-R series	MELSEC -Q series	MELSEC -L series	MELSEC iQ-R series	MELSEC -Q series	MELSEC -Q series	
CC-Link IE Field Network*1	MELSEC iQ-R series programmable controller	CC-Link IE Controller Network	○	○	×	○	○	×	×
		CC-Link IE Field Network	○	○	○	○	○*2	×	×
		MELSECNET/H network	○	○	×	○	○	○	×
		MELSECNET/10 network	○	○	×	○	○	○	×
		Ethernet	×	×	○	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×
	MELSEC iQ-R series C Controller module	CC-Link IE Controller Network	×	×	×	×	×	×	×
		CC-Link IE Field Network	×	×	×	×	×	×	×
		MELSECNET/H network	×	×	×	×	×	×	×
		MELSECNET/10 network	×	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×
	MELSEC-Q series programmable controller (Q mode)	CC-Link IE Controller Network*3	○	○*4	×	○	○	×	×
		CC-Link IE Field Network*3	○	○*4	○	○	○*2	×	×
		MELSECNET/H network	○	○*4	×	○	○	○	×
		MELSECNET/10 network	○	○*4	×	○	○	○	×
		MELSECNET(Ⅱ)	×	×	×	×	×	×	×
		Ethernet	×	×	○	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×
	MELSEC-Q series C Controller module	CC-Link IE Controller Network	×	×	×	×	×	×	×
		CC-Link IE Field Network	×	×	×	×	×	×	×
		MELSECNET/H network	×	×	×	×	×	×	×
		MELSECNET/10 network	×	×	×	×	×	×	×
		MELSECNET(Ⅱ)	×	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×

*1 A simple motion module (RD77GF4, RD77GF8, and RD77GF16) is not accessible.
*2 The following CPUs are accessible:
Q12DCCPU-V (Extended mode)
Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS
*3 The station number 65 or later is accessible only when all control CPUs on the network to be routed are universal model QCPUs.
*4 It is not accessible when the connected station CPU is Q00J/Q00/Q01CPU.

■Accessible devices

This following shows the devices accessible for communication via a CC-Link IE Field Network master or local module.

Point

- 'Batch' and 'Random' in the following table indicate as follows:
  Batch: Batch write (mdSendEx function), batch read (mdReceiveEx function)
  Random: Random write (mdRandWEx function), random read (mdRandREx function), bit set (mdDevSetEx function), bit reset (mdDevRstEx function), random write by using a label name (mdRandWLabelEx function), random read by using a label name (mdRandRLabelEx function)
- Bit devices can be accessed by using 'bit set' (mdDevSetEx function) and 'bit reset' (mdDevRstEx function).
- Device extension specifications (digit specification, bit specification and index specification) cannot be used.

- Accessing the own station

The accessible devices when accessing the CC-Link IE Field Network module controlled by C Controller module are shown in the following table.

○: Accessible, ×: Not accessible

Device	Access method	Access target CPU
		R12CCPU-V
RECV function	Batch	○
	Random	×

For details on replacement from the device types specified with an existing product, refer to the following section.

☞ Page 183 Replacement of device type

<For other than the RECV function>

To access a CC-Link IE Field Network module controlled by a C Controller module, use the method explained in the following section. Accessing the own station by using CC-Link IE Field Network communication will result in the station number/network number error.

☞ Page 183 Replacement of device type

- Accessing other stations

No.	Access target CPU
(1)	Basic model QCPU, high performance model QCPU, process CPU, redundant CPU, universal model QCPU
(2)	Q12DCCPU-V, Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS
(3)	WinCPU module and personal computer
(4)	L26CPU-BT, L02CPU, L02CPU-P, L26CPU-PBT, LJ72GF15-T2, NZ2GF-ETB, L02SCPU, L26CPU, and L06CPU
(5)	RCPU
(6)	R12CCPU-V

○: Accessible, ×: Not accessible

Device		Access method	Access target CPU					
			(1)	(2)	(3)	(4)	(5)	(6)
Input relay	X	Batch/random	○	○*1	×	○	○	○
Output relay	Y	Batch/random	○	○*1	×	○	○	○
Latch relay	L	Batch/random	○	×	×	○	○	×
Internal relay	M	Batch/random	○	○*1	×	○	○	○
Special relay	SM	Batch/random	○	○*1	×	○	○	○
Annunciator	F	Batch/random	○	×	×	○	○	×
Timer (Contact)	T	Batch/random	○	×	×	○	○	×
Long timer (Contact)	LT	Batch/random	×	×	×	×	○	×
Timer (Coil)	T	Batch/random	○	×	×	○	○	×
Long timer (Coil)	LT	Batch/random	×	×	×	×	○	×
Counter (Contact)	C	Batch/random	○	×	×	○	○	×
Long counter (Contact)	LC	Batch/random	×	×	×	×	○	×
Counter (Coil)	C	Batch/random	○	×	×	○	○	×
Long counter (Coil)	LC	Batch/random	×	×	×	×	○	×
Timer (Current value)	T	Batch/random	○	×	×	○	○	×
Long timer (Current value)	LT	Batch/random	×	×	×	×	○	×
Counter (Current value)	C	Batch/random	○	×	×	○	○	×
Long counter (Current value)	LC	Batch/random	×	×	×	×	○	×
Data register	D	Batch/random	○	○*1	×	○	○	○
Special register	SD	Batch/random	○	○*1	×	○	○	○
Index register	Z	Batch/random	○	×	×	○	○	×
Long index register	LZ	Batch/random	○	×	×	○	○	×
File register	R	Batch/random	○*2	×	×	○	○	×
	ZR	Batch/random	○*2	×	×	○	○	○
Refresh data register	RD	Batch/random	×	×	×	×	○	×
Link relay	B	Batch/random	○	○*1	×	○	○	○
Link register	W	Batch/random	○	○*1	×	○	○	○
Link special relay	SB	Batch/random	○	×	×	○	○	×
Retentive timer (Contact)	ST	Batch/random	○	×	×	○	○	×
Long retentive timer (Contact)	LST	Batch/random	×	×	×	×	○	×
Retentive timer (Coil)	ST	Batch/random	○	×	×	○	○	×
Long retentive timer (Coil)	LST	Batch/random	×	×	×	×	○	×
Link special register	SW	Batch/random	○	×	×	○	○	×
Edge relay	V	Batch/random	○	×	×	○	○	×
Own station random access buffer	—	Batch/random	×	×	×	×	×	×
Retentive timer (Current value)	ST	Batch/random	○	×	×	○	○	×
Long retentive timer (Current value)	LST	Batch/random	×	×	×	×	○	×
Own station link register (for sending)	—	Batch/random	×	×	×	×	×	×
Own station link register (for receiving)	—	Batch/random	×	×	×	×	×	×
Own station buffer memory	—	Batch/random	×	×	×	×	×	×
SEND function (with arrival confirmation)	—	Batch	○	○*1	×	○	○	○
		Random	×	×	×	×	×	×

Device		Access method	Access target CPU					
			(1)	(2)	(3)	(4)	(5)	(6)
SEND function (without arrival confirmation)	—	Batch	○	○[*1]	×	○	○	○
		Random	×	×	×	×	×	×
Link direct device (Link input)	Jn\X	Batch/random	○	○	×	×	○	○
Remote input for CC-Link IE Field	RX	Batch/random	○	○[*3]	×	○	○	○
Link direct device (Link output)	Jn\Y	Batch/random	○	○	×	×	○	○
Remote output for CC-Link IE Field	RY	Batch/random	○	○[*3]	×	○	○	○
Link direct device (Link relay)	Jn\B[*3]	Batch/random	○	○	×	×	○	○
Link direct device (Link register)	Jn\W[*3]	Batch/random	○	○	×	×	○	○
Remote register for CC-Link IE field (for sending)	RWw	Batch/random	○	○[*3]	×	○	○	○
Remote register for CC-Link IE field (for receiving)	RWr	Batch/random	○	○[*3]	×	○	○	○
Direct link special relay (other station side)	—	Batch/random	○	○[*3]	×	○	○	○
Direct link special register (other station side)	—	Batch/random	○	○[*3]	×	○	○	○
Intelligent function module device, module access device	Un\G	Batch/random	○	○[*3]	×	○	○	○
CPU shared memory, CPU buffer memory (CPU No.1 area)	U3E0\G	Batch	○	○	×	×	○	○
		Random	×	×			×	×
CPU shared memory, CPU buffer memory (CPU No.2 area)	U3E1\G	Batch	○	○	×	×	○	○
		Random	×	×			×	×
CPU shared memory, CPU buffer memory (CPU No.3 area)	U3E2\G	Batch	○	○	×	×	○	○
		Random	×	×			×	×
CPU shared memory, CPU buffer memory (CPU No.4 area)	U3E3\G	Batch	○	○	×	×	○	○
		Random	×	×			×	×
Fixed cycle communication area (CPU No.1 area)	U3E0\HG	Batch	×	×	×	×	○	○
		Random					×	×
Fixed cycle communication area (CPU No.2 area)	U3E1\HG	Batch	×	×	×	×	○	○
		Random					×	×
Fixed cycle communication area (CPU No.3 area)	U3E2\HG	Batch	×	×	×	×	○	○
		Random					×	×
Fixed cycle communication area (CPU No.4 area)	U3E3\HG	Batch	×	×	×	×	○	○
		Random					×	×
Other station buffer memory	—	Batch/random	×	×	×	×	×	×
Other station random access buffer	—	Batch/random	×	×	×	×	×	×
Remote input for CC-Link	RX	Batch/random	×	×	×	×	×	×
Remote output for CC-Link	RY	Batch/random	×	×	×	×	×	×
Other station link register	—	Batch/random	×	×	×	×	×	×
Link special relay for CC-Link	SB	Batch/random	×	×	×	×	×	×
Link special register for CC-Link	SW	Batch/random	×	×	×	×	×	×
Global label	GV	Batch	×	×	×	×	×	×
		Random	×	×	×	×	○	×
Safety input	SA\X	Batch/random	×	×	×	×	×	×
Safety output	SA\Y	Batch/random	×	×	×	×	×	×
Safety internal relay	SA\M	Batch/random	×	×	×	×	×	×
Safety link relay	SA\B	Batch/random	×	×	×	×	×	×
Safety timer	SA\T	Batch/random	×	×	×	×	×	×
Safety retentive timer	SA\ST	Batch/random	×	×	×	×	×	×
Safety counter	SA\C	Batch/random	×	×	×	×	×	×
Safety data register	SA\D	Batch/random	×	×	×	×	×	×
Safety link register	SA\W	Batch/random	×	×	×	×	×	×
Safety special relay	SA\SM	Batch/random	×	×	×	×	×	×
Safety special register	SA\SD	Batch/random	×	×	×	×	×	×

*1 The following CPUs are accessible:
Q12DCCPU-V with a serial number of which the first 5 digits are '12042' or later
Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS

*2 Q00JCPU is not accessible.

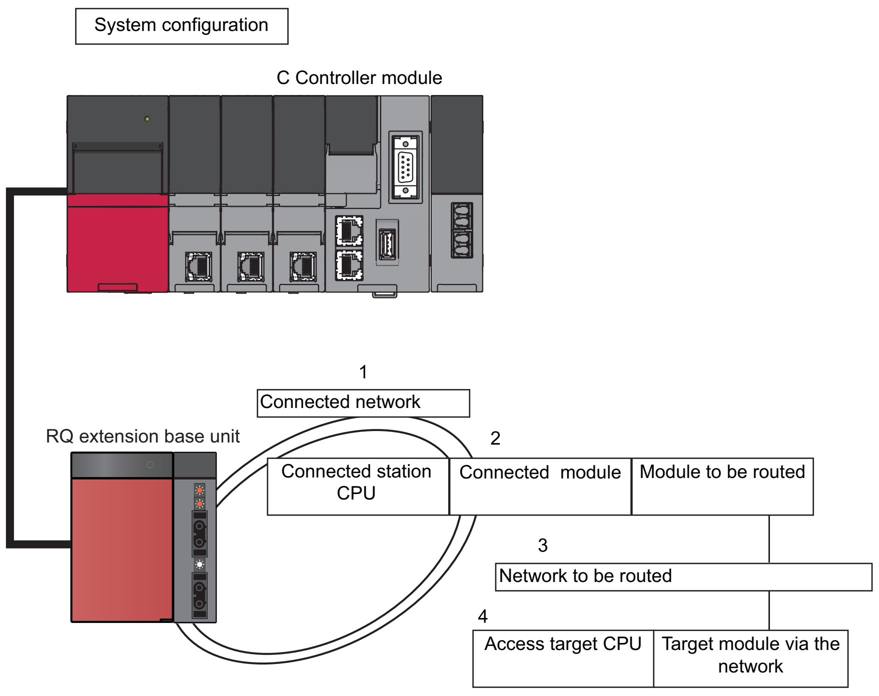
*3 The following CPUs are accessible:
Q12DCCPU-V (Extended mode)
Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS

MELSECNET/H network communication

The range and devices accessible for communication via a MELSECNET/H network module are shown below:

■Accessible range

The following shows the system configuration in the accessible range and the accessibility of each access target CPU via a MELSECNET/H network module.



■Applicable access

Accessibility is shown in the following table. The own station and the connected station CPU are accessible.

○: Accessible, ×: Not accessible

1. Connected network	2. Connected station CPU	3. Network to be routed	4. Access target CPU						
			Programmable controller			C Controller module		WinCPU module	Interface board for personal computer
			MELSEC iQ-R series	MELSEC-Q series	MELSEC-L series	MELSEC iQ-R series	MELSEC-Q series	MELSEC-Q series	
• MELSECNET/H network • MELSECNET/10 network	MELSEC iQ-R series programmable controller	CC-Link IE Controller Network	○	○	×	○	○	×	×
		CC-Link IE Field Network	○	○	○	○	○*1	×	×
		MELSECNET/H network	○	○	×	○	○	○	×
		MELSECNET/10 network	○	○	×	○	○	○	×
		Ethernet	○	○	○	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×
	MELSEC iQ-R series C Controller module	CC-Link IE Controller Network	×	×	×	×	×	×	×
		CC-Link IE Field Network	×	×	×	×	×	×	×
		MELSECNET/H network	×	×	×	×	×	×	×
		MELSECNET/10 network	×	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×
	MELSEC-Q series programmable controller (Q mode)	CC-Link IE Controller Network*2	○	○*3	×	○	○	×	×
		CC-Link IE Field Network*2	○	○*3	○	○	○*1	×	×
		MELSECNET/H network	○	○*3	×	○	○	○	×
		MELSECNET/10 network	○	○*3	×	○	○	○	×
		MELSECNET(II)	×	×	×	×	×	×	×
		Ethernet	○	○	○	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×
	MELSEC-Q series C Controller module	CC-Link IE Controller Network	×	×	×	×	×	×	×
		CC-Link IE Field Network	×	×	×	×	×	×	×
		MELSECNET/H network	×	×	×	×	×	×	×
		MELSECNET/10 network	×	×	×	×	×	×	×
		MELSECNET(II)	×	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×

*1 The following CPUs are accessible:
Q12DCCPU-V (Extended mode)
Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS

*2 The station number 65 or later is accessible only when all control CPUs on the network to be routed are universal model QCPUs.

*3 It is not accessible when the connected station CPU is Q00J/Q00/Q01CPU.

■Accessible devices
This following shows the devices accessible for communication via a MELSECNET/H network module.

Point

- 'Batch' and 'Random' in the following table indicate as follows:
  Batch: Batch write (mdSendEx function), batch read (mdReceiveEx function)
  Random: Random write (mdRandWEx function), random read (mdRandREx function), bit set (mdDevSetEx function), bit reset (mdDevRstEx function), random write by using a label name (mdRandWLabelEx function), random read by using a label name (mdRandRLabelEx function)
- Bit devices can be accessed by using 'bit set' (mdDevSetEx function) and 'bit reset' (mdDevRstEx function).
- Device extension specifications (digit specification, bit specification and index specification) cannot be used.

- Accessing the own station

The accessible devices when accessing the MELSECNET/H network module controlled by C Controller module are shown in the following table.

○: Accessible, ×: Not accessible

Device	Access method	Access target CPU
		R12CCPU-V
RECV function	Batch	○
	Random	×

For details on replacement from the device types specified with an existing product, refer to the following section.

☞ Page 183 Replacement of device type

<For other than the RECV function>

To access a MELSECNET/H network module controlled by a C Controller module, use the methods explained in the following section. Accessing the own station by using MELSECNET/H network communication will result in the station number/network number error.

☞ Page 183 Replacement of device type

- Accessing other stations

No.	Access target CPU
(1)	Basic model QCPU, high performance model QCPU, process CPU, redundant CPU, universal model QCPU
(2)	Q12DCCPU-V, Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS
(3)	WinCPU module and personal computer
(4)	L26CPU-BT, L02CPU, L02CPU-P, L26CPU-PBT, LJ72GF15-T2, NZ2GF-ETB, L02SCPU, L26CPU, and L06CPU
(5)	RCPU
(6)	R12CCPU-V

○: Accessible, ×: Not accessible

Device		Access method	Access target CPU					
			(1)	(2)	(3)	(4)	(5)	(6)
Input relay	X	Batch/random	○	○[*1]	×	○	○	○
Output relay	Y	Batch/random	○	○[*1]	×	○	○	○
Latch relay	L	Batch/random	○	×	×	○	○	×
Internal relay	M	Batch/random	○	○[*1]	×	○	○	○
Special relay	SM	Batch/random	○	○[*1]	×	○	○	○
Annunciator	F	Batch/random	○	×	×	○	○	×
Timer (Contact)	T	Batch/random	○	×	×	○	○	×
Long timer (Contact)	LT	Batch/random	○	×	×	○	○	×
Timer (Coil)	T	Batch/random	○	×	×	○	○	×
Long timer (Coil)	LT	Batch/random	○	×	×	○	○	×
Counter (Contact)	C	Batch/random	○	×	×	○	○	×
Long counter (Contact)	LC	Batch/random	○	×	×	○	○	×
Counter (Coil)	C	Batch/random	○	×	×	○	○	×
Long counter (Coil)	LC	Batch/random	○	×	×	○	○	×
Timer (Current value)	T	Batch/random	○	×	×	○	○	×
Long timer (Current value)	LT	Batch/random	○	×	×	○	○	×
Counter (Current value)	C	Batch/random	○	×	×	○	○	×
Long counter (Current value)	LC	Batch/random	○	×	×	○	○	×
Data register	D	Batch/random	○	○[*1]	×	○	○	○
Special register	SD	Batch/random	○	○[*1]	×	○	○	○
Index register	Z	Batch/random	○	×	×	×	○	×
Long index register	LZ	Batch/random	○	×	×	×	○	○
File register	R	Batch/random	○[*2]	×	×	○	○	×
	ZR	Batch/random	○[*2]	×	×	○	○	○
Refresh data register	RD	Batch/random	×	×	×	×	○	×
Link relay	B	Batch/random	○	○[*3]	×	○	○	○
Link register	W	Batch/random	○	○[*3]	×	○	○	○
Link special relay	SB	Batch/random	○	×	×	○	○	×
Retentive timer (Contact)	ST	Batch/random	○	×	×	○	○	×
Long retentive timer (Contact)	LST	Batch/random	×	×	×	×	○	×
Retentive timer (Coil)	ST	Batch/random	○	×	×	○	○	×
Long retentive timer (Coil)	LST	Batch/random	×	×	×	×	○	×
Link special register	SW	Batch/random	○	×	×	○	○	×
Edge relay	V	Batch/random	○	×	×	○	○	×
Own station random access buffer	—	Batch/random	×	×	×	×	×	×
Retentive timer (Current value)	ST	Batch/random	○	×	×	○	○	×
Long retentive timer (Current value)	LST	Batch/random	×	×	×	×	○	×
Own station link register (for sending)	—	Batch/random	×	×	×	×	×	×
Own station link register (for receiving)	—	Batch/random	×	×	×	×	×	×
Own station buffer memory	—	Batch/random	×	×	×	×	×	×
SEND function (with arrival confirmation)	—	Batch	○	○	×	○	○	○
		Random	×	×	×	×	×	×

Device		Access method	Access target CPU					
			(1)	(2)	(3)	(4)	(5)	(6)
SEND function (without arrival confirmation)	—	Batch	○	○	×	○	○	○
		Random	×	×	×	×	×	×
Link direct device (Link input)	Jn\X	Batch/random	○	○*4	×	×	○	×
Remote input for CC-Link IE Field	RX	Batch/random	○*3	○*3	×	○	○	×
Link direct device (Link output)	Jn\Y	Batch/random	○	○*4	×	×	○	×
Remote output for CC-Link IE Field	RY	Batch/random	○*3	○*3	×	○	○	×
Link direct device (Link relay)	Jn\B	Batch/random	○	○*4	×	×	○	×
Link direct device (Link register)	Jn\W	Batch/random	○	○*4	×	×	○	×
Remote register for CC-Link IE Field (for sending)	RWw	Batch/random	○*3	○*3	×	○	○	×
Remote register for CC-Link IE Field (for receiving)	RWr	Batch/random	○*3	○*3	×	○	○	×
Direct link special relay (other station side)	—	Batch/random	○	○*1	×	○	○	×
Direct link special register (other station side)	—	Batch/random	○	○*1	×	○	○	×
Intelligent function module device, module access device	Un\G	Batch/random	○	○*1	×	○	○	×
CPU shared memory, CPU buffer memory (CPU No.1 area)	U3E0\G	Batch	○	○	×	×	○	×
		Random	×	×			×	
CPU shared memory, CPU buffer memory (CPU No.2 area)	U3E1\G	Batch	○	○	×	×	○	×
		Random	×	×			×	
CPU shared memory, CPU buffer memory (CPU No.3 area)	U3E2\G	Batch	○	○	×	×	○	×
		Random	×	×			×	
CPU shared memory, CPU buffer memory (CPU No.4 area)	U3E3\G	Batch	○	○	×	×	○	×
		Random	×	×			×	
Fixed cycle communication area (CPU No.1 area)	U3E0\HG	Batch	×	×	×	×	○	×
		Random					×	
Fixed cycle communication area (CPU No.2 area)	U3E1\HG	Batch	×	×	×	×	○	×
		Random					×	
Fixed cycle communication area (CPU No.3 area)	U3E2\HG	Batch	×	×	×	×	○	×
		Random					×	
Fixed cycle communication area (CPU No.4 area)	U3E3\HG	Batch	×	×	×	×	○	×
		Random					×	
Other station buffer memory	—	Batch/random	×	×	×	×	×	×
Other station random access buffer	—	Batch/random	×	×	×	×	×	×
Remote input for CC-Link	RX	Batch/random	×	×	×	×	×	×
Remote output for CC-Link	RY	Batch/random	×	×	×	×	×	×
Other station link register	—	Batch/random	×	×	×	×	×	×
Link special relay for CC-Link	SB	Batch/random	×	×	×	×	×	×
Link special register for CC-Link	SW	Batch/random	×	×	×	×	×	×
Global label	GV	Batch	×	×	×	×	×	×
		Random	×	×	×	×	×	×
Safety input	SA\X	Batch/random	×	×	×	×	×	×
Safety output	SA\Y	Batch/random	×	×	×	×	×	×
Safety internal relay	SA\M	Batch/random	×	×	×	×	×	×
Safety link relay	SA\B	Batch/random	×	×	×	×	×	×
Safety timer	SA\T	Batch/random	×	×	×	×	×	×
Safety retentive timer	SA\ST	Batch/random	×	×	×	×	×	×
Safety counter	SA\C	Batch/random	×	×	×	×	×	×
Safety data register	SA\D	Batch/random	×	×	×	×	×	×
Safety link register	SA\W	Batch/random	×	×	×	×	×	×
Safety special relay	SA\SM	Batch/random	×	×	×	×	×	×
Safety special register	SA\SD	Batch/random	×	×	×	×	×	×

*1 The following CPUs are accessible:
Q12DCCPU-V with a serial number of which the first 5 digits are '12042' or later
Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS

*2 Q00JCPU is not accessible.

*3 The following CPUs are accessible:
Q12DCCPU-V (Extended mode)
Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS

*4 A message is send to a network module on the other station via a MELSECNET/H network module.
Access to a multiple CPU system (when a logical station number is specified) is not available.

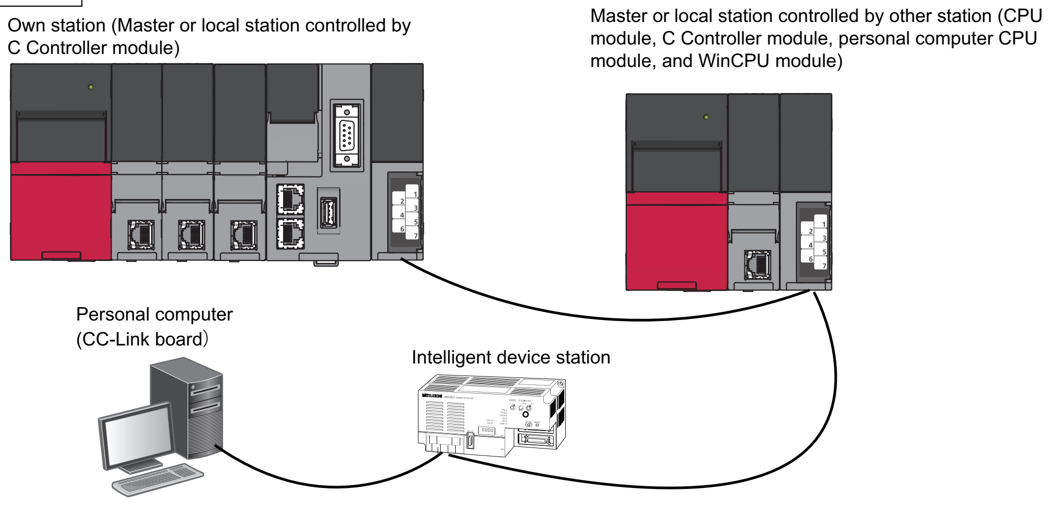
CC-Link communication

The accessible range and devices for CC-Link communication are shown below:

■Accessible range

The accessible range for CC-Link communication includes the own station (master station or local station controlled by C Controller module), master station or local station controlled by other stations (CPU module, C Controller module, PC CPU module, WinCPU module), intelligent device station and a personal computer on which a CC-Link board is installed.

System configuration


Own station (Master or local station controlled by C Controller module)
Master or local station controlled by other station (CPU module, C Controller module, personal computer CPU module, and WinCPU module)
Personal computer (CC-Link board)
Intelligent device station

Point

When the own station number is 64, other stations cannot be accessed.
Only the own station can be accessed.

■Accessible devices

The following explains the devices accessible for communication via a CC-Link module.

Point

- 'Batch' and 'Random' in the following table indicate as follows:
  Batch: Batch write (mdSendEx function), batch read (mdReceiveEx function)
  Random: Random write (mdRandWEx function), random read (mdRandREx function), bit set (mdDevSetEx function), bit reset (mdDevRstEx function), random write by using a label name (mdRandWLabelEx function), random read by using a label name (mdRandRLabelEx function)
- Bit devices can be accessed by using 'bit set' (mdDevSetEx function) and 'bit reset' (mdDevRstEx function).
- Device extension specifications (digit specification, bit specification and index specification) cannot be used.

- Accessing the own station

To access a CC-Link module controlled by a C Controller module, use the method explained in the following section.
Accessing the own station by using CC-Link communication will result in the station number/network number error.
☞ Page 183 Replacement of device type

• Accessing other stations

No.	Access target CPU
(1)	Basic model QCPU, high performance model QCPU, process CPU, redundant CPU, universal model QCPU
(2)	Q12DCCPU-V, Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS
(3)	PC CPU module, WinCPU module, personal computer, and intelligent device station
(4)	L02CPU, L26CPU-BT, L02CPU-P, L26CPU-PBT, L02SCPU, L26CPU, and L06CPU
(5)	RCPU
(6)	R12CCPU-V

○: Accessible, ×: Not accessible

Device		Access method	Access target CPU					
			(1)	(2)	(3)	(4)	(5)	(6)
Input relay	X	Batch/random	○	○*1	×	○	○	○
Output relay	Y	Batch/random	○	○*1	×	○	○	○
Latch relay	L	Batch/random	○	×	×	○	○	×
Internal relay	M	Batch/random	○	○*1	×	○	○	○
Special relay	SM	Batch/random	○	○*1	×	○	○	○
Annunciator	F	Batch/random	○	×	×	○	○	×
Timer (Contact)	T	Batch/random	○	×	×	○	○	×
Long timer (Contact)	LT	Batch/random	×	×	×	×	○	×
Timer (Coil)	T	Batch/random	○	×	×	○	○	×
Long timer (Coil)	LT	Batch/random	×	×	×	×	○	×
Counter (Contact)	C	Batch/random	○	×	×	○	○	×
Long counter (Contact)	LC	Batch/random	×	×	×	×	○	×
Counter (Coil)	C	Batch/random	○	×	×	○	○	×
Long counter (Coil)	LC	Batch/random	×	×	×	×	○	×
Timer (Current value)	T	Batch/random	○	×	×	○	○	×
Long timer (Current value)	LT	Batch/random	×	×	×	×	○	×
Counter (Current value)	C	Batch/random	○	×	×	○	○	×
Long counter (Current value)	LC	Batch/random	×	×	×	×	○	×
Data register	D	Batch/random	○	○*1	×	○	○	○
Special register	SD	Batch/random	○	○*1	×	○	○	○
Index register	Z	Batch/random	○	×	×	○	○	×
Long index register	LZ	Batch/random	×	×	×	×	○	×
File register	R	Batch/random	○*2	×	×	○	○	×
	ZR	Batch/random	○*2	×	×	○	○	○
Refresh data register	RD	Batch/random	×	×	×	×	○	×
Link relay	B	Batch/random	○	○*3	×	○	○	○
Link register	W	Batch/random	○	○*3	×	○	○	○
Link special relay	SB	Batch/random	○	×	×	○	○	×
Retentive timer (Contact)	ST	Batch/random	○	×	×	○	○	×
Long retentive timer (Contact)	LST	Batch/random	×	×	×	×	○	×
Retentive timer (Coil)	ST	Batch/random	○	×	×	○	○	×
Long retentive timer (Coil)	LST	Batch/random	×	×	×	×	○	×
Link special register	SW	Batch/random	○	×	×	○	○	×
Edge relay	V	Batch/random	○	×	×	○	○	×
Own station random access buffer	—	Batch/random	×	×	×	×	×	×
Retentive timer (Current value)	ST	Batch/random	○	×	×	○	○	×
Long retentive timer (Current value)	LST	Batch/random	×	×	×	×	○	×
Remote register for sending	RWw	Batch/random	×	×	×	×	×	×
Remote register for receiving	RWr	Batch/random	×	×	×	×	×	×
Own station buffer memory	—	Batch/random	×	×	×	×	×	×
SEND function (with arrival confirmation)	—	Batch/random	×	×	×	×	×	×
SEND function (without arrival confirmation)	—	Batch/random	×	×	×	×	×	×

Device		Access method	Access target CPU					
			(1)	(2)	(3)	(4)	(5)	(6)
Link direct device (Link input)	Jn\X	Batch/random	○	○*1	×	○	○	○
Link direct device (Link output)	Jn\Y	Batch/random	○	○*1	×	○	○	○
Link direct device (Link relay)	Jn\B	Batch/random	○	○*1	×	○	○	○
Link direct device (Link register)	Jn\W	Batch/random	○	○*1	×	○	○	○
Link direct device (Link special relay)	Jn\SB	Batch/random	○	○*1	×	○	○	○
Link direct device (Link special register)	Jn\SW	Batch/random	○	○*1	×	○	○	○
Intelligent function module device, module access device	Un\G	Batch/random	○	○*1	×	○	○	○
CPU shared memory, CPU buffer memory (CPU No.1 area)	—	Batch	○	○*1	×	×	○	○
		Random	×	×			×	×
CPU shared memory, CPU buffer memory (CPU No.2 area)	—	Batch	○	○*1	×	×	○	○
		Random	×	×			×	×
CPU shared memory, CPU buffer memory (CPU No.3 area)	—	Batch	○	○*1	×	×	○	○
		Random	×	×			×	×
CPU shared memory, CPU buffer memory (CPU No.4 area)	—	Batch	○	○*1	×	×	○	○
		Random	×	×			×	×
Fixed cycle communication area (CPU No.1 area)	—	Batch	×	×	×	×	○	○
		Random					×	×
Fixed cycle communication area (CPU No.2 area)	—	Batch	×	×	×	×	○	○
		Random					×	×
Fixed cycle communication area (CPU No.3 area)	—	Batch	×	×	×	×	○	○
		Random					×	×
Fixed cycle communication area (CPU No.4 area)	—	Batch	×	×	×	×	○	○
		Random					×	×
Global label	GV	Batch	×	×	×	×	×	×
		Random	×	×	×	×	○	×
Safety input	SA\X	Batch/random	×	×	×	×	×	×
Safety output	SA\Y	Batch/random	×	×	×	×	×	×
Safety internal relay	SA\M	Batch/random	×	×	×	×	×	×
Safety link relay	SA\B	Batch/random	×	×	×	×	×	×
Safety timer	SA\T	Batch/random	×	×	×	×	×	×
Safety retentive timer	SA\ST	Batch/random	×	×	×	×	×	×
Safety counter	SA\C	Batch/random	×	×	×	×	×	×
Safety data register	SA\D	Batch/random	×	×	×	×	×	×
Safety link register	SA\W	Batch/random	×	×	×	×	×	×
Safety special relay	SA\SM	Batch/random	×	×	×	×	×	×
Safety special register	SA\SD	Batch/random	×	×	×	×	×	×

*1 The following CPUs are accessible:
Q12DCCPU-V with a serial number of which first 5 digits are "12042" or later
Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS

*2 Q00JCPU is not accessible.

*3 The following CPUs are accessible:
Q12DCCPU-V (Extended mode),
Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, and Q26DHCCPU-LS

Argument specification

This section shows the argument specification of the MELSEC data link functions.

Channel

A channel implies a network and communication route to be used when communicating with a C Controller module.
It needs to be set for each module in a user program.
Channels to be used for MELSEC data link functions are as follows:

Channel number	Network	Communication route
12	Bus interface	Used for communication via a bus.
151 to 158	CC-Link IE Controller Network	Used for communication via a CC-Link IE Controller Network module controlled by a C Controller module.
181 to 188	CC-Link IE Field Network	Used for communication via a CC-Link IE Field Network module controlled by a C Controller module.
51 to 54	MELSECNET/H network	Used for communication via a MELSECNET/H network module controlled by a C Controller module.
81 to 88	CC-Link	Used for communication via a CC-Link module controlled by a C Controller module.

Network number and station number

■Network number and station number for MELSEC data link functions (excluding the mdControl function and the mdTypeRead function)

Network numbers and station numbers to be specified to MELSEC data link functions are as follows:

To specify station numbers for the mdControl function and mdTypeRead function, refer to "Network number and station number for MELSEC data link functions (the mdControl function and the mdTypeRead function)".

Communication	Specification method		Network number	Station number
Bus interface	Own station		0 (0H)	255 (FFH)*3
	Other station			1 (CPU No.1), 2 (CPU No.2), 3 (CPU No.3), and 4 (CPU No.4)
CC-Link IE Controller Network	Own station		0 (0H)	255 (FFH)
	Other station	Station number	1 (1H) to 239 (EFH)	1 (1H) to 120 (78H) 0 (0H)*7, 125 (7DH)*7
		Group number 1 to 32*4,*5		129 (81H) to 160 (A0H)
		All stations*4		240 (F0H)
	Logical station number*1		0 (0H)	65 (41H) to 239 (EFH)
CC-Link IE Field Network	Own station		0 (0H)	255 (FFH)
	Other station	Station number	1 (1H) to 239 (EFH)	0 (0H) to 120 (78H), 125 (7DH)*6
		All stations*4		240 (F0H)
	Logical station number*1		0 (0H)	65 (41H) to 239 (EFH)
MELSECNET/H network	Own station		0 (0H)	255 (FFH)
	Other station	Station number	1 (1H) to 239 (EFH)	1 (1H) to 64 (40H) 0 (0H)*7, 125 (7DH)*7
		Group number 1 to 32*4,*5		129 (81H) to 160 (A0H)
		All stations*4		240 (F0H)
	Logical station number*1		0 (0H)	65 (41H) to 239 (EFH)
CC-Link	Other station		0 (0H)	0(0H) to 63(3FH)*2
	Logical station number*1			65 (41H) to 239 (EFH)

*1 Logical station numbers are logical numbers which specifies "station number" in a user program (MELSEC data link functions).
Logical station numbers are used to access from an applicable module (channel number) to the other station CPU (other CPU of a multiple CPU system).
To access directly to a CPU module which controls other stations on MELSECNET/10 network, MELSECNET/H network, CC-Link IE Controller Network, and CC-Link IE Field Network, the logical station number is not required to be set. Use the station numbers of MELSECNET/10 network, MELSECNET/H network, CC-Link IE Controller Network, and CC-Link IE Field Network.
Also, for direct access to CC-Link other stations (station number 0 to 63) or the CPU module which controls CC-Link other stations, setting of the local station number is not required. Specify or use the station number of CC-Link.

*2 The station number 64 cannot be specified for CC-Link communication.

*3 Communication to C Controller module (own station) by using the MELSEC data link functions is possible; however, it may take longer to execute the functions compared to the C Controller module dedicated functions. Use the C Controller module dedicated functions to create a user program in which performance should be ensured (such as control program).

*4 The group number and all stations specification is valid when the SEND function (mdSendEX (message send function)) with 'no arrival confirmation' specification is used.

*5 The group number can be specified when CC-Link IE Controller Network and MELSECNET/H network are used.

*6 125 (7DH) is valid when CC-Link IE Field Network master station (station number 0) is specified by using the SEND/RECV function (mdSendEx (message send function) or mdReceiveEX (message receive function)). To access a master operating station (a station that is operating as the master station when the submaster function is used), specify the station number from 1 (1H) to 120 (78H).

*7 When '0 (0H)' or '125 (7DH)' is specified for the station number, a specified control station of the network, which is specified to the network number, is accessed. To access the current control station (a station that is actually operating as the control station), specify the station number from 1 (1H) to 120 (78H).

■Network number and station number for MELSEC data link functions (the mdControl function and the mdTypeRead function)

Network numbers and station numbers to be specified to the mdControl function or the mdTypeRead function are as follows:

Communication	Station number specification method
Bus interface	Own station: 255 (FFH) Other station: 1 (CPU No.1), 2 (CPU No.2), 3 (CPU No.3), 4 (CPU No.4)
CC-Link IE Controller Network	Own station: 255 (FFH) Other station: *1,*2
CC-Link IE Field Network	Own station: 255 (FFH) Other station: *1,*3
MELSECNET/H network	Own station: 255 (FFH) Other station: *1,*2
CC-Link	Own station: 255 (FFH) Other stations: 0 (0H) to 63 (3FH), 65 (41H) to 239 (EFH)*4,*5

*1 Station number setting for a CC-Link IE Controller Network module, CC-Link IE Field Network module, and MELSECNET/H network module

Upper	Lower

Upper/ lower	Setting item	Setting value	Description
Upper	Network number	1 (1H) to 239 (EFH)	Set this to specify other stations in the own network or each station on other networks. • Set this to issue a sending request to any of CC-Link IE Field Network, CC-Link IE Controller Network, MELSECNET/H network, or MELSECNET/10 network.
Lower	Station number, Group number, or All stations	1 (1H) to 120 (78H)	Set the station number of other stations. • For MELSECNET/H network, the setting range is from 1 to 64. • For CC-Link IE Field Network, the setting range is from 0 to 120.
		129 (81H) to 160 (A0H)	Set the group number 1 to 32. • For MELSECNET/10 network mode, 129 (81H) to 137 (89H): Group number 1 to 9. • The group number specification is valid when the SEND function (mdSendEx (message send function)) is used. (Cannot be specified if CC-Link IE Field Network is used.)
		240 (F0H)	Set all stations. • Specifying all stations is enabled when the SEND function (mdSendEx (message send function)) is used.

Logical station number setting method
Set '0' in the upper byte (network number) of the station number above, and specify a logical station number in the lower byte (station number). The setting range of the logical station number is 65 (41H) to 239 (EFH). Set the logical station number with the user program (MELSEC data link functions).

*2 When '0 (0H)' or '125 (7DH)' is specified for the station number, a specified control station of the network, which is specified to the network number, is accessed. To access the current control station (a station that is actually operating as the control station), specify the station number from 1 (1H) to 120 (78H).

*3 125 (7DH) is valid when CC-Link IE Field Network master station (station number 0) is specified by using the SEND/RECV function (mdSendEx (message send function) or mdReceiveEX (message receive function)). To access a master operating station (a station that is operating as the master station when the submaster function is used), specify the station number from 1 (1H) to 120 (78H).

*4 Station number setting for CC-Link module

Upper	Lower

Upper/ lower	Setting item	Setting value	Description
Upper	Network number	0	Set the value for CC-Link.
Lower	Station number	0 (0H) to 63 (3FH)	Set the station number of other stations.

Logical station number setting method
Set '0' in the upper byte (network number) of the station number above, and specify a logical station number in the lower byte (station number). The setting range of the logical station number is 65 (41H) to 239 (EFH). Set the logical station number with the user program (MELSEC data link functions).

*5 The station number 64 cannot be specified for CC-Link communication.
In addition, when the station number of the own station is 64, other stations cannot be set. (Only the own station can be accessed.)

Device type

The following tables show the device types specified to the MELSEC data link functions.
Devices are defined in the header file "MDFunc.h".

Point

Either a code or a device name can be specified as a device type.

■Common device types

Device (Device name)		Device type		
		Code		Device name
		Decimal	Hexadecimal	
Input relay (X)		1	1H	DevX
Output relay (Y)		2	2H	DevY
Latch relay (L)		3	3H	DevL
Internal relay (M)		4	4H	DevM
Special relay (SM)		5	5H	DevSM
CPU buffer memory[*1,*2]	CPU No.1 area (U3E0\G)	501	1F5H	DevSPB1
	CPU No.2 area (U3E1\G)	502	1F6H	DevSPB2
	CPU No.3 area (U3E2\G)	503	1F7H	DevSPB3
	CPU No.4 area (U3E3\G)	504	1F8H	DevSPB4
Fixed cycle communication area[*1,*2]	CPU No.1 area (U3E0\HG)	511	1FFH	DevHSPB1
	CPU No.2 area (U3E1\HG)	512	200H	DevHSPB2
	CPU No.3 area (U3E2\HG)	513	201H	DevHSPB3
	CPU No.4 area (U3E3\HG)	514	202H	DevHSPB4
Annunciator (F)		6	6H	DevF
Timer	Contact (T)	7	7H	DevTT
	Coil (T)	8	8H	DevTC
	Current value (T)	11	BH	DevTN
Long timer	Contact (LT)	41	29H	DevLTT
	Coil (LT)	42	2AH	DevLTC
	Current value (LT)	43	2BH	DevLTN
Counter	Contact (C)	9	9H	DevCT
	Coil (C)	10	AH	DevCC
	Current value (C)	12	CH	DevCN
Long counter (Contact)	Contact (LC)	44	2CH	DevLCT
	Coil (LC)	45	2DH	DevLCC
	Current value (LC)	46	2EH	DevLCN
Retentive timer	Contact (ST)	26	1AH	DevSTT
	Coil (ST)	27	1BH	DevSTC
	Current value (ST)	35	23H	DevSTN
Long retentive timer	Contact (LST)	47	2FH	DevLSTT
	Coil (LST)	48	30H	DevLSTC
	Current value (LST)	49	31H	DevLSTN
Data register (D)		13	DH	DevD
Special register (SD)		14	EH	DevSD
Index register (Z)[*3]		20	14H	DevZ
Long index register (LZ)[*3]		38	26H	DevLZ
File register (R)[*3]		22	16H	DevR
File register (ZR)[*3]		220	DCH	DevZR
Link relay (B)		23	17H	DevB
Link register (W)		24	18H	DevW
Link special relay (SB)[*3]		25	19H	DevQSB
Link special register (SW)[*3]		28	1CH	DevQSW

Device (Device name)		Device type		
		Code		Device name
		Decimal	Hexadecimal	
Edge relay (V)		30	1EH	DevQV
Refresh data register (RD)		39	27H	DevRD
Global label (GV)[*4]	For word, double word, and quad word size	600	258H	DevGV
	For bit 0	601	259H	DevGV_0
	For bit 1	602	25AH	DevGV_1
	For bit 2	603	25BH	DevGV_2
	For bit 3	604	25CH	DevGV_3
	For bit 4	605	25DH	DevGV_4
	For bit 5	606	25EH	DevGV_5
	For bit 6	607	25FH	DevGV_6
	For bit 7	608	260H	DevGV_7
	For bit 8	609	261H	DevGV_8
	For bit 9	610	262H	DevGV_9
	For bit A	611	263H	DevGV_A
	For bit B	612	264H	DevGV_B
	For bit C	613	265H	DevGV_C
	For bit D	614	266H	DevGV_D
	For bit E	615	267H	DevGV_E
	For bit F	616	268H	DevGV_F
Link direct device[*3][*5] Argument value of device name (1 to 255): Network number	Link input (J□\X)	1001 to 1255	3E9H to 4E7H	DevLX(1) to DevLX(255)
	Link output (J□\Y)	2001 to 2255	7D1H to 8CFH	DevLY(1) to DevLY(255)
	Link relay (J□\B)	23001 to 23255	59D9H to 5AD7H	DevLB(1) to DevLB(255)
	Link register (J□\W)	24001 to 24255	5DC1H to 5EBFH	DevLW(1) to DevLW(255)
	Link special relay (J□\SB)	25001 to 25255	61A9H to 62A7H	DevLSB(1) to DevLSB(255)
	Link special register (J□\SW)	28001 to 28255	6D61H to 6E5FH	DevLSW(1) to DevLSW(255)
Intelligent function module device[*3], module access device[*3] Argument value of device name (0 to 255): Start I/O No. divided by 16		29000 to 29255	7148H to 7247H	DevSPG(0) to DevSPG(255)
SEND function (with arrival confirmation) and RECV function		101	65H	DevMAIL
SEND function (without arrival confirmation)		102	66H	DevMAILNC

*1 For Q12DCCPU-V, it is categorized as the device type for Q bus interface.
(It is not accessible in CC-Link communication, CC-Link IE Controller Network communication and CC-Link IE Field Network communication.)

*2 The devices cannot be used for the mdRandREx, mdRandWEx, mdDevSetEx, and mdDevRstEx functions.

*3 Even if a non-existent device is specified in the mdRandREx function, the function may end normally.
(All of the bits turn ON in read data. For word devices, the read data is '-1'.)

*4 Only the mdRandRLabelEx and mdRandWLabelEx functions can be used.

*5 □: indicates a network number.

■Device types for CC-Link IE Controller Network module access

The device types shown in the following table can be specified in the user program:

- For sending/receiving message

Device	Device type		
	Code		Device name
	Decimal	Hexadecimal	
SEND function (with arrival confirmation) and RECV function	101	65H	DevMAIL
SEND function (without arrival confirmation)	102	66H	DevMAILNC

■Device types for CC-Link IE Field Network module access

The device types shown in the following table can be specified in the user program:

- For sending/receiving message

Device	Device type		
	Code		Device name
	Decimal	Hexadecimal	
SEND function (with arrival confirmation) and RECV function	101	65H	DevMAIL
SEND function (without arrival confirmation)	102	66H	DevMAILNC

■Device types for MELSECNET/H network module access

The device types shown in the following table can be specified in the user program:

- For sending/receiving message

Device	Device type		
	Code		Device name
	Decimal	Hexadecimal	
SEND function (with arrival confirmation) and RECV function	101	65H	DevMAIL
SEND function (without arrival confirmation)	102	66H	DevMAILNC

1.4 Considerations on Interrupt Service Routine (ISR)

Fully understand the restrictions of VxWorks, operating system, before creating a routine which is executed in an interrupt service routine (ISR: InterruptServiceRoutine). To use another dedicated function by synchronizing it to an interrupt, implement the notification processing in a user program and perform it in a task.

Point

Setting an inappropriate value to an argument of the C Controller module dedicated function for ISR or executing a function other than the C Controller module dedicated function for ISR from an interrupt service routine may cause the VxWorks runaway.

2 FUNCTION LIST

This chapter describes the functions that can be used for a C Controller module.

2.1 C Controller Module Dedicated Functions

The C Controller module dedicated functions are as listed below.

C Controller module dedicated functions

Function name	Function	Reference
CCPU_ChangeFileSecurity	Changes the file access restriction status of a C Controller module.	Page 48 CCPU_ChangeFileSecurity
CCPU_ClearError	Clears errors of a C Controller module.	Page 49 CCPU_ClearError
CCPU_Control	Performs remote operations (remote RUN/STOP/PAUSE) for the CPU module.	Page 50 CCPU_Control
CCPU_DedicatedDInst	Executes dedicated instructions categorized as 'D' or 'DP'.	Page 51 CCPU_DedicatedDInst
CCPU_DedicatedGInst	Executes dedicated instructions categorized as 'G' or 'GP'.	Page 53 CCPU_DedicatedGInst
CCPU_DedicatedJInst	Executes dedicated instructions categorized as 'J' or 'JP'.	Page 55 CCPU_DedicatedJInst
CCPU_DedicatedMInst	Executes dedicated instructions categorized as 'M' or 'MP'.	Page 57 CCPU_DedicatedMInst
CCPU_DisableInt	Disables the routine registered with the CCPU_EntryInt function.	Page 59 CCPU_DisableInt
CCPU_EnableInt	Enables the routine registered with the CCPU_EntryInt function.	Page 60 CCPU_EnableInt
CCPU_EntryInt	Registers a routine to be called when an interrupt occurs.	Page 61 CCPU_EntryInt
CCPU_EntryTimerEvent	Registers a timer event.	Page 62 CCPU_EntryTimerEvent
CCPU_EntryWDTInt	Registers a routine to be called when a user WDT error interrupt occurs.	Page 63 CCPU_EntryWDTInt
CCPU_FromBuf	Reads data from the CPU buffer memory of the CPU module and the buffer memory of the intelligent function module which are mounted on the specified module position. (FROM instruction)	Page 64 CCPU_FromBuf
CCPU_FromBufHG	Reads data from the fixed cycle communication area of the CPU module mounted on the specified module position.	Page 65 CCPU_FromBufHG
CCPU_GetConstantProcessStatu s	Obtains the fixed cycle processing status of a C Controller module.	Page 66 CCPU_GetConstantProcessStatus
CCPU_GetCounterMicros	Obtains a 1 µs counter value of a C Controller module.	Page 67 CCPU_GetCounterMicros
CCPU_GetCounterMillis	Obtains a 1 ms counter value of a C Controller module.	Page 68 CCPU_GetCounterMillis
CCPU_GetCpuStatus	Obtains the operating status of a C Controller module.	Page 69 CCPU_GetCpuStatus
CCPU_GetDotMatrixLED	Obtains the value displayed on the dot matrix LED of a C Controller module.	Page 70 CCPU_GetDotMatrixLED
CCPU_GetErrInfo	Obtains the error information of a C Controller module.	Page 71 CCPU_GetErrInfo
CCPU_GetFileSecurity	Obtains the file access mode of a C Controller module.	Page 72 CCPU_GetFileSecurity
CCPU_GetIDInfo	Obtains the individual identification information of a C Controller module.	Page 73 CCPU_GetIDInfo
CCPU_GetLEDStatus	Obtains the LED status of a C Controller module.	Page 74 CCPU_GetLEDStatus
CCPU_GetOpSelectMode	Obtains the operation selection mode of a C Controller module.	Page 76 CCPU_GetOpSelectMode
CCPU_GetPowerStatus	Obtains the power status of a C Controller module.	Page 77 CCPU_GetPowerStatus
CCPU_GetRTC	Obtains the clock data (local time) of a C Controller module.	Page 78 CCPU_GetRTC
CCPU_GetSerialNo	Obtains the serial number of a C Controller module.	Page 79 CCPU_GetSerialNo
CCPU_GetSwitchStatus	Obtains the switch status of a C Controller module.	Page 80 CCPU_GetSwitchStatus
CCPU_GetUnitInfo	Obtains the module configuration information.	Page 81 CCPU_GetUnitInfo
CCPU_MountMemoryCard	Mounts the SD memory card inserted to a C Controller module.	Page 84 CCPU_MountMemoryCard
CCPU_ReadDevice	Reads data from internal user devices and internal system devices of a C Controller module.	Page 85 CCPU_ReadDevice
CCPU_ReadLinkDevice	Reads data from the own station link devices of CC-Link IE Controller Network module, CC-Link IE Field Network module, Ethernet module (when CC-Link IE Field Network is selected), and MELSECNET/H network module.	Page 86 CCPU_ReadLinkDevice
CCPU_RegistEventLog	Registers event logs in the event history of a C Controller module.	Page 87 CCPU_RegistEventLog
CCPU_Reset	Resets the bus master CPU (CPU No.1).	Page 88 CCPU_Reset

Function name	Function	Reference
CCPU_ResetDevice	Resets internal user devices and internal system devices (bit devices) of C Controller module.	Page 89 CCPU_ResetDevice
CCPU_ResetWDT	Resets the user WDT of C Controller module.	Page 90 CCPU_ResetWDT
CCPU_SetDevice	Sets internal user devices and internal system devices (bit devices) of C Controller module.	Page 91 CCPU_SetDevice
CCPU_SetDotMatrixLED	Sets a value to be displayed on the dot matrix LED of C Controller module.	Page 92 CCPU_SetDotMatrixLED
CCPU_SetLEDStatus	Sets the LED status of a C Controller module.	Page 94 CCPU_SetLEDStatus
CCPU_SetOpSelectMode	Sets the operation selection mode of C Controller module.	Page 95 CCPU_SetOpSelectMode
CCPU_SetRTC	Sets the clock data (local time) of C Controller module.	Page 96 CCPU_SetRTC
CCPU_ShutdownRom	Shuts down the program memory and the data memory of a C Controller module.	Page 97 CCPU_ShutdownRom
CCPU_StartWDT	Sets and starts the user WDT of C Controller module.	Page 98 CCPU_StartWDT
CCPU_StopWDT	Stops the user WDT of C Controller module.	Page 99 CCPU_StopWDT
CCPU_SysClkRateGet	Reads the system clock rate specified with the CCPU_SysClkRateSet function from the backup RAM.	Page 100 CCPU_SysClkRateGet
CCPU_SysClkRateSet	Specifies a system clock rate, and stores it in the backup RAM.	Page 101 CCPU_SysClkRateSet
CCPU_ToBuf	Writes data to the CPU buffer memory of the CPU module (host CPU) and the buffer memory of the intelligent function module which are mounted on the specified module position. (TO instruction)	Page 102 CCPU_ToBuf
CCPU_ToBufHG	Writes data to the fixed cycle communication area of the CPU module mounted on the specified module position.	Page 103 CCPU_ToBufHG
CCPU_UnmountMemoryCard	Unmounts the SD memory card and USB Mass Storage Class-compliant device connected to C Controller module.	Page 104 CCPU_UnmountMemoryCard
CCPU_WaitEvent	Waits for an interrupt event notification from other CPUs.	Page 105 CCPU_WaitEvent
CCPU_WaitSwitchEvent	Waits for a switch interrupt event of C Controller module to occur.	Page 107 CCPU_WaitSwitchEvent
CCPU_WaitTimerEvent	Waits for a timer event to occur.	Page 108 CCPU_WaitTimerEvent
CCPU_WaitUnitEvent	Waits for an interrupt event notification from modules.	Page 109 CCPU_WaitUnitEvent
CCPU_WriteDevice	Writes data to internal user devices and internal system devices of C Controller module.	Page 111 CCPU_WriteDevice
CCPU_WriteLinkDevice	Writes data to own station link devices of CC-Link IE Controller Network module, CC-Link IE Field Network module, Ethernet module (when CC-Link IE Field Network is selected), and MELSECNET/H network module.	Page 112 CCPU_WriteLinkDevice
CCPU_X_In_BitEx	Reads the input signal (X) in bit (1-point) units.	Page 113 CCPU_X_In_BitEx
CCPU_X_In_WordEx	Reads the input signal (X) in word (16-point) units.	Page 114 CCPU_X_In_WordEx
CCPU_Y_In_BitEx	Reads the output signal (Y) in bit (1-point) units.	Page 115 CCPU_Y_In_BitEx
CCPU_Y_In_WordEx	Reads the output signal (Y) in word (16-point) units.	Page 116 CCPU_Y_In_WordEx
CCPU_Y_Out_BitEx	Outputs the output signal (Y) in bit (1-point) units.	Page 117 CCPU_Y_Out_BitEx
CCPU_Y_Out_WordEx	Outputs the output signal (Y) in word (16-point) units.	Page 118 CCPU_Y_Out_WordEx

C Controller module dedicated functions for ISR

Function name	Function	Reference
CCPU_DisableInt_ISR	Disables the routine registered with the CCPU_EntryInt function.	Page 119 CCPU_DisableInt_ISR
CCPU_EnableInt_ISR	Enables the routine registered with the CCPU_EntryInt function.	Page 120 CCPU_EnableInt_ISR
CCPU_FromBuf_ISR	Reads data from the CPU buffer memory of the CPU module and the buffer memory of the intelligent function module which are mounted on the specified module position. (FROM instruction)	Page 121 CCPU_FromBuf_ISR
CCPU_FromBufHG_ISR	Reads data from the fixed cycle communication area of the CPU module mounted on the specified module position.	Page 122 CCPU_FromBufHG_ISR
CCPU_GetCounterMicros_ISR	Obtains a 1 μs counter value of C Controller module.	Page 123 CCPU_GetCounterMicros_ISR
CCPU_GetCounterMillis_ISR	Obtains a 1 ms counter value of C Controller module.	Page 124 CCPU_GetCounterMillis_ISR
CCPU_GetDotMatrixLED_ISR	Obtains the value displayed on the dot matrix LED of C Controller module.	Page 125 CCPU_GetDotMatrixLED_ISR
CCPU_ReadDevice_ISR	Reads data from internal user devices and internal system devices of C Controller module.	Page 127 CCPU_ReadDevice_ISR
CCPU_RegistEventLog_ISR	Registers event logs in the event history of C Controller module.	Page 128 CCPU_RegistEventLog_ISR
CCPU_ResetDevice_ISR	Resets internal user devices and internal system devices (bit devices) of C Controller module.	Page 127 CCPU_ReadDevice_ISR
CCPU_SetDevice_ISR	Sets internal user devices and internal system devices (bit devices) of C Controller module.	Page 130 CCPU_SetDevice_ISR
CCPU_SetDotMatrixLED_ISR	Sets a value to be displayed on the dot matrix LED of C Controller module.	Page 131 CCPU_SetDotMatrixLED_ISR
CCPU_SetLEDStatus_ISR	Sets the LED status of a C Controller module.	Page 133 CCPU_SetLEDStatus_ISR
CCPU_ToBuf_ISR	Writes data to the CPU buffer memory of the CPU module (host CPU) and the buffer memory of the intelligent function module which are mounted on the specified module position. (TO instruction)	Page 134 CCPU_ToBuf_ISR
CCPU_ToBufHG_ISR	Writes data to the fixed cycle communication area of the CPU module mounted on the specified module position.	Page 135 CCPU_ToBufHG_ISR
CCPU_WriteDevice_ISR	Writes data to internal user devices and internal system devices of C Controller module.	Page 136 CCPU_WriteDevice_ISR
CCPU_X_In_Word_ISR	Reads the input signal (X) in word (16-point) units.	Page 137 CCPU_X_In_Word_ISR
CCPU_Y_In_Word_ISR	Reads the output signal (Y) in word (16-point) units.	Page 138 CCPU_Y_In_Word_ISR
CCPU_Y_Out_Word_ISR	Outputs the output signal (Y) in word (16-point) units.	Page 139 CCPU_Y_Out_Word_ISR

2.2 MELSEC Data Link Functions

The MELSEC data link functions are as listed below.

Function name	Function	Reference
mdClose	Closes a communication line (channel).	Page 140 mdClose
mdControl	Performs remote operations (remote RUN/STOP/PAUSE) for the CPU module.	Page 141 mdControl
mdDevRstEx	Resets bit devices.	Page 142 mdDevRstEx
mdDevSetEx	Sets bit devices.	Page 143 mdDevSetEx
mdGetLabelInfo	Obtains device information corresponding to label names.	Page 144 mdGetLabelInfo
mdInit	Initializes the communication route information.	Page 147 mdInit
mdOpen	Opens a communication line (channel).	Page 148 mdOpen
mdRandREx	Reads devices randomly.	Page 149 mdRandREx
mdRandRLabelEx	Reads devices corresponding to labels randomly.	Page 152 mdRandRLabelEx
mdRandWEx	Writes devices randomly.	Page 155 mdRandWEx
mdRandWLabelEx	Writes devices corresponding to labels randomly.	Page 157 mdRandWLabelEx
mdReceiveEx	Reads devices in batch.	Page 159 mdReceiveEx
mdReceiveEx	Receives messages. (RECV function)	Page 160 mdReceiveEx
mdSendEx	Writes devices in batch.	Page 162 mdSendEx
mdSendEx	Sends messages. (SEND function)	Page 163 mdSendEx
mdTypeRead	Reads the model code of CPU module.	Page 165 mdTypeRead

3 DETAILS OF FUNCTION

This chapter explains the details of the C Controller module dedicated function and the MELSEC data link function.

3.1 C Controller Module Dedicated Functions

This section explains the details of the C Controller module dedicated function.

C Controller module dedicated functions

CCPU_ChangeFileSecurity

This function changes the file access restriction status of a C Controller module.

■Format

short CCPU_ChangeFileSecurity(short sMode, char* pcPass);

■Argument

Argument	Name	Description	IN/OUT
sMode	File access mode	Specify the file access mode. (When 'Reserved' is specified, this function ends normally without processing.) • 0: Access restriction clear mode • 1: Access restriction mode • Others: Reserved	IN
pcPass	Password	Specify the security password.	IN

■Description

- Specify the file access restriction status to the file access mode (sMode).
- To change the file access mode (sMode), use the security password.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 72 CCPU_GetFileSecurity

CCPU_ClearError

This function clears errors of a C Controller module.

■Format

short CCPU_ClearError (long* plErrorInfo)

■Argument

Argument	Name	Description	IN/OUT
plErrorInfo	Error information	Unused (Even if a value is specified, the operation is not affected.)	IN

■Description

- This function clears errors occurred in a C Controller module.
- When no error occurs, this function ends normally.
- When a stop error has occurred, the error cannot be cleared. (This function ends normally.)

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 71 CCPU_GetErrInfo

CCPU_Control

This function performs remote operations (remote RUN/STOP/PAUSE) for a CPU module.

■Format

short CCPU_Control (short sCpuNo, short sCode)

■Argument

Argument	Name	Description	IN/OUT
sCpuNo	CPU number	Specify the target CPU module number. Specify '0' to execute the remote operation for the host CPU. (When the CPU module specified with 1 to 4 is the host CPU, an error is returned.) • 0: Host CPU • 1 to 4: Other CPUs	IN
sCode	Remote operation specification code	Specify the remote operation to be executed. • 0: Remote RUN • 1: Remote STOP • 2: Remote PAUSE	IN

■Description

- This function executes remote operations (remote RUN/STOP/PAUSE) for the CPU module or C Controller module specified to the CPU number (sCpuNo).
- The RUN/STOP/RESET switch has a higher priority to change the operating status of a C Controller module. Therefore, if the RUN/STOP/RESET switch is put in the STOP state, the operating status will remain STOP irrespective of the specified remote operation.
  However, since the remote operation by this function is effective even when the RUN/STOP/RESET switch is put in the STOP state, a C Controller module operates according to the last specified remote operation once the RUN/STOP/RESET switch is switched from STOP to RUN.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

CCPU_DedicatedDInst

This function executes dedicated instructions categorized as 'D' or 'DP'.

■Format

short CCPU_DedicatedDInst (char* pcInstName, short sCPUNo, short* psArg1, short sArg1Size, short* psArg2, short sArg2Size, short* psArg3, short sArg3Size, short* psArg4, short sArg4Size, short* psArg5, short sArg5Size, short* psArg6, short sArg6Size, short* psArg7, short sArg7Size, short* psArg8, short sArg8Size, short* psArg9, short sArg9Size)

■Argument

Argument	Name	Description	IN/OUT
pcInstName	Instruction name	Specify the instruction name of the dedicated instruction to be executed.	IN
sCPUNo	Start input/output number of the target CPU	Start input/output number of the target CPU divided by 16 • CPU No.1 to 4: 3E0H to 3E3H	IN
psArg1	Setting data (1st)*1	Specify the first setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg1Size	Setting data size (1st)*1	Specify the size of the first setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg2	Setting data (2nd)*1	Specify the second setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg2Size	Setting data size (2nd)*1	Specify the size of the second setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg3	Setting data (3rd)*1	Specify the third setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg3Size	Setting data size (3rd)*1	Specify the size of the third setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg4	Setting data (4th)*1	Specify the fourth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg4Size	Setting data size (4th)*1	Specify the size of the fourth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg5	Setting data (5th)*1	Specify the fifth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg5Size	Setting data size (5th)*1	Specify the size of the fifth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg6	Setting data (6th)*1	Specify the sixth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg6Size	Setting data size (6th)*1	Specify the size of the sixth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg7	Setting data (7th)*1	Specify the seventh setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg7Size	Setting data size (7th)*1	Specify the size of the seventh setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg8	Setting data (8th)*1	Specify the eighth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg8Size	Setting data size (8th)*1	Specify the size of the eighth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg9	Setting data (9th)*1	Specify the ninth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg9Size	Setting data size (9th)*1	Specify the size of the ninth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN

*1 Out of the setting data for the dedicated instruction to be executed, setting the "start input/output number of the target CPU divided by 16" is not required.

■Description

The dedicated instructions that can be specified to the instruction name (pcInstName) are as shown below.

For the specifications of each dedicated function and the completion status, refer to the programming manual of each module.

Instruction name	Description	Instruction
CHGA	Requests a motion CPU to change the current value.	D.CHGA, DP.CHGA
CHGAS	Requests a motion CPU to change the current value.	D.CHGAS, DP.CHGAS
CHGV	Requests a motion CPU to change the speed value.	D.CHGV, DP.CHGV
CHGVS	Requests a motion CPU to change the speed value.	D.CHGVS, DP.CHGVS
CHGT	Requests a motion CPU to change the torque limit value.	D.CHGT, DP.CHGT
DDRD	Reads data from the device of motion CPU.	D.DDRD, DP.DDRD
GINT*1	Issues an interrupt to motion CPU or C Controller module.	D.GINT, DP.GINT
SFCS	Requests a motion CPU to start a motion SFC program.	D.SFCS, DP.SFCS
SVST	Requests a motion CPU to start a servo program.	D.SVST, DP.SVST
DDWR	Writes data to the device of motion CPU.	D.DDWR, DP.DDWR
MCNST	Requests a motion CPU to start a machine program operation.	D.MCNST, DP.MCNST

*1 The arguments to issue an interrupt from a C Controller module by using the CCPU_DedicatedDInst function are the same as those to issue an interrupt to a C Controller module by using the CCPU_DedicatedMInst function.
For details on the arguments, refer to the following section.
☞ Page 57 CCPU_DedicatedMInst

Precautions

- To execute this function, make sure to reserve an area to store the completion device of the dedicated instruction and specify it to the argument. If the area and its size are not specified to the argument, an error occurs. (The instruction is not executed properly.)
- Depending on the type of dedicated instruction, the return value of this function may be normal even if wrong argument or size is specified. Make sure to check the completion status by referring to the manual of the dedicated instruction.
- Specifying an incorrect argument may result in unexpected operation. Make sure to refer to the manual for dedicated instruction to specify the argument.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 53 CCPU_DedicatedGInst
- Page 55 CCPU_DedicatedJInst
- Page 57 CCPU_DedicatedMInst

CCPU_DedicatedGInst

This function executes dedicated instructions categorized as 'G' or 'GP'.

■Format

short CCPU_DedicatedGInst (char* pcInstName, short sIoNo, short* psArg1, short sArg1Size, short* psArg2, short sArg2Size, short* psArg3, short sArg3Size, short* psArg4, short sArg4Size, short* psArg5, short sArg5Size, short* psArg6, short sArg6Size, short* psArg7, short sArg7Size, short* psArg8, short sArg8Size, short* psArg9, short sArg9Size)

■Argument

Argument	Name	Description	IN/OUT
pcInstName	Instruction name	Specify the instruction name of the dedicated instruction to be executed.	IN
sIoNo	Start input/output number of the own station	Specify the start input/output number of the own station divided by 16. (00H to FEH)	IN
psArg1	Setting data (1st)*1	Specify the first setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg1Size	Setting data size (1st)*1	Specify the size of the first setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg2	Setting data (2nd)*1	Specify the second setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg2Size	Setting data size (2nd)*1	Specify the size of the second setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg3	Setting data (3rd)*1	Specify the third setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg3Size	Setting data size (3rd)*1	Specify the size of the third setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg4	Setting data (4th)*1	Specify the fourth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg4Size	Setting data size (4th)*1	Specify the size of the fourth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg5	Setting data (5th)*1	Specify the fifth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg5Size	Setting data size (5th)*1	Specify the size of the fifth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg6	Setting data (6th)*1	Specify the sixth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg6Size	Setting data size (6th)*1	Specify the size of the sixth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg7	Setting data (7th)*1	Specify the seventh setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg7Size	Setting data size (7th)*1	Specify the size of the seventh setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg8	Setting data (8th)*1	Specify the eighth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg8Size	Setting data size (8th)*1	Specify the size of the eighth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg9	Setting data (9th)*1	Specify the ninth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg9Size	Setting data size (9th)*1	Specify the size of the ninth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN

*1 Out of the setting data for the dedicated instruction to be executed, setting the "start input/output number of the own station" is not required.

■Description

The dedicated instructions that can be specified to the instruction name (pcInstName) are as shown below.
For the specifications of each dedicated function and the completion status, refer to the programming manual of each module.

Instruction name	Description	Instruction
SEND	Sends data to other station programmable controller.	GP.SEND
RECV	Receives data from other station programmable controller.	GP.RECV
CCPASET	Sets parameters of mater station, submaster station, and local station of CC-Link IE Field Network.	G.CCPASET, GP.CCPASET

Precautions

- To execute this function, make sure to reserve an area to store the completion device of the dedicated instruction and specify it to the argument. If the area and its size are not specified to the argument, an error occurs. (The instruction is not executed properly.)
- Depending on the type of dedicated instruction, the return value of this function may be normal even if wrong argument or size is specified. Make sure to check the completion status by referring to the manual of the dedicated instruction.
- Specifying an incorrect argument may result in unexpected operation. Make sure to refer to the manual for dedicated instruction to specify the argument.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 51 CCPU_DedicatedDInst
- Page 55 CCPU_DedicatedJInst
- Page 57 CCPU_DedicatedMInst

CCPU_DedicatedJInst

This function executes dedicated instructions categorized as 'J' or 'JP'.

■Format

short CCPU_DedicatedJInst (char* pcInstName, short sNetNo, short* psArg1, short sArg1Size, short* psArg2, short sArg2Size, short* psArg3, short sArg3Size, short* psArg4, short sArg4Size, short* psArg5, short sArg5Size, short* psArg6, short sArg6Size, short* psArg7, short sArg7Size, short* psArg8, short sArg8Size, short* psArg9, short sArg9Size)

■Argument

Argument	Name	Description	IN/OUT
pcInstName	Instruction name	Specify the instruction name of the dedicated instruction to be executed.	IN
sNetNo	Network number of the own station	Specify the network number of own station. (1 to 239)	IN
psArg1	Setting data (1st)*1	Specify the first setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg1Size	Setting data size (1st)*1	Specify the size of the first setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg2	Setting data (2nd)*1	Specify the second setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg2Size	Setting data size (2nd)*1	Specify the size of the second setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg3	Setting data (3rd)*1	Specify the third setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg3Size	Setting data size (3rd)*1	Specify the size of the third setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg4	Setting data (4th)*1	Specify the fourth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg4Size	Setting data size (4th)*1	Specify the size of the fourth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg5	Setting data (5th)*1	Specify the fifth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg5Size	Setting data size (5th)*1	Specify the size of the fifth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg6	Setting data (6th)*1	Specify the sixth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg6Size	Setting data size (6th)*1	Specify the size of the sixth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg7	Setting data (7th)*1	Specify the seventh setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg7Size	Setting data size (7th)*1	Specify the size of the seventh setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg8	Setting data (8th)*1	Specify the eighth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg8Size	Setting data size (8th)*1	Specify the size of the eighth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg9	Setting data (9th)*1	Specify the ninth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg9Size	Setting data size (9th)*1	Specify the size of the ninth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN

*1 Out of the setting data for the dedicated instruction to be executed, setting the "network number of the own station" is not required.

■Description

The dedicated instructions that can be specified to the instruction name (pcInstName) are as shown below.

For the specifications of each dedicated function and the completion status, refer to the programming manual of each module.

Instruction name	Description	Instruction
SEND	Sends data to other station programmable controller.	JP.SEND
RECV	Receives data from other station programmable controller.	JP.RECV
REMTO[*1]	Writes data to the buffer memory of intelligent device station/remote device station.	JP.REMTO
REMFR[*1]	Reads data from the buffer memory of intelligent device station/remote device station.	JP.REMFR

*1 The error code is stored in SW0080 to SW009F instead of the completion status.
Obtain the error codes stored in SW0080 to SW009F by using the CCPU_ReadLinkDevice function.

Precautions

- To execute this function, make sure to reserve an area to store the completion device of the dedicated instruction and specify it to the argument. If the area and its size are not specified to the argument, an error occurs. (The instruction is not executed properly.)
- Depending on the type of dedicated instruction, the return value of this function may be normal even if wrong argument or size is specified. Make sure to check the completion status by referring to the manual of the dedicated instruction.
- Specifying an incorrect argument may result in unexpected operation. Make sure to refer to the manual for dedicated instruction to specify the argument.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 51 CCPU_DedicatedDInst
- Page 53 CCPU_DedicatedGInst
- Page 57 CCPU_DedicatedMInst

CCPU_DedicatedMInst

This function executes dedicated instructions categorized as 'M' or 'MP'.

■Format

short CCPU_DedicatedMInst (char* pcInstName, short sCPUNo, short* psArg1, short sArg1Size, short* psArg2, short sArg2Size, short* psArg3, short sArg3Size, short* psArg4, short sArg4Size, short* psArg5, short sArg5Size, short* psArg6, short sArg6Size, short* psArg7, short sArg7Size, short* psArg8, short sArg8Size, short* psArg9, short sArg9Size)

■Argument

Argument	Name	Description	IN/OUT
pcInstName	Instruction name	Specify the instruction name of the dedicated instruction to be executed.	IN
sCPUNo	Start input/output number of the target CPU	Start input/output number of the target CPU divided by 16 (CPU No.1: 3E0H, CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H)	IN
psArg1	Setting data (1st)*1	Specify the first setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg1Size	Setting data size (1st)*1	Specify the size of the first setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg2	Setting data (2nd)*1	Specify the second setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg2Size	Setting data size (2nd)*1	Specify the size of the second setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg3	Setting data (3rd)*1	Specify the third setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg3Size	Setting data size (3rd)*1	Specify the size of the third setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg4	Setting data (4th)*1	Specify the fourth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg4Size	Setting data size (4th)*1	Specify the size of the fourth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg5	Setting data (5th)*1	Specify the fifth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg5Size	Setting data size (5th)*1	Specify the size of the fifth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg6	Setting data (6th)*1	Specify the sixth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg6Size	Setting data size (6th)*1	Specify the size of the sixth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg7	Setting data (7th)*1	Specify the seventh setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg7Size	Setting data size (7th)*1	Specify the size of the seventh setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg8	Setting data (8th)*1	Specify the eighth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg8Size	Setting data size (8th)*1	Specify the size of the eighth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN
psArg9	Setting data (9th)*1	Specify the ninth setting data for the dedicated instruction to be executed. Specify 'NULL' if there is no setting data.	IN/OUT
sArg9Size	Setting data size (9th)*1	Specify the size of the ninth setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN

*1 Out of the setting data for the dedicated instruction to be executed, setting the "start input/output number of the target CPU divided by 16" is not required.

■Description

The dedicated instructions that can be specified to the instruction name (pcInstName) are as shown below.

For the specifications of each dedicated function and the completion status, refer to the programming manual of each module.

Instruction name	Description	Instruction
GINT	Issues an interrupt to motion CPU and C Controller module.	M.GINT, MP.GINT
MCNST	Requests a motion CPU to start a machine program operation.	M.MCNST, MP.MCNST

To issue an interrupt from a C Controller module, specify the following values to each argument.

Argument	Description
pcInstName	Specify "GINT".
sCPUNo	Specify the target CPU module. • CPU No.1: 3E0H • CPU No.2: 3E1H • CPU No.3: 3E2H • CPU No.4: 3E3H
psArg1	Specify the interrupt pointer number to psArg1[0]. (0 to 15)
sArg1Size	Specify '1'.
psArg2	Specify the area to store the completion device (2 words). • Completion of the instruction processing: psArg2[1]=0, psArg2[0]=1 • Abnormal completion of the instruction processing: psArg2[1]=1, psArg2[0]=1
sArg2Size	Specify '2'.
psArg3	Specify the area to store the completion status (1 word). When the interrupt pointer number specified to psArg1[0] is other than 0 to 15, the psArg3[0] will be '2282H'.
sArg3Size	Specify '1'.
psArg4 to psArg9	Specify NULL.
sArg4Size to sArg9Size	Specify '0'.

Point

The completion status and completion device can be omitted at the same time. (Omitting only one of them is not available.)

To omit them, specify "0" to sArg2Size and sArg3Size, and "NULL" to psArg2 and psArg3.

Precautions

- To execute this function, make sure to reserve an area to store the completion device of the dedicated instruction and specify it to the argument. If the area and its size are not specified to the argument, an error occurs. (The instruction is not executed properly.)
- Depending on the type of dedicated instruction, the return value of this function may be normal even if wrong argument or size is specified. Make sure to check the completion status by referring to the manual of the dedicated instruction.
- Specifying an incorrect argument may result in unexpected operation. Make sure to refer to the manual for dedicated instruction to specify the argument.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 51 CCPU_DedicatedDInst
- Page 53 CCPU_DedicatedGInst
- Page 55 CCPU_DedicatedJInst

CCPU_DisableInt

This function disables the routine registered by using the CCPU_EntryInt function.

■Format

short CCPU_DisableInt (short sSINo)

■Argument

Argument	Name	Description	IN/OUT
sSINo	Interrupt pointer number	Specify the interrupt pointer number.	IN

■Description

- This function disables the routine registered by using the CCPU_EntryInt function. (The routine is not executed when an interrupt occurs.)
- Specify the interrupt pointer number (sSINo) specified in the CCPU_EntryInt function to Interrupt pointer number (sSINo).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 61 CCPU_EntryInt
- Page 60 CCPU_EnableInt

CCPU_EnableInt

This function enables the routine registered by using the CCPU_EntryInt function.

■Format

short CCPU_EnableInt (short sSINo)

■Argument

Argument	Name	Description	IN/OUT
sSINo	Interrupt pointer number	Specify the interrupt pointer number.	IN

■Description

- This function enables the routine registered by using the CCPU_EntryInt function. (The routine is executed when an interrupt occurs.)
- Specify the interrupt pointer number (sSINo) specified in the CCPU_EntryInt function to Interrupt pointer number (sSINo).
- Since an interrupt does not occur while a stop error is occurring in a C Controller module, the routine registered by using the CCPU_EntryInt function will not be executed even if it is enabled.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 61 CCPU_EntryInt
- Page 59 CCPU_DisableInt

CCPU_EntryInt

This function registers a routine to be called when an interrupt occurs.

■Format

short CCPU_EntryInt (short sSINo, CCPU_FUNCPTR pFuncPtr)

■Argument

Argument	Name	Description	IN/OUT
sSINo	Interrupt pointer number	Specify the interrupt pointer number.	IN
pFuncPtr	Registered routine	Specify the routine to be registered. (The routine is deregistered by specifying NULL.)	IN

- The data type of the registered routine (pFuncPtr) is defined as void type in the header file, "CCPUFunc.h".

The specification method of the interrupt pointer number (sSINo) is as follows:

sSINo	Description
0 to 15	Interrupt by module
44	Inter-module synchronization interrupt
45	Multiple CPU synchronization interrupt
50 to 1023	Interrupt by module

■Description

- This function registers the routine specified to the registered routine (pFuncPtr) in the interrupt specified with interrupt pointer number (sSINo).
- When NULL is specified to the registered routine (pFuncPtr), the routine is deregistered.
- Enable the routine registered with the CCPU_EntryInt function by using the CCPU_EnableInt function.
  If the registered routine is disabled, the routine will not be called.

Precautions

- The registered routine is not executed while the operating system is in the interrupt-disabled state.
- For processing a routine to be registered in the registered routine (pFuncPtr), note the following:
  A routine to be registered must not have an argument. (Do not pass an argument from an interrupt.)
  When registering a routine, observe the considerations on the interrupt service routine (ISR).
  Register minimal processing of a routine so that the processing time is as short as possible.
  Only the C Controller module dedicated function for ISR can be used for a routine to be registered. Do not use any other function. (An error of a function is not checked.)
- When the CCPU_EntryInt function is executed multiple times by specifying the same interrupt pointer number (sSINo), the routine specified to the registered routine (pFuncPtr) last is registered. (Multiple routines cannot be registered.)
- The routine is disabled after the registration is done by this function.
- Calling an interrupt from other CPUs, a multiple CPU synchronization interrupt, and a WDT error interrupt is delayed while the routine registered by using the CCPU_EntryInt function is in operation.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 60 CCPU_EnableInt
- Page 59 CCPU_DisableInt

CCPU_EntryTimerEvent

This function registers a timer event.

■Format

short CCPU_EntryTimerEvent (long* plEvent)

■Argument

Argument	Name	Description	IN/OUT
plEvent	Registered event	Specify a timer event to be registered.	IN

The specification method of the registered event (plEvent) is as follows:

plEvent	Description	
plEvent[0]	Number of timer event settings (1 to 16)	
plEvent[1]	First timer event number (1 to 16)	First timer event setting
plEvent[2]	Cycle of the first timer event (0: Clear, 1 to 60,000: Cycle [ms])	
plEvent[3]	Synchronization type of the first timer event (0: Batch synchronization, 1: Individual synchronization)	
plEvent[4]	Second timer event number (1 to 16)	Second timer event setting
plEvent[5]	Cycle of the second timer event (0: Clear, 1 to 60,000: Cycle [ms])	
plEvent[6]	Synchronization type of the second timer event (0: Batch synchronization, 1: Individual synchronization)	
plEvent[7]	Third timer event number (1 to 16)	Third timer event setting
plEvent[8]	Cycle of the third timer event (0: Clear, 1 to 60,000: Cycle [ms])	
plEvent[9]	Synchronization type of the third timer event (0: Batch synchronization, 1: Individual synchronization)	
⋮	⋮	⋮

When setting the timer event cycle, only the following specification method is applicable.

- For 1 to 1000: Specify multiples of 5 (5 ms units)
- For 1000 to 60,000: Specify multiples of 1000 (1 s units)

■Description

- This function sets the cycle and synchronization type for the timer event registration.
- When '0' is specified to the cycle of the registered event (plEvent), the timer event is deregistered (the occurrence is cleared). Deregistration will clear the events that have occurred before that.
- Up to 16 timer events can be set. The cycle (1 to 60,000 [ms]) and synchronization type (batch synchronization or individual synchronization) can be specified for each timer event. For the synchronization type, refer to the description of the CCPU_WaitTimerEvent function.
- Specify the timer event number without duplication. Otherwise, an error will be returned.
- To change the cycle of a timer event number that the cycle is already set, clear it (specify '0' to the cycle), and then register the cycle (set the cycle) again. Otherwise, an error will be returned.
- The registered timer event can be placed into the wait state by using the CCPU_WaitTimerEvent function.
- All the timer events are cleared at the initial status.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 108 CCPU_WaitTimerEvent

CCPU_EntryWDTInt

This function registers a routine to be called when a user WDT error interrupt occurs.

■Format

short CCPU_EntryWDTInt (short sType, CCPU_FUNCPTR pFuncPtr)

■Argument

Argument	Name	Description	IN/OUT
sType	WDT type	Specify the WDT type. (If reserve is specified, an error is returned.) • 0: User WDT • Others: Reserved	IN
pFuncPtr	Registered routine	Specify the routine to be registered. (The routine is deregistered by specifying NULL.)	IN

- The data type of the registered routine (pFuncPtr) is defined as void type in the header file, "CCPUFunc.h".

■Description

- This function registers a routine to call when a user WDT error interrupt of C Controller module occurs.
- Specify the routine to be registered to the registered routine (pFuncPtr).
- When this function is executed several times, the last registered routine will be in effect.
- The routine registered with this function is executed as an interrupt service routine (ISR) when a user WDT error occurs. (If the CCPU_ResetWDT function is not executed within the time interval specified in the CCPU_StartWDT function, the WDT error interrupt will occur.)

Precautions

- The registered routine is not executed while the operating system is in the interrupt-disabled state.
- For processing a routine to be registered in the registered routine (pFuncPtr), note the following:
  A routine to be registered must not have an argument. (Do not pass an argument from an interrupt.)
  When registering a routine, observe the considerations on the interrupt service routine (ISR).
  Register minimal processing of a routine so that the processing time is as short as possible.
  Only the C Controller module dedicated function for ISR can be used for a routine to be registered. Do not use any other function. (An error of a function is not checked.)

■⚠WARNING

When a routine that does not observe the considerations on interrupt service routine (ISR) is registered, the operating system may be runaway.
Make sure to use the routine after carefully verifying the operation and performance.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 98 CCPU_StartWDT
- Page 90 CCPU_ResetWDT
- Page 99 CCPU_StopWDT

CCPU_FromBuf

This function reads data from the CPU buffer memory of the CPU module and the buffer memory of the intelligent function module which are mounted on the specified module position. (FROM instruction)

■Format

short CCPU_FromBuf (unsigned short usIoNo, unsigned long ulOffset, unsigned long ulSize, unsigned short* pusDataBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
usIoNo	Module position	Specify the module position. Start I/O number divided by 16 (0H to FFH, 3E0H to 3E3H)	IN
ulOffset	Offset	Specify the offset in word units.	IN
ulSize	Data size	Specify the read data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of read data.	OUT
ulBufSize	Data storage destination size	Specify the data storage destination size in word units.	IN

■Description

- This function reads data equivalent to the size specified to the data size (ulSize) from the CPU buffer memory of a CPU module and the buffer memory of an intelligent function module which are specified to the module position (usIoNo), and stores the read data in the data storage destination (pusDataBuf).
  Data is read by specifying an offset address (ulOffset) from the start of the CPU buffer memory of a CPU module and the buffer memory of an intelligent function module.
- To access the CPU buffer memory of the module in a multiple CPU system (CPU No.1 to No.4), specify 3E0H to 3E3H (CPU No.1 to No.4) to the module position (usIoNo). However, the CPU buffer memory can be accessed only when the multiple CPU setting is configured.

Precautions

Note that the size of data storage destination (ulBufSize) should be equal to or bigger than the data size (ulSize).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 102 CCPU_ToBuf

CCPU_FromBufHG

This function reads data from the fixed cycle communication area of the CPU module mounted on the specified module position.

■Format

short CCPU_FromBufHG(unsigned short usIoNo, unsigned long short ulOffset, unsigned long ulSize,
unsigned short* pusDataBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
usIoNo	Module position	Specify the module position. Start I/O number divided by 16 (3E0H to 3E3H)	IN
ulOffset	Offset	Specify the offset in word units.	IN
ulSize	Data size	Specify the read data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of read data.	OUT
ulBufSize	Data storage destination size	Specify the data storage destination size in word units.	IN

■Description

- This function reads data equivalent to the size specified to the data size (ulSize) from the fixed cycle communication area of a CPU module specified to the module position (usIoNo), and stores the read data in the data storage destination (pusDataBuf).
  Data is read by specifying an offset address (ulOffset) from the start of the fixed cycle communication area.
- The fixed cycle communication area can be accessed only when the fixed cycle communication area setting under the multiple CPU setting is configured.

Precautions

Note that the size of data storage destination (ulBufSize) should be equal to or bigger than the data size (ulSize).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 64 CCPU_FromBuf
- Page 102 CCPU_ToBuf
- Page 103 CCPU_ToBufHG

CCPU_GetConstantProcessStatus

This function obtains the fixed cycle processing status of C Controller module.

■Format

short CCPU_GetConstantProcessStatus(unsigned short * pusStatusBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
pusStatusBuf	Fixed cycle processing status storage destination	Specify the fixed cycle processing status storage destination.	OUT
ulBufSize	Fixed cycle processing status storage destination size	Specify the fixed cycle processing status storage destination size in word units. (When '0' is specified, this function ends normally without processing.)	IN

■Description

- This function obtains the fixed cycle processing status of C Controller module and stores it in the fixed cycle processing status storage destination (pusStatusBuf).
- It obtains the information for the size specified to the fixed cycle processing status storage destination size (ulBufSize).
- The information to be stored in the fixed cycle processing status storage destination (pusStatusBuf) is as follows.

pusStatusBuf	Description
pusStatusBuf[0]	Fixed cycle processing cycle [ms] (setting value)*1
pusStatusBuf[1]	Current fixed cycle processing time [ms]
pusStatusBuf[2]	Current fixed cycle processing time [μs]
pusStatusBuf[3]	Minimum fixed cycle processing time [ms]
pusStatusBuf[4]	Minimum fixed cycle processing time [μs]
pusStatusBuf[5]	Maximum fixed cycle processing time [ms]
pusStatusBuf[6]	Maximum fixed cycle processing time [μs]

*1 The fixed cycle processing includes the refresh processing with Network modules, the reset processing of watchdog timer, and the self-diagnostic processing. For more details on the functions, refer to the following manual.
MELSEC iQ-R C Controller Module User's Manual (Application)

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. Page 168 ERROR CODE LIST

■Relevant functions

- Page 71 CCPU_GetErrInfo

CCPU_GetCounterMicros

This function obtains a 1 μs counter value of the C Controller module.

■Format

short CCPU_GetCounterMicros(unsigned long* pulMicros)

■Argument

Argument	Name	Description	IN/OUT
pulMicros	1μs counter value storage destination	Specify the storage destination of the 1μs counter value.	OUT

■Description

- This function obtains a 1μs counter value of C Controller module and stores the value in the 1μs counter value storage destination (pulMicros).
- The 1 μs counter value increases by 1 every 1 μs after the power is turned ON.
- The count cycles between 0 and 4294967295.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 68 CCPU_GetCounterMillis

CCPU_GetCounterMillis

This function obtains a 1 ms counter value of C Controller module.

■Format

short CCPU_GetCounterMillis(unsigned long* pulMillis)

■Argument

Argument	Name	Description	IN/OUT
pulMillis	1 ms counter value storage destination	Specify the storage destination of the 1 ms counter value.	OUT

■Description

- This function obtains a 1 ms counter value of C Controller module and stores the value in the 1 ms counter value storage destination (pulMillis).
- The 1 ms counter value increases by 1 every 1 ms after the power is turned ON.
- The count cycles between 0 and 4294967295.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 67 CCPU_GetCounterMicros

CCPU_GetCpuStatus

This function obtains the operating status of C Controller module.

■Format

short CCPU_GetCpuStatus(long* plStatusBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
plStatusBuf	Operating status storage destination	Specify the storage destination of the operating status.	OUT
ulBufSize	Operating status storage destination size	Specify the size of area reserved in the operating status storage destination in double word units. (When '0' is specified, this function ends normally without processing.)	IN

■Description

- This function obtains the operating status of a C Controller module, and stores it to the operating status storage destination (plStatusBuf).
- It obtains the information for the size specified to the operating status storage destination size (ulBufSize).
- The information to be stored in the operating status storage destination (plStatusBuf) is as follows.
  (If information to be stored is not supported, '0' is set as its status.)

plStatusBuf	Description		
	Storage position		Status
plStatusBuf[0]	bit31 to 8	Reserved	—
	bit7 to 4	Cause of STOP/PAUSE	• 0: RUN/STOP/RESET switch • 1: Reserved • 2: Reserved • 3: Execution of the CCPU_Control function from a user program • 4: Error • 5. Remote operations • Others: Reserved
	bit3 to 0	CPU operating status	• 0: RUN state • 1: Reserved • 2: STOP state • 3: PAUSE state • Others: Reserved
plStatusBuf[1]	bit31 to 16	Reserved	—
	bit15 to 7	Reserved	—
	bit6	USB Mass Storage Class-compliant device status	• 0: Inserted (mounted) • 1: Inserted (unmounted) • 2: Not inserted
	bit5		
	bit4	SD memory card status	• 0: Inserted (mounted) • 1: Inserted (unmounted) • 2: Not inserted
	bit3		
	bit2	Reserved	—
	bit1	Program memory shutdown status	• 0: Shutdown not performed • 1: Shutdown completed
	bit0	Data memory shutdown status	• 0: Shutdown not performed • 1: Shutdown completed
plStatusBuf[2]	bit31 to 0	Index value for number of data memory write cycle	—

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 71 CCPU_GetErrInfo

CCPU_GetDotMatrixLED

This function obtains the value displayed on the dot matrix LED of a C Controller module, and stores it to the LED data storage destination (pcData).

■Format

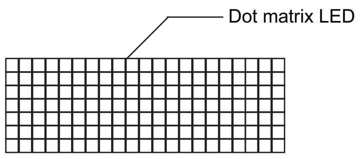
short CCPU_GetDotMatrixLED(char* pcData, unsigned long ulDataSize)

■Argument

Argument	Name	Description	IN/OUT
pcData	LED data storage destination	Specify the storage destination of LED data.	OUT
ulDataSize	LED data storage destination size	Specify the LED data storage destination size in byte units. (When '0' is specified, this function ends normally without processing.)	IN

■Description

- This function obtains the value displayed on the dot matrix LED, and stores it in the LED data storage destination (pcData).
- It obtains the information for the size specified to the LED data storage destination size (ulDataSize).
- The value displayed on the dot matrix LED is stored in the LED data storage destination (pcData) as shown below.



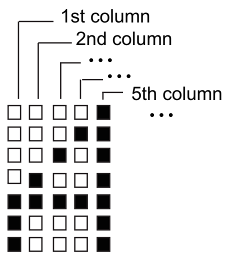
pcData[0] to pcData[19]: Data of the dot matrix LED (7×20)

The value displayed in the following format is obtained.

Data format for each column: Bit pattern in which '0' is for the upper one bit, and '1' (when LED is ON) or '0' (when LED is OFF) is for lower seven bits

Ex.

The bit pattern shown below is displayed on the dot matrix LED:



1st column: 0000 0111b = 07H → pcData[0] = 0x07

2nd column: 0000 1100b = 0cH → pcData[1] = 0x0c

3rd column: 0001 0100b = 14H → pcData[2] = 0x14

4th column: 0010 0100b = 24H → pcData[3] = 0x24

5th column: 0111 1111b = 7fH → pcData[4] = 0x7f

6th column to 20th column: 0000 0000b = 00H → pcData[5] to pcData[19] = 0x00

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 92 CCPU_SetDotMatrixLED

CCPU_GetErrInfo

This function obtains the error information of C Controller module.

■Format

short CCPU_GetErrInfo(unsigned short* pusErrorInfo, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
pusErrorInfo	Error information storage destination	Specify the error information storage destination .	OUT
ulBufSize	Error information storage destination size	Specify the error information storage destination size in word units. (When '0' is specified, this function ends normally without processing.)	IN

■Description

- This function obtains the error information of C Controller module and stores it in the error information storage destination (pusErrorInfo).
- It obtains the information for the size specified to the error information storage destination size (ulBufSize).
- The information to be stored in the error information storage destination (pusErrorInfo) is as follows.

pusErrorInfo	Description
pusErrorInfo[0]	Self-diagnostics error code 1
pusErrorInfo[1]	Self-diagnostics error code 2
pusErrorInfo[2]	Self-diagnostics error code 3
pusErrorInfo[3]	Self-diagnostics error code 4
pusErrorInfo[4]	Self-diagnostics error code 5
pusErrorInfo[5]	Self-diagnostics error code 6
pusErrorInfo[6]	Self-diagnostics error code 7
pusErrorInfo[7]	Self-diagnostics error code 8
pusErrorInfo[8]	Self-diagnostics error code 9
pusErrorInfo[9]	Self-diagnostics error code 10
pusErrorInfo[10]	Self-diagnostics error code 11
pusErrorInfo[11]	Self-diagnostics error code 12
pusErrorInfo[12]	Self-diagnostics error code 13
pusErrorInfo[13]	Self-diagnostics error code 14
pusErrorInfo[14]	Self-diagnostics error code 15
pusErrorInfo[15]	Self-diagnostics error code 16

Point

Up to 16 error codes for errors occurred in the self-diagnostics are stored in order from pusErrorInfo[0]. The error code which has already been stored is not stored.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 49 CCPU_ClearError

CCPU_GetFileSecurity

This function obtains the file access mode of a C Controller module.

■Format

short CCPU_GetFileSecurity(short* psMode);

■Argument

Argument	Name	Description	IN/OUT
psMode	File access mode	Stores the file access mode. • 0: Access restriction clear mode • 1: Access restriction mode	OUT

■Description

This function obtains the current file access mode and stores it to the file access mode (psMode).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

• Page 48 CCPU_ChangeFileSecurity

CCPU_GetIDInfo

This function obtains the individual identification information of C Controller module.

■Format

short CCPU_GetIDInfo(unsigned char *pucGetData, unsigned long ulBufSize);

■Argument

Argument	Name	Description	IN/OUT
pucGetData	Individual identification information storage destination	Specify the individual identification information storage destination.	OUT
ulBufSize	Individual identification information storage destination size	Specify the individual identification information storage destination size in word units.	IN

■Description

- This function obtains the individual identification information of C Controller module, and stores it in the individual identification information storage destination (pucGetData).
- It obtains the information for the size specified to the individual identification information storage destination size (ulBufSize).
- The information to be stored in the individual identification information storage destination (pucGetData) is as follows.

pucGetData	Description
pucGetData[0]	Individual identification information for CH1
pucGetData[1]	
pucGetData[2]	
pucGetData[3]	
pucGetData[4]	
pucGetData[5]	
pucGetData[6]	Individual identification information for CH2
pucGetData[7]	
pucGetData[8]	
pucGetData[9]	
pucGetData[10]	
pucGetData[11]	

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 79 CCPU_GetSerialNo

CCPU_GetLEDStatus

This function obtains the LED status of a C Controller module.

■Format

short CCPU_GetLEDStatus(long lLed, unsigned short* pusLedInfo, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
lLed	Target LED	Specify the target LED. (When 'Reserved' is specified, this function ends normally without processing.) • 0: READY LED • 1: ERROR LED • 2: BUS RUN LED • 3: CARD RDY LED • 4: USER LED • 5: USB RDY LED • 6: RS SD/RD LED • -1: All of the LEDs above • Others: Reserved	IN
pusLedInfo	LED status storage destination	Specify the storage destination of the LED status.	OUT
ulBufSize	LED status storage destination size	Specify the LED status storage destination size in word units. (When '0' is specified, this function ends normally without processing.)	IN

■Description

- This function obtains the LED status of the C Controller module specified to the target LED (lLed), and stores it in the LED status storage destination (pusLedInfo).
- It obtains the information for the size specified in the LED status storage destination (ulBufSize).
- If the LED is not supported, '0' is set to the LED status.
- The LED status to be stored in the LED status storage destination (pusLedInfo) is as follows.

pusLedInfo	Description
0	OFF
1	ON (Red)
2	Flashing at low speed (Red)
3	Flashing at high speed (Red)
4	ON (Green)
5	Flashing at low speed (Green)
6	Flashing at high speed (Green)

- When '-1' is specified to the target LED (lLed), the LED status stored in the LED status storage destination (pusLedInfo) is as follows:
  (When '0' to '6' is specified, the specified LED status is stored in pusLedInfo[0].)

pusLedInfo	Description
pusLedInfo[0]	READY LED status
pusLedInfo[1]	ERROR LED status
pusLedInfo[2]	BUS RUN LED status
pusLedInfo[3]	CARD RDY LED status
pusLedInfo[4]	USER LED status
pusLedInfo[5]	USB RDY LED status
pusLedInfo[6]	RS SD/RD LED status

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 71 CCPU_GetErrInfo

CCPU_GetOpSelectMode

This function obtains the operation selection mode of a C Controller module.

■Format

short CCPU_GetOpSelectMode(long lModeInfo, long* plSelectMode)

■Argument

Argument	Name	Description	IN/OUT
lModeInfo	Mode information	Specify the mode information. • 1: Operation selection mode when the MODE/SELECT switch is held in the SELECT position • 2: Display mode of a dot matrix LED • Others: Reserved	IN
plSelectMode	Operation selection mode	Specify the storage destination of the obtained operation mode.	OUT

■Description

- This function stores the operation selection mode of a C Controller module in the operation selection mode (plSelectMode).
- The information which is stored to the operation selection mode (plSelectMode) when 1 is specified to the mode information (lModeInfo) is as follows:

plSelectMode	Description
1	Notifies event to the user program.
2	Unmounts SD memory card forcibly.
3	Unmounts USB Mass Storage Class-compliant devices forcibly.
4	Unmounts SD memory card/USB Mass Storage Class-compliant devices forcibly.

- The information which is stored to the operation selection mode (plSelectMode) when 2 is specified to the mode information (lModeInfo) is as follows:

plSelectMode	Description
1	Displays the specified content on the dot matrix LED.
2	Displays an error code on the dot matrix LED.
3	Displays the IP address of CH1 on the dot matrix LED.
4	Displays the IP address of CH2 on the dot matrix LED.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 95 CCPU_SetOpSelectMode

CCPU_GetPowerStatus

This function obtains the power status of C Controller module.

■Format

short CCPU_GetPowerStatus(long* plStatusBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
plStatusBuf	Power status storage destination	Specify the storage destination of the power status.	OUT
ulBufSize	Power status storage destination size	Specify the power status storage destination size in double word units. (When '0' is specified, this function ends normally without processing.)	IN

■Description

- This function obtains the power status of C Controller module and stores it in the power status storage destination (plStatusBuf).
- It obtains the information for the size specified to the power status storage destination size (ulBufSize).
- The information to be stored in the power status storage destination (plStatusBuf) is as follows.

plStatusBuf	Description		
	Storage position		Status
plStatusBuf[0]	bit31 to 16	Reserved	—
	bit15 to 0	Number of detected momentary power failures	—

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 71 CCPU_GetErrInfo

CCPU_GetRTC

This function obtains the clock data (local time) of C Controller module.

■Format

short CCPU_GetRTC(short* psGetData,unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
psGetData	Clock data storage destination	Specify the storage destination of the clock data (local time).	OUT
ulBufSize	Clock data storage destination size	Specify the clock data (local time) storage destination size in word units. (When '0' is specified, this function ends normally without processing.)	IN

■Description

- This function obtains the clock data (local time) of C Controller module and stores it in the clock data storage destination (psGetData).
- It obtains the information for the size specified in the clock data storage destination size (ulBufSize).
- The clock data (local time) are stored in the clock data storage destination (psGetData) as follows.

(Available range: January 1, 1980 to December 31, 2079)

psGetData	Description
psGetData[0]	Year data (1980 to 2079)
psGetData[1]	Month data (1 to 12)
psGetData[2]	Day data (1 to 31)
psGetData[3]	Hour data (0 to 23)
psGetData[4]	Minute data (0 to 59)
psGetData[5]	Second data (0 to 59)
psGetData[6]	Day data (0 to 6) ('0: Sunday, 1: Monday, 2: Tuesday, 3: Wednesday, 4: Thursday, 5: Friday, 6: Saturday)
psGetData[7][*1]	Time zone (Unit: minute)
psGetData[8][*1]	Daylight saving time status flag (0 to 1) (0: Not during daylight saving time, 1: During daylight saving time)

*1 Information can be acquired when using a C Controller module with the firmware version '06' or later. For the firmware version '05' or earlier, the area is not overwritten even when information is acquired.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 96 CCPU_SetRTC

CCPU_GetSerialNo

This function obtains the serial number of C Controller module.

■Format

short CCPU_GetSerialNo(char* pcGetData, unsigned long ulDataSize)

■Argument

Argument	Name	Description	IN/OUT
pcGetData	Serial number storage destination	Specify the serial number storage destination.	OUT
ulDataSize	Serial number storage destination size	Specify the serial number storage destination in byte units. (When '0' is specified, this function ends normally without processing.)	IN

■Description

- This function obtains the serial number (16-digits) of C Controller module and store it in the serial number storage destination (pcGetData).
- It obtains the information for the size specified to the serial number storage destination size (ulDataSize).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 73 CCPU_GetIDInfo

CCPU_GetSwitchStatus

This function obtains the switch status of C Controller module.

■Format

short CCPU_GetSwitchStatus(long* plStatusBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
plStatusBuf	Switch status storage destination	Specify the switch status storage destination.	OUT
ulBufSize	Switch status storage destination size	Specify the switch status storage destination size in double word units. (When '0' is specified, this function ends normally without processing.)	IN

■Description

- This function obtains the switch status of C Controller module and stores it in the switch status storage destination (plStatusBuf).
- It obtains the information for the size specified to the switch status storage destination size (ulBufSize).
- The information to be stored in the switch status storage destination (plStatusBuf) is as follows.

plStatusBuf	Description		
	Storage position		Status
plStatusBuf[0]	bit31-6	Reserved	—
	bit5-3	MODE/SELECT switch status	• 000: MODE state • 010: NEUTRAL state • 100: SELECT state • Others: Reserved
	bit2-0	RUN/STOP/RESET switch status	• 000: RESET state • 010: STOP state • 100: RUN state • Others: Reserved

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

CCPU_GetUnitInfo

This function obtains the module configuration information.

■Format

short CCPU_GetUnitInfo (unsigned short* pusUnitInfo1, unsigned short* pusUnitInfo2, unsigned short* pusUnitInfo3)

■Argument

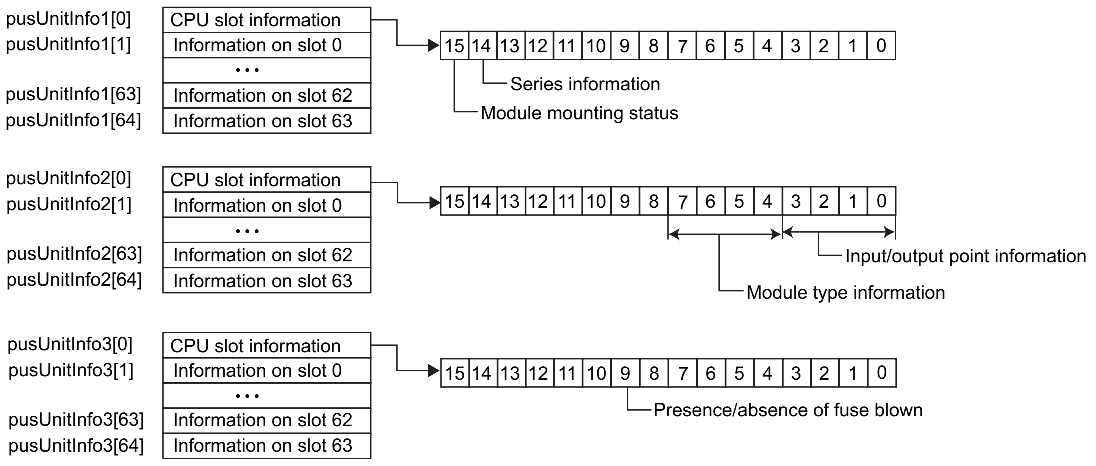
Argument	Name	Description	IN/OUT
pusUnitInfo1	Module configuration information 1	Specify the storage destination of module configuration information 1.	OUT
pusUnitInfo2	Module configuration information 2	Specify the storage destination of module configuration information 2.	OUT
pusUnitInfo3	Module configuration information 3	Specify the storage destination of module configuration information 3.	OUT

■Description

This function reads module configuration information (65 slots) and stores to the module configuration information 1 (pusUnitInfo1), pumodule configuration information 2 (sUnitInfo2), and module configuration information 3 (pusUnitInfo3).
Module configuration information to be stored differs depending on the series information.

■Series information is MELSEC iQ-R series

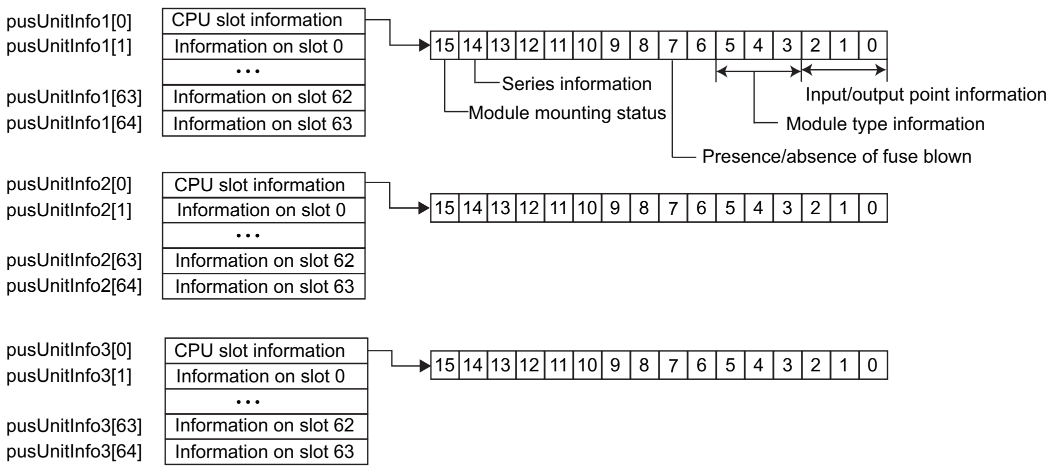
The series can be checked in the 14th bit of pusUnitInfo1[0-64].



pusUnitInfo	Description		
	Storage position		Status
pusUnitInfo1[0-64]	bit15	Module mounting status	• 0: Not mounted • 1: Mounted
	bit14	Series information	1: MELSEC iQ-R series (0: MELSEC-Q series)
	bit13-0	Reserved	—
pusUnitInfo2[0-64]	bit15-8	Reserved	—
	bit7-4	Module type information	• 0000: Input module • 0001: Power • 0010: Output module • 0011: Base unit • 0100: Reserved • 0101: Reserved • 0110: I/O combined module • 0111: Empty • 1000: Intelligent function module • 1001: CPU • 1010: Bus extension module • 1011: Reserved • 1100: Reserved • 1101: Reserved • 1110: Reserved • 1111: Module other than above
	bit3-0	Input/output point information	• 0000: 16 points • 0001: 32 points • 0010: 48 points • 0011: 64 points • 0100: 128 points • 0101: 256 points • 0110: 512 points • 0111: 1024 points • 1000: 2048 points • 1001: 4096 points • 1111: 0
pusUnitInfo3[0-64]	bit15-10	Reserved	—
	bit9	Presence/absence of fuse blown	• 0: Normal • 1: Fuse blown
	bit8-0	Reserved	—

■Series information is MELSEC-Q series

The series can be checked in the 14th bit of pusUnitInfo1[0-64].



pusUnitInfo	Description		
	Storage position		Status
pusUnitInfo1[0-64]	bit15	Module mounting status	• 0: Not mounted • 1: Mounted
	bit14	Series information	0: MELSEC-Q series (1: MELSEC iQ-R series)
	bit7	Presence/absence of fuse blown	• 0: Normal • 1: Fuse blown
	bit6	Reserved	—
	bit5-3	Module type information	• 000: Input module • 001: Output module • 010: I/O combined module • 011: Intelligent function module • 111: Module other than above
	bit2-0	Input/output point information	• 000: 16 points • 001: 32 points • 010: 48 points • 011: 64 points • 100: 128 points • 101: 256 points • 110: 512 points • 111: 1024 points
pusUnitInfo2[0-64]	bit15-0	Reserved	—
pusUnitInfo3[0-64]	bit15-0	Reserved	—

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

CCPU_MountMemoryCard

This function mounts an SD memory card inserted to a C Controller module.

■Format

short CCPU_MountMemoryCard (short sDrive)

■Argument

Argument	Name	Description	IN/OUT
sDrive	Target drive	Specify a target drive. (When 'Reserved' is specified, this function ends normally without processing.) • 1: SD memory card • Others: Reserved	IN

■Description

- This function mounts the drive specified to the target drive (sDrive).
- The CARD RDY LED keeps flashing during the mount processing, and it turns ON once the mount processing is completed.
- This function can be executed when the status of an SD memory card is "Inserted (unmounted)".
  (The status can be checked by using the CCPU_GetCpuStatus function.)
- When an SD memory card has already been mounted, this function ends normally without processing.

Point

Use this function to access an SD memory card without removing it after unmounting the SD memory card by using the CCPU_UnmountMemoryCard function while the power is ON.
This function does not need to be executed since an SD memory card is automatically mounted when it is replaced.

Precautions

The USB Mass Storage Class-compliant device cannot be mounted using this function.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 104 CCPU_UnmountMemoryCard
- Page 69 CCPU_GetCpuStatus

CCPU_ReadDevice

This function reads data from internal user devices and internal system devices of C Controller module.

■Format

short CCPU_ReadDevice (short sDevType, unsigned long ulDevNo, unsigned long ulSize, unsigned short* pusDataBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
sDevType	Device type	Specify the device type. ☞ Page 9 Argument specification	IN
ulDevNo	Start device number	Specify the start device number. (Only multiples of 16 can be specified for bit devices.)	IN
ulSize	Data size	Specify the read data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of read data.	OUT
ulBufSize	Data storage destination size	Specify the data storage destination size in word units.	IN

■Description

This function reads data equivalent to the size specified to the data size (ulSize) from a device specified to the device type (sDevType) and the start device number (ulDevNo) and subsequent devices, and stores the read data in the data storage destination (pusDataBuf).

Precautions

Note that the size of data storage destination (ulBufSize) should be equal to or bigger than the data size (ulSize).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 111 CCPU_WriteDevice

CCPU_ReadLinkDevice

This function reads data from the own station link devices of CC-Link IE Controller Network module, CC-Link IE Field Network module, Ethernet module (when CC-Link IE Field Network is selected), and MELSECNET/H network module.

■Format

short CCPU_ReadLinkDevice (unsigned short usIoNo, short sDevType, unsigned long ulDevNo, unsigned long ulSize, unsigned short* pusDataBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
usIoNo	Module position	Specify the module position as follows. Start I/O number divided by 16 (0H to FFH)	IN
sDevType	Device type	Specify the device type. ☞ Page 9 Argument specification	IN
ulDevNo	Start device number	Specify the start device number. (Only multiples of 16 can be specified for bit devices.)	IN
ulSize	Data size	Specify the read data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of read data.	OUT
ulBufSize	Data storage destination size	Specify the data storage destination size in word units.	IN

■Description

This function reads data equivalent to the size specified to the data size (ulSize) from a device specified to the device type (sDevType) and the start device number (ulDevNo) and subsequent devices of a CC-Link IE Controller Network module, CC-Link IE Field Network module, Ethernet module (when CC-Link IE Field Network is selected), and MELSECNET/H network module which are specified to the module position (usIoNo), and stores the read data in the data storage destination (pusDataBuf).

Precautions

Note that the size of data storage destination (ulBufSize) should be equal to or bigger than the data size (ulSize).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 112 CCPU_WriteLinkDevice

CCPU_RegistEventLog

This function registers event logs in the event history of C Controller module.

■Format

short CCPU_RegistEventLog (long lEventCode, char* pcEventMsg)

■Argument

Argument	Name	Description	IN/OUT
lEventCode	Detailed code	Specify a detailed event code to be registered in the event history.	IN
pcEventMsg	Detailed information	Specify detailed information character string data of an event to be registered in the event history. (The detailed information character string data of an event can be specified up to 200 bytes. When 'NULL' is specified, the detailed information is not registered.)	IN

■Description

This function registers event logs in the event history of C Controller module.

The contents to be registered on the event history screen of CW Configurator are as follows:

Item	Description
Occurrence Date	Event registered date and time
Event Type	Operation (Fixed)
Status	Information (Fixed)
Event Code	25000 (Fixed)
Overview	Registration from the user program (Fixed)
Source	R12CCPU-V (Fixed)
Start I/O No.	Input/output number of the C Controller module that executed the CCPU_RegistEventLog function.
Detailed event code information	Detailed code (hexadecimal) specified to the detailed code (lEventCode)
Detailed event log information	Detailed information specified to pcEventMsg
Cause	The event history was registered from the C Controller module detailed function. (Fixed)

- The event history can be stored for the size of the event history file specified with CW Configurator.
  Note that old data is deleted if the event history exceeds the specified file size.
- An error occurs if the character string data specified to the detailed information (pcEventMsg) is 201 bytes or bigger.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

CCPU_Reset

This function resets the bus master CPU (CPU No.1).

■Format

short CCPU_Reset (void)

■Argument

None

■Description

- This function resets the bus master CPU (CPU No.1).
- Use this function only to reset the bus master CPU due to an error or others.
- Do not execute this function while a file in the program memory, SD memory card, and USB Mass Storage Class-compliant device is being accessed. Doing so may result in data corruption or file system failure.
- Perform the following before executing this function while a file is being accessed.

File access destination	Description
Program memory	Close a user file.
SD memory card, USB Mass Storage Class-compliant device	Close a user file, and unmount an SD memory card and a USB Mass Storage Class-compliant device.

- This function can be executed only when all the following conditions are satisfied.
  If the condition is not satisfied, the error codes indicated in the brackets below will be returned.

Host CPU	Description
Bus master CPU (CPU No.1)	"Enable" is set to "Remote Reset" in the bus master CPU (CPU No.1). (Unset: 16523H) The operating status of the bus master CPU (CPU No.1) is in the STOP state. (RUN/PAUSE state: -222H)
CPUs other than the bus master (CPU No.1)	The bus master CPU (CPU No.1) is a CPU module: (C Controller module: -222H) "Enable" is set to "Remote Reset" in the bus master CPU (CPU No.1). (Unset: -222H) The operating status of the bus master CPU (CPU No.1) is in the STOP state. (RUN/PAUSE state: -222H)

Precautions

- When remote STOP is performed to the bus master CPU (CPU No.1) from other peripheral devices (such as GX Works3), the bus master CPU (CPU No.1) cannot be reset by using the CCPU_Reset function.
  For the remote operation and the operating status of a C Controller module, refer to the manual.
  (📖 MELSEC iQ-R C Controller Module User's Manual (Application))
- When this function is executed, no value is returned because a C Controller module is restarted from an operating system. (All programs are forcibly terminated.)

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 104 CCPU_UnmountMemoryCard
- Page 97 CCPU_ShutdownRom

CCPU_ResetDevice

This function resets internal user devices and internal system devices (bit devices) of C Controller module.

■Format

short CCPU_ResetDevice (short sDevType, unsigned long ulDevNo)

■Argument

Argument	Name	Description	IN/OUT
sDevType	Device type	Specify the device type. ☞ Page 9 Argument specification	IN
ulDevNo	Start device number	Specify the start device number.	IN

■Description

This function resets (turns OFF) the device of a C Controller module specified to the device type (sDevType) and the start device number (ulDevNo).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

• Page 91 CCPU_SetDevice

CCPU_ResetWDT

This function resets the user WDT of C Controller module.

■Format

short CCPU_ResetWDT (short sType)

■Argument

Argument	Name	Description	IN/OUT
sType	WDT type	Specify the WDT type. (If reserve is specified, an error is returned.) • 0: User WDT • Others: Reserved	IN

■Description

- This function resets the user WDT.
- When this function is executed without starting the user WDT, an error will be returned.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 98 CCPU_StartWDT
- Page 99 CCPU_StopWDT
- Page 63 CCPU_EntryWDTInt

CCPU_SetDevice

This function sets internal user devices and internal system devices (bit devices) of C Controller module.

■Format

short CCPU_SetDevice (short sDevType, unsigned long ulDevNo)

■Argument

Argument	Name	Description	IN/OUT
sDevType	Device type	Specify the device type. ☞ Page 9 Argument specification	IN
ulDevNo	Device number	Specify the device number.	IN

■Description

This function sets (turns ON) the device of a C Controller module specified to the device type (sDevType) and the start device number (ulDevNo).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 89 CCPU_ResetDevice

CCPU_SetDotMatrixLED

This function sets a value to be displayed on the dot matrix LED of C Controller module.

■Format

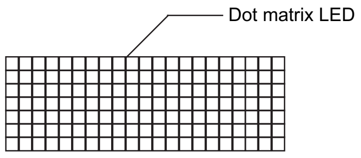
short CCPU_SetDotMatrixLED(unsigned short usLedMode, char* pcData)

■Argument

Argument	Name	Description	IN/OUT
usLedMode	Output mode	Specify the output mode to the dot matrix LED. (When 'Reserved' is specified, this function ends normally without processing.) • 0: Dot mode • 1: ASCII mode • Others: Reserved	IN
pcData	LED data	Specify the LED data.	IN

• Specify the LED data (pcData) as follows.

· Mode 0: Dot mode



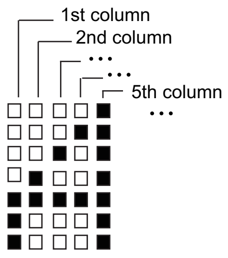
pcData[0] to pcData[19]: Data of the dot matrix LED (7×20)

The data specified in the following format is displayed.

Data format for each column: Bit pattern in which '0' is for the upper one bit, and '1' (when LED is ON) or '0' (when LED is OFF) is for lower seven bits

Ex.

The bit pattern shown below is output to the dot matrix LED:



1st column: 0000 0111b = 07H → pcData[0] = 0x07

2nd column: 0000 1100b = 0cH → pcData[1] = 0x0c

3rd column: 0001 0100b = 14H → pcData[2] = 0x14

4th column: 0010 0100b = 24H → pcData[3] = 0x24

5th column: 0111 1111b = 7fH → pcData[4] = 0x7f

6th column to 20th column: 0000 0000b = 00H → pcData[5] to pcData[19] = 0x00

·Mode 1: ASCII mode

The specified character strings are displayed in pcData[0] to pcData[3].

Available characters (ASCII code) are shown below.

×: character string specification not allowed

Bit		Upper four bits															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
Lower four bits	0	×	×	SP	0	×	P	×	×	×	×	×	×	×	×	×	×
	1	×	×	×	1	A	Q	×	×	×	×	×	×	×	×	×	×
	2	×	×	×	2	B	R	×	×	×	×	×	×	×	×	×	×
	3	×	×	×	3	C	S	×	×	×	×	×	×	×	×	×	×
	4	×	×	×	4	D	T	×	×	×	×	×	×	×	×	×	×
	5	×	×	%	5	E	U	×	×	×	×	×	×	×	×	×	×
	6	×	×	×	6	F	V	×	×	×	×	×	×	×	×	×	×
	7	×	×	×	7	G	W	×	×	×	×	×	×	×	×	×	×
	8	×	×	×	8	H	X	×	×	×	×	×	×	×	×	×	×
	9	×	×	×	9	I	Y	×	×	×	×	×	×	×	×	×	×
	A	×	×	×	×	J	Z	×	×	×	×	×	×	×	×	×	×
	B	×	×	×	×	K	×	×	×	×	×	×	×	×	×	×	×
	C	×	×	×	×	L	×	×	×	×	×	×	×	×	×	×	×
	D	×	×	-	×	M	×	×	×	×	×	×	×	×	×	×	×
	E	×	×	.	×	N	×	×	×	×	×	×	×	×	×	×	×
	F	×	×	/	×	O	×	×	×	×	×	×	×	×	×	×	×

When a character other than above is specified, an error will be returned.

When the character strings are null-terminated, data after the NULL character are not displayed (blank). (They are displayed with left-aligned.)

■Description

This function displays the value specified to the LED data (pcData) on the dot matrix LED according to the output mode (usLedMode).

Precautions

- To display data on the dot matrix LED, selecting 'USER' in the operation selection mode is required. (📖 MELSEC iQ-R C Controller Module User's Manual (Startup))
- When checking an operation or checking the selected operation with the MODE/SELECT switch, an error occurs at the time of executing the CCPU_SetDotMatrixLED function even when "USER" is selected in the operation selection mode.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 70 CCPU_GetDotMatrixLED

CCPU_SetLEDStatus

This function sets the LED status of a C Controller module.

■Format

short CCPU_SetLEDStatus(long lLed, unsigned short usLedInfo)

■Argument

Argument	Name	Description	IN/OUT
lLed	Target LED	Specify the target LED. (When 'Reserved' is specified, this function ends normally without processing.) • 0: USER LED • Others: Reserved	IN
usLedInfo	LED status information	Specify the LED status information.	IN

The specification method of the LED status information (usLedInfo) is as follows:

usLedInfo	Description
0	OFF
1	ON (Red)
2	Flashing at low speed (Red)
3	Flashing at high speed (Red)
4	ON (Green)
5	Flashing at low speed (Green)
6	Flashing at high speed (Green)

■Description

This function controls the USER LED of C Controller module to the status specified to the LED status information (usLedInfo).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

• Page 74 CCPU_GetLEDStatus

CCPU_SetOpSelectMode

This function sets the operation selection mode of C Controller module.

■Format

short CCPU_SetOpSelectMode(long lModeInfo, long lSelectMode)

■Argument

Argument	Name	Description	IN/OUT
lModeInfo	Mode information	Specify the mode information.	IN
lSelectMode	Operation selection mode	Specify the operation selection mode.	IN

The specification method of the mode information (lModeInfo) and operation selection mode (lSelectMode) is as follows:

lModeInfo	lSelectMode	Description
1	1	Notifies event to the user program when holding the MODE/SELECT switch in the SELECT position.
	2	Unmounts SD memory card forcibly when holding the MODE/SELECT switch in the SELECT position.
	3	Unmounts USB Mass Storage Class-compliant devices forcibly when holding the MODE/SELECT switch in the SELECT position.
	4	Unmounts SD memory card/USB Mass Storage Class-compliant devices forcibly when holding the MODE/SELECT switch in the SELECT position.
	Others	Reserved
2	1	Displays the specified content on the dot matrix LED.
	2	Displays an error code on the dot matrix LED.
	3	Displays the IP address of CH1 on the dot matrix LED.
	4	Displays the IP address of CH2 on the dot matrix LED.
	Others	Reserved
Others	—	Reserved

■Description

- This function sets the operation selection mode of C Controller module to the status specified to the operation selection mode (lSelectMode).
- The operation selection mode setting will be valid after this function is executed.
- If the operation selection mode is changed with both of this function and the switch operation, the mode set the last will be valid.
- An error occurs if this function is executed while the MODE/SELECT switch is being operated to select an operation.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 76 CCPU_GetOpSelectMode

CCPU_SetRTC

This function sets the clock data (local time) of C Controller module.

■Format

short CCPU_SetRTC(short* psSetData)

■Argument

Argument	Name	Description	IN/OUT
psSetData	Clock data	Specify the clock data (local time) to be set.	IN

- Specify the clock data (local time) to the clock data (psSetData) as follows.

(Available range: January 1, 1980 to December 31, 2079)

psSetData	Description
psSetData[0]	Year data (1980 to 2079)
psSetData[1]	Month data (1 to 12)
psSetData[2]	Day data (1 to 31)
psSetData[3]	Hour data (0 to 23)
psSetData[4]	Minute data (0 to 59)
psSetData[5]	Second data (0 to 59)

■Description

- This function sets the clock data (local time) specified to the clock data (psSetData) to C Controller module.
- If the clock data (psSetData) is out of the range, an error is returned.
- Once the clock data (local time) is set, the history set to the event history is registered.
- When the daylight saving time function is enabled, an error is returned if clock data less than one hour from the start date and time of daylight saving time is set.

Precautions

- The clock data (local time) set with this function are not reflected to the clock of the operating system (VxWorks).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 78 CCPU_GetRTC

CCPU_ShutdownRom

This function shuts down the program memory and data memory of a C Controller module.

■Format

short CCPU_ShutdownRom (void)

■Argument

None

■Description

- This function shuts down the program memory and data memory of a C Controller module. The BUS RUN LED starts flashing at high speed after the shutdown. (The shutdown status can be checked by using the CCPU_GetCpuStatus function.)
- This function is used to shut down the program memory and data memory before powering OFF a C Controller module. File operations (creating, deleting, and overwriting a file) on the program memory and data memory are disabled after the shutdown. However, reference to the program memory and data memory is possible.
- When calling this function, make sure to stop accessing files in the program memory and data memory, and close all files. Otherwise, data corruption or a file system error may occur.
- Always power OFF a system or reset a CPU module after checking that a shutdown is completed. If operation is continued, an error occurs when accessing files in the program memory and the data memory.
- Shutdown processing is performed in the order from the program memory to the data memory. If the program memory fails to be shut down, the data memory will not be shut down.
- When the program memory and data memory are in the shutdown completed status, this function ends normally without processing.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 84 CCPU_MountMemoryCard
- Page 104 CCPU_UnmountMemoryCard

CCPU_StartWDT

This function sets and starts the user WDT of a C Controller module.

■Format

short CCPU_StartWDT(short sType, short sInterval)

■Argument

Argument	Name	Description	IN/OUT
sType	WDT type	Specify the WDT type. (If reserve is specified, an error is returned.) • 0: User WDT • Others: Reserved	IN
sInterval	WDT interval	Specify the interval of WDT in 10 ms units. (Available range is between 10 to 1000 (100 to 10000 [ms]).)	IN

■Description

- The user WDT is the timer for detecting a hardware failure or program error.
- This function sets an interval of the WDT to the WDT interval (sInterval) ×10 ms and starts the user WDT.
- If the user WDT is not reset periodically within the set time (by execution of the CCPU_ResetWDT function), a user WDT error will occur. When a user WDT error occurs, a C Controller module will be in the stop error state. (The BUS RUN LED turns OFF, and the ERROR LED starts flashing.)
- When this function is executed while the WDT is running, an error will be returned.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 90 CCPU_ResetWDT
- Page 99 CCPU_StopWDT
- Page 63 CCPU_EntryWDTInt

CCPU_StopWDT

This function stops the user WDT of C Controller module.

■Format

short CCPU_StopWDT(short sType)

■Argument

Argument	Name	Description	IN/OUT
sType	WDT type	Specify the WDT type. (If reserve is specified, an error is returned.) • 0: User WDT • Others: Reserved	IN

■Description

- This function stops the user WDT.
- When this function is executed without starting the user WDT, this function ends normally.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 98 CCPU_StartWDT
- Page 90 CCPU_ResetWDT
- Page 63 CCPU_EntryWDTInt

CCPU_SysClkRateGet

This function reads the system clock rate specified with the CCPU_SysClkRateSet function from the backup RAM.

■Format

short CCPU_SysClkRateGet(short* psTicks)

■Argument

Argument	Name	Description	IN/OUT
psTicks	Clock rate	Stores the system clock rate in the unit of clock frequency (Hz) per one second. • 0: Default value (60 Hz) • 60 to 1000: Specified clock rate value	OUT

■Description

This function reads the system clock rate specified with the CCPU_SysClkRateSet function from the backup RAM.

Precautions

The read value may not correspond to the system clock rate in operation.
To check the system clock rate in operation, use the sysClkRateGet function of VxWorks.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

• Page 101 CCPU_SysClkRateSet

CCPU_SysClkRateSet

This function specifies a system clock rate, and stores it in the backup RAM.

■Format

short CCPU_SysClkRateSet(short sTicks, short* psRestart)

■Argument

Argument	Name	Description	IN/OUT
sTicks	Clock rate	Specify the system clock rate in the unit of clock frequency (Hz) per one second. • 0: Default value (60 Hz) • 60 to 1000: Specified clock rate value	IN
psRestart	Restart necessity flag	Stores the necessity to restart a C Controller module after the execution of this function. (When 'NULL' is specified, the restart necessity flag is not stored.) • 0: Restart is not required. (C Controller module has already been running at the specified clock rate.) • 1: Restart is required. (C Controller module operates at the specified clock rate after restarting it.)	OUT

■Description

- This function specifies a system clock rate, and stores it in the backup RAM.
  The specified system clock rate will be enabled after a C Controller module is restarted.
- When the output to the restart necessity flag (psRestart) is '0' (restart is not required), continue the application processing.
- When the output to the restart necessity flag (psRestart) is '1' (restart is required), stop the application processing, and reset or power OFF to ON a C Controller module.
- For more details on system clock rate, refer to the manual for VxWorks.

Precautions

- Execute this function only once after a C Controller module is started.
  If this function is executed with the same clock rate value as the one specified for the first time, the restart necessity flag (psRestart) will be '0' (restart is not required) regardless of the system clock rate in operation.
- Use this function to change the system clock rate.
  If the sysClkRateSet function of VxWorks is used, VxWorks operation will be unstable.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 100 CCPU_SysClkRateGet

CCPU_ToBuf

This function writes data to the CPU buffer memory of the CPU module (host CPU) and the buffer memory of the intelligent function module which are mounted on the specified module position. (TO instruction)

■Format

short CCPU_ToBuf (unsigned short usIoNo, unsigned long ulOffset, unsigned long ulSize, unsigned short* pusDataBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
usIoNo	Module position	Specify the module position as follows. For the CPU buffer memory, only the host CPU can be accessed. Start I/O number divided by 16 (0H to FFH, 3E0H to 3E3H)	IN
ulOffset	Offset	Specify the offset in word units.	IN
ulSize	Data size	Specify the write data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of write data.	IN
ulBufSize	Data storage destination size	Specify '0'.	IN

■Description

- This function writes data in the data storage destination (pusDataBuf) equivalent to the size specified to the data size (ulSize) to the CPU buffer memory of a CPU module (host CPU) and the buffer memory of an intelligent function module which are specified to the module position (usIoNo).
  Data is written by specifying an offset address (ulOffset) from the start of the CPU buffer memory of a CPU module (host CPU) and the buffer memory of an intelligent function module.
- To access the CPU buffer memory (host CPU) of the module in a multiple CPU system (CPU No.1 to No.4), specify 3E0H (CPU No.1) to 3E3H (CPU No.4) to the module position (usIoNo). However, the CPU buffer memory (host CPU) can be accessed only when the multiple CPU setting is configured.
- When executing this function while the operating status of a CPU module is not RUN, the STOP/PAUSE error (-28640) occurs.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 64 CCPU_FromBuf

CCPU_ToBufHG

This function writes data to the fixed cycle communication area of the CPU module mounted on the specified module position.

■Format

short CCPU_ToBufHG(unsigned short usIoNo, unsigned long ulOffset, unsigned long ulSize, unsigned short* pusDataBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
usIoNo	Module position	Specify the module position as follows. Start I/O number divided by 16 (3E0H to 3E3H)	IN
ulOffset	Offset	Specify the offset in word units.	IN
ulSize	Data size	Specify the write data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of write data.	IN
ulBufSize	Data storage destination size	Specify '0'.	IN

■Description

- This function writes data in the data storage destination (pusDataBuf) equivalent to the size specified to the data size (ulSize) to the fixed cycle communication area of a CPU module specified to the module position (usIoNo). Data is written by specifying an offset address (ulOffset) from the start of the fixed cycle communication area.
- The fixed cycle communication area can be accessed only when the fixed cycle communication area setting under the multiple CPU setting is configured.
- When executing this function while the operating status of a CPU module is not RUN, the STOP/PAUSE error (-28640) occurs.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 64 CCPU_FromBuf
- Page 102 CCPU_ToBuf
- Page 65 CCPU_FromBufHG

CCPU_UnmountMemoryCard

This function unmounts the SD memory card and USB Mass Storage Class-compliant device inserted to C Controller module.

■Format

short CCPU_UnmountMemoryCard (short sDrive)

■Argument

Argument	Name	Description	IN/OUT
sDrive	Target drive	Specify a target drive. (When 'Reserved' is specified, this function ends normally without processing.) • 1: SD memory card • 2: USB Mass Storage Class-compliant device • Others: Reserved	IN

■Description

- This function unmounts the drive specified to the target drive (sDrive).
- The CARD RDY LED keeps flashing during the unmount processing of the SD memory card, and it turns OFF after the unmount processing is completed.
- The USB RDY LED keeps flashing during the unmount processing of the USB Mass Storage Class-compliant device, and it turns OFF after the unmount processing is completed.
- This function can be executed when the status of the drive specified to the target drive (sDrive) is "Inserted (mounted)". (The status can be checked by using the CCPU_GetCpuStatus function.)
- When the drive specified to the target drive (sDrive) has already been unmounted, this function ends normally without processing.

Precautions

Design a program so that accessing files in the target drive is stopped and all files are closed before calling this function. If this function is called while the files are open, data corruption or a file system error may occur.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 84 CCPU_MountMemoryCard
- Page 69 CCPU_GetCpuStatus

CCPU_WaitEvent

This function waits for an interrupt event notification from other CPUs.

■Format

short CCPU_WaitEvent (short* psEvent, unsigned long ulTimeout, short* psSetEventNo)

■Argument

Argument	Name	Description	IN/OUT
psEvent	Interrupt event setting	Specify the interrupt event.	IN
ulTimeout	Timeout value	Specify the timeout value in ms units (0H to FFFFFFFFH). (When FFFFFFFFH is specified, the function waits for an event infinitely.)	IN
psSetEventNo	Occurred event	Stores the occurred event. Stores the CPU number and event number (interrupt pointer number) of the notified interrupt event.	OUT

• The specification method of the interrupt event setting (psEvent) is as follows:

psEvent	Description	
psEvent[0]	Number of interrupt event settings (1 to 64)	
psEvent[1]	CPU number of the first interrupt event (1 to 4)	First event setting
psEvent[2]	Event number (interrupt pointer number) of the first interrupt event (0 to 15)	
psEvent[3]	CPU number of the second interrupt event (1 to 4)	Second event setting
psEvent[4]	Event number (interrupt pointer number) of the second interrupt event (0 to 15)	
psEvent[5]	CPU number of the third interrupt event (1 to 4)	Third event setting
psEvent[6]	Event number (interrupt pointer number) of the third interrupt event (0 to 15)	
⋮	⋮	⋮

• The following value is stored in the occurred event (psSetEventNo).

psSetEventNo	Description
psSetEventNo[0]	CPU number of the notified interrupt event
psSetEventNo[1]	Event number (interrupt pointer number) of the notified interrupt event

■Description

- This function waits for an interrupt event specified to the interrupt event setting (psEvent) for the amount of time equivalent to the time specified to the timeout value (ulTimeout).
- When multiple interrupt events occur, the interrupt events are notified in ascending order of the event number.
- If an interrupt event has already been notified at a time when this function is called, this function immediately ends normally. When a reset operation is performed, any interrupt event that occurred prior to reset is discarded.
- If multiple interrupt events have been notified for the same event number (interrupt pointer number) at a time when this function is called, it is notified as a single interrupt event.
- Set the event number (interrupt pointer number) without duplication. Otherwise, an error will be returned.
- The specified timeout value is rounded to the tick unit. Specify a timeout value of one tick or more.
- Specify the programmable controller CPU or C Controller module to the CPU number. Otherwise, an error will be returned.
- Design a program so that this function is not called simultaneously by specifying the same event number (interrupt pointer number) from multiple tasks. Otherwise, the execution of the interrupt event notified task is unpredictable.

Ex.

Setting of psEvent to wait for interrupt event 0 and 1 for CPU No.1, and interrupt event 10 for CPU No.2

```
psEvent[0] = 3;
psEvent[1] = 1;
psEvent[2] = 0;
psEvent[3] = 1;
psEvent[4] = 1;
psEvent[5] = 2;
psEvent[6] = 10;
```

When interrupt event 10 for CPU No.2 occurs, '2' and '10' are returned to psSetEventNo[0] and psSetEventNo[1], respectively.

Precautions

Do not set the clock data of a C Controller module while executing this function. Otherwise, this function does not operate properly. (This function may not be completed.)

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 109 CCPU_WaitUnitEvent

CCPU_WaitSwitchEvent

This function waits for a switch interrupt event of C Controller module to occur.

■Format

short CCPU_WaitSwitchEvent(short sSwitch, unsigned long ulTimeout)

■Argument

Argument	Name	Description	IN/OUT
sSwitch	Switch interrupt event type	Specify the switch interrupt event type. • 0: RUN switch interrupt event • 1: STOP switch interrupt event • 2: SELECT switch interrupt event	IN
ulTimeout	Timeout	Specify the timeout value in ms units (0H to FFFFFFFFH). (When FFFFFFFFH is specified, the function waits for an event infinitely.)	IN

■Description

- This function waits for a switch interrupt event specified to the switch interrupt event type (sSwitch).
- If an interrupt event has already been notified at a time when this function is called, this function immediately ends normally.
- If the same switch interrupt event has been notified several times at a time when this function is called, the user program executes processing as a single switch interrupt event notification.
- The specified timeout value is rounded to the tick unit. Specify a timeout value of one tick or more.

Precautions

- To issue the switch interrupt event by holding the MODE/SELECT switch in the SELECT position, selecting "EVENT" in the operation selection mode is required. (📖 MELSEC iQ-R C Controller Module User's Manual (Startup))
- For the SELECT switch interrupt event, an event issuance status cannot be judged from the appearance. To check the issued status of SELECT switch interrupt event, implement the processing such as receiving a switch interrupt event using this function and making the USER LED turn ON.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- None

CCPU_WaitTimerEvent

This function waits for a timer event to occur.

■Format

short CCPU_WaitTimerEvent (long lEventNo)

■Argument

Argument	Name	Description	IN/OUT
lEventNo	Timer event number	Specify a timer event number that waits for a timer event to occur. (1 to 16)	IN

■Description

- This function waits for a timer event specified to the timer event number (lEventNo) to occur.
- The occurrence cycle of the timer every number (1 to 16) can be set, changed, or cleared by the CCPU_EntryTimerEvent function.
- When reset operation is performed, any event that has occurred prior to reset is discarded.
- Using this function enables a cycle timer task. However, even though an event occurs, the waiting task may not be operated immediately due to the system status (such as the interrupt).
- If waiting for an event with this function to a cleared timer event, the wait status will not be cleared until an event occurs after the registration of the event (and the specified cycle has elapsed) with CCPU_EntryTimerEvent function.

Precautions

Note that operation of waiting for event (function completion) using this function will vary. This operation variation depends on the specified value of synchronization type of the timer event number with the CCPU_EntryTimerEvent function.

- When the synchronization type is batch synchronization, the wait state of all tasks that are waiting for an event is canceled. However, if there is no task in the wait state at event occurrence, the wait state will not be canceled even if the CCPU_WaitTimerEvent function is called.
- When the synchronization type is individual synchronization, the wait state of one task in the tasks that are waiting for an event is canceled. If multiple tasks are waiting for the same event, the wait state will be canceled in order of priority of a task (in order of execution of wait when the priority is same). However, if there is no task in the wait state at the time of event occurrence, the wait state will not be canceled even if the CCPU_WaitTimerEvent function is called later.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 62 CCPU_EntryTimerEvent

CCPU_WaitUnitEvent

This function waits for an interrupt event notification from modules.

■Format

short CCPU_WaitUnitEvent (short* psEvent, unsigned long ulTimeout, short* psSetEventNo)

■Argument

Argument	Name	Description	IN/OUT
psEvent	Event setting	Specify the interrupt event.	IN
ulTimeout	Timeout value	Specify the timeout value in ms units (0H to FFFFFFFFH). (When FFFFFFFFH is specified, the function waits for an event infinitely.)	IN
psSetEventNo	Occurred event	Stores the occurred event. Stores the event number (interrupt pointer number) of the notified interrupt event.	OUT

- The specification method of the event setting (psEvent) is as follows:

psEvent	Description
psEvent[0]	Number of interrupt event settings (1 to 64)
psEvent[1]	Interrupt pointer number of the first interrupt event (0 to 15, 50 to 1023)
psEvent[2]	Interrupt pointer number of the second interrupt event (0 to 15, 50 to 1023)
psEvent[3]	Interrupt pointer number of the third interrupt event (0 to 15, 50 to 1023)
⋮	⋮

■Description

- This function waits for an interrupt event specified to the event setting (psEvent) for the amount of time equivalent to the time specified to the timeout value (ulTimeout).
- When multiple interrupt events occur, the interrupt events are notified in ascending order of the event number.
- If an interrupt event has already been notified at a time when this function is called, this function immediately ends normally. When a reset operation is performed, any interrupt event that occurred prior to reset is discarded.
- If multiple interrupt events have been notified for the same event number (interrupt pointer number) at a time when this function is called, it is notified as a single interrupt event.
- Set the event number (interrupt pointer number) without duplication. Otherwise, an error will be returned.
- The specified timeout value is rounded to tick unit. Specify a timeout value of one tick or more.
- Design a program so that this function is not called simultaneously by specifying the same interrupt event (interrupt pointer number) from multiple tasks. Otherwise, the execution of the interrupt event notified task is unpredictable.
- When an interrupt event is notified (return value of this function is normal), the event number of the notified interrupt event is returned to the occurred event (psSetEventNo).
- An interrupt event is not notified while a stop error is occurring in a C Controller module.

Ex.

Setting of psEvent when waiting for interrupt event 0, interrupt event 1, interrupt event 50, and interrupt event 51

psEvent[0] = 4;

psEvent[1] = 0;

psEvent[2] = 1;

psEvent[3] = 50;

psEvent[4] = 51;

When event 51 occurs, 51 is returned to psSetEventNo.

The event numbers (interrupt pointer numbers) are as follows.

Event number (Interrupt pointer number)	Interrupt factor	Notes
0 to 15	Interrupt by module	—
16 to 49	Reserved	—
50 to 1023	Interrupt by module	Set with CW Configurator

Precautions

Do not set the clock data of a C Controller module while executing this function. Otherwise, this function does not operate properly. (This function may not be completed.)

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 105 CCPU_WaitEvent

CCPU_WriteDevice

This function writes data to internal user devices and internal system devices of C Controller module.

■Format

short CCPU_WriteDevice (short sDevType, unsigned long ulDevNo, unsigned long ulSize, unsigned short* pusDataBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
sDevType	Device type	Specify the device type. ☞ Page 9 Argument specification	IN
ulDevNo	Start device number	Specify the start device number. (Only multiples of 16 can be specified for bit devices.)	IN
ulSize	Data size	Specify the write data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of write data.	IN
ulBufSize	Data storage destination size	Specify '0'.	IN

■Description

This function writes data in the data storage destination (pusDataBuf) equivalent to the size specified to the data size (ulSize) to a device specified to the device type (sDevType) and the start device number (ulDevNo) and subsequent devices.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 85 CCPU_ReadDevice

CCPU_WriteLinkDevice

This function writes data to the own station link devices of CC-Link IE Controller Network module, CC-Link IE Field Network module, Ethernet module (when CC-Link IE Field Network is selected), and MELSECNET/H network module.

■Format

short CCPU_WriteLinkDevice (unsigned short usIoNo, short sDevType, unsigned long ulDevNo, unsigned long ulSize, unsigned short* pusDataBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
usIoNo	Module position	Specify the module position as follows. Start I/O number divided by 16 (0H to FFH)	IN
sDevType	Device type	Specify the device type. ☞ Page 9 Argument specification	IN
ulDevNo	Start device number	Specify the start device number. (Only multiples of 16 can be specified for bit devices.)	IN
ulSize	Data size	Specify the write data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of write data.	IN
ulBufSize	Data storage destination size	Specify '0'.	IN

■Description

- This function writes data in the data storage destination (pusDataBuf) equivalent to the size specified to the data size (ulSize) to a device specified to the device type (sDevType) and the start device number (ulDevNo) and subsequent devices of a CC-Link IE Controller Network module, CC-Link IE Field Network module, Ethernet module (when CC-Link IE Field Network is selected), and MELSECNET/H network module which are specified to the module position (usIoNo).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 86 CCPU_ReadLinkDevice

CCPU_X_In_BitEx

This function reads an input signal (X) in bit (1-point) units.

■Format

short CCPU_X_In_BitEx (short sFlg, unsigned short usXNo, unsigned short* pusData)

■Argument

Argument	Name	Description	IN/OUT
sFlg	Access flag	Specify an access flag. • 0: Normal access • Others: Reserved	IN
usXNo	Input signal	Specify an input signal (X).	IN
pusData	Data storage destination	Specify the storage destination of read data. Either of the following values is stored depending on the value of the input signal (X). • 0: OFF • 1: ON	OUT

■Description

- This function reads an input signal (X) specified to the input signal (usXNo) in bit (1-point) units.
- A read value of an input signal (X) is stored in the data storage destination (pusData).
- This function operates to the mounted module corresponding to the specified input signal (usXNo) regardless of the type of the parameter setting (I/O assignment). When the specified area is "empty", this function ends normally without processing (read data: 0). When it is "output module", the I/O assignment error occurs.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 114 CCPU_X_In_WordEx
- Page 117 CCPU_Y_Out_BitEx
- Page 118 CCPU_Y_Out_WordEx
- Page 115 CCPU_Y_In_BitEx
- Page 116 CCPU_Y_In_WordEx

CCPU_X_In_WordEx

This function reads an input signal (X) in word (16-point) units.

■Format

short CCPU_X_In_WordEx (short sFlg, unsigned short usXNo, unsigned short usSize, unsigned short* pusDataBuf, unsigned short usBufSize)

■Argument

Argument	Name	Description	IN/OUT
sFlg	Access flag	Specify an access flag. • 0: Normal access • Others: Reserved	IN
usXNo	Start input signal	Specify a start input signal (X). (Specify a multiple of 16.)	IN
usSize	Read data size	Specify the read data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of read data.	OUT
usBufSize	Data storage destination size	Specify the data storage destination size in word units.	IN

■Description

- This function reads an input signal (X) equivalent to the size specified to the read data size (usSize) from the start input signal (X) specified to the start input signal (usXNo), and stores the read data in the data storage destination (pusDataBuf).
- Specify an area size of the data storage destination (pusDataBuf) to the data storage destination size (usBufSize).
- This function operates to the mounted module corresponding to the specified input signal (usXNo) regardless of the type of the parameter setting (I/O assignment). When the specified area is "empty", this function ends normally without processing (read data: 0). When it is "output module", the I/O assignment error occurs.
- Read data is stored in the data storage destination (pusDataBuf) in ascending order from the lower bit as shown below.

pusDataBuf	Description
pusDataBuf[0]	Data of usXNo+FH to usXNo
pusDataBuf[1]	Data of usXNo+1FH to usXNo+10H
⋮	⋮
pusDataBuf[usSize-1]	Data of usXNo+(usSize-1)×16+FH to usXNo+(usSize-1)×16

Precautions

Note that the size of data storage destination (usBufSize) should be equal to or bigger than the read data size (usSize).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 113 CCPU_X_In_BitEx
- Page 117 CCPU_Y_Out_BitEx
- Page 118 CCPU_Y_Out_WordEx
- Page 115 CCPU_Y_In_BitEx
- Page 116 CCPU_Y_In_WordEx

CCPU_Y_In_BitEx

This function reads an output signal (Y) in bit (1-point) units.

■Format

short CCPU_Y_In_BitEx (short sFlg, unsigned short usYNo, unsigned short* pusData)

■Argument

Argument	Name	Description	IN/OUT
sFlg	Access flag	Specify an access flag. • 0: Normal access • Others: Reserved	IN
usYNo	Output signal	Specify an output signal (Y).	IN
pusData	Data storage destination	Specify the storage destination of read data. Either of the following values is stored depending on the value of the output signal (Y). • 0: OFF • 1: ON	OUT

■Description

- This function reads an output signal (Y) specified to the output signal (usYNo) in bit (1-point) units.
- A read value of an output signal (Y) is stored in the data storage destination (pusData).
- This function operates to the mounted module corresponding to the specified output signal (usYNo) regardless of the type of the parameter setting (I/O assignment). When the specified area is "empty", this function ends normally without processing (read data: 0). When it is "input module", the I/O assignment error occurs.
- No error will occur even if this function is executed when the operating status of a CPU module is STOP or PAUSE. An output signal (Y) is read at execution of this function.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 113 CCPU_X_In_BitEx
- Page 114 CCPU_X_In_WordEx
- Page 117 CCPU_Y_Out_BitEx
- Page 118 CCPU_Y_Out_WordEx
- Page 116 CCPU_Y_In_WordEx

CCPU_Y_In_WordEx

This function reads an output signal (Y) in word (16-point) units.

■Format

short CCPU_Y_In_WordEx (short sFlg, unsigned short usYNo, unsigned short usSize, unsigned short* pusDataBuf, unsigned short usBufSize)

■Argument

Argument	Name	Description	IN/OUT
sFlg	Access flag	Specify an access flag. • 0: Normal access • Others: Reserved	IN
usYNo	Start output signal	Specify a start output signal (Y). (Specify a multiple of 16.)	IN
usSize	Read data size	Specify the read data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of read data.	OUT
usBufSize	Data storage destination size	Specify the data storage destination size in word units.	IN

■Description

- This function reads an output signal (Y) equivalent to the size specified to the read data size (usSize) from the start output signal (Y) specified to the start output signal (usYNo), and stores the read data in the data storage destination (pusDataBuf).
- Specify an area size of the data storage destination (pusDataBuf) to the data storage destination size (usBufSize).
- This function operates to the mounted module corresponding to the specified output signal (usYNo) regardless of the type of the parameter setting (I/O assignment). When the specified area is "empty", this function ends normally without processing (read data: 0). When it is "input module", the I/O assignment error occurs.
- No error will occur even if this function is executed when the operating status of a CPU module is STOP or PAUSE. An output signal (Y) is read at execution of this function.
- Read data is stored in the data storage destination (pusDataBuf) in ascending order from the lower bit as shown below.

pusDataBuf	Description
pusDataBuf[0]	Data of usYNo+FH to usYNo
pusDataBuf[1]	Data of usYNo+1FH to usYNo+10H
⋮	⋮
pusDataBuf[usSize-1]	Data of usYNo+(usSize-1)×16+FH to usYNo+(usSize-1)×16

Precautions

Note that the size of data storage destination (usBufSize) should be equal to or bigger than the read data size (usSize).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 113 CCPU_X_In_BitEx
- Page 114 CCPU_X_In_WordEx
- Page 117 CCPU_Y_Out_BitEx
- Page 118 CCPU_Y_Out_WordEx
- Page 115 CCPU_Y_In_BitEx

CCPU_Y_Out_BitEx

This function outputs an output signal (Y) in bit (1-point) units.

■Format

short CCPU_Y_Out_BitEx (short sFlg, unsigned short usYNo, unsigned short usData)

■Argument

Argument	Name	Description	IN/OUT
sFlg	Access flag	Specify an access flag. • 0: Normal access • Others: Reserved	IN
usYNo	Output signal	Specify an output signal (Y).	IN
usData	Data storage destination	Specify the storage destination of output data. (Specify the value of bit 0.) • 0: OFF • 1: ON	IN

■Description

- This function outputs (turns ON/OFF) an output signal (Y) specified to the output signal (usYNo) in bit (1-point) units.
- An output signal (Y) turns ON/OFF according to a value specified to bit 0 in the data storage destination (usData). (Values of bit 1 to 7 are ignored.)
- When executing this function while the operating status of a CPU module is not RUN, the STOP/PAUSE error occurs.
- When this function is executed to "input module", the I/O assignment error occurs.
- Do not specify an output module controlled by other CPUs to the output signal (usYNo).
  If it is specified, no operation is performed to the output module.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 113 CCPU_X_In_BitEx
- Page 114 CCPU_X_In_WordEx
- Page 118 CCPU_Y_Out_WordEx
- Page 115 CCPU_Y_In_BitEx
- Page 116 CCPU_Y_In_WordEx

CCPU_Y_Out_WordEx

This function outputs an output signal (Y) in word (16-point) units.

■Format

short CCPU_Y_Out_WordEx (short sFlg, unsigned short usYNo, unsigned short usSize, unsigned short* pusDataBuf, unsigned short usBufSize)

■Argument

Argument	Name	Description	IN/OUT
sFlg	Access flag	Specify an access flag. • 0: Normal access • Others: Reserved	IN
usYNo	Start output signal	Specify a start output signal (Y). (Specify a multiple of 16.)	IN
usSize	Output size	Specify the output size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of output data.	IN
usBufSize	Data storage destination size	Specify '0'.	IN

■Description

- This function outputs (turns ON/OFF) data in the data storage destination (pusDataBuf) from the start output signal (Y) specified to the start output signal (usYNo) to the output signal (Y) equivalent to the size specified to the data size (usSize).
- When executing this function while the operating status of a CPU module is not RUN, the STOP/PAUSE error occurs.
- When this function is executed to "input module", the I/O assignment error occurs.
- Do not specify an output module controlled by other CPUs to the output signal (usYNo).
  If it is specified, no operation is performed to the output module.
- Store output data in the data storage destination (pusDataBuf) in ascending order from the lower bit as shown below.

pusDataBuf	Description
pusDataBuf[0]	Data of usYNo+FH to usYNo
pusDataBuf[1]	Data of usYNo+1FH to usYNo+10H
⋮	⋮
pusDataBuf[usSize-1]	Data of usYNo+(usSize-1)×16+FH to usYNo+(usSize-1)×16

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 113 CCPU_X_In_BitEx
- Page 114 CCPU_X_In_WordEx
- Page 117 CCPU_Y_Out_BitEx
- Page 115 CCPU_Y_In_BitEx
- Page 116 CCPU_Y_In_WordEx

C Controller module dedicated functions for ISR

CCPU_DisableInt_ISR

This function disables the routine registered with the CCPU_EntryInt function.

■Format

short CCPU_DisableInt (short sSINo)

■Argument

Argument	Name	Description	IN/OUT
sSINo	Interrupt pointer number	Specify the interrupt pointer number.	IN

■Description

- This function disables the routine registered with the CCPU_EntryInt function. (The routine is not executed when an interrupt occurs.)
- Specify the interrupt pointer number (sSINo) specified in the CCPU_EntryInt function to Interrupt pointer number (sSINo).

■⚠WARNING

If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

- Page 61 CCPU_EntryInt
- Page 120 CCPU_EnableInt_ISR

CCPU_EnableInt_ISR

This function enables the routine registered with the CCPU_EntryInt function.

■Format

short CCPU_EnableInt (short sSINo)

■Argument

Argument	Name	Description	IN/OUT
sSINo	Interrupt pointer number	Specify the interrupt pointer number.	IN

■Description

- This function enables the routine registered with the CCPU_EntryInt function. (The routine is executed when an interrupt occurs.)
- Specify the interrupt pointer number (sSINo) specified in the CCPU_EntryInt function to Interrupt pointer number (sSINo).
- Since an interrupt does not occur while a stop error is occurring in a C Controller module, the routine registered by using the CCPU_EntryInt function will not be executed even if it is enabled.

■⚠WARNING

If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

- Page 61 CCPU_EntryInt
- Page 119 CCPU_DisableInt_ISR

CCPU_FromBuf_ISR

This function reads data from the CPU buffer memory of the CPU module and the buffer memory of the intelligent function module which are mounted on the specified module position. (FROM instruction)

■Format

short CCPU_FromBuf_ISR (unsigned short usIoNo, unsigned long ulOffset, unsigned long ulSize,
unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
usIoNo	Module position	Specify the module position as follows. Start I/O number divided by 16 (0H to FFH, 3E0H to 3E3H)	IN
ulOffset	Offset	Specify the offset in word units.	IN
ulSize	Data size	Specify the read data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of read data.	OUT

■Description

- This function reads data equivalent to the size specified to the data size (ulSize) from the CPU buffer memory of a CPU module and the buffer memory of an intelligent function module which are specified to the module position (usIoNo), and stores the read data in the data storage destination (pusDataBuf).
  Data is read by specifying an offset address (ulOffset) from the start of the CPU buffer memory of a CPU module and the buffer memory of an intelligent function module.
- To access the CPU buffer memory of the module in a multiple CPU system (CPU No.1 to No.4), specify 3E0H (CPU No.1) to 3E3H (CPU No.4) to the module position (usIoNo). However, the CPU buffer memory can be accessed only when the multiple CPU setting is configured.

Restriction

Do not execute this function in a routine other than an interrupt routine.

■⚠WARNING

- If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.
- This function does not check the specified argument.
  When creating a program, note the following:
  An address specified to the read data is a multiple of 2.
  A data area for the size (words) of the read data is reserved.
  A non-existent CPU buffer memory is not specified.
  A non-existent buffer memory is not specified.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 134 CCPU_ToBuf_ISR

CCPU_FromBufHG_ISR

This function reads data from the fixed cycle communication area of the CPU module mounted on the specified module position.

■Format

short CCPU_FromBufHG_ISR (unsigned short usIoNo, unsigned long ulOffset, unsigned long ulSize, unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
usIoNo	Module position	Specify the module position as follows. Start I/O number divided by 16 (3E0H to 3E3H)	IN
ulOffset	Offset	Specify the offset in word units.	IN
ulSize	Data size	Specify the read data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of read data.	OUT

■Description

- This function reads data equivalent to the size specified to the data size (ulSize) from the fixed cycle communication area of a CPU module specified to the module position (usIoNo), and stores the read data in the data storage destination (pusDataBuf). Data is read by specifying an offset address (ulOffset) from the start of the fixed cycle communication area.
- The fixed cycle communication area can be accessed only when the fixed cycle communication area setting under the multiple CPU setting is configured.

Restriction

Do not execute this function in a routine other than an interrupt routine.

■⚠WARNING

- If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.
- This function does not check the specified argument.
  When creating a program, note the following:
  An address specified to the read data is a multiple of 2.
  A data area for the size (words) of the read data is reserved.
  A non-existent fixed cycle communication area is not specified.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 121 CCPU_FromBuf_ISR
- Page 134 CCPU_ToBuf_ISR
- Page 135 CCPU_ToBufHG_ISR

CCPU_GetCounterMicros_ISR

This function obtains a 1μs counter value of C Controller module.

■Format

short CCPU_GetCounterMicros_ISR(unsigned long* pulMicros)

■Argument

Argument	Name	Description	IN/OUT
pulMicros	1μs counter value storage destination	Specify the storage destination of the 1μs counter value.	OUT

■Description

- This function obtains a 1μs counter value of C Controller module and stores the value in the 1μs counter value storage destination (pulMicros).
- The 1 μs counter value increases by 1 every 1 μs after the power is turned ON.
- The count cycles between 0 and 4294967295.

Restriction

Do not execute this function in a routine other than an interrupt routine.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

- Page 124 CCPU_GetCounterMillis_ISR

CCPU_GetCounterMillis_ISR

This function obtains a 1 ms counter value of C Controller module.

■Format

short CCPU_GetCounterMillis_ISR(unsigned long* pulMillis)

■Argument

Argument	Name	Description	IN/OUT
pulMillis	1 ms counter value storage destination	Specify the storage destination of the 1 ms counter value.	OUT

■Description

- This function obtains a 1 ms counter value of C Controller module and stores the value in the 1 ms counter value storage destination (pulMillis).
- The 1 ms counter value increases by 1 every 1 ms after the power is turned ON.
- The count cycles between 0 and 4294967295.

Restriction

Do not execute this function in a routine other than an interrupt routine.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

- Page 123 CCPU_GetCounterMicros_ISR

CCPU_GetDotMatrixLED_ISR

This function obtains the value displayed on the dot matrix LED of a C Controller module, and stores it to the LED data storage destination (pcData).

■Format

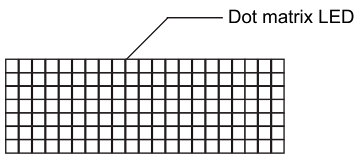
short CCPU_GetDotMatrixLED_ISR (char* pcData, unsigned long ulDataSize)

■Argument

Argument	Name	Description	IN/OUT
pcData	LED data storage destination	Specify the storage destination of LED data.	OUT
ulDataSize	LED data storage destination size	Specify the LED data storage destination size in byte units.	IN

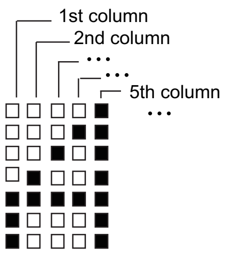
■Description

- This function obtains the value displayed on the dot matrix LED, and stores it in the LED data storage destination (pcData).
- It obtains the information for the size specified to the LED data storage destination size (ulDataSize).
- The value displayed on the dot matrix LED is stored in the LED data storage destination (pcData) as shown below.



pcData[0] to pcData[19]: Data of the dot matrix LED (7×20)
The value displayed in the following format is obtained.
Data format for each column: Bit pattern in which '0' is for the upper one bit, and '1' (when LED is ON) or '0' (when LED is OFF) is for lower seven bits

Ex.
The bit pattern shown below is displayed on the dot matrix LED:



1st column: 0000 0111b = 07H → pcData[0] = 0x07
2nd column: 0000 1100b = 0cH → pcData[1] = 0x0c
3rd column: 0001 0100b = 14H → pcData[2] = 0x14
4th column: 0010 0100b = 24H → pcData[3] = 0x24
5th column: 0111 1111b = 7fH → pcData[4] = 0x7f
6th column to 20th column: 0000 0000b = 00H → pcData[5] to pcData[19] = 0x00

Restriction

Do not execute this function in a routine other than an interrupt routine.

■⚠WARNING
If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function
- Page 131 CCPU_SetDotMatrixLED_ISR

CCPU_ReadDevice_ISR

This function reads data from internal user devices and internal system devices of C Controller module.

■Format

short CCPU_ReadDevice_ISR (short sDevType, unsigned long ulDevNo, unsigned long ulSize, unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
sDevType	Device type	Specify the device type. ☞ Page 9 Argument specification	IN
ulDevNo	Start device number	Specify the start device number. (Only multiples of 16 can be specified for bit devices.)	IN
ulSize	Data size	Specify the read data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of read data.	OUT

■Description

This function reads data equivalent to the size specified to the data size (ulSize) from a device specified to the device type (sDevType) and the start device number (ulDevNo) and subsequent devices, and stores the read data in the data storage destination (pusDataBuf).

Restriction

Do not execute this function in a routine other than an interrupt routine.

■⚠WARNING

- If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.
- This function does not check the specified argument.
  When creating a program, note the following:
  A data area for the size (words) of the read data is reserved.
  A device which is out of the range is not specified.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

- Page 136 CCPU_WriteDevice_ISR

CCPU_RegistEventLog_ISR

This function registers event logs in the event history of C Controller module.

■Format

short CCPU_RegistEventLog_ISR (long lEventCode, char* pcEventMsg)

■Argument

Argument	Name	Description	IN/OUT
lEventCode	Detailed code	Specify a detailed event code to be registered in the event history.	IN
pcEventMsg	Detailed information	Specify detailed information character string data of an event to be registered in the event history. (The detailed information character string data of an event can be specified up to 200 bytes. When 'NULL' is specified, the detailed information is not registered.)	IN

■Description

This function registers event logs in the event history of C Controller module.

The contents to be registered on the event history screen of CW Configurator are as follows:

Item	Description
Occurrence Date	Event registered date and time
Event Type	Operation (Fixed)
Status	Information (Fixed)
Event Code	25000 (Fixed)
Overview	Registration from the user program (Fixed)
Source	R12CCPU-V (Fixed)
Start I/O No.	Input/output number of the C Controller module that executed the CCPU_RegistEventLog_ISR function.
Detailed event code information	Detailed code (hexadecimal) specified to the detailed code (lEventCode)
Detailed event log information	Detailed information specified to the detailed information (pcEventMsg)
Cause	The event history was registered from the C Controller module detailed function. (Fixed)

- The event history can be stored for the size of the event history file specified with CW Configurator.
  Note that the data is deleted in order from older data if the specified file size is exceeded.

Restriction

Do not execute this function in a routine other than an interrupt routine.

■⚠WARNING

- If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.
- This function does not check the specified argument.
  When creating a program, note the following:
  Detailed information which is out of the range is not specified.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

CCPU_ResetDevice_ISR

This function resets internal user devices and internal system devices (bit devices) of C Controller module.

■Format

short CCPU_ResetDevice_ISR(short sDevType, unsigned long ulDevNo)

■Argument

Argument	Name	Description	IN/OUT
sDevType	Device type	Specify the device type. ☞ Page 9 Argument specification	IN
ulDevNo	Start device number	Specify the start device number.	IN

■Description

This function resets (turns OFF) a device of a C Controller module specified to the device type (sDevType) and the start device number (ulDevNo).

Restriction

Do not execute this function in a routine other than an interrupt routine.

■⚠WARNING

- If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.
- This function does not check the specified argument.
  When creating a program, note the following:
  A device which is out of the range is not specified.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

- Page 130 CCPU_SetDevice_ISR

CCPU_SetDevice_ISR

This function sets internal user devices and internal system devices (bit devices) of C Controller module.

■Format

short CCPU_SetDevice_ISR (short sDevType, unsigned long ulDevNo)

■Argument

Argument	Name	Description	IN/OUT
sDevType	Device type	Specify the device type. ☞ Page 9 Argument specification	IN
ulDevNo	Device number	Specify the device number.	IN

■Description

This function sets (turns ON) the device of a C Controller module specified to the device type (sDevType) and the start device number (ulDevNo).

Restriction

Do not execute this function in a routine other than an interrupt routine.

■⚠WARNING

- If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.
- This function does not check the specified argument.
  When creating a program, note the following:
  A device which is out of the range is not specified.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

- Page 129 CCPU_ResetDevice_ISR

CCPU_SetDotMatrixLED_ISR

This function sets a value to be displayed on the dot matrix LED of C Controller module.

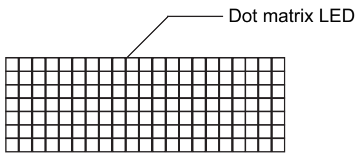
■Format

short CCPU_SetDotMatrixLED_ISR (unsigned short usLedMode, char* pcData)

■Argument

Argument	Name	Description	IN/OUT
usLedMode	Output mode	Unused (Even if a value is specified, the operation is not affected.)	IN
pcData	LED data	Specify the LED data.	IN

Specify the LED data (pcData) as follows.



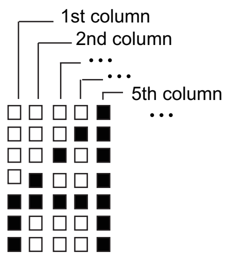
pcData[0] to pcData[19]: Data of the dot matrix LED (7×20)

The data specified in the following format is displayed.

Data format for each column: Bit pattern in which '0' is for the upper one bit, and '1' (when LED is ON) or '0' (when LED is OFF) is for lower seven bits

Ex.

The bit pattern shown below is output to the dot matrix LED:



1st column: 0000 0111b = 07H → pcData[0] = 0x07

2nd column: 0000 1100b = 0cH → pcData[1] = 0x0c

3rd column: 0001 0100b = 14H → pcData[2] = 0x14

4th column: 0010 0100b = 24H → pcData[3] = 0x24

5th column: 0111 1111b = 7fH → pcData[4] = 0x7f

6th column to 20th column: 0000 0000b = 00H → pcData[5] to pcData[19] = 0x00

■Description

This function displays the value specified to the LED data (pcData) on the dot matrix LED.

Restriction

- Do not execute this function in a routine other than an interrupt routine.
- Do not execute this function when a mode other than "USER" is selected in the operation selection mode. An unintended value may be displayed on the dot matrix LED.
- Do not execute this function while checking an operation or checking the selected operation with the MODE/ SELECT switch. An unintended value may be displayed on the dot matrix LED.

Precautions

When this function is executed, an image of the dot matrix LED in which data are being written may be displayed on the [Module Diagnostics (CPU diagnostics)] screen of CW Configurator.

■⚠WARNING
If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

- Page 125 CCPU_GetDotMatrixLED_ISR

CCPU_SetLEDStatus_ISR

This function sets the LED status of a C Controller module.

■Format

short CCPU_SetLEDStatus_ISR(long lLed, unsigned short usLedInfo)

■Argument

Argument	Name	Description	IN/OUT
lLed	Target LED	Unused (Even if a value is specified, the operation is not affected.)	IN
usLedInfo	LED status information	Specify the LED status information.	IN

The specification method of the LED status information (usLedInfo) is as follows:

usLedInfo	Description
0	OFF
1	ON (Red)
2	Flashing at low speed (Red)
3	Flashing at high speed (Red)
4	ON (Green)
5	Flashing at low speed (Green)
6	Flashing at high speed (Green)

■Description

This function controls the USER LED of C Controller module to the status specified to the LED status information (usLedInfo).

Restriction

Do not execute this function in a routine other than an interrupt routine.

■⚠WARNING

If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

- Page 94 CCPU_SetLEDStatus

CCPU_ToBuf_ISR

This function writes data to the CPU buffer memory of the CPU module (host CPU) and the buffer memory of the intelligent function module which are mounted on the specified module position. (TO instruction)

■Format

short CCPU_ToBuf_ISR (unsigned short usIoNo, unsigned long ulOffset, unsigned long ulSize, unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
usIoNo	Module position	Specify the module position as follows. For the CPU buffer memory, only the host CPU can be accessed. Start I/O number divided by 16 (0H to FFH, 3E0H to 3E3H)	IN
ulOffset	Offset	Specify the offset in word units.	IN
ulSize	Data size	Specify the write data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of write data.	IN

■Description

- This function writes data in the data storage destination (pusDataBuf) equivalent to the size specified to the data size (ulSize) to the CPU buffer memory of a CPU module (host CPU) and the buffer memory of an intelligent function module which are specified to the module position (usIoNo).
  Data is written by specifying an offset address (ulOffset) from the start of the CPU buffer memory of a CPU module (host CPU) and the buffer memory of an intelligent function module.
- To access the CPU buffer memory (host CPU) of the module in a multiple CPU system (CPU No.1 to No.4), specify 3E0H (CPU No.1) to 3E3H (CPU No.4) to the module position (usIoNo). However, the CPU buffer memory (host CPU) can be accessed only when the multiple CPU setting is configured.
- When executing this function while the operating status of a CPU module is not RUN, the STOP/PAUSE error (-28640) occurs.

Restriction

- Do not execute this function in a routine other than the one registered in the interrupt.
- When data is written to the same CPU buffer memory (host CPU) from a routine other than an interrupt routine, the output value may be overlapped, resulting in an invalid value. Manage the resource so that data is not written to the same CPU buffer memory (host CPU).

■⚠WARNING

- If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.
- This function does not check the specified argument.
  When creating a program, note the following:
  An address specified to the write data is a multiple of 2.
  A non-existent CPU buffer memory (host CPU) is not specified.
  A non-existent buffer memory is not specified.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 121 CCPU_FromBuf_ISR

CCPU_ToBufHG_ISR

This function writes data to the fixed cycle communication area of the CPU module mounted on the specified module position.

■Format

short CCPU_ToBufHG_ISR (unsigned short usIoNo, unsigned long ulOffset, unsigned long ulSize, unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
usIoNo	Module position	Specify the module position as follows. Start I/O number divided by 16 (3E0H to 3E3H)	IN
ulOffset	Offset	Specify the offset in word units.	IN
ulSize	Data size	Specify the write data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of write data.	IN

■Description

- This function writes data in the data storage destination (pusDataBuf) equivalent to the size specified to the data size (ulSize) to the fixed cycle communication area of a CPU module specified to the module position (usIoNo). Data is written by specifying an offset address (ulOffset) from the start of the fixed cycle communication area.
- The fixed cycle communication area can be accessed only when the fixed cycle communication area setting under the multiple CPU setting is configured.
- When executing this function while the operating status of a CPU module specified to the module position (usIoNo) is not RUN, the STOP/PAUSE error (-28640) occurs.

Restriction

- Do not execute this function in a routine other than an interrupt routine.
- When data is written to the same fixed cycle communication area from a routine other than an interrupt routine, the output value may be overlapped, resulting in an invalid value. Manage the resource so that data is not written to the same fixed cycle communication area.

■⚠WARNING

- If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.
- This function does not check the specified argument.
  When creating a program, note the following:
  An address specified to the write data is a multiple of 2.
  A non-existent fixed cycle communication area is not specified.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 121 CCPU_FromBuf_ISR
- Page 134 CCPU_ToBuf_ISR
- Page 122 CCPU_FromBufHG_ISR

CCPU_WriteDevice_ISR

This function writes data to internal user devices and internal system devices of C Controller module.

■Format

short CCPU_WriteDevice_ISR (short sDevType, unsigned long ulDevNo, unsigned long ulSize, unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
sDevType	Device type	Specify the device type. ☞ Page 9 Argument specification	IN
ulDevNo	Start device number	Specify the start device number. (Only multiples of 16 can be specified for bit devices.)	IN
ulSize	Data size	Specify the write data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of write data.	IN

■Description

- This function writes data in the data storage destination (pusDataBuf) equivalent to the size specified to the data size (ulSize) to a device specified to the device type (sDevType) and the start device number (ulDevNo) and subsequent devices.

Restriction

- Do not execute this function in a routine other than an interrupt routine.
- When data is written to the same device from a routine other than an interrupt routine, the output value may be overlapped, resulting in an invalid value. Manage the resource so that data is not written to the same device.

■⚠WARNING

- If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.
- This function does not check the specified argument.
  When creating a program, note the following:
  An address specified to the write data is a multiple of 2.
  A device which is out of the range is not specified.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

- Page 127 CCPU_ReadDevice_ISR

CCPU_X_In_Word_ISR

This function reads an input signal (X) in word (16-point) units.

■Format

short CCPU_X_In_Word_ISR (unsigned short usXNo, unsigned short usSize, unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
usXNo	Start input signal	Specify a start input signal (X). (Specify a multiple of 16.)	IN
usSize	Read data size	Specify the read data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of read data.	OUT

■Description

- This function operates to the mounted module corresponding to the specified start input signal (usXNo) regardless of the type of the parameter setting (I/O assignment). When the specified area is "empty" or "output module", this function ends normally without processing (read data: 0).
- The input status controlled by other CPUs is not imported.
  (The setting to import the out-group input status for the multiple CPU setting is ignored.)
- This function reads an input signal (X) equivalent to the size specified to the read data size (usSize) from the start input signal (X) specified to the start input signal (usXNo), and stores the read data in the data storage destination (pusDataBuf).
- Specify the start input signal (usXNo) in multiples of 16. (The remainder divided by 16 is discarded.)
- Read data is stored in the data storage destination (pusDataBuf) in ascending order from the lower bit as shown below.

pusDataBuf	Description
pusDataBuf[0]	Data of usXNo+FH to usXNo
pusDataBuf[0]	Data of usXNo+1FH to usXNo+10H
⋮	⋮
pusDataBuf[usSize-1]	Data of usXNo+(usSize-1)×16+FH to usXNo+(usSize-1)×16

Restriction

- Do not execute this function in a routine other than an interrupt routine.
- Do not execute this function to an I/O assignment on which an intelligent function module or an interrupt module is mounted.

■⚠WARNING

- If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.
- This function does not check the specified argument.
  When creating a program, note the following:
  An address specified to the read data is a multiple of 2.
  A data area for the size (words) of the read data is reserved.
  An input signal (X) which is out of the range (excluding 0H to FFFH) is not specified.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 138 CCPU_Y_In_Word_ISR
- Page 139 CCPU_Y_Out_Word_ISR

CCPU_Y_In_Word_ISR

This function reads an output signal (Y) in word (16-point) units.

■Format

short CCPU_Y_In_Word_ISR (unsigned short usYNo, unsigned short usSize, unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
usYNo	Start output signal	Specify a start output signal (Y). (Specify a multiple of 16.)	IN
usSize	Read data size	Specify the read data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of read data.	OUT

■Description

- This function operates to the mounted module corresponding to the specified output signal (usYNo) regardless of the type of the parameter setting (I/O assignment). When the specified area is "empty" or "input module", this function ends normally without processing (read data: 0).
- The input status controlled by other CPUs is not imported.
  (The setting to import the out-group input status for the multiple CPU setting is ignored.)
- This function reads an output signal (Y) equivalent to the size specified to the read data size (usSize) from the start output signal (Y) specified to the start output signal (usYNo), and stores the read data in the data storage destination (pusDataBuf).
- Specify the start output signal (usYNo) in multiples of 16. (The remainder divided by 16 is discarded.)
- Read data is stored in the data storage destination (pusDataBuf) in ascending order from the lower bit as shown below.

pusDataBuf	Description
pusDataBuf[0]	Data of usYNo+FH to usYNo
pusDataBuf[1]	Data of usYNo+1FH to usYNo+10H
⋮	⋮
pusDataBuf[usSize-1]	Data of usYNo+(usSize-1)×16+FH to usYNo+(usSize-1)×16

Restriction

- Do not execute this function in a routine other than an interrupt routine.
- Do not execute this function to an I/O assignment on which an intelligent function module or an interrupt module is mounted.

■⚠WARNING

- If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.
- This function does not check the specified argument.
  When creating a program, note the following:
  An address specified to the read data is a multiple of 2.
  A data area for the size (words) of the read data is reserved.
  An output signal (Y) which is out of the range (excluding 0H to FFFH) is not specified.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 137 CCPU_X_In_Word_ISR
- Page 139 CCPU_Y_Out_Word_ISR

CCPU_Y_Out_Word_ISR

This function outputs an output signal (Y) in word (16-point) units.

■Format

short CCPU_Y_Out_Word_ISR (unsigned short usYNo, unsigned short usSize, unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
usYNo	Start output signal	Specify a start output signal (Y). (Specify a multiple of 16.)	IN
usSize	Output size	Specify the output size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of output data.	IN

■Description

- This function outputs (turns ON/OFF) data in the data storage destination (pusDataBuf) from the start output signal (Y) specified to the start output signal (usYNo) to the output signal (Y) equivalent to the size specified to the data size (usSize).
- Specify the start output signal (usYNo) in multiples of 16. (The remainder divided by 16 is discarded.)
- Do not specify an output module controlled by other CPUs to the output signal (usYNo).
  If it is specified, no operation is performed to the output module.
- Store output data in the data storage destination (pusDataBuf) in ascending order from the lower bit as shown below.

pusDataBuf	Description
pusDataBuf[0]	Data of usYNo+FH to usYNo
pusDataBuf[1]	Data of usYNo+1FH to usYNo+10H
⋮	⋮
pusDataBuf[usSize-1]	Data of usYNo+(usSize-1)×16+FH to usYNo+(usSize-1)×16

Restriction

- Do not execute this function in a routine other than an interrupt routine.
- Do not execute this function to an I/O assignment on which an intelligent function module or an interrupt module is mounted.
- When data is output to the same output signal (Y) from a routine other than an interrupt routine, the output value may be overlapped, resulting in an invalid value. Manage the resource so that data is not output to the same output signal (Y).

■⚠WARNING

- If any function in which an invalid argument is specified is executed, an error such as hardware failure (3C02H) may occur on C Controller module.
- This function does not check the specified argument.
  When creating a program, note the following:
  An address specified to the write data is a multiple of 2.
  An output signal (Y) which is out of the range (excluding 0H to FFFH) is not specified.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant function

- Page 137 CCPU_X_In_Word_ISR
- Page 138 CCPU_Y_In_Word_ISR

3.2 MELSEC Data Link Functions

This section explains the details of the MELSEC data link function.

mdClose

This function closes a communication line (channel).

■Format

short mdClose(long lPath)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN

■Description

- This function closes the channel opened by the mdOpen function.
- When using multiple channels, close the channel one by one.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 148 mdOpen

mdControl

This function performs remote operations (remote RUN/STOP/PAUSE) for the CPU module.

■Format

short mdControl(long lPath, short sStNo, short sCode)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN
sStNo	Station number	Specify a network number and station number of the target module. ☞ Page 9 Argument specification	IN
sCode	Instruction code	Specify the contents of the remote operation in numerical value.	IN

The specification method of the instruction code (sCode) is as follows:

sCode (decimal)	Description
0	Remote RUN
1	Remote STOP
2	Remote PAUSE

■Description

This function changes the status of a CPU module specified to the station number (sStNo) to the one specified to the instruction code (sCode).

Restriction

This function cannot be executed for C Controller module, PC CPU module, and WinCPU module.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose

mdDevRstEx

This function resets bit devices.

■Format

long mdDevRstEx(long lPath, long lNetNo, long lStNo, long lDevType, long lDevNo)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN
lNetNo	Network number	Specify the network number of target module.	IN
lStNo	Station number	Specify the station number of target module. ☞ Page 9 Argument specification	IN
lDevType	Device type	Specify the device type of bit device. ☞ Page 9 Argument specification	IN
lDevNo	Device number	Specify the device number of bit device.	IN

■Description

- This function resets (turns OFF) the bit device of the module specified to the network number (lNetNo), the station number (lStNo), the device type (lDevType), and the device number (lDevNo).
- This function is dedicated for bit devices such as link relay (B), internal relay (M).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose
- Page 143 mdDevSetEx

mdDevSetEx

This function sets bit devices.

■Format

long mdDevSetEx (long lPath, long lNetNo, long lStNo, long lDevType, long lDevNo)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN
lNetNo	Network number	Specify the network number of target module.	IN
lStNo	Station number	Specify the station number of target module. ☞ Page 9 Argument specification	IN
lDevType	Device type	Specify the device type of bit device. ☞ Page 9 Argument specification	IN
lDevNo	Device number	Specify the device number of bit device.	IN

■Description

- This function sets (turns ON) the bit device of the module specified to the network number (lNetNo), the station number (lStNo), the device type (lDevType), and the device number (lDevNo).
- This function is dedicated for bit devices such as link relay (B), internal relay (M).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose
- Page 142 mdDevRstEx

mdGetLabelInfo

This function obtains device information corresponding to label names.

■Format

long mdGetLabelInfo (long lPath, long lNetNo, long lStNo, long lLbCnt, void* pLbLst, long* plDevLst, unsigned long long* pullLbCode)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN
lNetNo	Network number	Specify the network number of target module. ☞ Page 9 Argument specification	IN
lStNo	Station number	Specify the station number of target module. ☞ Page 9 Argument specification	IN
lLbCnt	Number of labels	Specify the number of labels. (Up to 10240) Number of labels can be specified up to 10240.	IN
pLbLst	Label name array	Specify the storage address of label name for each label. Specify a label name in Unicode (UTF-16).	IN
plDevLst	Device name array	Specify a device to store device information which is obtained. (Device information assigned to labels specified to the label name array (pLbLst) is stored in a randomly selected device format.)	OUT
pullLbCode	Label code	A value to identify whether the label of a CPU module is changed or not is stored. (Whether the label setting is changed or not can be checked by whether this value is changed or not. However, even when converting all in a CPU module, the value changes.)	OUT

Device information assigned to labels specified to the label name array (pLbLst) is stored in a device specified to the device name array (plDevLst) in a randomly selected device format listed below.

plDevLst	Description	
plDevLst[0]	Number of blocks	
plDevLst[1]	Device type	Block 1
plDevLst[2]	Start device number	
plDevLst[3]	Number of read points	
plDevLst[4]	Device type	Block 2
plDevLst[5]	Start device number	
plDevLst[6]	Number of read points	
⋮	⋮	⋮
plDevLst[3n+1]	Device type	Block n
plDevLst[3n+2]	Start device number	
plDevLst[3n+3]	Number of read points	

• One block comprises of three elements such as device type, start device number, and number of read points, and the total number of blocks will be stored in the first element of the device name array (plDevLst).

■Description

- This function reads labels of a CPU module specified to the network number (lNetNo) and the station number (lStNo).
- Reserve the area for the device name array (plDevLst) in the call source.
- Reserve the size of area equivalent to the size (lLbCnt × 3 + 1) for the device name array (plDevLst).
- If any of the labels of which the label information cannot be obtained exists in the label name specified to the label name array (pLbLst), this function returns any of the following errors. For the device type, start device number, and the number of read points of the label, '0' is stored.

Error code	Error occurrence
-82 (FFB2H)	• A non-existent label was specified. • Devices assigned to labels do not support random read/write. • The specification method for devices assigned to labels is incorrect.
-84 (FFB4H)	The specification method for devices assigned to labels is incorrect.

- The error response is returned in order of detection.
  If two labels (Label1: non-existent label name, Label2: incorrect device specification method by digit specification) are specified, only an error of Label1 (the first detected label) (-82) is returned.
- Even if the mdGetLabelInfo function returns the error (-82 or -84) the value is stored in the device name array (plDevLst) for the label that obtained device information successfully.
- The specification method of the label name to specify to the device name array (pLbLst) is as follows:

○: Possible, ×: Impossible

Label type	Specification possibility	Specification method	Specification example
Label of the simple data type	○	Specify the label name.	Label1
Element specification of the array label	○	Specify in the following format. • One-dimensional array: Label name [m] • Two-dimensional array: Label name [m, n] • Three-dimensional array: Label name [m, n, l]	• One-dimensional array: Label1 [10] • Two-dimensional array: Label2 [10, 20] • Three-dimensional array: Label3 [10, 20, 30]
Whole specification of the structure label	×	—	—
Member of the structure label	○	Specify in the following format. Label name.Element name. to Element name	Str1.Elem1. to Elem3
Array member of the structure label	○	Specify in the following format. Label name.Element name [m]	Str1.Elem[10]
Bit specification of label	×	—	—
Digit specification of label	×	—	—
Label of timer type, retentive timer type, and counter type	○	Specify in the following format. • Contact: Label name.S • Coil: Label name.C • Current value: Label name.N	• Contact: Label1.S • Coil: Label2.C • Current value: Label3.N

Precautions

- In CW Workbench, Unicode character strings cannot be entered and source codes including Unicode character strings cannot be compiled. Create a file with Unicode (UTF-16) character strings entered in an application (such as Notepad) of Windows.
- When a device is specified such as the bit specification of word device or the digit specification of label, the device information cannot be obtained.
- When a label to which a device is not assigned is specified, 'DevGV' is stored in the device type.
- The DevGV can be specified only by using the functions supporting label access (mdRandRLabelEx/mdRandWLabelEx).
- For CPU modules of which labels can be accessed, refer to the following manual.
  MELSEC iQ-R C Controller Module User's Manual (Application)

■Example

The following tables show the examples of values specified to the label name array (pLbLst) and data read to the device name array (plDevLst). (For five labels to be read: Label 1 to 5).

1. Describe a label name to use in a text file, and save it by specifying Unicode (UTF-16).

2. Read the label name in a binary format on a user program from the saved text file, and store the address of the label name to pass to the label name array (pLbLst) on the memory.

- Values specified to pLbLst

pLbLst	Specified value	Description
pLbLst[0]	First (Label1) label name storage address	Label name
pLbLst[1]	Second (Label2) label name storage address	Label name
pLbLst[2]	Third (Label3) label name storage address	Label name
pLbLst[3]	Fourth (Label4) label name storage address	Label name
pLbLst[4]	Fifth (Label5) label name storage address	Label name

- Value to be read to plDevLst

plDev	Value to be read	Description
plDev[0]	4	Number of blocks
plDev[1]	DevD	Device type
plDev[2]	10	Start device number
plDev[3]	1	Number of read points
plDev[4]	DevD	Device type
plDev[5]	11	Start device number
plDev[6]	1	Number of read points
plDev[7]	DevM	Device type
plDev[8]	100	Start device number
plDev[9]	1	Number of read points
plDev[10]	DevM	Device type
plDev[11]	101	Start device number
plDev[12]	1	Number of read points
plDev[13]	DevM	Device type
plDev[14]	102	Start device number
plDev[15]	1	Number of read points

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter.[*1] ☞ Page 168 ERROR CODE LIST

*1 For return values which does not exist in the reference, refer to the manual for the CPU module. (📖 MELSEC iQ-R CPU Module User's Manual (Application))

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose
- Page 152 mdRandRLabelEx
- Page 157 mdRandWLabelEx

mdInit

This function initializes communication route information.

■Format

short mdInit(long lPath)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN

■Description

This function clears communication route information using the path of the specified channel.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose

mdOpen

This function opens a communication line (channel).

■Format

short mdOpen(short sChan, short sMode, long* plPath)

■Argument

Argument	Name	Description	IN/OUT
sChan	Channel	Specify a communication line (channel). ☞ Page 9 Argument specification	IN
sMode	Mode	Specify '-1'.	IN
plPath	Path of channel	Specify the storage destination (address) of the path of the channel. (The path of the opened channel is stored.)	OUT

■Description

- When executing a MELSEC data link function, use the path of a channel opened by using this function.
- To end a user program, close the opened path of a channel by using the mdClose function.
- When using multiple channels, open the channel one by one.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 140 mdClose

mdRandREx

This function reads devices randomly.

■Format

long mdRandREx(long lPath, long lNetNo, long lStNo, long* plDev, short* psBuf, long lBufSize)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN
lNetNo	Network number	Specify the network number of target module. ☞ Page 9 Argument specification	IN
lStNo	Station number	Specify the station number of target module. ☞ Page 9 Argument specification	IN
plDev	Randomly selected device	Specify the number of blocks, device type, start device number, and device points of devices to be read.	IN
psBuf	Read data storage destination	Specify the storage destination (address) of read data.	OUT
lBufSize	Read data storage destination size	Specify the size of area allocated in the read data storage destination in byte units.	IN

The specification method of the randomly selected device (plDev) is as follows:

plDev	Description	
plDev[0]	Number of blocks	
plDev[1]	Device type	Block 1
plDev[2]	Start device number	
plDev[3]	Number of read points	
plDev[4]	Device type	Block 2
plDev[5]	Start device number	
plDev[6]	Number of read points	
⋮	⋮	⋮
plDevLst[3n+1]	Device type	Block n
plDevLst[3n+2]	Start device number	
plDevLst[3n+3]	Number of read points	

■Description

- This function reads devices specified to the randomly selected device (plDev) from the module specified to the network number (lNetNo) and the station number (lStNo).
- The read data is stored in the read data storage destination (psBuf) in word units in order of the specification to the randomly selected device (plDev). A bit device is stored per 16 points, a word device is stored per 1 point, and a double-word device is stored in word units.
- Specify so that the total number of read points specified by each block is 10240 points or less. If specified more than the maximum number, size error (-5) will occur.
- Communication time can very significantly depending on the contents specified to the randomly selected device (plDev). To reduce communication time, use the mdReceiveEx function.
- To access the own station, set the station number to 255. When the actual station number is used, an error will occur.

■Example

The following tables show the examples of values specified to the randomly selected device (plDev) and read to the read data storage destination (psBuf), and the number of read data bytes.

Device to be read randomly	Current value
M100 to M115	All bits are OFF.
D10 to D13	10 is stored to D10, 200 is stored to D11, 300 is stored to D12, and 400 is stored to D13.
M0 to M13	All bits are ON.
T10 current value	'10' is stored in T10.
LCN100 to LCN101	0x1 is stored to LCN100 and 0x10000 is stored to LCN101.

Values specified to the randomly selected device (plDev)

plDev	Specified value	Description	
plDev[0]	5	Number of blocks = 5	—
plDev[1]	DevM	Device type = M	Block 1: M100 to M115
plDev[2]	100	Start device number = 100	
plDev[3]	16	Number of read points = 16	
plDev[4]	DevD	Device type = D	Block 2: D10 to D13
plDev[5]	10	Start device number = 10	
plDev[6]	4	Number of read points = 4	
plDev[7]	DevM	Device type = M	Block 3: M0 to M13
plDev[8]	0	Start device number = 0	
plDev[9]	14	Number of read points = 14	
plDev[10]	DevTN	Device type = T	Block 4: T10
plDev[11]	10	Start device number = 10	
plDev[12]	1	Number of read points = 1	
plDev[13]	DevLCN	Device type = LCN	Block 5: LCN100 to LCN101
plDev[14]	100	Start device number = 100	
plDev[15]	2	Number of read points = 2	

Values to be read to the read data storage destination (psBuf)

psBuf	Read device	Value to be read	Description
psBuf[0]	M100 to M115	0	All the bit devices from M100 to M115 are OFF.
psBuf[1]	D10	10	D10 = 10
psBuf[2]	D11	200	D11 = 200
psBuf[3]	D12	300	D12 = 300
psBuf[4]	D13	400	D13 = 400
psBuf[5]	M0 to M13	3FFFH	All the bit devices from M0 to M13 are ON.
psBuf[6]	T10	10	T10=10
psBuf[7]	LCN100	0x1	Lower bit of LCN100 = 0x0001
psBuf[8]			Upper bit of LCN100 = 0x0000
psBuf[9]	LCN101	0x10000	Lower bit of LCN101 = 0x0000
psBuf[10]			Upper bit of LCN101 = 0x0001

Number of bytes of read data

(psBuf[0] to psBuf[10] = 11) × 2 = 22

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose
- Page 155 mdRandWEx

mdRandRLabelEx

Reads devices corresponding to labels randomly.

■Format

long mdRandRLabelEx(long lPath, long lNetNo, long lStNo, long* plDev, short* psBuf, long lBufSize, unsigned long long ullLbCode)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN
lNetNo	Network number	Specify the network number of target module. ☞ Page 9 Argument specification	IN
lStNo	Station number	Specify the station number of target module. ☞ Page 9 Argument specification	IN
plDev	Randomly selected device	Specify the number of blocks, device type, start device number, and device points of devices to be read. (Specify the value obtained by using the mdGetLabelInfo function.)	IN
psBuf	Read data storage destination	Specify the storage destination (address) of read data.	OUT
lBufSize	Read data storage destination size	Specify the area size reserved in the read data storage destination in byte units.	IN
ullLbCode	Label code	Specify the label code obtained by using the mdGetLabelInfo function.	IN

The specification method of the randomly selected device (plDev) is as follows:

plDev	Description	
plDev[0]	Number of blocks	
plDev[1]	Device type	Block 1
plDev[2]	Start device number	
plDev[3]	Number of read points	
plDev[4]	Device type	Block 2
plDev[5]	Start device number	
plDev[6]	Number of read points	
⋮	⋮	⋮
plDevLst[3n+1]	Device type	Block n
plDevLst[3n+2]	Start device number	
plDevLst[3n+3]	Number of read points	

- One block comprises of three elements such as device type, start device number, and number of read points, the total number of blocks will be stored in the first element of the randomly-specified device (plDev).

■Description

- This function reads devices specified to the randomly selected device (plDev) from the module specified to the network number (lNetNo) and the station number (lStNo).
- The read data is stored in the read data storage destination (psBuf) in word units in order of the specification to the randomly selected device (plDev). A bit device and a word device are stored per 1 point, and a double-word device is stored in word units.
- Specify so that the total number of read points specified by each block is 10240 points or less. If specified more than the maximum number, size error (-5) will occur.
- When '0' is specified to the label code (ullLbCode), the device is read without checking the label code.

■Example

The following tables show the examples of values specified to the randomly selected device (plDev) and read to the read data storage destination (psBuf), and the number of read data bytes.

Device to be read randomly	Current value
M100	Bit is OFF.
D10 to D13	10 is stored to D10, 200 is stored to D11, 300 is stored to D12, and 400 is stored to D13.
M0	Bit is ON.
T10 current value	'10' is stored in T10.
LCN100 to LCN101	0x1 is stored to LCN100 and 0x10000 is stored to LCN101.

Values specified to the randomly selected device (plDev)

plDev	Specified value	Description	
plDev[0]	5	Number of blocks = 5	—
plDev[1]	DevM	Device type = M	Block 1: M100
plDev[2]	100	Start device number = 100	
plDev[3]	1	Number of read points = 1	
plDev[4]	DevD	Device type = D	Block 2: D10 to D13
plDev[5]	10	Start device number = 10	
plDev[6]	4	Number of read points = 4	
plDev[7]	DevM	Device type = M	Block 3: M0
plDev[8]	0	Start device number = 0	
plDev[9]	1	Number of read points = 1	
plDev[10]	DevTN	Device type = T	Block 4: T10
plDev[11]	10	Start device number = 10	
plDev[12]	1	Number of read points = 1	
plDev[13]	DevLCN	Device type = LCN	Block 5: LCN100 to LCN101
plDev[14]	100	Start device number = 100	
plDev[15]	2	Number of read points = 2	

Values to be read to the read data storage destination (psBuf)

psBuf	Read device	Value to be read	Description
psBuf[0]	M100	0	M100 = OFF
psBuf[1]	D10	10	D10 = 10
psBuf[2]	D11	200	D11 = 200
psBuf[3]	D12	300	D12 = 300
psBuf[4]	D13	400	D13 = 400
psBuf[5]	M0	1	M0 = ON
psBuf[6]	T10	10	T10=10
psBuf[7]	LCN100	0x1	Lower bit of LCN100 = 0x0001
psBuf[8]			Upper bit of LCN100 = 0x0000
psBuf[9]	LCN101	0x10000	Lower bit of LCN101 = 0x0000
psBuf[10]			Upper bit of LCN101 = 0x0001

Number of bytes of read data
(psBuf[0] to psBuf[10] = 11) × 2 = 22

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter.[*1] ☞ Page 168 ERROR CODE LIST

*1 For return values which does not exist in the reference, refer to the following manual. (📖 MELSEC iQ-R CPU Module User's Manual (Application))

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose
- Page 144 mdGetLabelInfo
- Page 157 mdRandWLabelEx

mdRandWEx

This function writes devices randomly.

■Format

long mdRandWEx(long lPath, long lNetNo, long lStNo, long* plDev, short* psBuf, long lBufSize)

■Argument

Argument	Name	Description	IN/OUT
lPath	Channel	Specify the path of the channel.	IN
lNetNo	Network number	Specify the network number of target module. ☞ Page 9 Argument specification	IN
lStNo	Station number	Specify the station number of target module. ☞ Page 9 Argument specification	IN
plDev	Randomly selected device	Specify the number of blocks, device type, start device number, and device points of devices to be written.	IN
psBuf	Write data storage destination	Specify the storage destination (address) of write data. Allocate successive area to the write data storage destination.	IN
lBufSize	Write data storage destination size	Unused (Even if a value is specified, the operation is not affected.)	IN

The specification method of the randomly selected device (plDev) is as follows:

plDev	Description	
plDev[0]	Number of blocks	
plDev[1]	Device type	Block 1
plDev[2]	Start device number	
plDev[3]	Number of write points	
plDev[4]	Device type	Block 2
plDev[5]	Start device number	
plDev[6]	Number of write points	
⋮	⋮	⋮
plDev[3n+1]	Device type	Block n
plDev[3n+2]	Start device number	
plDev[3n+3]	Number of write points	

■Description

- This function writes data to the device, which is specified to the randomly selected device (plDev), of the module specified to the network number (lNetNo) and the station number (lStNo).
- The data to be written is stored to the write data storage destination (psBuf) in word units. A bit device is stored per 16 points, a word device is stored per 1 point, and a double-word device is stored in word units.
- Specify so that the total number of write points specified by each block is 10240 points or less. If specified more than the maximum number, size error (-5) will occur.
- Note that the extension comment information will be deleted when the data is written to the block to which an extension comment is assigned (extension file register).
- Also, note that sub 2 or sub 3 program will be deleted when data is written to a block (extension file register) overlapping with the program setting area for sub 2 or sub 3.

■Example

The following tables show the examples of values specified to the randomly selected device (plDev) and the write data storage destination (psBuf), and the number of write data bytes.

Device to be written randomly	Description
M100 to M115	Turns all the bits OFF.
D10 to D13	Stores 10 in D10, 200 in D11, 300 in D12, and 400 in D13.
LCN100 to LCN101	Stores 0x1 to LCN100, and 0x10000 to LCN101.

Values specified to the randomly selected device (plDev)

plDev	Specified value	Description	
plDev[0]	3	Number of blocks = 3	—
plDev[1]	DevM	Device type = M	Block 1: M100 to M115
plDev[2]	100	Start device number = 100	
plDev[3]	16	Number of write points = 16	
plDev[4]	DevD	Device type = D	Block 2: D10 to D13
plDev[5]	10	Start device number = 10	
plDev[6]	4	Number of write points = 4	
plDev[7]	DevLCN	Device type = LCN	Block 3: LCN100 to LCN101
plDev[8]	100	Start device number = 100	
plDev[9]	2	Number of write points = 2	

Values specified to the write data storage destination (psBuf)

psBuf	Specified value	Description
psBuf[0]	0	Turns all bit devices from M100 to M115 OFF.
psBuf[1]	10	D10 = 10
psBuf[2]	200	D11 = 200
psBuf[3]	300	D12 = 300
psBuf[4]	400	D13 = 400
psBuf[5]	0x0001	Lower bit of LCN100
psBuf[6]	0x0000	Upper bit of LCN100
psBuf[7]	0x0000	Lower bit of LCN101
psBuf[8]	0x0001	Upper bit of LCN101

Number of bytes of write data

(psBuf[0] to psBuf[8] = 9) × 2 = 18

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose
- Page 149 mdRandREx

mdRandWLabelEx

Writes devices corresponding to labels randomly.

■Format

long mdRandWLabelEx(long lPath, long lNetNo, long lStNo, long* plDev, short* psBuf, long lBufSize, unsigned long long ullLbCode)

■Argument

Argument	Name	Description	IN/OUT
lPath	Channel	Specify the path of the channel.	IN
lNetNo	Network number	Specify the network number of target module. ☞ Page 9 Argument specification	IN
lStNo	Station number	Specify the station number of target module. ☞ Page 9 Argument specification	IN
plDev	Randomly selected device	Specify the number of blocks, device type, start device number, and device points of devices to be written.	IN
psBuf	Write data storage destination	Specify the storage destination (address) of write data. Allocate successive area to the write data storage destination.	IN
lBufSize	Write data storage destination size	Unused (Even if a value is specified, the operation is not affected.)	IN
ullLbCode	Label code	Specify the label code obtained by using the mdGetLabelInfo function.	IN

The specification method of the randomly selected device (plDev) is as follows:

plDev	Description	
plDev[0]	Number of blocks	
plDev[1]	Device type	Block 1
plDev[2]	Start device number	
plDev[3]	Number of write points	
plDev[4]	Device type	Block 2
plDev[5]	Start device number	
plDev[6]	Number of write points	
⋮	⋮	⋮
plDev[3n+1]	Device type	Block n
plDev[3n+2]	Start device number	
plDev[3n+3]	Number of write points	

- One block comprises of three elements such as device type, start device number, and number of write points, the total number of blocks will be stored in the first element of the randomly-specified device (plDev).

■Description

- This function writes data to the device, which is specified to the randomly selected device (plDev), of the module specified to the network number (lNetNo) and the station number (lStNo).
- The data to be written is stored to the write data storage destination (psBuf) in word units. A bit device and a word device are stored per 1 point, and a double-word device is stored in word units.
- Specify so that the total number of write points specified by each block is 10240 points or less. If specified more than the maximum number, size error (-5) will occur.
- Note that the extension comment information will be deleted when the data is written to the block to which an extension comment is assigned (extension file register).
- Also, note that sub 2 or sub 3 program will be deleted when data is written to a block (extension file register) overlapping with the program setting area for sub 2 or sub 3.
- When '0' is specified to the label code (ullLbCode), the device is written without checking the label code.

■Example

The following tables show the examples of values specified to the randomly selected device (plDev) and the write data storage destination (psBuf), and the number of write data bytes.

Device to be written randomly	Description
M100	Turns the bit OFF.
D10 to D13	Stores 10 in D10, 200 in D11, 300 in D12, and 400 in D13.
LCN100 to LCN101	Stores 0x1 to LCN100, and 0x10000 to LCN101.

Values specified to the randomly selected device (plDev)

plDev	Specified value	Description	
plDev[0]	3	Number of blocks = 3	—
plDev[1]	DevM	Device type = M	Block 1: M100
plDev[2]	100	Start device number = 100	
plDev[3]	1	Number of write points = 1	
plDev[4]	DevD	Device type = D	Block 2: D10 to D13
plDev[5]	10	Start device number = 10	
plDev[6]	4	Number of write points = 4	
plDev[7]	DevLCN	Device type = LCN	Block 3: LCN100 to LCN101
plDev[8]	100	Start device number = 100	
plDev[9]	2	Number of write points = 2	

Values specified to the write data storage destination (psBuf)

psBuf	Specified value	Description
psBuf[0]	0	M100 = OFF
psBuf[1]	10	D10 = 10
psBuf[2]	200	D11 = 200
psBuf[3]	300	D12 = 300
psBuf[4]	400	D13 = 400
psBuf[5]	0x0001	Lower bit of LCN100
psBuf[6]	0x0000	Upper bit of LCN100
psBuf[7]	0x0000	Lower bit of LCN101
psBuf[8]	0x0001	Upper bit of LCN101

Number of bytes of write data

(psBuf[0] to psBuf[8] = 9) × 2 = 18

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter.[*1] ☞ Page 168 ERROR CODE LIST

*1 For return values which does not exist in the reference, refer to the following manual. (📖 MELSEC iQ-R CPU Module User's Manual (Application))

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose
- Page 144 mdGetLabelInfo
- Page 152 mdRandRLabelEx

mdReceiveEx

This function reads devices in batch.

■Format

long mdReceiveEx(long lPath, long lNetNo, long lStNo, long lDevType, long lDevNo, long* plSize, short* psData)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN
lNetNo	Network number	Specify the network number of target module. ☞ Page 9 Argument specification	IN
lStNo	Station number	Specify the station number of target module. ☞ Page 9 Argument specification	IN
lDevType	Device type	Specify the device type for device to be read in batch.	IN
lDevNo	Start device number	Specify the start device number for device to be read in batch. (For bit devices, specify the device number in multiples of 8.)	IN
plSize	Read data size	Specify the read data size in byte units. (Specify the value in multiples of 4 when a double-word device (LZ, LTN, LCN, LSTN) is specified, or specify the value in multiples of 2 when a word device or bit device is specified. If the value other than that is specified, the size error (-5) will occur.)	IN/OUT
psData	Read data storage destination	Specify the storage destination (address) of read data.	OUT

■Description

- This function reads data equivalent to the size specified to the read data size (plSize) from a device specified to the device type (lDevType) and the start device number (lDevNo) of a module specified to the network number (lNetNo) and the station number (lStNo).
- When the read data size exceeds the device range, a readable size is returned to the read data size (plSize).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose
- Page 162 mdSendEx (Device batch write function)

mdReceiveEx

This function receives messages. (RECV function)

■Format

long mdReceiveEx(long lPath, long lNetNo, long lStNo, long lDevType, long lDevNo, long* plSize, short* psData)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN
lNetNo	Network number	Specify '0' (0H).	IN
lStNo	Station number	Specify own station 255 (FFH).	IN
lDevType	Device type	Specify the device type for device to be read in batch. Only "RECV function: 101 (65H and DevMAIL)" is available.	IN
lDevNo	Channel number	Specify the channel number. • CC-Link IE Controller Network: 1 to 8 • CC-Link IE Field Network: 1 to 2 • MELSECNET/H network: 1 to 8	IN
plSize	Receive message size	Specify the receive message size in byte units. (2 to 1920) (Specify a size with an even number. If it is specified with an odd number, the size error (-5) will occur.)	IN/OUT
psData	Receive data storage destination	Specify the storage destination (address) of receive data. Data equivalent to 6 + a value specified to plSize (bytes) are specified.	OUT

■Description

- This function supports the RECV instruction, the dedicated instruction for a CC-Link IE Controller Network module, a CC-Link IE Field Network module, or a MELSECNET/H network module, when a value specified to the path of channel (lPath) is a value returned by using the mdOpen function by specifying CC-Link IE Controller Network (channel No.151 to 158), CC-Link IE Field Network (channel No.181 to 188), or MELSECNET/H network (channel No.51 to 54), and "RECV function: 101" is specified for the device type.
- It receives the message to the channel specified to the channel number (lDevNo) from among the messages sent to a CC-Link IE Controller Network module, a CC-Link IE Field Network module, or a MELSECNET/H network module.
- Confirm that the RECV execution request flag of Network module is ON before executing this function.
- For more advanced RECV function, use the C Controller module dedicated functions. This mdReceiveEx function reads the message to the specified channel number in the order it was received.
- When the actual size of a received message is smaller than the value specified to the receive message size (plSize), data of the actual size is stored in the receive data storage destination (psData[3] or higher), and the data size of the received message is returned to the receive message size (plSize).
- When the actual size of a received message is bigger than the value specified to the receive message size (plSize), data up to the specified size is stored in the receive data storage destination (psData[3] or higher).
- A received message is stored in the receive data storage destination (psData).
  Information on a message send source (network number, station number, and used channel of a send station) is stored in psData[0] to psData[2]. Therefore, the storage size of a received message is '6' + (plSize) byte.

psData	Description
psData[0]	Send station network number
psData[1]	Send station number
psData[2]	Channel used by send station
psData[3] or higher	• Received message (actual data) • (2 to 1920 bytes)

• The arguments of this function correspond to the control data (device) of the dedicated instruction (RECV) as shown below:

Device	Item	Corresponding argument and return value
+0	Error completion type	—
+1	Completion status	sRet
+2	Own station storage channel	lDevNo
+3	Channel used by send station	psData[2]
+4	Send station network number	psData[0]
+5	Send station number	psData[1]
+6	Not used	—
+7	Not used	—
+8	Arrival monitoring time	—
+9	Receive data length	plSize
+10	Not used	—
+11	Clock setting flag	—
+12	Clock data (Set only in an abnormal state.)	—
+13		—
+14		—
+15		—
+16	Error detection network number	—
+17	Error-detected station number	—

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose
- Page 163 mdSendEx (Message send function)
- Page 53 CCPU_DedicatedGInst
- Page 55 CCPU_DedicatedJInst

mdSendEx

This function writes devices in batch.

■Format

long mdSendEx(long lPath, long lNetNo, long lStNo, long lDevType, long lDevNo, long* plSize, short* psData)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN
lNetNo	Network number	Specify the network number of target module. ☞ Page 9 Argument specification	IN
lStNo	Station number	Specify the station number of target module. ☞ Page 9 Argument specification	IN
lDevType	Device type	Specify the device type for device to be written in batch.	IN
lDevNo	Start device number	Specify the start device number to be written in batch. (For bit devices, specify the device number in multiples of 8.)	IN
plSize	Write data size	Specify the write data size in byte units. (Specify the value in multiples of 4 when a double-word device (LZ, LTN, LCN, LSTN) is specified, or specify the value in multiples of 2 when a word device or bit device is specified. If the value other than that is specified, the size error (-5) will occur.)	IN/OUT
psData	Write data storage destination	Specify the storage destination (address) of write data. Allocate successive area to the write data storage destination.	IN

■Description

- This function writes data equivalent to the size specified to the write data size (plSize) from a device specified to the device type (lDevType) and the start device number (lDevNo) of a module specified to the network number (lNetNo) and the station number (lStNo).
- It checks the arguments and verifies whether the address + size determined by the arguments is within the device memory range.
- When the write data size exceeds the device range, a writable size is returned to the write data size (plSize).
- Note that the extension comment information will be deleted when the data is written to the block to which an extension comment is assigned (extension file register).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose
- Page 159 mdReceiveEx (Device batch read function)

mdSendEx

This function sends messages. (SEND function)

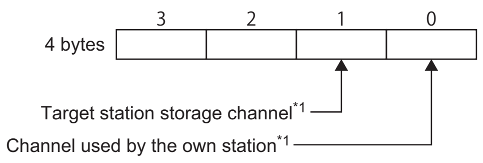
■Format

long mdSendEx(long lPath, long lNetNo, long lStNo, long lDevType, long lDevNo, long* plSize, short* psData)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN
lNetNo	Network number	Specify the network number of target module. A logical station number cannot be specified. ☞ Page 9 Argument specification	IN
lStNo	Station number	Specify the station number of target module. A logical station number cannot be specified. ☞ Page 9 Argument specification	IN
lDevType	Device type	Specify the device type for device to be written in batch. When "Group number" or "All stations" is specified for the station number, only "Without arrival confirmation" is valid. • With arrival confirmation: 101 (65H, DevMAIL) • Without arrival confirmation: 102 (66H, DevMAILNC)	IN
lDevNo	Channel number	Specify the channel number.	IN
plSize	Send data size	Specify the send data size in byte units. (2 to 1920) (Specify a size with an even number.)	IN/OUT
psData	Send data storage destination	Specify the storage destination (address) of send data. Allocate successive area to the send data storage destination.	IN

• Specify the channel number as follows.


3
2
1
0
4 bytes
Target station storage channel*1
Channel used by the own station*1

*1 CC-Link IE Controller Network: 1 to 8
CC-Link IE Field Network: 1 to 2
MELSECNET/H network: 1 to 8

■Description

- This function supports the SEND instruction, the dedicated instruction for a CC-Link IE Controller Network module, a CC-Link IE Field Network module, or a MELSECNET/H network module, when a value specified to the path of channel (lPath) is a value returned by using the mdOpen function by specifying CC-Link IE Controller Network (channel No.151 to 158), CC-Link IE Field Network (channel No.181 to 188), or MELSECNET/H network (channel No.51 to 54), and "With arrival confirmation: 101" or "Without arrival confirmation: 102" is specified for the device type.
- This function sends a message from a CC-Link IE Controller Network module, a CC-Link IE Field Network module, or a MELSECNET/H network module to the target (network number/station/channel) specified to the station number (lStNo) and the device type (lDevNo).
- For more advanced SEND functions, use the C Controller module dedicated functions.
- An error occurs if a message data is sent to a channel currently in use.

- The arguments of this function correspond to the control data (device) of the dedicated instruction (SEND) as shown below:

Device	Item	Corresponding argument and return value
+0	Execution/error completion type	lDevType
+1	Completion status	sRet
+2	Own station channel	lDevNo
+3	Target station storage channel	lDevNo
+4	Target station network number	lNetNo
+5	Target station number	lStNo
+6	Not used	—
+7	Number of retransmission (retry)	—
+8	Arrival monitoring time	—
+9	Send data length	plSize
+10	Not used	—
+11	Clock setting flag	—
+12	Clock data	—
+13		—
+14		—
+15		—
+16	Error detection network number	—
+17	Error-detected station number	—

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose
- Page 160 mdReceiveEx (Message receive function)
- Page 53 CCPU_DedicatedGInst
- Page 55 CCPU_DedicatedJInst

mdTypeRead

This function reads the model code of a CPU module.

■Format

short mdTypeRead(long lPath, short sStNo, short* psCode)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN
sStNo	Station number	Specify the network number and station number of the target module. ☞ Page 9 Argument specification	IN
psCode	Model code	Specify the storage destination (address) of the model code. (Stores the read model code.)	OUT

■Description

This function reads the model name of the CPU module with the specified station number to the station number (sStNo).

For CPU modules other than the following, the model code is undefined.

Model code (hexadecimal)	CPU module
0041H	Q02CPU, Q02HCPU
0042H	Q06HCPU
0043H	Q12HCPU
0044H	Q25HCPU
0049H	Q12PHCPU
004AH	Q25PHCPU
004BH	Q12PRHCPU
004CH	Q25PRHCPU
004DH	Q02PHCPU
004EH	Q06PHCPU
0250H	Q00JCPU
0251H	Q00CPU
0252H	Q01CPU
0260H	Q00UJCPU
0261H	Q00UCPU
0262H	Q01UCPU
0263H	Q02UCPU
0266H	Q10UDHCPU
0267H	Q20UDHCPU
0268H	Q03UDCPU
0269H	Q04UDHCPU
026AH	Q06UDHCPU
026BH	Q13UDHCPU
026CH	Q26UDHCPU
02E6H	Q10UDEHCPU
02E7H	Q20UDEHCPU
02E8H	Q03UDECPU
02E9H	Q04UDEHCPU
02EAH	Q06UDEHCPU
02EBH	Q13UDEHCPU
02ECH	Q26UDEHCPU
02EDH	Q50UDEHCPU
02EEH	Q100UDEHCPU
0366H	Q03UDVCPU
0367H	Q04UDVCPU
0368H	Q06UDVCPU

Model code (hexadecimal)	CPU module
036AH	Q13UDVCPU
036CH	Q26UDVCPU
0541H	L02CPU
0543H	L02SCPU
0544H	L06CPU
0545H	L26CPU
0548H	L26CPU-BT
0549H	L02CPU-P
054AH	L26CPU-PBT
0641H	LJ72GF15-T2
0642H	NZ2GF-ETB
2014H	Q172DCPU(-S1)
2015H	Q173DCPU(-S1)
2018H	Q172DSCPU
2019H	Q173DSCPU
2043H	Q12DCCPU-V
2044H	Q24DHCCPU-V
2045H	Q24DHCCPU-LS
2046H	Q24DHCCPU-VG
2047H	Q26DHCCPU-LS
4800H	R04CPU
4801H	R08CPU
4802H	R16CPU
4803H	R32CPU
4804H	R120CPU
4805H	R04ENCPU
4806H	R08ENCPU
4807H	R16ENCPU
4808H	R32ENCPU
4809H	R120ENCPU
4820H	R12CCPU-V
4C00H	R16MTCPU
4C01H	R32MTCPU
4C02H	R64MTCPU
4841H	R08PCPU
4842H	R16PCPU
4843H	R32PCPU
4844H	R120PCPU
4891H	R08SFCPU
4892H	R16SFCPU
4893H	R32SFCPU
4894H	R120SFCPU

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details on the error, refer to the following chapter. ☞ Page 168 ERROR CODE LIST

■Relevant functions

- Page 148 mdOpen
- Page 140 mdClose

4 ERROR CODE LIST

This chapter shows the codes for errors occurred in the dedicated function library and the corrective actions.

4.1 Common Error Codes

The following table shows the common error codes for the dedicated function library.

Error code*1		Description	Corrective action
Decimal	Hexadecimal		
1	0001H	■Driver not started The driver is not started.	• Check the channel number. • Correct the error that occurred when the driver is started. • Check the status of the system drive of the C Controller module. • Check if the operating system is running normally.
2	0002H	■Timeout error • A timeout occurred while waiting for the response. • During CC-Link communication, the request was issued to other stations when the station number of the own station is '64'. • The module specified as the communication target is not supported.	• Review the operating status and mounting condition of the accessed station. • Retry on the user program. • Increase the timeout value of MELSEC data link function. • When issuing a request to other stations during CC-Link communication, set the station number of the own station other than '64'. • Check if the module specified as the communication target is supported.
66	0042H	■Already opened error The specified channel has already been opened.	The open processing is required only one time. (If this error occurred, the path of the correct channel will be returned to the argument.)
67	0043H	■Already closed error The specified channel has already been closed.	The close processing is required only one time.
69	0045H	■Unsupported function performing error An unsupported function in the target station was performed.	• Check the path of the channel, network number, and station number. • Check if the function performed in the target station is supported.
70	0046H	■Station number error • The specified station number is incorrect. • The request for other stations was issued to the own station, or the network number was not '0' even though the station number was the own station (FFH).	Correct the network number and station number specified in the user program.
77	004DH	■Memory reservation error ■Resource shortage error ■Task over error Reserving sufficient memory failed, or there are too many tasks using the dedicated function library.	• The memory may be insufficient. End another running task or reduce the access size. • Reduce the number of tasks using the dedicated function library and retry the operation. • Review the size or number specified to the arguments of the user program. • Check if the C Controller module is running normally. • Reset or power OFF to ON the C Controller module.
85	0055H	■Network channel number error (When a SEND/RECV request is issued.) Channel number error	Check the specified channel number when the SEND/RECV request is issued.
102	0066H	■Data send error ■Restart error Sending data failed, or an attempt was made to send data during restart.	• Retry. • Retry after completion of the restart. • Check if the C Controller module is running normally. • Reset the C Controller system.
103	0067H	■Reception error Receiving data failed.	• Retry. • Check if the C Controller module is running normally. • Reset or power OFF to ON the C Controller module.
130	0082H	■Device number error • The specified device number is out of range. • The specified bit device number is not a multiple of 8.	Check the device number.
131	0083H	■Number of device points error • The specified number of device points is out of range. • The specified number of bit device points is not a multiple of 8.	Check the specified number of device points.

Error code[*1]		Description	Corrective action
Decimal	Hexadecimal		
16384 to 20479	4000H to 4FFFH	■Errors detected in the access target CPU module	Refer to the user's manual of the access target CPU module.
-25056	9E20H	■Processing code error A request which cannot be performed in the request destination was issued.	Check the network number and station number of the request destination.
-26334	9922H	■Reset error • Another task using the same channel was reset while accessing another station. • Reset operation was performed while monitoring with CW Configurator.	• Retry. • Monitor again.
-26336	9920H	■Routing request error for unsupported station A routing request to another loop was issued to a station which does not support the routing function.	Check the settings of routing parameters.
-28150	920AH	■Device access error during data link stop The devices (RX, RY, RWw, and RWr) of the own station were accessed when the data link was not performed.	• Check the specified device start number and size, or the device range of the parameter for the master station. • Restart the date link. (Note that data is written/read despite this error, however, the contents of the data will not be guaranteed.)
-28151	9209H	■Abnormal data reception error Abnormal response data received.	Check if an error occurred in the request destination CPU module or link module. (If the status is normal, try again.)
-28158	9202H	■WDT error WDT (system/user) error occurred.	Reset or power OFF to ON the C Controller module.
-28410	9106H	■Target CPU busy error The target CPU module is busy.	• Add the processing to wait for the completion of the target operation or to retry the operation in the user program. • Increase the timeout time specified to the argument in the user program.
-28412	9104H	■Target CPU unsupported error An unsupported request was issued to the target CPU module.	Change the target CPU module specified in the user program.
-28413	9103H	■Target CPU down error The target CPU module is down.	Check the operating status of the target CPU and troubleshoot the error according to the user's manual of the CPU module.
-28414	9102H	■Target CPU abnormal start error A request was issued to the CPU module which is not operating normally.	Check the operating status of the target CPU and troubleshoot the error according to the user's manual of the CPU module.
-28415	9101H	■Target CPU major error A request was issued to the CPU module in which a major error occurred.	Check the operating status of the target CPU and troubleshoot the error according to the user's manual of the CPU module.
-28416	9100H	■Target CPU mounting error A request was issued by specifying the CPU number in the state where no CPU module is mounted.	• Check the mounting condition of the target CPU module. • Change the target CPU number specified in the user program.
-28622	9032H	■Target module busy error • The target module is busy. • The own station channel or the target station storage channel is used for other instructions, or multiple identical instructions are being executed.	Add the processing to wait for the completion of the target operation or to retry the operation in the user program.
-28624	9030H	■Function unsupported error • Any processing was performed to the module that does not support the block guarantee function of cyclic data per station. • Any processing was performed to the module that is not set the block guarantee function of cyclic data per station. • An attempt was made to access a module which was not controlled by the host CPU module.	• Check if the target module, CC-Link module, supports the block guarantee function of cyclic data per station. • Check if the block guarantee function of cyclic data per station is set for the target module. • Check if the control CPU of the target module is the host CPU module.
-28625	902FH	■Intelligent function module offline error An attempt was made to access the intelligent function module when it was offline.	Check the status of the intelligent function module and access the module while it is online.
-28626	902EH	■Control data setting value out of range error The specified control data is out of range.	Check the value set to the control data.
-28627	902DH	■Transient unsupported error Transient transmission cannot be performed via the specified communication route and target. (Another station was specified when the station number of the own station is '64' during CC-Link communication.)	• Check the communication route and target which support the transient request. • Change the station number of the own station.

Error code[*1]		Description	Corrective action
Decimal	Hexadecimal		
-28628	902CH	■Pointer address specification error An incorrect address was specified to the argument pointer.	Check the address of the specified pointer.
-28629	902BH	■WDT not started error An attempt was made to reset a WDT before starting it.	Reset the WDT after starting it.
-28630	902AH	■WDT startup error An attempt was made to start WDT while the other WDT is starting up.	Start the WDT after stopping the WDT which is starting up.
-28631	9029H	■Buffer access range error • The specified offset is out of range. • The specified offset and its size is out of range.	• Check the specified offset. • Check the specified buffer size. • Check the offset and its size.
-28632	9028H	■I/O number error • The specified I/O number is out of range. • No accessible module is mounted on the specified I/O number.	Check the specified I/O number.
-28633	(9027H)	■Non-controlled module read error An attempt was made to access a module which is not controlled by the host CPU when reading data from a non-controlled module is not allowed.	Check if the control CPU of the specified module is the host CPU module (C Controller module/PC CPU module/WinCPU module).
-28634	9026H	■Intelligent function module down error The intelligent function module has an error.	• Check the mounting condition of the target CPU module. • Replace the intelligent function module or base unit.
-28635	9025H	■Intelligent function module error No intelligent function module is mounted on the accessed slot with the specified I/O number.	• Check the specified I/O number and the slot. • Check the mounting condition of the target CPU module.
-28636	9024H	■Control bus error The control bus to the intelligent function module has an error.	• Check if an error occurs in the bus master CPU (CPU No.1) in the multiple CPU system. • Check the mounting condition of the target CPU module. • Replace the intelligent function module or base unit.
-28638	9022H	■Multiple CPU unsupported operation error	Reset the bus master CPU (CPU No.1).
-28640	9020H	■STOP/PAUSE error The request of output or of writing to the buffer memory is issued when the operating status of the CPU module is STOP or PAUSE.	Change the operation status of the CPU module to RUN.
-28653	9013H	■I/O assignment error • An attempt was made to read the input value (X) from an output module. • An attempt was made to write the output value (Y) to an input module. • An attempt was made to read the output value (Y) from an input module.	Check the input number (X) and output number (Y).
-28654	9012H	■Non-controlled module write error An attempt was made to access a module which is not controlled by the host CPU.	Check if the control CPU of the specified module is the host CPU module (C Controller module/PC CPU module/WinCPU module).
-28660	900CH	■Access size error The specified size is out of range.	Review the specified offset and size.
-28661	900BH	■Inaccessible error Inaccessible area was specified.	Review the specified offset and size.
-28662	900AH	■CPU number specification error The specified CPU number is out of range or unavailable.	• Review the specified CPU number. • Check the operating status of the specified CPU module.
-28663	9009H	■Base unit number specification error The specified base unit number is out of range.	Review the specified base unit number.
-28664	9008H	■Data send area occupied	Retry.
-28665	9007H	■No registration data error	Reset or power OFF to ON the C Controller module.
-28666	9006H	■Data length error	Reset or power OFF to ON the C Controller module.
-28668	9004H	■Reply data stored error	Resend the request.
-28669	9003H	■Area number error The specified area number, offset address, and mode are out of range.	Review the area number, offset address, and mode.
-28671	9001H	■Module identification error	• Review the parameters. • Check the specified module. • Reset or power OFF to ON the C Controller module.
-28672	9000H	■Processing code error	Reset or power OFF to ON the C Controller module.

*1 When the function of which the return value is a long-type, the value will be eight digits in hexadecimal.

4.2 C Controller Module Dedicated Functions

The following table shows the error codes of the C Controller module dedicated functions.

Error code		Description	Corrective action
Decimal	Hexadecimal		
-201	FF37H	■Module identification error The specified module identification is unavailable.	Check the specified module identification.
-203	FF35H	■I/O number error The specified I/O signal is out of range.	Check the specified I/O signal.
-204	FF34H	■I/O access size error The specified access size of I/O signal is out of range.	Check the specified access size of I/O signal (I/O number and read/write size in words).
-205	FF33H	■I/O number error The specified I/O number is out of range.	Check the specified I/O number.
-206	FF32H	■Program execution type error The specified execution type of the program is out of range.	Check the specified execution type of the program.
-208	FF30H	■Offset error • The specified offset is out of range. • An AnS series module (buffer memory) was accessed.	• Check the specified offset. • Check the specified I/O number.
-209	FF2FH	■Buffer memory size error • The specified offset and its size is out of range. • The address of data storage buffer pointer is 0. • The specified size is 0.	• Check the specified buffer memory size. • Check the offset and its size. • Check the specified data storage buffer pointer.
-210	FF2EH	■Read area size error The read area size is smaller than the read size.	• Check the read size. • Check the read area size.
-211	FF2DH	■Time setting error The specified time is out of range.	Check the specified time.
-214	FF2AH	■Intelligent function module error A slot with the specified I/O number was accessed in the state where no intelligent module was mounted.	• Check the specified I/O number and the slot. • Check the mounting condition of the target CPU module.
-217	FF27H	■Driver not started The driver is not started.	Check if the driver is started.
-219	FF25H	■Program name error The specified program name is unavailable. (e.g. the specified program does not exist or the parameters are not registered.)	Check the specified program name.
-220	FF24H	■WDT type error The specified WDT type is out of range.	Check the specified WDT type.
-222	FF22H	■Bus master CPU reset error Remote reset of the bus master CPU (CPU No.1) failed.	• Enable the setting to allow remote reset (set "Remote Reset" to "Enable") for the bus master CPU (CPU No.1). • Change the operating status of the bus master CPU (CPU No.1) to STOP. • Check if the bus master CPU (CPU No.1) is a programmable controller CPU or C Controller module.
-223	FF21H	■Memory reservation error Reserving sufficient memory failed.	Check if sufficient memory is available.
-224	FF20H	■LED setting value error The specified LED setting value is out of range.	Check the specified LED setting value.
-225	FF1FH	■Event number specification error The specified event number is out of range or duplicated.	Check the specified event number.
-227	FF1DH	■Control code send error Sending control code failed.	• Retry. • Check if the C Controller module is running normally. • Reset the C Controller system.
-231	FF19H	■Event timeout error A timeout occurred while waiting for an event.	• Increase the timeout time. • Check if the interrupt event number (interrupt pointer number) is set correctly.
-232	FF18H	■CPU number specification error • The specified CPU number is incorrect. • The specified CPU cannot issue the request. • The host CPU module in which the remote operation is performed is specified by a number for specifying other stations (1 to 4).	• Change the specified CPU number. • Do not issue a request, that generated an error, to the specified CPU. • Specify '0' (host CPU) to perform the remote operation on the host CPU module.

Error code		Description	Corrective action
Decimal	Hexadecimal		
-234	FF16H	■Event wait error An error other than timeout occurred while the function waits for the event.	• Check if a program is forcibly being terminated. • Check if the C Controller module is running normally. • Reset or power OFF to ON the C Controller module.
-235	FF15H	■Number of event settings specification error The specified number of event settings is out of range.	Check the number of specified event settings.
-236	FF14H	■Remote operation specification code error The remote operation specification code is out of range.	Check the specified remote operation specification code.
-237	FF13H	■Detailed information character string specification error The length of the specified character string was out of range or characters which cannot be specified was specified.	Correct the length of the specified character string or character string data.
		■Application code specification error Five or more digits of the hexadecimal number was specified for the application code.	Change the specified application code.
-238	FF12H	■Event log registration error Registering an event log failed.	Reset or power OFF to ON the C Controller module.
-239	FF11H	■Drive insertion error Either of the following functions executed with no drive inserted. • CCPU_UnmountMemoryCard • CCPU_mountMemoryCard	Check if a drive is inserted.
-240	FF10H	■Clock data incorrect error The clock data to be set or the read clock data is incorrect.	• Check the clock data to be set. • If this error occurs when reading the clock data, set the data again.
-241	FF0FH	■Cycle specification error • The specified cycle is out of range. • The cycle was set even when it had already been set.	• Check the specified cycle. • Check if the cycle has been already set.
-242	FF0EH	■Synchronization type specification error The specified synchronization type is out of range.	Check the specified synchronization type.
-245	FF0BH	■Not executable during interrupt service routine A function was executed from interrupt service routine without specifying '1' (ISR) to the call source flag.	Specify '1' to the call source flag (ISR) and execute the function again.
-246	FF0AH	■Timer event registration error Registering a timer event failed.	• Retry. • Check if the C Controller module is running normally. • Reset or power OFF to ON the C Controller module.
-247	FF09H	■Program number specification error The specified program number is out of range or unavailable. • SFC program number (0 to 255) • Servo program number (0 to 4095)	Correct the specified program number.
-248	FF08H	■Number of starting axes specification error 9 or more starting axes were specified.	Correct the specified number of starting axes.
-249	FF07H	■Axis type specification error An axis type other than the axis/encoder axis/cam axis was specified.	Correct the specified axis type.
-250	FF06H	■Axis number specification error The specified axis number is out of the available range.	Correct the specified axis number.
-252	FF04H	■Torque limit value specification error The specified torque limit value is out of the available range.	Correct the specified torque limit value.
-253	FF03H	■Device number specification error • The specified device number is out of range. • The specified bit device number is not a multiple of 16.	Correct the start device number of the specified device.
-254	FF02H	■Device type specification error The specified device type is unavailable.	Correct the specified device type.
-255	FF01H	■Size specification error • The specified number of words is out of range. • The specified size is 0.	Correct the specified start device number and number of words.
-256	FF00H	■Response completion wait timeout error A timeout occurred while waiting for completion of a response of a processing requested to other CPU modules.	• Increase the timeout time specified to the argument. • Review and correct the user program (including other tasks which execute motion CPU interaction functions). • Review the program used for the request destination CPU module and correct it to perform the processing requested from other CPU modules, for example, by adding the WAIT instruction.

Error code		Description	Corrective action
Decimal	Hexadecimal		
-257	FEFFH	■Interrupt event type specification error The value specified to the interrupt event type is out of range.	Check the specified value.
-258	FEFEH	■Interrupt pointer number specification error The value specified as the interrupt pointer number is out of range.	Check the specified value.
-259	FEFDH	■Interrupt service routine unregistered error • The processing was not registered when enabling the processing corresponding to an event (interrupt). • The specified CPU number is incorrect.	• Register the processing for the event (interrupt) and perform the operation again. • Check the specified CPU number.
-260	FEFCH	■Drive mount error ■Drive unmount error The mount processing or unmount processing of the drive failed.	• Retry. • Check if the drive is damaged. • Replace the drive.
-263	FEF9H	■Caller flag error The value specified to the caller flag is out of range.	Review the specified value, and specify a value within the range.
-264	FEF8H	■Pointer error The address of the specified pointer is incorrect.	Check the address of the specified pointer.
-265	FEF7H	■Target system specification error The value specified in the target system is out of range.	Check the specified value.
-266	FEF6H	■WDT start error The user WDT is being started.	Check the user WDT settings.
-267	FEF5H	■Authentication error The password is incorrect.	Check the specified password.
-268	FEF4H	■Security error The setting content of the security function is incorrect.	Check the settings of the specified security function.
-269	FEF3H	■Network number error The specified network number is out of range.	Check the specified network number.
-270	FEF2H	■Channel number error The specified channel number is out of range.	Check the specified channel number.
-271	FEF1H	■Target station number error The specified target station number is out of range.	Check the specified target station number.
-279	FEE9H	■File specification error • The specified file does not exist. • A file with the same name already exists. (The existing file is overwrite-protected.) • A file cannot be created in the specified path, or the specified path does not exist.	• Check the specified file. • Check if the existing file is overwrite-protected. • Check if no files exist with the same name as the file to be created.
-280	FEE8H	■File access error The specified file is in use.	Check if the specified file is in use.
-281	FEE7H	■Instruction name error The instruction name is incorrect.	Check the specified instruction name.
-282	FEE6H	■Mode information error The specified mode information is out of range.	Check the specified mode information.
-283	FEE5H	■Operation selection mode The specified operation selection mode is out of range.	Check the specified operation selection mode.
-288	FEE0H	■Individual identification information read error Reading individual identification information failed.	• Check if the C Controller module is running normally. • Reset or power OFF to ON the C Controller module.
-289	FEDFH	■Dot matrix LED mode selection error An operation other than "USER" was selected by using the CCPU_SetOpSelectMode function or with the MODE/SELECT switch.	Select "USER" by using the CCPU_SetOpSelectMode function or with the MODE/SELECT switch.
-290	FEDEH	■MODE is being selected by switch operation The CCPU_SetOpSelectMode function or the CCPU_SetDotMatrixLED function was executed while an operation was being selected with the MODE/SELECT switch.	Execute the function after selecting an operation.
-291	FEDDH	■Fixed cycle communication area unreserved error An attempt was made to access a CPU module in which fixed cycle communication area is not reserved.	Use the fixed cycle communication function in the multiple CPU setting of system parameter, and check if 1K word or more is set for the fixed cycle communication area.

Error code		Description	Corrective action
Decimal	Hexadecimal		
-292	FEDCH	■Program memory shutdown error The shutdown processing of the program memory failed.	• Check if files in the program memory are being accessed. • Check if all files in the program memory have been closed.
-293	FEDBH	■Data memory shutdown error The shutdown processing of the data memory failed.	• Check if files in the data memory are being accessed. • Check if all files in the data memory have been closed.
-295	FED9H	■Selected operation is being checked The CCPU_SetDotMatrixLED function was executed while checking the selected operation.	Execute the CCPU_SetDotMatrixLED function after checking the operation.
-296	FED8H	■Setting data size error The setting data size is out of range.	Check the setting data size.
-297	FED7H	■Input/output number, network number incorrect specification An input/output number out of the range (other than 000H to FFFH or 3E0H to 3E3H) was specified.	Correct the argument of the function.
-298	FED6H	■Input/output number, network number incorrect specification An input/output number with no module was specified.	
-299	FED5H	■Input/output number, network number incorrect specification • An input/output number of a module which does not support the function was specified. • The dedicated instruction was specified with the specified module or mode.	Check the applicability of the dedicated function (such as the support status and executable mode) by referring to the manual for relevant modules.
-300	FED4H	■Input/output number, network number incorrect specification An input/output number of a module, that cannot be specified, was specified.	Correct the argument of the function.
-301	FED3H	■Input/output number, network number incorrect specification A network number out of the range (other than 1 to 239) was specified.	
-302	FED2H	■Input/output number, network number incorrect specification A network number which does not exist was specified.	
-303	FED1H	■Input/output number, network number incorrect specification An I/O module or intelligent function module controlled by other CPU modules was specified.	• Correct the argument of the function. • Delete a module controlled by other CPU modules, which has been specified by the link direct device, from the program. • Specify the network module controlled by the host CPU module with a link direct device.
-304	FED0H	■Input/output number, network number incorrect specification The target module cannot be identified by using a function to specify an I/O module or intelligent function module. (The character strings to specify the target module are incorrect.)	Correct the argument of the function.
-305	FECFH	■Input/output number, network number incorrect specification The specified I/O module or intelligent function module is in the state where it cannot execute the function.	The possible cause is a hardware failure of the I/O module or intelligent function module specified. Please consult your local Mitsubishi representative.
-306	FECEH	■Device, buffer memory incorrect specification The specified device exceeded the usable range.	Correct the argument of the function.
-307	FECDH	■Device, buffer memory incorrect specification A device that cannot be specified was specified.	
-308	FECCH	■Program fault The specified argument structure is incorrect.	
-309	FECBH	■Program fault The specified number of devices is incorrect.	
-310	FECAH	■Operation error An unusable character string for a function is specified.	
-311	FEC9H	■Operation error Data out of the specifiable range was entered.	

Error code		Description	Corrective action
Decimal	Hexadecimal		
-312	FEC8H	■Operation error The CCPU_DedicatedDInst function was executed in the state where the fixed cycle communication function is set to "Not Use" in the multiple CPU setting of system parameters.	Change the fixed cycle communication function in the multiple CPU setting to "Use".
-313	FEC7H	■Operation error In a multiple CPU system, the number of data points which exceeds the usable system area size in each CPU was specified.	Correct the number of data points for the CCPU_DedicatedDInst function.
-314	FEC6H	■Module major error An error was detected in the intelligent function module when a function was executed.	The possible cause is a hardware failure of the intelligent function module where the error was detected. Please consult your local Mitsubishi representative.
-315	FEC5H		
-316	FEC4H	■Other CPU module major error An error was detected in other CPU modules when a function was executed.	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 other CPU modules where the error was detected. Please consult your local Mitsubishi representative.
-317	FEC3H		
-318	FEC2H	■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.
-319	FEC1H	■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.
-320	FEC0H	■Clock rate specification error The specified clock rate is out of range.	Check the specified clock rate.
-321	FEBFH	■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. Please consult your local Mitsubishi representative.
-322	FEBEH	■System bus error An error was detected on the system bus.	• 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. Please consult your local Mitsubishi representative.
-323	FEBDH	■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. Please consult your local Mitsubishi representative.
-324	FEBCH	■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. Please consult your local Mitsubishi representative.
-325	FEBBH	■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. Please consult your local Mitsubishi representative.
-326	FEBAH	■System bus error	• 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.
-327	FEB9H	■Module major error • A major error was notified from the intelligent function module. • The I/O module or intelligent function module is not mounted properly or was removed during operation.	• Check the connection status of the extension cable. • 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.

4.3 MELSEC Data Link Functions

The following table shows the error codes of MELSEC data link functions.

Error code[*1]		Description	Corrective action
Decimal	Hexadecimal		
-1	FFFFH	■Path error • The specified path is unavailable. • The taskDelete was executed in the task using a MELSEC data link function. • The task using a MELSEC data link function was deleted with the taskDelete.	• Use a path pointer returned by using the mdOpen function. • Check if the taskDelete is executed in the task using the MELSEC data link function. • Check if the task using the MELSEC data link function is deleted with the taskDelete.
-2	FFFEH	■Device number error • The specified device number is out of range. • The specified bit device number is not a multiple of 8. • The device number and the points for the same block specified for reading/writing device randomly exceeds the device range.	• Check the start device number of the specified device. • Check the device number plus the number of points. • Specify the start device number of bit device in multiples of 8. • Check if the specified device is available in the CPU module on the target station.
-3	FFFDH	■Device type error The specified device type is unavailable.	• Check the specified device type. • Check if the specified device is available in the target station.
-5	FFFBH	■Size error • The device number and the size exceeds the device range. • The device number and the size exceeds the range for the same block. • The access was made with an odd-number bytes. • The total of the points that are specified for each block number of the mdRandREx function or the mdRandWEx function exceeds 10240.	• Check the specified device size. • Check the device number and the size. • Specify an even-number byte. • Reduce the total points that are specified for each block number of the mdRandREx function or the mdRandWEx function 10240 or less.
-6	FFFAH	■Number of blocks error The number of blocks specified to the function for reading/writing device randomly is out of range.	Check the number of the specified blocks.
-8	FFF8H	■Channel number error The channel number specified by using the mdOpen function is unavailable.	Check the specified channel number.
-11	FFF5H	■Insufficient buffer area error The area size of the read data storage is smaller than the read data size.	Check the area size of the data storage destination and read data size.
-12	FFF4H	■Block number error The specified block number is unavailable.	• Check the block number (device type) of the specified device. • Check if the specified device and block number are available in the target.
-13	FFF3H	■Write protect error The specified block number of the extension file register overlaps with the write protect area of the memory card.	• Check the block number (device type) of the extension file register. • Check the write protect switch on the memory card.
-16	FFF0H	■Station number/network number error • The specified station number or network number is out of range. • A device which cannot be accessed by the target station is specified.	• Check the specified station number and network number. • Check the devices which can be accessed by the target station.
-17	FFEFH	■All stations/group number specification error A function which does not support specifying all stations and group number was specified.	• Check if the function allows specifying all stations and group number. • Specify the device type to "Without arrival confirmation" when "All stations" or "Group number" is specified for the station number.
-18	FFEEH	■Remote operation error The specification code specified by using the mdControl function is unavailable.	Check the specified specification code.
-19	FFEDH	■SEND/RECV channel number error The channel number specified in the SEND/RECV function is out of range.	Specify the channel number within the range. • CC-Link IE Controller Network: 1 to 8 • CC-Link IE Field Network: 1 to 2
-31	FFE1H	■Module load error Loading modules required for executing functions failed.	• The memory may be insufficient. End another running task or reduce the access size. • Check the status of the system drive of the C Controller module.

Error code[*1]		Description	Corrective action
Decimal	Hexadecimal		
-32	FFE0H	■Resource timeout error The resource is being used by another task/thread and is not released within 30 seconds.	• Retry. • The memory may be insufficient. End another running task. • Check if the C Controller module is running normally. • Reset or power OFF to ON the C Controller module.
-33	FFDFH	■Communication target unsupported error The module specified as the communication target by a network number and station number is not supported.	• Check if the module specified as the communication target by a network number and station number is supported. • Check the settings of the access target set in CW Configurator.
-34	FFDEH	■Registry open error Opening parameter files in the registry failed.	Check if the access target is correctly set with CW Configurator.
-35	FFDDH	■Registry read error Reading parameter files from the registry failed.	• Check if the access target is correctly set with CW Configurator. • Check if the setting for the channel number is enabled. • Reset or power OFF to ON the C Controller module after checking the parameters with CW Configurator again and writing them. • Check if the access target module supports dedicated function libraries. (📖 MELSEC iQ-R C Controller Module User's Manual (Startup))
-36	FFDCH	■Registry write error Writing parameter files to the registry failed.	• Check if the standard ROM has already been shutdown. • Reset or power OFF to ON the C Controller module.
-37	FFDBH	■Communications initialization error Initializing the setting for communication failed.	• Retry. • The memory may be insufficient. End another running task. • Check the available memory capacity. • Check if the C Controller module is running normally. • Reset or power OFF to ON the C Controller module.
-42	FFD6H	■Close error Communications cannot be closed.	• Retry. • Check if the C Controller module is running normally. • Reset or power OFF to ON the C Controller module.
-43	FFD5H	■ROM operation error A TC setting value was written to the CPU module during ROM operation.	Change the TC setting value during RAM operation.
-52	FFCCH	■MELSEC data link function service error MELSEC data link function service is disabled.	Enable the MELSEC data link function service with CW Configurator.
-53	FFCBH	■Timeout value error The specified timeout value is out of range.	Check the specified time out value.
-54	FFCAH	■I/O number error The specified I/O number is out of range.	Check the specified I/O number.
-55	FFC9H	■Logical station number error The specified logical station number is out of range.	Check the specified logical station number.
-56	FFC8H	■Target CPU error The specified target CPU is out of range.	Check the specified target CPU.
-80	FFB0H	■Connection destination CPU error The connection destination CPU is not an RCPU.	Connect an RCPU.
-81	FFB1H	■Label code mismatch error Label assignment information of the CPU module was changed.	Obtain label information by using the mdGetLabelInfo function again.
-82	FFB2H	■Label incorrect value error An incorrect label name was specified. • Non-existent label name • Label name assigned to a device which does not support random read/write. • Label name assigned to a device which is specified by the inappropriate specification method (index modification or indirect specification)	Check the specified label name or the device specification method.
-83	FFB3H	■Size error The number of labels exceeded the range.	Check the number of labels.
-84	FFB4H	■Device specification method error A device was specified by inappropriate specification method (bit specification or digit specification).	Check the device specification method.
-475 to -3839	FE25H to F101H	Refer to the following manual. 📖 Q Corresponding MELSECNET/H Network System Reference Manual (PLC to PLC network)	

Error code[*1]		Description	Corrective action
Decimal	Hexadecimal		
-4097 to -8192	EFFFH to E000H	Refer to the following manuals. 📖 MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application) 📖 MELSEC-Q CC-Link IE Controller Network Reference Manual	
-8193 to -12288	DFFFH to D000H	Refer to the following manuals. 📖 MELSEC iQ-R CC-Link IE Field Network User's Manual (Application) 📖 MELSEC-Q CC-Link IE Field Network Master/Local Module User's Manual 📖 MELSEC-L CC-Link IE Field Network Master/Local Module User's Manual	
-16385 to -20480	BFFFH to B000H	Refer to the following manuals. 📖 MELSEC iQ-R CC-Link System Master/Local Module User's Manual (Application) 📖 MELSEC-Q CC-Link System Master/Local Module User's Manual 📖 MELSEC-L CC-Link System Master/Local Module User's Manual	

*1 When the function of which the return value is a long-type, the value will be eight digits in hexadecimal.

4.4 Error Codes Different From Conventional Functions

Error code (return value) for replaced functions may differ from the one for conventional functions. Be sure to refer to the list of error code in this manual (☞ Page 168 ERROR CODE LIST) to perform the troubleshooting.

APPENDIX

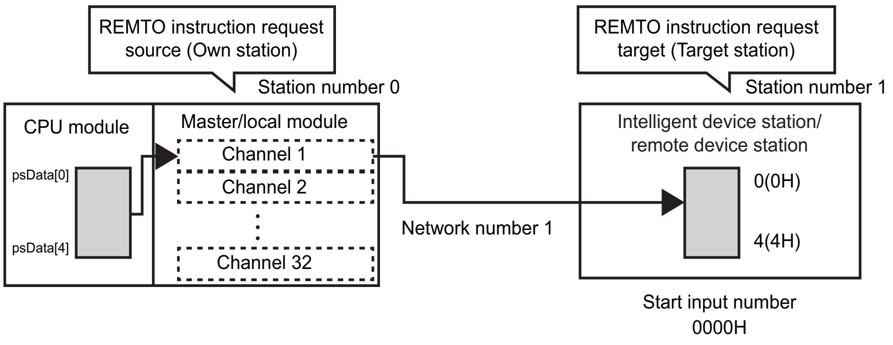
Appendix 1 Example for Replacing Ladder by C Language

This section shows program examples for replacing ladder by C language.

Program example

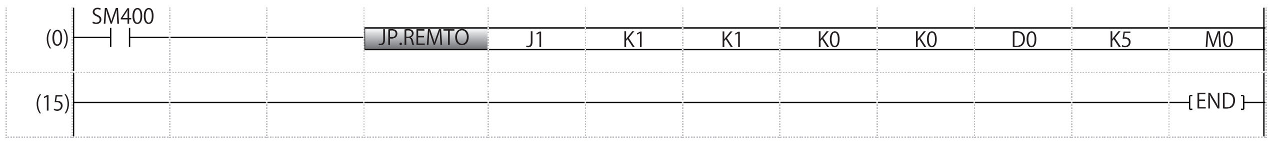
The following shows a program for writing the data in psData[5] of the station number 0 (own station) to the buffer memory (address: 0 to 4) of the intelligent device station/remote device station on the station number 1 (target station).

System configuration example



The CPU module is a programmable controller CPU

■Example for using the ladder program



Precautions

For a programmable controller CPU, in the program example, the data for five words (D0 to D4) is written since the start address of the buffer memory is specified with a device.

The CPU module is C Controller module

■Example for using the C Controller module dedicated function

```
short CCPU_DedicatedJInstSample(void){
  short sRet=0;                    /* Return value of the CCPU_DedicatedJInst function */
  char pcInstName[8]="REMTO";  /* Instruction code */
  short sNetNo=1;                  /* Target network number (1 to 239) */
  short sChan=1;                   /* Channel used by the own station (1 to 32) */
  short sStNo=1;                   /* Target station number (1 to 120) */
  short sIoNo=0x0000;              /* Start input/output number of Intelligent function module (0x0000 to 0x00FE) */
  short sAdd=0;                    /* Start address of buffer memory (0 to 65535) */
  short psData[5]={1,2,3,4,5};     /* Write data */
  short sSize=5;                   /* Number of write data (1 to 240 words) */
  short psCmp[2]={0,0};            /* Instruction completion result */

/* Dedicated instruction execution */
sRet=CCPU_DedicatedJInst(
     pcInstName,
     sNetNo,
     &sChan,
     1,
     &sStNo,
     1,
     &sIoNo,
     1,
     &sAdd,
     1,
     psData,
     5,
     &sSize,
     1,
     psCmp,
     2,
     NULL,
     0,
     NULL,
     0
     );

     return sRet;
   }
```

Appendix 2 How to Replace an Existing Product

Replacement of projects

Import the Q12DCCPU-V projects by using the Import function of CW Workbench (SW1DND-CWWR-E/EZ/EVZ). Select the "Build Support and Specs" tab on the screen of property for the imported project, and change "Active build spec" to "ARMARCH7gnu_SMP".*1

*1 For details on importing projects and changing "Active build spec", refer to the following manual.
📖 CW Workbench/CW-Sim Operating manual

Replacement of VxWorks standard API functions

The operating system of R12CCPU-V has been upgraded from Q12DCCPU-V.
(Q12DCCPU-V: VxWorks 6.4 → VxWorks 6.9)
To replace VxWorks standard API function, refer to "MIGRATION GUIDE" of VxWorks.*1

*1 PDF file of VxWorks "MIGRATION GUIDE" is included in CW Workbench.

Replacement of functions

If any of the functions listed in Page 187 Correspondence Table to Conventional Functions is used in the user program, replace the function. *1

*1 Check the specifications of the function before replacement since changing arguments may be required for the replacement in some case.

Replacement of device type

The device types listed on the following table are deleted from R12CCPU-V.
If any of the following device type is used in the user program to be replaced, perform the alternative method shown in the "Alternative method" column. The methods described in the following section are available as the alternative methods.
☞ Page 185 Alternative method

Bus interface function

■Device types for CC-Link IE Controller Network module access

Device type deleted from R12CCPU-V			Alternative method
Device		Device name specification	
Link input internal buffer	—	QBFDev_LXBuf	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with the following method. ☞ Page 185 Refreshing device, The access target is Network module on the own station.
Link output internal buffer	—	QBFDev_LYBuf	
Link relay internal buffer	—	QBFDev_LBBuf	
Link register internal buffer	—	QBFDev_LWBuf	

MELSEC data link function

■Device types for CC-Link module access

Device type deleted from R12CCPU-V			Alternative method
Device		Device name specification	
Own station remote input	RX	DevX	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with either of the methods below. ☞ Page 185 Refreshing device, The access target is Network module on the own station. ☞ Page 185 Module access device, The access target is Network module on the own station.
Own station remote output	RY	DevY	
Own station link register (for sending)	—	DevWw	
Own station link register (for receiving)	—	DevWr	
Own station link special relay[*1]	SB	DevSM	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with the following method. ☞ Page 185 Module access device, The access target is Network module on the own station.
Own station link special register[*2]	SW	DevSD	
Own station link special relay[*1]	SB	DevQSB	
Own station link special register[*2]	SW	DevQSW	
Own station random access buffer	—	DevMRB	
Own station buffer memory	—	DevSPB	
Other station buffer memory	—	DevRBM	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with the following method. ☞ Page 185 Module access device, The access target is Network module on the other station.
Other station random access buffer	—	DevRAB	
Other station remote input	—	DevRX	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with either of the methods below. ☞ Page 185 Refreshing device, The access target is Network module on the other station. ☞ Page 185 Module access device, The access target is Network module on the other station.
Other station remote output	—	DevRY	
Other station link register	—	DevRW	
Other station link special relay	—	DevSB	
Other station link special register	—	DevSW	

*1 The own station link special relay (SB) has two device type specifications; DevSM and DevQSB. Either of them can be specified for the same operation.
*2 The own station link special register (SW) has two device type specifications; DevSD, DevQSW. Either of them can be specified for the same operation.

■Device types for CC-Link IE Controller Network module access

Device type deleted from R12CCPU-V			Alternative method
Device		Device name specification	
Own station link input internal buffer (LX buffer)	—	DevX	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with either of the methods below. ☞ Page 185 Refreshing device, The access target is Network module on the own station. ☞ Page 185 CCPU_ReadLinkDevice/ CCPU_WriteLinkDevice
Own station link output internal buffer (LY buffer)	—	DevY	
Own station link relay internal buffer (LB buffer)	—	DevB	
Own station link register internal buffer (LW buffer)	—	DevW	
Own station direct link input	LX	DevLX(0)	
Own station direct link output	LY	DevLY(0)	
Own station direct link relay	LB	DevLB(0)	
Own station direct link register	LW	DevLW(0)	
Own station direct link special relay[*1]	SB	DevSM, DevQSB, DevLSB(0)	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with the following method. ☞ Page 185 CCPU_ReadLinkDevice/ CCPU_WriteLinkDevice
Own station direct link special register[*2]	SW	DevSD, DevQSW, DevLSW(0)	
Buffer memory	—	—	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with the following method. ☞ Page 185 Module access device, The access target is Network module on the own station.

*1 The own station direct link special relay (SB) has three device type specifications; DevSM, DevQSB, and DevLSB(0). Any of them can be specified for the same operation.
*2 The own station direct link special register (SW) has three device type specifications (DevSD, DevQSW, and DevLSW(0). Any of them can be specified for the same operation.

■Device types for CC-Link IE Field Network module access

Device type deleted from R12CCPU-V			Alternative method
Device		Device name specification	
Own station remote input	RX	DevLX(0)	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with any of the methods below. ☞ Page 185 Refreshing device, The access target is Network module on the own station. ☞ Page 185 Module access device, The access target is Network module on the own station. ☞ Page 185 CCPU_ReadLinkDevice/CCPU_WriteLinkDevice
Own station remote output	RY	DevLY(0)	
Own station remote register (for sending)	RWw	DevLW(0)	
Own station remote register (for receiving)	RWr		
Own station direct link special relay[*1]	SB	DevSM, DevQSB, DevLSB(0)	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with either of the methods below. ☞ Page 185 Refreshing device, The access target is Network module on the own station. ☞ Page 185 CCPU_ReadLinkDevice/CCPU_WriteLinkDevice
Own station direct link special register[*2]	SW	DevSD, DevQSW, DevLSW(0)	
Buffer memory	—	DevSPB	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with the following method. ☞ Page 185 Module access device, The access target is Network module on the own station.

*1 The own station direct link special relay (SB) has three device type specifications; DevSM, DevQSB, and DevLSB(0). Any of them can be specified for the same operation.

*2 The own station direct link special register (SW) has three device type specifications; DevSD, DevQSW, and DevLSW(0). Any of them can be specified for the same operation.

Alternative method

■Refreshing device

Access target	Alternative method
The access target is Network module on the own station.	Configure the refresh setting so that a device of C Controller module, M, B, D, W, or ZR is refreshed by a link device of Network module.
	Use the MELSEC data link functions to access a device of C Controller module, M, B, D, W or ZR.
The access target is Network module on the other station.	Configure the refresh setting for CPU module on the other station so that a device of C Controller module on the other station is refreshed by a link device of Network module.
	Specify the other station to the network number and station number in the MELSEC data link functions to access a device of CPU module on the other station.

■Module access device

Access target	Alternative method
The access target is Network module on the own station.	Open a communication line by specifying 'bus interface' to the channel in the mdOpen function.
	Specify the module access device (DevSPG) to the device type in the MELSEC data link functions, and access the area to which link devices are assigned[*1] in the buffer memory of Network module.
The access target is Network module on the other station.	Open a communication line by specifying the channel name corresponding to each network to the channel in the mdOpen function.
	Specify the other station to the network number and station number in the MELSEC data link functions.
	Specify the module access device (DevSPG) to the device type in the MELSEC data link functions, and access the area to which link devices are assigned[*1] in the buffer memory of Network module.

*1 For details on the buffer memory address to which link devices are assigned, refer to the manual for the Network module to be accessed.

■CCPU_ReadLinkDevice/CCPU_WriteLinkDevice

- Access the own station link device of Network module by using the CCPU_ReadLinkDevice/CCPU_WriteLinkDevice function. For details, refer to the relevant functions.

☞ Page 86 CCPU_ReadLinkDevice,Page 112 CCPU_WriteLinkDevice

Compilation of replaced project

Compile the replaced project in CW Workbench.

Appendix 3 Correspondence Table to Conventional Functions

○: Conventional functions can be used. ×: Conventional functions cannot be used.
—: Replacement of functions is not required. Not available: No functions are available to be replaced.

C Controller module dedicated functions

Function name (conventional)	Mode type	Availability in R12CCPU-V	Function name (replaced)
CCPU_ClearError	Extended mode	○	—
CCPU_EntryWDTInt	Extended mode	○	—
CCPU_Get7SegLED	Extended mode	×	CCPU_GetDotMatrixLED
CCPU_GetCpuStatus	Extended mode	○	—
CCPU_GetErrInfo	Extended mode	○	—
CCPU_GetLEDStatus	Extended mode	○	—
CCPU_GetPowerStatus	Extended mode	○	—
CCPU_GetRefreshStatus	Extended mode	×	CCPU_GetConstantProcessStatus
CCPU_GetRTC	Extended mode	○	—
CCPU_GetSwitchStatus	Extended mode	○	—
CCPU_MountMemoryCard	Extended mode	○	—
CCPU_ReadSRAM	Extended mode	×	CCPU_ReadDevice*1
CCPU_RegistEventLog	Extended mode	○	—
CCPU_ResetWDT	Extended mode	○	—
CCPU_Set7SegLED	Extended mode	×	CCPU_SetDotMatrixLED
CCPU_SetLEDStatus	Extended mode	○	—
CCPU_SetRTC	Extended mode	○	—
CCPU_StartWDT	Extended mode	○	—
CCPU_StopWDT	Extended mode	○	—
CCPU_UnmountMemoryCard	Extended mode	○	—
CCPU_WriteSRAM	Extended mode	×	CCPU_WriteDevice*1
CCPU_ChangeFileSecurity	Extended mode	○	—
CCPU_GetFileSecurity	Extended mode	○	—
CCPU_CommunicateMCProtocol	Extended mode	×	—
CCPU_SetOpenNoMCProtocol	Extended mode	×	—

*1 Use ZR device as a substitute.

C Controller module dedicated function for ISR

Function name (conventional)	Mode type	Availability in R12CCPU-V	Function name (replaced)
CCPU_Get7SegLED_ISR	Extended mode	×	CCPU_GetDotMatrixLED_ISR
CCPU_Set7SegLED_ISR	Extended mode	×	CCPU_SetDotMatrixLED_ISR
CCPU_ReadSRAM_ISR	Extended mode	×	CCPU_ReadDevice_ISR*1
CCPU_WriteSRAM_ISR	Extended mode	×	CCPU_WriteDevice_ISR*1
CCPU_SetLEDStatus_ISR	Extended mode	○	—

*1 Use ZR device as a substitute.

Bus interface functions

Function name (conventional)	Mode type	Availability in R12CCPU-V	Function name (replaced)
QBF_Close	Basic mode/Extended mode	×	Not available
QBF_ControlEx	Basic mode/Extended mode	×	CCPU_Control
QBF_ControlProgram	Basic mode/Extended mode	×	Not available
QBF_FromBuf	Basic mode/Extended mode	×	CCPU_FromBuf CCPU_FromBufHG
QBF_GINT	Basic mode/Extended mode	×	CCPU_DedicatedDInst
QBF_MotionCHGA	Basic mode/Extended mode	×	CCPU_DedicatedDInst
QBF_MotionCHGT	Basic mode/Extended mode	×	CCPU_DedicatedDInst
QBF_MotionCHGT2	Basic mode/Extended mode	×	CCPU_DedicatedDInst
QBF_MotionCHGV	Basic mode/Extended mode	×	CCPU_DedicatedDInst
QBF_MotionDDRD	Basic mode/Extended mode	×	CCPU_DedicatedDInst
QBF_MotionDDWR	Basic mode/Extended mode	×	CCPU_DedicatedDInst
QBF_MotionSFCS	Basic mode/Extended mode	×	CCPU_DedicatedDInst
QBF_MotionSVST	Basic mode/Extended mode	×	CCPU_DedicatedDInst
QBF_Open	Basic mode/Extended mode	×	Not available
QBF_ReadDevice	Basic mode/Extended mode	×	CCPU_ReadDevice
QBF_ReadLinkDevice	Basic mode/Extended mode	×	CCPU_ReadLinkDevice
QBF_RECV	Basic mode/Extended mode	×	CCPU_DedicatedGInst, CCPU_DedicatedJInst
QBF_RefreshLinkDevice	Basic mode/Extended mode	×	Not available
QBF_Reset	Basic mode/Extended mode	×	CCPU_Reset
QBF_ResetDevice	Basic mode/Extended mode	×	CCPU_ResetDevice
QBF_SEND	Basic mode/Extended mode	×	CCPU_DedicatedGInst, CCPU_DedicatedJInst
QBF_SetDevice	Basic mode/Extended mode	×	CCPU_SetDevice
QBF_ToBuf	Basic mode/Extended mode	×	CCPU_ToBuf, CCPU_ToBufHG
QBF_UnitInfo	Basic mode/Extended mode	×	CCPU_GetUnitInfo
QBF_WaitEvent	Basic mode/Extended mode	×	CCPU_WaitEvent
QBF_WaitUnitEvent	Basic mode/Extended mode	×	CCPU_WaitUnitEvent
QBF_WriteDevice	Basic mode/Extended mode	×	CCPU_WriteDevice
QBF_WriteLinkDevice	Basic mode/Extended mode	×	CCPU_WriteLinkDevice
QBF_X_In_BitEx	Basic mode/Extended mode	×	CCPU_X_In_BitEx
QBF_X_In_WordEx	Basic mode/Extended mode	×	CCPU_X_In_WordEx
QBF_Y_In_BitEx	Basic mode/Extended mode	×	CCPU_Y_In_BitEx
QBF_Y_In_WordEx	Basic mode/Extended mode	×	CCPU_Y_In_WordEx
QBF_Y_Out_BitEx	Basic mode/Extended mode	×	CCPU_Y_Out_BitEx
QBF_Y_Out_WordEx	Basic mode/Extended mode	×	CCPU_Y_Out_WordEx
QBF_MotionCHGVS	Basic mode/Extended mode	×	CCPU_DedicatedDInst
QBF_MotionCHGAS	Basic mode/Extended mode	×	CCPU_DedicatedDInst
QBF_REMTO	Extended mode	×	CCPU_DedicatedJInst
QBF_REMFR	Extended mode	×	CCPU_DedicatedJInst
QBF_DisableCpuInt	Basic mode/Extended mode	×	Not available
QBF_DisableMultiCPUSyncInt	Basic mode/Extended mode	×	CCPU_DisableInt
QBF_DisableUnitInt	Basic mode/Extended mode	×	CCPU_DisableInt
QBF_EnableCpuInt	Basic mode/Extended mode	×	Not available
QBF_EnableMultiCPUSyncInt	Basic mode/Extended mode	×	CCPU_EnableInt
QBF_EnableUnitInt	Basic mode/Extended mode	×	CCPU_EnableInt
QBF_EntryCpuInt	Basic mode/Extended mode	×	Not available
QBF_EntryMultiCPUSyncInt	Basic mode/Extended mode	×	CCPU_EntryInt
QBF_EntryUnitInt	Basic mode/Extended mode	×	CCPU_EntryInt

Function name (conventional)	Mode type	Availability in R12CCPU-V	Function name (replaced)
QBF_ClearError	Basic mode/Extended mode	×	CCPU_ClearError
QBF_Control	Basic mode/Extended mode	×	CCPU_Control
QBF_Control7SegLED	Basic mode/Extended mode	×	CCPU_SetDotMatrixLED
QBF_ControlLED	Basic mode/Extended mode	×	CCPU_SetLEDStatus
QBF_EntryTimerEvent	Basic mode/Extended mode	×	CCPU_EntryTimerEvent
QBF_EntryWDTInt	Basic mode/Extended mode	×	CCPU_EntryWDTInt
QBF_GetTime	Basic mode/Extended mode	×	CCPU_GetRTC
QBF_MountCfCard	Basic mode/Extended mode	×	CCPU_MountMemoryCard
QBF_ReadSRAM	Basic mode/Extended mode	×	CCPU_ReadDevice*1
QBF_ReadStatusEx	Basic mode/Extended mode	×	CCPU_GetCpuStatus
QBF_RegistEventLog	Basic mode/Extended mode	×	CCPU_RegistEventLog, CCPU_RegistEventLog_ISR
QBF_ResetWDT	Basic mode/Extended mode	×	CCPU_ResetWDT
QBF_SetTime	Basic mode/Extended mode	×	CCPU_SetRTC
QBF_ShutdownRom	Basic mode/Extended mode	×	CCPU_ShutdownRom
QBF_StartWDT	Basic mode/Extended mode	×	CCPU_StartWDT
QBF_StopWDT	Basic mode/Extended mode	×	CCPU_StopWDT
QBF_UnmountCfCard	Basic mode/Extended mode	×	CCPU_UnmountMemoryCard
QBF_WaitTimerEvent	Basic mode/Extended mode	×	CCPU_WaitTimerEvent
QBF_WriteSRAM	Basic mode/Extended mode	×	CCPU_WriteDevice*1

*1 Use ZR device as a substitute.

Bus interface functions for ISR

Function name (conventional)	Mode type	Availability in R12CCPU-V	Function name (replaced)
QBF_DisableCpuInt_ISR	Basic mode/Extended mode	×	Not available
QBF_DisableMultiCPUSyncInt_ISR	Basic mode/Extended mode	×	CCPU_DisableInt_ISR
QBF_DisableUnitInt_ISR	Basic mode/Extended mode	×	CCPU_DisableInt_ISR
QBF_EnableCpuInt_ISR	Basic mode/Extended mode	×	Not available
QBF_EnableMultiCPUSyncInt_ISR	Basic mode/Extended mode	×	CCPU_EnableInt_ISR
QBF_EnableUnitInt_ISR	Basic mode/Extended mode	×	CCPU_EnableInt_ISR
QBF_FromBuf_ISR	Basic mode/Extended mode	×	CCPU_FromBuf_ISR
QBF_ReadDevice_ISR	Basic mode/Extended mode	×	CCPU_ReadDevice_ISR
QBF_ResetDevice_ISR	Basic mode/Extended mode	×	CCPU_ResetDevice_ISR
QBF_SetDevice_ISR	Basic mode/Extended mode	×	CCPU_SetDevice_ISR
QBF_ToBuf_ISR	Basic mode/Extended mode	×	CCPU_ToBuf_ISR
QBF_WriteDevice_ISR	Basic mode/Extended mode	×	CCPU_WriteDevice_ISR
QBF_X_In_Word_ISR	Basic mode/Extended mode	×	CCPU_X_In_Word_ISR
QBF_Y_In_Word_ISR	Basic mode/Extended mode	×	CCPU_Y_In_Word_ISR
QBF_Y_Out_Word_ISR	Basic mode/Extended mode	×	CCPU_Y_Out_Word_ISR
QBF_ControlLED_ISR	Basic mode/Extended mode	×	CCPU_SetLEDStatus_ISR
QBF_Control7SegLED_ISR	Basic mode/Extended mode	×	CCPU_SetDotMatrixLED_ISR
QBF_WriteSRAM_ISR	Basic mode/Extended mode	×	CCPU_WriteDevice_ISR*1
QBF_ReadSRAM_ISR	Basic mode/Extended mode	×	CCPU_ReadDevice_ISR*1

*1 Use ZR device as a substitute.

MELSEC data link functions

Function name (conventional)	Mode type	Availability in R12CCPU-V	Function name (replaced)
mdClose	Basic mode/Extended mode	○	—
mdControl	Basic mode/Extended mode	○	—
mdDevRstEx	Basic mode/Extended mode	○	—
mdDevSetEx	Basic mode/Extended mode	○	—
mdInit	Basic mode/Extended mode	○	—
mdOpen	Basic mode/Extended mode	○	—
mdRandREx	Basic mode/Extended mode	○	—
mdRandWEx	Basic mode/Extended mode	○	—
mdReceiveEx (Device batch read function)	Basic mode/Extended mode	○	—
mdReceiveEx (Message receive function)	Basic mode/Extended mode	○	—
mdSendEx (Device batch write function)	Basic mode/Extended mode	○	—
mdSendEx (Message send function)	Basic mode/Extended mode	○	—
mdTypeRead	Basic mode/Extended mode	○	—
mdDevRst	Basic mode	×	mdDevRstEx
mdDevSet	Basic mode	×	mdDevSetEx
mdRandR	Basic mode	×	mdRandREx
mdRandW	Basic mode	×	mdRandWEx
mdReceive (Device batch read function)	Basic mode	×	mdReceiveEx (Device batch read function)
mdReceive (Message receive function)	Basic mode	×	mdReceiveEx (Message receive function)
mdSend (Device batch write function)	Basic mode	×	mdSendEx (Device batch write function)
mdSend (Message send function)	Basic mode	×	mdSendEx (Message send function)

MEMO

A

INDEX

B

Bus interface communication 10,13

C

C Controller module dedicated function 6,7
CC-Link communication. 10,34
CC-Link IE Controller Network communication
. . . . 10,16
CC-Link IE Field Network communication 10,22
Channel 37
CW Workbench 6
CW-Sim 6

D

Dedicated function library. 6,7
Device type 9,40
Dummy access 10,11

H

Header file 7,9,40,61,63

I

ISR 43

M

MELSEC data link function. 6,7
MELSECNET/H network communication 10,28

T

Task 11

U

User watchdog timer 8

V

VxWorks 6
VxWorks standard API function 7

MEMO

FUNCTION INDEX

C

CCPU_ChangeFileSecurity 48
CCPU_ClearError 49
CCPU_Control 50
CCPU_DedicatedDInst 51
CCPU_DedicatedGInst 53
CCPU_DedicatedJInst. 55
CCPU_DedicatedMInst 57
CCPU_DisableInt 59
CCPU_DisableInt_ISR. 119
CCPU_EnableInt 60
CCPU_EnableInt_ISR 120
CCPU_EntryInt 61
CCPU_EntryTimerEvent 62
CCPU_EntryWDTInt 63
CCPU_FromBuf 64
CCPU_FromBufHG 65
CCPU_FromBufHG_ISR 122
CCPU_FromBuf_ISR. 121
CCPU_GetConstantProcessStatus 66
CCPU_GetCounterMicros 67
CCPU_GetCounterMicros_ISR. 123
CCPU_GetCounterMillis 68
CCPU_GetCounterMillis_ISR 124
CCPU_GetCpuStatus 69
CCPU_GetDotMatrixLED. 70
CCPU_GetDotMatrixLED_ISR 125
CCPU_GetErrInfo 71
CCPU_GetFileSecurity 72
CCPU_GetIDInfo 73
CCPU_GetLEDStatus 74
CCPU_GetOpSelectMode 76
CCPU_GetPowerStatus. 77
CCPU_GetRTC 78
CCPU_GetSerialNo 79
CCPU_GetSwitchStatus 80
CCPU_GetUnitInfo 81
CCPU_MountMemoryCard 84
CCPU_ReadDevice 85
CCPU_ReadDevice_ISR 127
CCPU_ReadLinkDevice. 86
CCPU_RegistEventLog 87
CCPU_RegistEventLog_ISR 128
CCPU_Reset 88
CCPU_ResetDevice 89
CCPU_ResetDevice_ISR. 129
CCPU_ResetWDT 90
CCPU_SetDevice 91
CCPU_SetDevice_ISR 130
CCPU_SetDotMatrixLED 92
CCPU_SetDotMatrixLED_ISR 131
CCPU_SetLEDStatus 94
CCPU_SetLEDStatus_ISR. 133
CCPU_SetOpSelectMode 95
CCPU_SetRTC. 96
CCPU_ShutdownRom 97
CCPU_StartWDT 98
CCPU_StopWDT 99
CCPU_SysClkRateGet 100
CCPU_SysClkRateSet 101
CCPU_ToBuf 102
CCPU_ToBufHG 103
CCPU_ToBufHG_ISR. 135
CCPU_ToBuf_ISR 134
CCPU_UnmountMemoryCard 104
CCPU_WaitEvent. 105
CCPU_WaitSwitchEvent 107
CCPU_WaitTimerEvent. 108
CCPU_WaitUnitEvent 109
CCPU_WriteDevice .111
CCPU_WriteDevice_ISR 136
CCPU_WriteLinkDevice 112
CCPU_X_In_BitEx .113
CCPU_X_In_WordEx 114
CCPU_X_In_Word_ISR 137
CCPU_Y_In_BitEx .115
CCPU_Y_In_WordEx 116
CCPU_Y_In_Word_ISR 138
CCPU_Y_Out_BitEx 117
CCPU_Y_Out_WordEx 118
CCPU_Y_Out_Word_ISR 139

M

mdClose 140
mdControl 141
mdDevRstEx 142
mdDevSetEx 143
mdGetLabelInfo 144
mdInit 147
mdOpen 148
mdRandREx 149
mdRandRLabelEx .152
mdRandWEx 155
mdRandWLabelEx .157
mdReceiveEx. 159,160
mdSendEx. 162,163
mdTypeRead 165

MEMO

REVISIONS

*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
February 2015	SH(NA)-081371ENG-A	First edition
April 2015	SH(NA)-081371ENG-B	Structural change of the manual
May 2015	SH(NA)-081371ENG-C	■Added or modified parts TERMS, Section 3.1
October 2015	SH(NA)-081371ENG-D	■Added or modified parts INTRODUCTION, TERMS, Section 1.2, Section 1.3, Section 2.2, Section 3.2, Section 4.3
September 2016	SH(NA)-081371ENG-E	■Added or modified parts TERMS, Section 1.3, Section 3.1, Section 3.2, Section 4.3

Japanese manual number: SH-081370-E

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

(1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.

(2) Even within the gratis warranty term, repairs shall be charged for in the following cases.

1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
2. Failure caused by unapproved modifications, etc., to the product by the user.
3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

(1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.

(2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to:

(1) Damages caused by any cause found not to be the responsibility of Mitsubishi.

(2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.

(3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.

(4) Replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

TRADEMARKS

SH(NA)-081371ENG-E(1609)
MODEL:RCCPU-P-E

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN