

Mitsubishi Programmable Controller

MELSEC iQ-R
series

MELSEC iQ-R C Controller Module Programming Manual



-SW1DND-CWWR-E

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.

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

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

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

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

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

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.

CONTENTS

SAFETY PRECAUTIONS	1
CONDITIONS OF USE FOR THE PRODUCT	1
CONSIDERATIONS FOR USE	2
INTRODUCTION	2
RELEVANT MANUAL	5
TERMS	6
CHAPTER 1 COMMON ITEMS	7
1.1 Header Files	7
1.2 C Controller Module Dedicated Functions	8
Argument specification	8
Considerations	9
1.3 MELSEC Data Link Functions	10
Argument specification	10
Accessible range and devices	15
Considerations	31
1.4 Considerations on Interrupt Service Routine	32
CHAPTER 2 FUNCTION LIST	33
2.1 C Controller Module Dedicated Functions	33
C Controller module dedicated functions for ISR	36
2.2 MELSEC Data Link Functions	37
CHAPTER 3 DETAILS OF FUNCTION	38
3.1 C Controller Module Dedicated Functions	38
C Controller module dedicated functions for ISR	112
3.2 MELSEC Data Link Functions	133
CHAPTER 4 ERROR CODE LIST	152
4.1 Common Error Codes	152
4.2 C Controller Module Dedicated Functions	155
4.3 MELSEC Data Link Functions	160
4.4 Error Codes Different From Conventional Functions	162
APPENDIX	163
Appendix 1 Example for Replacing Ladder by C Language	163
Program example	163
Appendix 2 How to Replace an Existing Product	165
Replacement of projects	165
Replacement of VxWorks standard API functions	165
Replacement of functions	165
Replacement of device type	165
Compilation of replaced project	167
Appendix 3 Correspondence Table to Conventional Functions	168
C Controller module dedicated functions	168
C Controller module dedicated function for ISR	168
Bus interface functions	169

Bus interface functions for ISR	170
MELSEC data link functions	171

FUNCTION INDEX	172
-----------------------	------------

REVISIONS	174
WARRANTY	175
TRADEMARKS	176

RELEVANT MANUAL

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 functions of C Controller module.	e-Manual EPUB 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 EPUB 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 EPUB PDF
CW Workbench/CW-Sim Operating Manual [SH-081373ENG]	Explains the system configuration, specifications, functions, and troubleshooting of CW Workbench/CW-Sim.	e-Manual EPUB 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 EPUB PDF

Point

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

e-Manual has the following features:

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

TERMS


Unless otherwise specified, this manual uses the following terms.

Term	Description
Basic model QCPU	A generic term for Q00JCPU, Q00CPU, and Q01CPU
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
C Controller module	An abbreviation for MELSEC iQ-R series C Controller module
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 engineering tool, CW Workbench
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
Process CPU	A generic term for Q02PHCPU, Q06PHCPU, Q12PHCPU, and Q25PHCPU
RCPU	A generic term for R04CPU, R08CPU, R16CPU, R32CPU, and R120CPU
Redundant CPU	A generic term for Q12PRHCPU and Q25PRHCPU
SD memory card	A memory card that is compliant with the SD standards designed and developed by the SD Association
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 VxWorks standard API functions^{*1} and dedicated function library provided by 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"

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 header file shown below to use the dedicated function libraries provided by C Controller module.

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

For storage destination of header file, refer to the following manual.

 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.

Argument specification

Either a code or a device name can be used as a device type specified to the C Controller module dedicated functions. Device names are defined in the include file "CCPUFunc.h" of this functions.

Device types for CC-Link IE Controller Network module access

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

Device		Device type	
		Code specification	Device name specification
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 functions; CCPU_WriteLinkDevice and CCPU_ReadLinkDevice.

Device		Device type	
		Code specification	Device name specification
Direct link input	RX	1000 (3E8H)	Dev_LX
Direct link output	RY	2000 (7D0H)	Dev_LY
Direct link register ^{*1}	RWr, RWw	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 functions; CCPU_WriteDevice, CCPU_ReadDevice, CCPU_SetDevice, CCPU_ResetDevice, CCPU_WriteDevice_ISR, CCPU_ReadDevice_ISR, CCPU_SetDevice_ISR, and CCPU_ResetDevice_ISR.

Device		Device type	
		Code specification	Device name specification
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

Considerations

Considerations for user WDT

■Output when a user WDT error occurs

If the user WDT cannot be reset due to a user program runaway, a user WDT error occurs and the outputs turn OFF.

■User WDT setting range

User WDT can be set within the following range:

User WDT setting range: 100 ms to 10,000 ms

■Countermeasures for user WDT error

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 system once the above corrective actions have been taken.

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.

Considerations for the dedicated instructions

Considerations for argument are shown below:

- For the first argument, "pclInstName", categorizing the dedicated instructions (D/DP/J/JP/G/GP/M) is not required.
- Own station devices cannot be specified. Reserve the required area with a user program, and specify the start address of the area to the relevant argument.
- Specify other station devices 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.

1.3 MELSEC Data Link Functions

MELSEC data link functions are the integrated communication libraries which are independent of the communication protocols.

By using MELSEC data link functions, communication can be performed regardless of the target hardware or communication protocols for creating programs to communicate to a programmable controller CPU.

Communication function	Description
Bus interface communication	Accesses the CPU module mounted on the base unit of the own station from C Controller module.
CC-Link communication	Accesses the CPU module connected to the network from the CC-Link module controlled by C Controller module.
CC-Link IE Controller Network communication	Accesses the CPU module connected to the network from the CC-Link IE Controller Network module controlled by C Controller module.
CC-Link IE Field Network communication	Accesses the CPU module connected to the network from the CC-Link IE Field Network module controlled by C Controller module.

Argument specification

Channel

A channel shows the connection configuration to media and a target communicating with C Controller module.

A channel is set for each module in a user program (MELSEC data link functions).

The channels to be used by MELSEC data link functions are as follows:

Channel number	Channel name	Description
12	Bus interface	Used for communication via bus.
81 to 88	CC-Link	Used for communication via CC-Link module controlled by C Controller module.
151 to 158	CC-Link IE Controller Network	Used for communication via CC-Link IE Controller Network module controlled by C Controller module.
181 to 188	CC-Link IE Field Network	Used for communication via CC-Link IE Field Network module controlled by 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		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 station * ⁴ ,* ⁵		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).
For direct access to the CPU module which controls MELSECNET/10 other station, MELSECNET/H other stations (station number 1 to 64) and MELSECNET/10 other station, MELSECNET/H other station, CC-Link IE Controller Network other station, or CC-Link IE Field Network other station, setting of the logical station number is not required. Specify or use the station number of MELSECNET/10, MELSECNET/H, 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 is 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	Own station: 255 (FFH) Other stations: 0 (0H) to 63 (3FH), 65 (41H) to 239 (EFH)*1,*2
CC-Link IE Controller Network	Own station: 255 (FFH) Other station: *3,*4
CC-Link IE Field Network	Own station: 255 (FFH) Other station: *3,*5

*1 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 other station number.

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

In addition, when own station number is 64, other stations cannot be set. (Only own station can be accessed.)

*3 Station number setting for CC-Link IE Controller Network module or CC-Link IE Field Network module

Upper	Lower
-------	-------

Upper/lower	Setting item	Setting value	Description
Upper	Network number	1 (1H) to 239 (EFH)	Set this to specify the other station in the own network or each station on the other network. • Set this to issue a sending request to any of CC-Link IE Field Network, CC-Link IE Controller Network, MELSECNET/H, or MELSECNET/10.
Lower	Station number, Group number, or All stations	1 (1H) to 120 (78H)	Set the station number of other station. • For MELSECNET/H, 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 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. • The all stations specification is valid 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).

*4 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).

*5 125 (7DH) is valid when CC-Link IE Field Network master station (station number 0) is specified by using the SEND/RCV 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).

Device type

Either a code or device name can be used as a device type to be specified to MELSEC data link functions.

Device names are defined in the include file "MDFunc.h" of this functions.

Common device types

Device		Device type	
		Code	Device name specification
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 (CPU No.1 area) ^{*1,*2}	U3E0\G	501 (1F5H)	DevSPB1
CPU buffer memory (CPU No.2 area) ^{*1,*2}	U3E1\G	502 (1F6H)	DevSPB2
CPU buffer memory (CPU No.3 area) ^{*1,*2}	U3E2\G	503 (1F7H)	DevSPB3
CPU buffer memory (CPU No.4 area) ^{*1,*2}	U3E3\G	504 (1F8H)	DevSPB4
Fixed cycle communication area (CPU No.1 area) ^{*1,*2}	U3E0\HG	511 (1FFH)	DevHSPB1
Fixed cycle communication area (CPU No.2 area) ^{*1,*2}	U3E1\HG	512 (200H)	DevHSPB2
Fixed cycle communication area (CPU No.3 area) ^{*1,*2}	U3E2\HG	513 (201H)	DevHSPB3
Fixed cycle communication area (CPU No.4 area) ^{*1,*2}	U3E3\HG	514 (202H)	DevHSPB4
Annunciator	F	6 (6H)	DevF
Timer (Contact)	T	7 (7H)	DevTT
Long timer (Contact)	LT	41 (29H)	DevLTT
Timer (Coil)	T	8 (8H)	DevTC
Long timer (Coil)	LT	42 (2AH)	DevLTC
Counter (Contact)	C	9 (9H)	DevCT
Long counter (Contact)	LC	44 (2CH)	DevLCT
Counter (Coil)	C	10 (AH)	DevCC
Long counter (Coil)	LC	45 (2DH)	DevLCC
Timer (Current value)	T	11 (BH)	DevTN
Long timer (Current value)	LT	43 (2BH)	DevLTN
Counter (Current value)	C	12 (CH)	DevCN
Long counter (Current value)	LC	46 (2EH)	DevLCN
Data register	D	13 (DH)	DevD
Special register	SD	14 (EH)	DevSD
Index register ^{*3}	Z	20 (14H)	DevZ
Long index register ^{*3}	LZ	38 (26H)	DevLZ
File register ^{*3}	R	22 (16H)	DevR
	ZR	220 (DCH)	DevZR
Link relay	B	23 (17H)	DevB
Link register	W	24 (18H)	DevW
Link special relay ^{*3}	SB	25 (19H)	DevQSB
Retentive timer (Contact)	ST	26 (1AH)	DevSTT
Long retentive timer (Contact)	LST	47 (2FH)	DevLSTT
Retentive timer (Coil)	ST	27 (1BH)	DevSTC
Long retentive timer (Coil)	LST	48 (30H)	DevLSTC
Link special register ^{*3}	SW	28 (1CH)	DevQSW
Edge relay	V	30 (1EH)	DevQV
Retentive timer (Current value)	ST	35 (23H)	DevSTN
Long retentive timer (Current value)	LST	49 (31H)	DevLSTN
Module refresh register	RD	39 (27H)	DevRD
SEND function (with arrival confirmation) and RECV function	—	101 (65H)	DevMAIL

Device		Device type	
		Code	Device name specification
SEND function (without arrival confirmation)	—	102 (66H)	DevMAILNC
Link direct device (Link input) Argument value of device name (1 to 255 ^{*3}): Network number	Jn\X	1001 to 1255 (3E9H to 4E7H)	DevLX(1) to DevLX(255)
Link direct device (Link output) Argument value of device name (1 to 255 ^{*3}): Network number	Jn\Y	2001 to 2255 (7D1H to 8CFH)	DevLY(1) to DevLY(255)
Link direct device (Link relay) Argument value of device name (1 to 255 ^{*3}): Network number	Jn\B	23001 to 23255 (59D9H to 5AD7H)	DevLB(1) to DevLB(255)
Link direct device (Link register) Argument value of device name (1 to 255 ^{*3}): Network number	Jn\W	24001 to 24255 (5DC1H to 5EBFH)	DevLW(1) to DevLW(255)
Link direct device (Link special relay) Argument value of device name (1 to 255 ^{*3}): Network number	Jn\SB	25001 to 25255 (61A9H to 62A7H)	DevLSB(1) to DevLSB(255)
Link direct device (Link special register) Argument value of device name (1 to 255 ^{*3}): Network number	Jn\SW	28001 to 28255 (6D61H to 6E5FH)	DevLSW(1) to DevLSW(255)
Intelligent function module device, module access device Argument value of device name (0 to 255 ^{*3}): Start I/O No. divided by 16	—	29000 to 29255 (7148H to 7247H)	DevSPG(0) to DevSPG(255)

*1 The CPU buffer memory for Q12DCCPU-V is categorized as the device type dedicated to Q bus interface. It is not accessible during CC-Link communication, CC-Link IE Controller Network communication or 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'.)

■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 specification
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 specification
SEND function (with arrival confirmation) and RECV function		101 (65H)	DevMAIL
SEND function (without arrival confirmation)		102 (66H)	DevMAILNC

Accessible range and devices

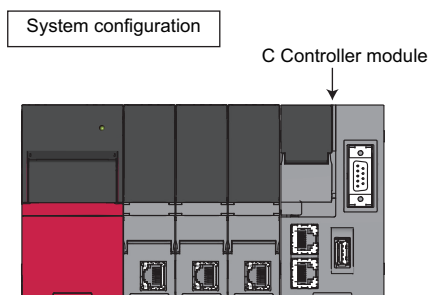
The following explains the accessible range and accessible devices when using MELSEC data link functions.

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.



■ Accessible devices

The following explains the accessible devices during communication via bus.

Point

- "Batch" and "Random" in the 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)
- Only bit devices are accessible to 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 or index specification) cannot be used.

• Accessing host CPU

The accessible devices when accessing the host CPU are shown in the following table.

○: Accessible, ×: Not accessible

Device		Access method	Access target
			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 accessible devices when accessing the other CPU (the CPU module and C Controller module in a multiple CPU system) are shown in the following table.

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

○: Accessible, ×: Not accessible

Device		Access method	Access target	
			(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	○	○
Module refresh 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	○	○

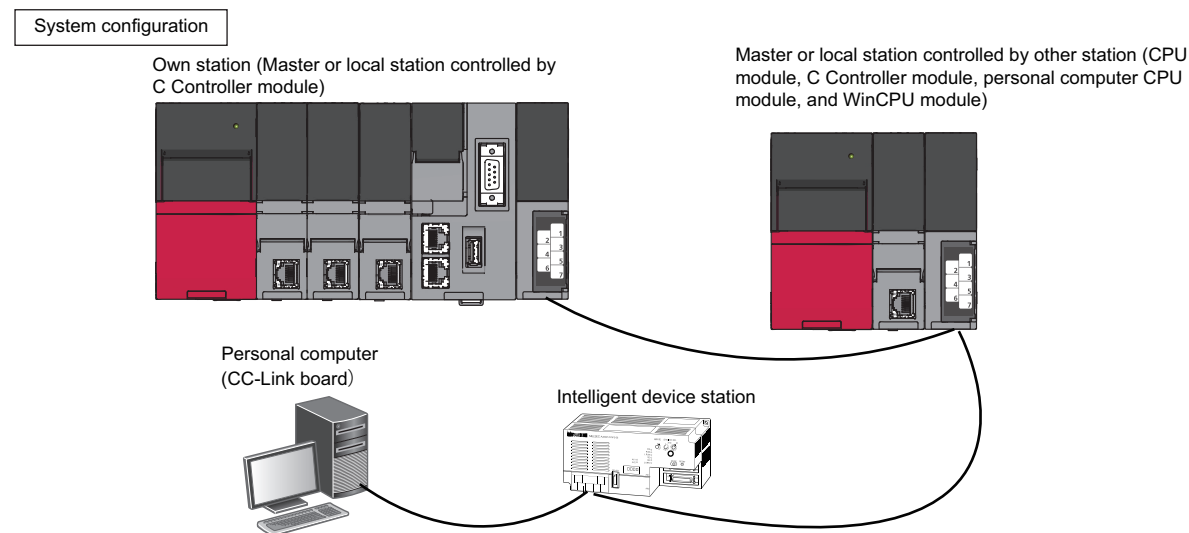
Device		Access method	Access target	
			(1)	(2)
Link direct device (Link special relay)	Jn\SB	Batch/random	○	○
Link direct device (Link special register)	Jn\SW	Batch/random	○	○
Intelligent function module device, module access device	Un\G	Batch/random	○	○
Other station buffer memory	—	Batch/random	×	×
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	×	×

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 own station (master station or local station controlled by C Controller module), master station or local station controlled by other station (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.



Point

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

■ Accessible devices

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

Point

- "Batch" and "Random" in the 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)

- Only bit devices are accessible to bit set (mdDevSetEx function) and bit reset (mdDevRstEx function).
- Device extension specifications (digit specification, bit specification or index specification) cannot be used.

- Accessing own station

To access the CC-Link module controlled by C Controller module, use the method explained in the following section.

Accessing the own station using CC-Link communication will result in the station number error, network number error (-16).

📖 Page 165 Replacement of device type

• Accessing other station

No.	Access target
(1)	Basic model QCPU, High Performance model QCPU, Process CPU, Redundant CPU, Universal model QCPU
(2)	Q12DCCPU-V, Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG
(3)	PC CPU module, WinCPU module, personal computer, Intelligent device station
(4)	L02CPU, L26CPU-BT, L02CPU-P, L26CPU-PBT, L02SCPU, L26CPU, L06CPU
(5)	RCPU
(6)	R12CCPU-V

○: Accessible, ×: Not accessible

Device		Access method	Access target					
			(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}	×	×	○	○	○
Module refresh 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					
			(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					×	×

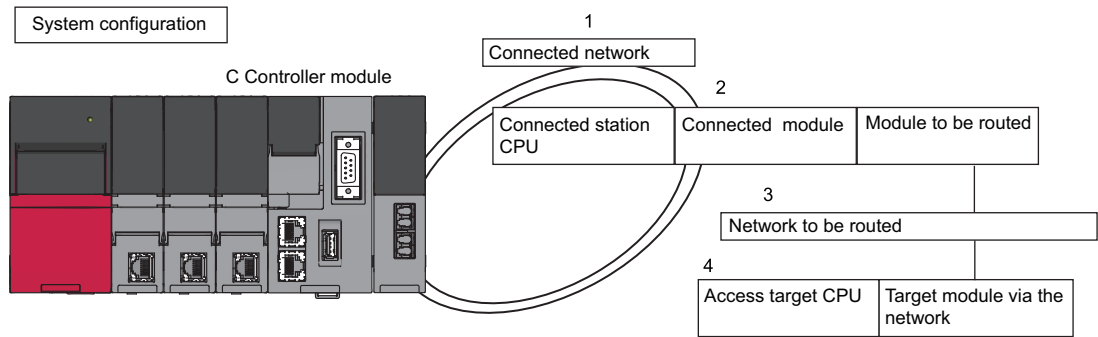
- *1 The following modules are accessible:
Q12DCCPU-V with a serial number of which first 5 digits are "12042" or later
Q24DHCCPU-V, Q24DHCCPU-LS, and Q24DHCCPU-VG
- *2 Q00JCPU cannot be accessed.
- *3 The following modules are accessible:
Q12DCCPU-V (Extended mode),
Q24DHCCPU-V, Q24DHCCPU-LS, and Q24DHCCPU-VG

CC-Link IE Controller Network communication

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

■Accessible range

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



■Applicable access

Accessibility is shown in the following table. All of 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 programmabl e controller	CC-Link IE Controller Network	○	○	×	○	○	×	×
		CC-Link IE Field Network	○	○	○	○	○ ^{*1}	×	×
		MELSECNET/H	○	○	×	×	○	○	×
		MELSECNET/10	○	○	×	×	○	○	×
		Ethernet	○	○	○	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×
	MELSEC iQ- R series C Controller module	CC-Link IE Controller Network	×	×	×	×	×	×	×
		CC-Link IE Field Network	×	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×	×
		MELSECNET/10	×	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×
	MELSEC-Q series programmabl e controller (Q mode)	CC-Link IE Controller Network ^{*2}	○	○ ^{*3}	×	○	○	×	×
		CC-Link IE Field Network ^{*2}	○	○ ^{*3}	○	○	○ ^{*1}	×	×
		MELSECNET/H	○	○ ^{*3}	×	×	○	○	×
		MELSECNET/10	○	○ ^{*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	×	×	×	×	×	×	×
		MELSECNET/10	×	×	×	×	×	×	×
		MELSECNET(II)	×	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×

*1 The following modules are accessible:

Q12DCCPU-V (Extended mode),
Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG

*2 The station number 65 or higher is accessible only when all of the control CPUs on the network to be routed are Universal model QCPU.

*3 If the connected station CPU is Q00J/Q00/Q01CPU, it is not accessible.

■ Accessible devices

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

Point

- "Batch" and "Random" in the 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)

- Only bit devices are accessible to bit set (mdDevSetEx function) and bit reset (mdDevRstEx function).
- Device extension specifications (digit specification, bit specification or index specification) cannot be used.

- Accessing 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
		R12CCPU-V
Own station link input internal buffer (LX buffer)	Batch/random	×
Own station link output internal buffer (LY buffer)	Batch/random	×
Own station link relay internal buffer (LB buffer)	Batch/random	×
Own station link register internal buffer (LW buffer)	Batch/random	×
RECV function	Batch	○
	Random	×

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

☞ Page 165 Replacement of device type

<For other than the RECV function>

To access the CC-Link IE Controller Network module controlled by C Controller module, use the method explained in the following section. Accessing the own station using CC-Link IE Controller Network communication will result in the station number error, network number error (-16).

☞ Page 165 Replacement of device type

• Accessing other station

No.	Access target
(1)	Basic model QCPU, High Performance model QCPU, Process CPU, Redundant CPU, Universal model QCPU
(2)	Q12DCCPU-V, Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG
(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					
			(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}	×	×	○	○	○
Module refresh 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					
			(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	×	×	×	×	×	×

*1 The following modules are accessible:

Q12DCCPU-V (Extended mode),
Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG

*2 Q00JCPU cannot be accessed.

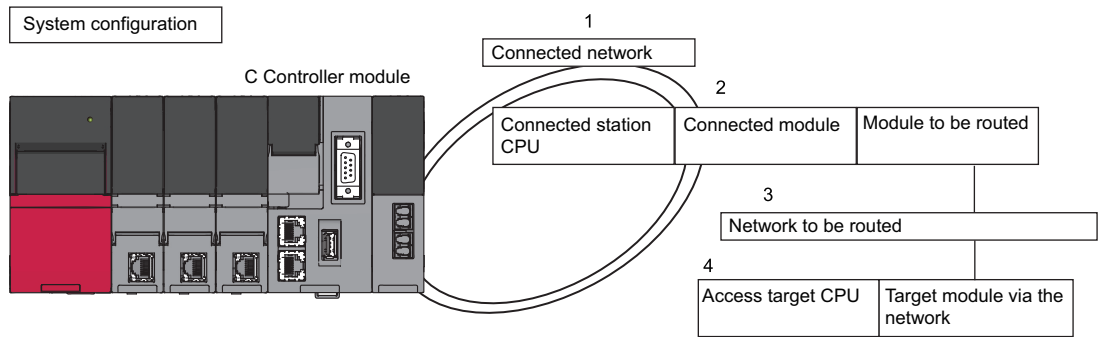
*3 This is a function to send a message to a network module on the other station via CC-Link IE Controller Network module. Access to a multiple CPU system (when a logical station number is specified) is not available.

CC-Link IE Field Network communication

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

■Accessible range

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



■Applicable access

Accessibility is shown in the following table. All of 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	MELSEC iQ- R series programmabl e controller	CC-Link IE Controller Network	○	○	×	○	○	×	×
		CC-Link IE Field Network	○	○	○	○	○ ^{*3}	×	×
		MELSECNET/H	○	○	×	×	○	○	×
		MELSECNET/10	○	○	×	×	○	○	×
		Ethernet	×	×	○	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×
	MELSEC iQ- R Series C Controller module	CC-Link IE Controller Network	×	×	×	×	×	×	×
		CC-Link IE Field Network	×	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×	×
		MELSECNET/10	×	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×
	MELSEC-Q series programmabl e controller (Q mode)	CC-Link IE Controller Network ^{*1}	○	○ ^{*2}	×	○	○	×	×
		CC-Link IE Field Network ^{*1}	○	○ ^{*2}	○	○	○ ^{*3}	×	×
		MELSECNET/H	○	○ ^{*2}	×	×	○	○	×
		MELSECNET/10	○	○ ^{*2}	×	×	○	○	×
		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	×	×	×	×	×	×	×
		MELSECNET/10	×	×	×	×	×	×	×
		MELSECNET(II)	×	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×	×
		Serial communication	×	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×	×

*1 The station number 65 or higher is accessible only when all of the control CPUs on the network to be routed are Universal model QCPU.

*2 If the connected CPU is Q00J/Q00/Q01CPU, it is not accessible.

*3 The following modules are accessible:

Q12DCCPU-V (Extended mode),

Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG

■ Accessible devices

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

Point

- "Batch" and "Random" in the 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)

- Only bit devices are accessible for bit set (mdDevSetEx function) and bit reset (mdDevRstEx function).
- Device extension specifications (digit specification, bit specification or index specification) cannot be used.

- Accessing 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
		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 165 Replacement of device type

<For other than the RECV function>

To access the CC-Link IE Field Network module controlled by C Controller module, use the method explained in the following section. Accessing the own station using CC-Link IE Field Network communication will result in the station number error, network number error (-16).

☞ Page 165 Replacement of device type

• Accessing other station

No.	Access target
(1)	Basic model QCPU, High Performance model QCPU, Process CPU, Redundant CPU, Universal model QCPU
(2)	Q12DCCPU-V, Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG
(3)	WinCPU module, personal computer
(4)	L26CPU-BT, L02CPU, L02CPU-P, L26CPU-PBT, LJ72GF15-T2, NZ2GF-ETB, L02SCPU, L26CPU, L06CPU
(5)	RCPU
(6)	R12CCPU-V

○: Accessible, ×: Not accessible

Device		Access method	Access target					
			(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}	×	×	○	○	○
Module refresh 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					
			(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	×	×	×	×	×	×

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

*2 Q00JCPU cannot be accessed.

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

Considerations

Programming

■Open/close processing of 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.

■Functions

The function which accesses first after the mdOpen function takes longer to be executed since it obtains the information of the CPU module. A dummy access at the first execution saves the succeeding processing time.

■When using user programs

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 from C Controller module, limit it to 32 or less stations on the same task.

Accessing own station devices and other station CPU module devices

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

■Access to own station devices

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 module	When the following conditions are met: <ul style="list-style-type: none"> • Module abnormal (Xn0) is OFF (normal) • Module Ready (XnF) is ON (activated) • Own station data link status (Xn1) bit is ON (data linking)
CC-Link IE Controller Network module	When the following conditions are met:
CC-Link IE Field Network master/local module	<ul style="list-style-type: none"> • Own station data link status (SB49) bit is OFF (data linking) • Cyclic transmission status (bit applicable to the communication target station from SWB0 to B7 read from the own station) is OFF (normal communication)


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 module	When the following conditions are met: <ul style="list-style-type: none"> • The condition that turns the interlock ON when accessing devices in the own station is met. • Data link status of the station to be accessed (bit applicable to the communication target station from SW80 to 83 read from the own station) is OFF (normal communication)
CC-Link IE Controller Network module	When the following conditions are met:
CC-Link IE Field Network master/local module	<ul style="list-style-type: none"> • Own station baton pass status (SB47) bit is OFF (normal) • Baton pass status of the station to be accessed (bit applicable to the communication station from SWA0 to A7 read from own station) is OFF (normal communication)

1.4 Considerations on Interrupt Service Routine

Fully understand the restrictions of VxWorks, the operating system for C Controller module, before creating execution routines from an interrupt service routine (ISR). ( Page 36 C Controller module dedicated functions for ISR)

In addition, to synchronize the other dedicated function provided by C Controller module with an interrupt, create a user program with notification processing implemented, and process it with the task.

Point

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

2 FUNCTION LIST

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

2.1 C Controller Module Dedicated Functions

2

The C Controller module dedicated functions are as listed below.

Function name	Function	Interrupt service routine	Reference
CCPU_ChangeFileSecurity	Changes the access restriction for a file of C Controller module.	—	Page 38 CCPU_ChangeFileSecurity
CCPU_ClearError	Clears errors of C Controller module.	—	Page 39 CCPU_ClearError
CCPU_Control	Performs remote operations (remote RUN/STOP/PAUSE) for the specified CPU module.	—	Page 40 CCPU_Control
CCPU_DedicatedDInst	Executes dedicated instructions categorized as 'D' or 'DP'.	—	Page 41 CCPU_DedicatedDInst
CCPU_DedicatedGInst	Executes dedicated instructions categorized as 'G' or 'GP'.	—	Page 44 CCPU_DedicatedGInst
CCPU_DedicatedJInst	Executes dedicated instructions categorized as 'J' or 'JP'.	—	Page 46 CCPU_DedicatedJInst
CCPU_DedicatedMInst	Executes dedicated instructions categorized as 'M' or 'MP'.	—	Page 48 CCPU_DedicatedMInst
CCPU_DisableInt	Disables a registered routine for each interrupt.	—	Page 50 CCPU_DisableInt
CCPU_EnableInt	Enables a registered routine for each interrupt.	—	Page 51 CCPU_EnableInt
CCPU_EntryInt	Registers a routine that corresponds to each interrupt.	—	Page 52 CCPU_EntryInt
CCPU_EntryTimerEvent	Registers a timer event.	—	Page 54 CCPU_EntryTimerEvent
CCPU_EntryWDTInt	Registers a routine which is called when a user WDT error interrupt of C Controller module occurs.	—	Page 55 CCPU_EntryWDTInt
CCPU_FromBuf	Reads data from the CPU buffer memory and the intelligent function module buffer memory of the module on the specified module position. (FROM instruction)	—	Page 56 CCPU_FromBuf
CCPU_FromBufHG	Reads data from the fixed cycle communication area of the module on the specified module position.	—	Page 57 CCPU_FromBufHG
CCPU_GetConstantProcessStatus	Obtains the fixed cycle processing status of C Controller module.	—	Page 58 CCPU_GetConstantProcessStatus
CCPU_GetCounterMicros	Obtains a 1 μ s counter value of C Controller module.	—	Page 59 CCPU_GetCounterMicros
CCPU_GetCounterMillis	Obtains a 1 ms counter value of C Controller module.	—	Page 60 CCPU_GetCounterMillis
CCPU_GetCpuStatus	Obtains the operating status of C Controller module.	—	Page 61 CCPU_GetCpuStatus
CCPU_GetDotMatrixLED	Obtains the value displayed on the dot matrix LED of C Controller module.	—	Page 62 CCPU_GetDotMatrixLED

Function name	Function	Interrupt service routine	Reference
CCPU_GetErrInfo	Obtains the error information of C Controller module.	—	Page 63 CCPU_GetErrInfo
CCPU_GetFileSecurity	Obtains the file access mode.	—	Page 64 CCPU_GetFileSecurity
CCPU_GetIDInfo	Obtains the individual identification information of C Controller module.	—	Page 65 CCPU_GetIDInfo
CCPU_GetLEDStatus	Obtains the indicator LED status of C Controller module.	—	Page 66 CCPU_GetLEDStatus
CCPU_GetOpSelectMode	Obtains the operation selection mode currently set to C Controller module.	—	Page 68 CCPU_GetOpSelectMode
CCPU_GetPowerStatus	Obtains the power status of C Controller module.	—	Page 69 CCPU_GetPowerStatus
CCPU_GetRTC	Obtains the clock data (local time) of C Controller module.	—	Page 70 CCPU_GetRTC
CCPU_GetSerialNo	Obtains the serial number of C Controller module.	—	Page 71 CCPU_GetSerialNo
CCPU_GetSwitchStatus	Obtains the switch status of C Controller module.	—	Page 72 CCPU_GetSwitchStatus
CCPU_GetUnitInfo	Obtains the module configuration information.	—	Page 73 CCPU_GetUnitInfo
CCPU_MountMemoryCard	Mounts the SD memory card inserted to C Controller module.	—	Page 76 CCPU_MountMemoryCard
CCPU_ReadDevice	Reads data from internal user devices and internal system devices of C Controller module.	—	Page 77 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, or Ethernet module (when CC-Link IE Field Network is selected).	—	Page 78 CCPU_ReadLinkDevice
CCPU_RegistEventLog	Registers event logs in the event history of C Controller module.	—	Page 79 CCPU_RegistEventLog
CCPU_Reset	Resets the bus master CPU (CPU No.1).	—	Page 81 CCPU_Reset
CCPU_ResetDevice	Resets internal user devices and internal system devices (bit devices) of C Controller module.	—	Page 82 CCPU_ResetDevice
CCPU_ResetWDT	Resets the user WDT of C Controller module.	—	Page 83 CCPU_ResetWDT
CCPU_SetDevice	Sets internal user devices and internal system devices (bit devices) of C Controller module.	—	Page 84 CCPU_SetDevice
CCPU_SetDotMatrixLED	Sets a value to be displayed on the dot matrix LED of C Controller module.	—	Page 85 CCPU_SetDotMatrixLED
CCPU_SetLEDStatus	Sets the status of the indicator LED on the C Controller module.	—	Page 87 CCPU_SetLEDStatus
CCPU_SetOpSelectMode	Sets the operation selection mode of C Controller module.	—	Page 88 CCPU_SetOpSelectMode
CCPU_SetRTC	Sets the clock data (local time) of C Controller module.	—	Page 89 CCPU_SetRTC
CCPU_ShutdownRom	Shuts down the program/data memory of C Controller module and allows to power OFF.	—	Page 90 CCPU_ShutdownRom
CCPU_StartWDT	Sets and starts the user WDT of C Controller module.	—	Page 91 CCPU_StartWDT

Function name	Function	Interrupt service routine	Reference
CCPU_StopWDT	Stops the user WDT of C Controller module.	—	Page 92 CCPU_StopWDT
CCPU_SysClkRateGet	Reads the system clock rate specified with the CCPU_SysClkRateSet function from the backup RAM.	—	Page 93 CCPU_SysClkRateGet
CCPU_SysClkRateSet	Stores the specified system clock rate into the backup RAM.	—	Page 94 CCPU_SysClkRateSet
CCPU_ToBuf	Writes data to the CPU buffer memory (host CPU) and the intelligent function module buffer memory of the module on the specified module position. (TO instruction)	—	Page 95 CCPU_ToBuf
CCPU_ToBufHG	Writes data to the fixed cycle communication area of the module on the specified module position.	—	Page 96 CCPU_ToBufHG
CCPU_UnmountMemoryCard	Unmounts the SD memory card inserted to C Controller module.	—	Page 97 CCPU_UnmountMemoryCard
CCPU_WaitEvent	Waits for an interrupt event notification from other CPUs.	—	Page 98 CCPU_WaitEvent
CCPU_WaitSwitchEvent	Waits for a switch interrupt event of C Controller module to occur.	—	Page 100 CCPU_WaitSwitchEvent
CCPU_WaitTimerEvent	Waits for a timer event to occur.	—	Page 101 CCPU_WaitTimerEvent
CCPU_WaitUnitEvent	Waits for an interrupt event notification from modules.	—	Page 102 CCPU_WaitUnitEvent
CCPU_WriteDevice	Writes data to internal user devices and internal system devices of C Controller module.	—	Page 104 CCPU_WriteDevice
CCPU_WriteLinkDevice	Writes data to own station link devices of CC-Link IE Controller Network module, CC-Link IE Field Network module, or Ethernet module (when CC-Link IE Field Network is selected).	—	Page 105 CCPU_WriteLinkDevice
CCPU_X_In_BitEx	Reads the input signal (X) in bit (1-point) units.	—	Page 106 CCPU_X_In_BitEx
CCPU_X_In_WordEx	Reads the input signal (X) in word (16-point) units.	—	Page 107 CCPU_X_In_WordEx
CCPU_Y_In_BitEx	Reads the output signal (Y) in bit (1-point) units.	—	Page 108 CCPU_Y_In_BitEx
CCPU_Y_In_WordEx	Reads the output signal (Y) in word (16-point) units.	—	Page 109 CCPU_Y_In_WordEx
CCPU_Y_Out_BitEx	Outputs the output signal (Y) in bit (1-point) units.	—	Page 110 CCPU_Y_Out_BitEx
CCPU_Y_Out_WordEx	Outputs the output signal (Y) in word (16-point) units.	—	Page 111 CCPU_Y_Out_WordEx

C Controller module dedicated functions for ISR

Function name	Function	Interrupt service routine	Reference
CCPU_DisableInt_ISR	Disables a registered routine for each interrupt.	○	Page 112 CCPU_DisableInt_ISR
CCPU_EnableInt_ISR	Enables a registered routine for each interrupt.	○	Page 113 CCPU_EnableInt_ISR
CCPU_FromBuf_ISR	Reads data from the CPU buffer memory and the intelligent function module buffer memory of the module on the specified module position. (FROM instruction)	○	Page 114 CCPU_FromBuf_ISR
CCPU_FromBufHG_ISR	Reads data from the fixed cycle communication area of the module on the specified module position.	○	Page 115 CCPU_FromBufHG_ISR
CCPU_GetCounterMicros_ISR	Obtains a 1 μs counter value of C Controller module.	○	Page 116 CCPU_GetCounterMicros_ISR
CCPU_GetCounterMillis_ISR	Obtains a 1 ms counter value of C Controller module.	○	Page 117 CCPU_GetCounterMillis_ISR
CCPU_GetDotMatrixLED_ISR	Obtains the value displayed on the dot matrix LED of C Controller module.	○	Page 118 CCPU_GetDotMatrixLED_ISR
CCPU_ReadDevice_ISR	Reads data from internal user devices and internal system devices of C Controller module.	○	Page 119 CCPU_ReadDevice_ISR
CCPU_RegistEventLog_ISR	Registers event logs in the event history of C Controller module.	○	Page 120 CCPU_RegistEventLog_ISR
CCPU_ResetDevice_ISR	Resets internal user devices and internal system devices (bit devices) of C Controller module.	○	Page 119 CCPU_ReadDevice_ISR
CCPU_SetDevice_ISR	Sets internal user devices and internal system devices (bit devices) of C Controller module.	○	Page 123 CCPU_SetDevice_ISR
CCPU_SetDotMatrixLED_ISR	Sets a value to be displayed on the dot matrix LED of C Controller module.	○	Page 124 CCPU_SetDotMatrixLED_ISR
CCPU_SetLEDStatus_ISR	Sets the status of the indicator LED on the C Controller module.	○	Page 126 CCPU_SetLEDStatus_ISR
CCPU_ToBuf_ISR	Writes data to the CPU buffer memory (host CPU) and the intelligent function module buffer memory of the module on the specified module position. (TO instruction)	○	Page 127 CCPU_ToBuf_ISR
CCPU_ToBufHG_ISR	Writes data to the fixed cycle communication area of the module on the specified module position.	○	Page 128 CCPU_ToBufHG_ISR
CCPU_WriteDevice_ISR	Writes data to internal user devices and internal system devices of C Controller module.	○	Page 129 CCPU_WriteDevice_ISR
CCPU_X_In_Word_ISR	Reads the input signal (X) in word (16-point) units.	○	Page 130 CCPU_X_In_Word_ISR
CCPU_Y_In_Word_ISR	Reads the output signal (Y) in word (16-point) units.	○	Page 131 CCPU_Y_In_Word_ISR
CCPU_Y_Out_Word_ISR	Outputs the output signal (Y) in word (16-point) units.	○	Page 132 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.	Page 133 mdClose
mdControl	Performs remote operations (RUN/STOP/PAUSE) for programmable controller CPU.	Page 134 mdControl
mdDevRstEx	Resets bit device(s).	Page 135 mdDevRstEx
mdDevSetEx	Sets bit device(s).	Page 136 mdDevSetEx
mdInit	Initializes the communication route information.	Page 137 mdInit
mdOpen	Opens a communication line.	Page 138 mdOpen
mdRandREx	Reads device(s) randomly.	Page 139 mdRandREx
mdRandWEx	Writes device(s) randomly.	Page 142 mdRandWEx
mdReceiveEx	Reads devices in batch.	Page 144 mdReceiveEx
mdReceiveEx	Receives messages.	Page 145 mdReceiveEx
mdSendEx	Writes devices in batch.	Page 147 mdSendEx
mdSendEx	Sends messages.	Page 148 mdSendEx
mdTypeRead	Reads the model name of CPU module.	Page 150 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.

CCPU_ChangeFileSecurity

This function changes the access restriction for a file of 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 for restricting access to a file. 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

- This function changes the mode of file access to the specified mode to sMode.
- To change the file access mode, use the security password set in the CPU parameter.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 64 CCPU_GetFileSecurity

CCPU_ClearError

This function clears errors of C Controller module.

■Format

short CCPU_ClearError(long* pErrorInfo)


■Argument

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

■Description

- This function clears errors.
- When no error occurs, this function ends normally.
- When a stop error has occurred, the error cannot be cleared by execution of this function. (This function ends normally.)

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 63 CCPU_GetErrInfo

CCPU_Control

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

■Format

short CCPU_Control (short sCpuNo, short sCode)


■Argument

Argument	Name	Description	IN/OUT
sCpuNo	CPU number	Specify the target CPU number. (0: Host CPU, 1 to 4: Other CPUs) Specify '0' to execute the remote operation for the host CPU. When the CPU specified with 1 to 4 is the host CPU, an error is returned.	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 sCpuNo in a multiple CPU system.
- The RUN/STOP/RESET switch has a higher priority to change the status of C Controller module. Therefore, if the RUN/STOP/RESET switch is put in the STOP state, the execution state will remain STOP irrespective of the specified remote operation. Since the remote operation specification by this function is effective even when the RUN/STOP/RESET switch is put in STOP; however, C Controller module operates according to the last specified remote operation status 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

None

CCPU_DedicatedDInst

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

■Format

short CCPU_DedicatedDInst (char* pClnstName, 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
pClnstName	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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} setting data for the dedicated instruction to be executed in word unit. Specify '0' if there is no setting data.	IN

Argument	Name	Description	IN/OUT
psArg9	Setting data (9th) ^{*1}	Specify the ninth ^{*1} 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 ^{*1} 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 pclInstName 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 motion CPU to change the current value.	D.CHGA, DP.CHGA
CHGAS	Requests motion CPU to change the current value.	D.CHGAS, DP.CHGAS
CHGV	Requests motion CPU to change the speed value.	D.CHGV, DP.CHGV
CHGVS	Requests motion CPU to change the speed value.	D.CHGVS, DP.CHGVS
CHGT	Requests 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 motion CPU to start a motion SFC program.	D.SFCS, DP.SFCS
SVST	Requests motion CPU to start a servo program.	D.SVST, DP.SVST
DDWR	Writes data to the device of motion CPU.	D.DDWR, DP.DDWR

*1 The arguments to issue an interrupt from C Controller module using the CCPU_DedicatedDInst function are the same as those to issue an interrupt to C Controller module using the CCPU_DedicatedMInst function. Refer to the following section.

☞ Page 48 CCPU_DedicatedMInst

■Precautions

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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter. ☞ Page 152 ERROR CODE LIST

■Relevant function

- Page 44 CCPU_DedicatedGInst
- Page 46 CCPU_DedicatedJInst
- Page 48 CCPU_DedicatedMInst

CCPU_DedicatedGInst

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

■Format

short CCPU_DedicatedGInst (char* pcInstName, short sloNo, 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
sloNo	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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 pclnstName 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

3

■Precautions

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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 41 CCPU_DedicatedDInst
- Page 46 CCPU_DedicatedJInst
- Page 48 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 `pclInstName` are as shown below. For the specifications of each dedicated function and the completion status^{*1}, 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 for the REMTO/REMFR instruction is stored in SW0080 to SW009F instead of the completion status. Obtain the error codes stored in SW0080 to SW009F by executing the `CCPU_ReadLinkDevice` function.

■Precautions

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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 41 `CCPU_DedicatedDInst`
- Page 44 `CCPU_DedicatedGInst`
- Page 48 `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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 ^{*1} 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 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*1	Issues an interrupt to motion CPU and C Controller module.	M.GINT, MP.GINT

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


Argument	Description
pcInstName	Specify "GINT".
sCPUNo	Specify 3E0H as CPU No.1, 3E1H as CPU No.2, 3E2H as CPU No.3, or 3E3H as CPU No.4.
psArg1	Specify the interrupt pointer number 0 to 15 to psArg1[0].
sArg1Size	Specify '1'.
psArg2	Specify the area to store the completion device (2 words). ^{*2} The psArg2[1] will be '0' and the psArg2[0] will be '1' upon the completion of the instruction processing. The psArg2[1] will be '1', and the psArg2[0] will be '1' upon the abnormal completion of the instruction processing.
sArg2Size	Specify '2'. ^{*2}
psArg3	Specify the area to store the completion status (1 word). ^{*2} When the interrupt pointer number specified to psArg1[0] is other than 0 to 15, the psArg3[0] will be '2282H'.
sArg3Size	Specify '1'. ^{*2}
psArg4 to psArg9	Specify NULL.
sArg4Size to sArg9Size	Specify '0'.

*2 The completion device and completion status can be omitted at the same time. (Omitting only one of them is not available.)
To omit them, specify "NULL" to psArg2 and psArg3, and "0" to sArg2Size and sArg3Size, respectively.

■Precautions

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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 41 CCPU_DedicatedDInst
- Page 44 CCPU_DedicatedGInst
- Page 46 CCPU_DedicatedJInst

CCPU_DisableInt

This function disables a registered routine for each interrupt.

■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 registered routine using the CCPU_EntryInt function. (The routine is not executed when interrupt occurs.)
- Specify the interrupt pointer number specified with the CCPU_EntryInt function to 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
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 52 CCPU_EntryInt
- Page 51 CCPU_EnableInt

CCPU_EnableInt

This function enables a registered routine for each interrupt.

■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 registered routine using the CCPU_EntryInt function. (The routine is executed when interrupt occurs.)
- Specify the interrupt pointer number specified in the CCPU_EntryInt function to sSINo.
- Since an interrupt does not occur at the time of the stop error, the registered routine will not be executed even if the routine 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
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 52 CCPU_EntryInt
- Page 50 CCPU_DisableInt

CCPU_EntryInt

This function registers a routine that corresponds to each interrupt.

■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*1	Registered routine	Specify the routine pointer to be registered (Deregistered by specifying NULL.)	IN

*1 The data type is defined as void type in the include file, "CCPUFunc.h" of the C Controller module dedicated functions.

Set sSINo in the following format.

Interrupt pointer number	Interrupt factor
0 to 15	Interrupt by module
44	Inter-module synchronization interrupt
45	Multiple CPU synchronization interrupt
50 to 1023	Interrupt by module

An error occurs if the interrupt point number other than above is specified.

■Description

- This function registers the routine specified to pFuncPtr in the interrupt specified with interrupt pointer number.
- When NULL is specified to pFuncPtr, the routine is deregistered.
- Use the CCPU_EnableInt function and enable the registered routine with this function.

Precautions

■When the operating system is in an interrupt disabled state

- The registered routine is not executed.


■Processing of a routine to be registered in pFuncPtr

- The routine to be registered must not have an argument. (Prevent an argument from being passed from an interrupt.)
- When registering a routine in pFuncPtr, observe the considerations on interrupt service routine (ISR).
- Register minimal routine processing so that the processing time is as short as possible.
- The following functions can be used for a routine to be registered
C Controller module dedicated functions executable from an interrupt service routine (ISR)
Do not use C Controller module dedicated functions other than the above, or MELSEC data link functions in the routine to be registered. (A function to be registered is not checked for an error.)
- The routine is disabled after the registration is done by this function.
- During the operation of routine registration, calling/execution of the following interrupt service routine (ISR) is delayed.
Routine registered in an interrupt from other CPUs
Routine registered in a multiple CPU synchronization interrupt
Routine registered in a user WDT error interrupt

■When this function is executed more than once with the same interrupt pointer number specified

- The routine specified to pFuncPtr last is registered. (Multiple routines cannot be registered.)

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 51 CCPU_EnableInt
- Page 50 CCPU_DisableInt

CCPU_EntryTimerEvent

This function registers a timer event.

■Format

short CCPU_EntryTimerEvent (long* pEvent)

■Argument

Argument	Name	Description	IN/OUT
pEvent	Registered event	Specify a timer event to be registered.	IN
pEvent	Description		
pEvent[0]	Number of timer event settings (1 to 16)		
pEvent[1]	First timer event number (1 to 16)	First event setting	
pEvent[2]	Cycle of the first timer event (0: Clear, 1 to 60000: Cycle [ms])		
pEvent[3]	Synchronization type of the first timer event (0: Batch synchronization, 1: Individual synchronization)		
pEvent[4]	Second timer event number (1 to 16)	Second event setting	
pEvent[5]	Cycle of the second timer event (0: Clear, 1 to 60000: Cycle [ms])		
pEvent[6]	Synchronization type of the second timer event (0: Batch synchronization, 1: Individual synchronization)		
pEvent[7]	Third timer event number (1 to 16)	Third event setting	
pEvent[8]	Cycle of the third timer event (0: Clear, 1 to 60000: Cycle [ms])		
pEvent[9]	Synchronization type of the third timer event (0: Batch synchronization, 1: Individual synchronization)		
.	.	.	.
.	.	.	.

Only a multiple of 5 (in 5 ms units), or only a multiple of 1000 (in 1 s units) is valid when 10 to 1000, or 1000 to 60000 is specified respectively to the cycle. Otherwise, an error will be returned.

■Description

- This function sets the cycle/synchronization type for the timer event registration.
- When '0' is specified to pEvent, 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 ms to 60 s) and synchronization type (batch synchronization or individual synchronization) can be specified for each event. For the synchronization type, refer to the description of the CCPU_WaitTimerEvent function.
- Specify the timer event number to pEvent without duplication. Otherwise, an error will be returned.
- To change the cycle that is already set to a timer event number, clear the registration of the timer event (specify '0' to the cycle), and then register the cycle (set the cycle) again. Otherwise, an error will be returned.
- The timer event registered by this function waits for the event with 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 101 CCPU_WaitTimerEvent

CCPU_EntryWDTInt

This function registers a routine which is called when a user WDT error interrupt of C Controller module 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*1	Registered routine	Specify the routine to be registered. The routine is deregistered by specifying NULL.	IN

*1 The data type is defined as void type in the include file, "CCPUFunc.h" of the C Controller module dedicated functions.

■Description

- This function registers a routine which is called when a user WDT error interrupt occurs.
- Specify the routine to be registered to 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

■When the operating system is in an interrupt disabled state


The registered routine is not executed.

■Processing of a routine to be registered in pFuncPtr

- The routine to be registered in pFuncPtr must not have an argument. (Prevent an argument from being passed from an interrupt.)
- When registering the routine in pFuncPtr, observe the considerations on interrupt service routine (ISR).
- Register minimal routine processing so that the processing time is as short as possible.

■Functions can be used for the routine to be registered in pFuncPtr


The C Controller module dedicated functions executable from an interrupt service routine (ISR) can be used for the routine to be registered in pFuncPtr.

 Page 32 Considerations on Interrupt Service Routine


Do not use C Controller module dedicated functions other than the interrupt service routine (ISR) or MELSEC data link functions. (A function to be registered is not checked for an error.)

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. For more details on the considerations on interrupt service routine (ISR), refer to the following section.

 Page 32 Considerations on Interrupt Service Routine

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 91 CCPU_StartWDT
- Page 83 CCPU_ResetWDT
- Page 92 CCPU_StopWDT

CCPU_FromBuf

This function reads data from the CPU buffer memory and the intelligent function module buffer memory of the module 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 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
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 ulSize (words) from the CPU buffer memory of the CPU number and the buffer memory of the Intelligent function module specified by the start I/O number divided by 16, and stores the read data in pusDataBuf. Data is read from the specified offset by offset address (ulOffset) to the start address of the CPU buffer memory and the intelligent function module buffer memory. Note that the size of ulBufSize should be equal to or bigger than the ulSize.
- 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 usIoNo. However, the CPU buffer memory can be accessed only when the multiple CPU setting is configured.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- None

CCPU_FromBufHG

This function reads data from the fixed cycle communication area of the module 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 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
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 ulSize (words) from the fixed cycle communication area of the CPU number specified by the start I/O number divided by 16, and stored the read data to pusDataBuf. Data is read from the specified offset by offset address (ulOffset) to the start of the fixed cycle communication area. Note that the size of ulBufSize should be equal to or bigger than the ulSize.
- The fixed cycle communication area can be accessed only when the fixed cycle communication area setting under the multiple CPU setting is configured.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 56 CCPU_FromBuf
- Page 95 CCPU_ToBuf
- Page 96 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 pusStatusBuf.
- This function obtains the information for the size specified to ulBufSize.
- The information to be stored in pusStatusBuf is as follows.

Information storage location	Stored information
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, refer to the following chapter. 📖 Page 152 ERROR CODE LIST

■Relevant functions

- Page 63 CCPU_GetErrInfo

CCPU_GetCounterMicros

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

■Format

short CCPU_GetCounterMicros(unsigned long* pulMicros)


■Argument

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

■Description

- This function obtains a 1 μ s counter value of C Controller module and stores the value in pulMicros.
- The 1 μ s counter value increases by 1 every 1 μ s 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 60 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 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 59 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.	IN

■Description

- This function obtains the operating status of C Controller module and stores it to plStatusBuf.
- It obtains the information for the size specified to ulBufSize. When '0' is specified, this function ends normally without processing.
- The information to be stored in plStatusBuf is as follows.
- If information to be stored is not supported, '0' is set as its status.

Information storage location	Stored information	Status
plStatusBuf[0]	bit31-8	Reserved
	bit7-4	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
	bit3-0	CPU operating status 0: RUN state 1: Reserved 2: STOP state 3: PAUSE state Others: Reserved
plStatusBuf[1]	bit31-16	Reserved
	bit15-7	Reserved
	bit6,5	USB mass storage class standard compliant-device status (USB1) 0: Inserted (mounted) 1: Inserted (unmounted) 2: Not inserted
	bit4,3	SD memory card status 0: Inserted (mounted) 1: Inserted (unmounted) 2: Not inserted
	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]	bit31-0	Index value for number of data memory write cycle

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 63 CCPU_GetErrInfo

CCPU_GetDotMatrixLED

This function obtains the value displayed on the dot matrix LED of C Controller module.

■Format

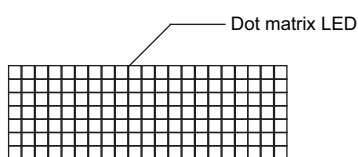
short CCPU_GetDotMatrixLED(char* pcData, unsigned long ulDataSize)

■Argument

Argument	Name	Description	IN/OUT
pcData	LED data	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 pcData.
- It obtains the information for the size specified to ulDataSize.
- The LED data are stored in pcData as shown below.



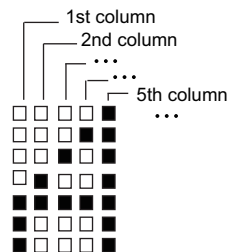
pcData[0] to pcData[19]: Data of the dot matrix LED (7 × 20)

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, refer to the following chapter. ☞ Page 152 ERROR CODE LIST

■Relevant function

- Page 85 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 pusErrorInfo.
- It obtains the information for the size specified to ulBufSize.
- The information to be stored in pusErrorInfo is as follows.
- The error code for an error occurred in the self-diagnostics (up to 16 types) is stored in order starting from pusErrorInfo[0].
- The error code which has already been stored is not stored.

Information storage location	Stored information
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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 39 CCPU_ClearError

CCPU_GetFileSecurity

This function obtains the file access mode.

■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 psMode.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 38 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 pucGetData.
- It obtains information for the size specified to ulBufSize.
- The individual identification information is stored in pucGetData as shown below.

Argument	Individual identification information
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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 71 CCPU_GetSerialNo

CCPU_GetLEDStatus

This function obtains the indicator LED status of 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.	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

- Specify lLed as follows.

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

When 'Reserved' is specified, this function ends normally without processing.

■Description

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

lLed: 0 to 6

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)

lLed: -1

Information storage location	Stored information
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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 63 CCPU_GetErrInfo

CCPU_GetOpSelectMode

This function obtains the operation selection mode currently set to C Controller module.

■Format

short CCPU_GetOpSelectMode(long IModelInfo, long* pSelectMode)

■Argument

Argument	Name	Description	IN/OUT
IModelInfo	Mode information	Specify the mode information.	IN
pSelectMode	Operation selection mode	Specify the storage destination of the obtained operation mode.	OUT

- Specify the argument as follows:

IModelInfo	Mode information
1	Operation selection mode when holding the MODE/SELECT switch in the SELECT position
2	Display mode of dot matrix LED
Others	Reserved

■Description


- This function stores the operation selection mode set to C Controller module to pSelectMode.
- The information which is stored to pSelectMode when 1 is specified to IModelInfo is as follows:

pSelectMode	Operation selection information
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 devices forcibly when holding the MODE/SELECT switch in the SELECT position.
4	Unmounts SD memory card/USB devices forcibly when holding the MODE/SELECT switch in the SELECT position.

- The information which is stored to pSelectMode when 2 is specified to IModelInfo is as follows:

pSelectMode	Operation selection information
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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 88 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 plStatusBuf.
- This function obtains the information for the size specified to ulBufSize.
- The information to be stored in plStatusBuf is as follows.

Information storage location		Stored information	Status
plStatusBuf[0]	bit31-16	Reserved	—
	bit15-0	Number of detected momentary power failures	—

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 63 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 psGetData.
- It obtains the information for the size specified in ulBufSize.
- The clock data (local time) are stored in psGetData as follows.

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

Information storage location	Stored information
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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 89 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 pcGetData.
- This function obtains the information for the size specified to ulDataSize.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 65 CCPU_GetIDInfo

CCPU_GetSwitchStatus

This function obtains the switch status of C Controller module.

■Format

short CCPU_GetSwitchStatus(long* pIStatusBuf, unsigned long ulBufSize)

■Argument


Argument	Name	Description	IN/OUT
pIStatusBuf	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 pIStatusBuf.
- This function obtains the information for the size specified to ulBufSize.
- The information to be stored in pIStatusBuf is as follows.

Information storage location		Stored information	Status
pIStatusBuf[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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

None

CCPU_GetUnitInfo

This function obtains the module configuration information.

■Format

```
short CCPU GetUnitInfo (unsigned short* pusUnitInfo1, unsigned short* pusUnitInfo2, unsigned short* pusUnitInfo3)
```

■Argument

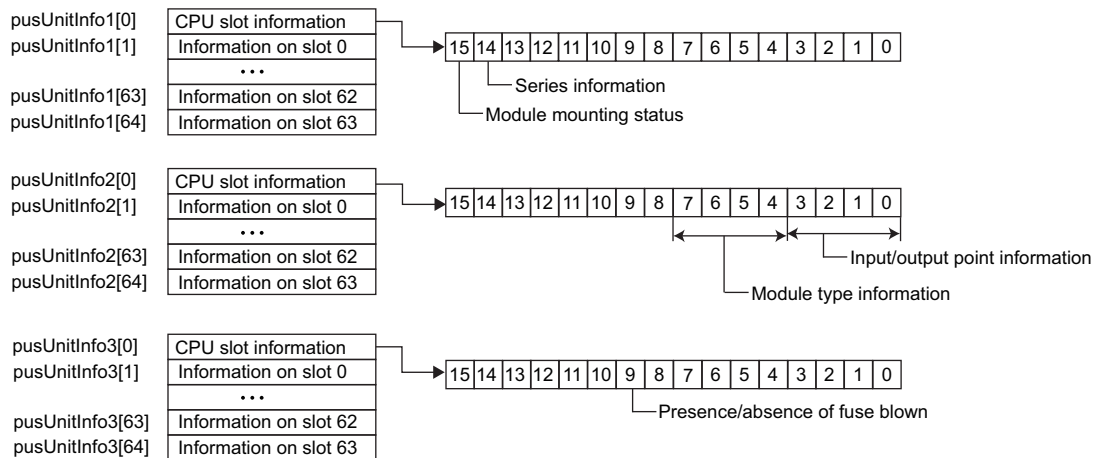
Argument	Name	Description	IN/OUT
pusUnitInfo1	Module configuration information 1	Specify the storage destination of module configuration information 1.	OUT
pusUnitInfo2	Module configuration information 2	Specify the storage destination of module configuration information 2.	OUT
pusUnitInfo3	Module configuration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

■Description

- This function reads module configuration information (65 slots) and stores to pusUnitInfo1, pusUnitInfo2, and pusUnitInfo3. The read module configuration information is stored as shown below. '0' is stored to 'Reserve'.

■Series information is MELSEC iQ-R series

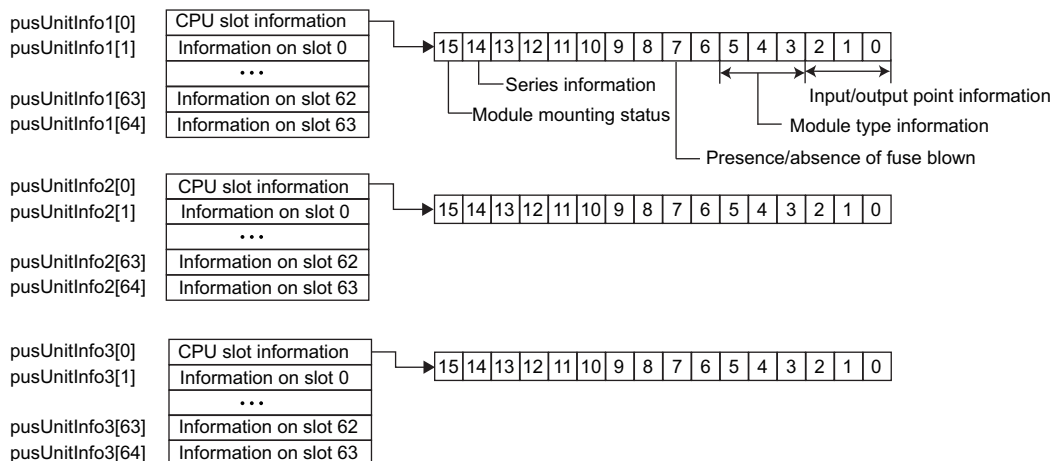
The series can be checked in the 14th bit of pusUnitInfo1[0-64].



Information storage location		Stored information	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
	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].



Information storage location		Stored information	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, refer to the following chapter. 📖 Page 152 ERROR CODE LIST

■Relevant functions

- None

CCPU_MountMemoryCard

This function mounts the SD memory card inserted to the 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 SD memory card in the drive specified to sDrive.
- The CARD READY LED keeps flashing during the mount processing, and it turns ON once the mount processing is completed.
- This function is available when the memory card status is "Inserted (unmounted)". (The SD memory card status can be confirmed by the CCPU_GetCpuStatus function.)
- When the SD memory card has already been mounted, this function ends normally without processing.
- Use this function to access (read/write data from/to) the SD memory card without the replacement after unmounting the SD memory card with the CCPU_UnmountMemoryCard function while the power is ON. (When the SD memory card is inserted, it is mounted automatically. Therefore, this function is unnecessary to be executed.)
- The USB Mass Storage Class standard-compliant devices (USB1) cannot be mounted using this function.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 97 CCPU_UnmountMemoryCard
- Page 61 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. For the specification method, refer to the following section.  Page 8 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 ulSize (words) from the specified device (sDevType, ulDevNo) and subsequent device of C Controller module, and the read data is stored in pusDataBuf. Note that the size of ulBufSize should be equal to or bigger than the ulSize.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 104 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, or Ethernet module (when CC-Link IE Field Network is selected).

■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. For the specification method, refer to the following section.  Page 8 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 ulSize (words) from the specified link device (sDevType, ulDevNo) and subsequent devices of CC-Link IE Controller Network module, CC-Link IE Field Network module, or Ethernet module (when CC-Link IE Field Network is selected) that is specified by the start I/O number divided by 16, and the read data is stored in pusDataBuf. Note that the size of ulBufSize should be equal to or bigger than the ulSize.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 105 CCPU_WriteLinkDevice

CCPU_RegistEventLog

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

■Format

short CCPU_RegistEventLog (long IEventCode, char* pcEventMsg)

■Argument

Argument	Name	Description	IN/OUT
IEventCode	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 are as follows.

- ① Occurrence Date: Event registered date and time
- ② Event Type: Operation (Fixed)
- ③ Status: Information (Fixed)
- ④ Event Code: 25000 (Fixed)
- ⑤ Overview: Registration from the user program (Fixed)
- ⑥ Source: R12CCPU-V (Fixed)
- ⑦ Start I/O No.: Input/output number of the C Controller module that executed the CCPU_RegistEventLog function.
- ⑧ Detailed event code information: Detailed code specified to IEventCode (hexadecimal)
- ⑨ Detailed event log information: Detailed information specified to pcEventMsg
- ⑩ Cause: The event history was registered from the C Controller module detailed function. (Fixed)

No.	Occurrence Date ①	Event Type ②	Status ③	Event Code ④	Overview ⑤	
00001	2015/02/28 15:11:42.090	Operation		25000	Registration from the user program	R
00002	2015/02/28 15:10:33.435	Operation		24100	Operating status change (RUN)	R
00003	2015/02/28 15:10:32.020	System		00400	Power-on and reset	R
00004	2015/02/28 15:10:17.117	System		01000	Power shutoff	R
00005	2015/02/28 15:10:06.534	Operation		20200	Event history clear	R
<div><div></div><div></div><div></div></div>						

Legend

Major

Moderate

Minor

Warning

Information

Clear All

Detailed Information	Detailed event code information	Detailed event log information	-
	Detail code :12345678	Detail information :Detail information of event history.	-
Cause	The event history was registered from the C controller module dedicated function.		
Corrective Action	-		

	Event Type	Status	Event Code	Overview	Source ⁶	Start I/O No. ⁷
090	Operation		25000	Registration from the user program	R12CCPU-V	3E00
435	Operation		24100	Operating status change (RUN)	R12CCPU-V	3E00
020	System		00400	Power-on and reset	R12CCPU-V	3E00
117	System		01000	Power shutoff	R12CCPU-V	3E00
534	Operation		20200	Event history clear	R12CCPU-V	3E00

Legend
 Major
 Moderate
 Minor
 Warning
 Information

Clear All

Detailed Information	Detailed event code information ⁸	Detailed event log information ⁹	-
	Detail code :12345678	Detail information :Detail information of event history.	-
Cause ¹⁰	The event history was registered from the C controller module dedicated function.		
Corrective Action	-		

- Note that the event history can be stored for the size of the event history file specified with CW Configurator. (Old data are deleted if the event history exceeds the event history file size).

An error occurs if the character string data specified to pcEventMsg is 201 bytes or bigger.

Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter. Page 152 ERROR CODE LIST

Relevant functions

None

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 and restart the bus master CPU due to an error other others.
- If this function is executed while a user file in the program memory or SD memory card is being written, data may be corrupted or a file system failure may occur.
- Perform the following before executing this function while a user file is being written.

When a file in the program memory is being written: Close the file.

When a file in the SD memory card is being written: Close the file and unmount the SD memory card.

- This function can be executed only when all the following conditions are met.

If any condition is not met, the error code indicated in the parentheses will be returned.

Host CPU is the bus master CPU (CPU No.1):

The parameter ("Enable" is set to "Remote Reset") is set for the bus master CPU (CPU No.1). (Unset: 16523 is returned.)

The operating status of the bus master CPU (CPU No.1) is 'STOP'. (If the operating status is 'RUN' or 'PAUSE', 16400 will be returned.)

Host CPU is other than the bus master CPU (CPU No.1):

The parameter ("Enable" is set to "Remote Reset") is set for the bus master CPU (CPU No.1). (Unset: -222 is returned.)

The operating status of the bus master CPU (CPU No.1) is 'STOP'. (If the operating status is 'RUN' or 'PAUSE', -222 will be returned.)

The bus master CPU (CPU No.1) is a CPU module. (If the bus master CPU is C Controller module, -222 will be returned.)

Precautions


When remote STOP is executed for the CPU module (bus master CPU (CPU No.1)) from other peripheral devices (such as GX Works3), the CPU module (bus master CPU (CPU No.1)) cannot be reset.

For details on the relationship between the remote operations and the operating status of C Controller module, refer to the following manual.

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

- Note that no value is returned because the module is rebooted from the operating system when this function is executed normally. (All programs are forcibly terminated.)

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 97 CCPU_UnmountMemoryCard
- Page 90 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. For the specification method, refer to the following section.  Page 8 Argument specification	IN
ulDevNo	Start device number	Specify the start device number.	IN

■Description

- This function resets (turns OFF) the specified bit device (sDevType, ulDevNo) of C Controller module.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 84 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 91 CCPU_StartWDT
- Page 92 CCPU_StopWDT
- Page 55 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. For the specification method, refer to the following section.  Page 8 Argument specification	IN
ulDevNo	Device number	Specify the device number.	IN

■Description

- This function sets (turns ON) the specified bit device (sDevType, ulDevNo) of C Controller module.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 82 CCPU_ResetDevice

CCPU_SetDotMatrixLED

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

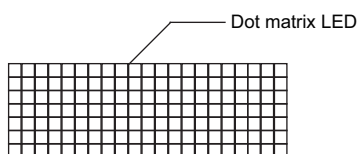
■Format

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 pcData as follows.
- Mode 0: Dot mode



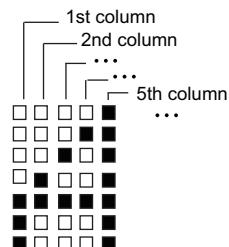
pcData[0] to pcData[19]: Data of the dot matrix LED (7 × 20)

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 pcData on the dot matrix LED according to the method specified to usLedMode.


Precautions

- To display data on the dot matrix LED, selecting 'USER' in the operation selection mode is required. For the selection method in the operation selection mode, refer to the following manual.

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

- Even when "USER" has been selected in the operation selection mode, this function results in an error when selecting the operation or checking the operation being selected with the MODE/SELECT switch.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 62 CCPU_GetDotMatrixLED

CCPU_SetLEDStatus

This function sets the indicator LED status of 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

- Specify usLedInfo as follows.

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 usLedInfo.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 66 CCPU_GetLEDStatus

CCPU_SetOpSelectMode

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

■Format

short CCPU_SetOpSelectMode(long IModelInfo, long ISelectMode)

■Argument

Argument	Name	Description	IN/OUT
IModelInfo	Mode information	Specify the mode information.	IN
ISelectMode	Operation selection mode	Specify the operation selection mode.	IN


- Specify the arguments as follows:

IModelInfo	ISelectMode	Operation selection information
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 devices forcibly when holding the MODE/SELECT switch in the SELECT position.
	4	Unmounts SD memory card/USB 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 ISelectMode.
- 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 68 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 psSetData as follows.

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

Information storage location	Stored information
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 psSetData to C Controller module.
- If the value of 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.

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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 70 CCPU_GetRTC

CCPU_ShutdownRom

This function shuts down the program/data memory of C Controller module and allows to power OFF.

■Format

short CCPU_ShutdownRom (void)


■Argument

None

■Description

- This function shuts down the program/data memory of C Controller module and allows to power OFF with the BUS RUN LED flashing (high speed). (The shutdown status can be checked with the function, CCPU_GetCpuStatus.)
- This function is used to perform the shutdown operation before turning the power OFF, and disables file operations (creating, deleting, and overwriting a file) on the program memory and data memory. Reference to the program memory and data memory is possible.
- Before calling this function, it is necessary to ensure the access to (reading from/writing to) the program/data memory stop processing and all files close processing is called. Otherwise, data in the program/data memory may corrupt or file system error may occur.
- When the program/data memory is in the shutdown completed status, this function ends normally without processing.
- Always turn the power OFF or reset CPU after executing this function and check the shutdown completed status. Operation without power OFF or CPU reset will cause an error at the time of writing files to the program/data memory.
- The shutdown is performed in the order of the program memory and the data memory. If the program memory fails to be shut down, both the program memory and data memory will not be shut down.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 76 CCPU_MountMemoryCard
- Page 97 CCPU_UnmountMemoryCard

CCPU_StartWDT

This function sets and starts the user WDT of 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 ms to 10000 ms [10 s]).	IN

■Description

- This function sets an interval of the WDT to $sInterval \times 10$ ms and starts the user WDT.
- The user WDT is the timer for detecting a hardware failure or program error. When the WDT is not reset periodically within the set time (by execution of the CCPU_ResetWDT function), the user WDT error will occur. When the user WDT error occurs, C Controller module will be in the stop error status. In addition, 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 83 CCPU_ResetWDT
- Page 92 CCPU_StopWDT
- Page 55 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 91 CCPU_StartWDT
- Page 83 CCPU_ResetWDT
- Page 55 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.
- The read value may not correspond to the actual system clock rate in operation. To check the actual system clock rate in operation, use the sysClkRateGet function of VxWorks.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

- Page 94 CCPU_SysClkRateSet

CCPU_SysClkRateSet

This function stores the specified system clock rate into the backup RAM. The specified system clock rate will be enabled after C Controller module is restarted.

■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 C Controller module after the execution of this function. Specify 'NULL' when restart of C Controller module is not required. 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 stores the specified system clock rate into the backup RAM.
The specified system clock rate will be enabled after C Controller module is restarted.
- When the output to the restart necessity flag is "0" (restart is not required), continue the application processing.
- When the output to the restart necessity flag is "1" (restart is required), stop the application processing, and restart the C Controller module (power OFF → ON).
- For more details on system clock rate, refer to the manual for VxWorks.

Restriction

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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 93 CCPU_SysClkRateGet

CCPU_ToBuf

This function writes data to the CPU buffer memory (host CPU) and the intelligent function module buffer memory of the module 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. Start I/O number divided by 16 (0H to FFH, 3E0H to 3E3H) Only the host CPU can be accessed.	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	Fixed to 0.	IN

■Description

- This function writes data equivalent to the size specified to ulSize (words) in pusDataBuf to the CPU buffer memory (host CPU) of the CPU number and the buffer memory of the Intelligent function module specified by the start I/O number divided by 16. Data is written to the specified offset by offset address (ulOffset) from the start of the CPU buffer memory (host CPU) and the intelligent function module buffer memory.
- To access the CPU buffer memory (host CPU) of the module in multiple CPU system (CPU No.1 to No.4), specify 3E0H (CPU No.1) to 3E3H (CPU No.4) to usIoNo. However, the CPU buffer memory can be accessed only when the multiple CPU setting is configured.
- When executing this function while the CPU operating status is not RUN, the STOP/PAUSE error (-28640) occurs.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 56 CCPU_FromBuf

CCPU_ToBufHG

This function writes data to the fixed cycle communication area of the module 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	Fixed to 0.	IN

■Description

- This function write data equivalent to the size specified to ulSize (words) in pusDataBuf to the fixed cycle communication area of the CPU number specified by the start I/O number divided by 16. Data is written to the specified offset by 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 CPU operating status is not RUN, the STOP/PAUSE error (-28640) occurs.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 56 CCPU_FromBuf
- Page 95 CCPU_ToBuf
- Page 57 CCPU_FromBufHG

CCPU_UnmountMemoryCard

This function unmounts the SD memory card inserted to the 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 standard-compliant device (USB1) Others: Reserved	IN


■Description

- This function unmounts the SD memory card in the drive specified to sDrive.
- The CARD READY LED keeps flashing during the unmount processing, and it turns OFF after the unmount processing is completed.
- This function is available when the SD memory card status is "Inserted (mounted)". (The SD memory card status can be confirmed by the CCPU_GetCpuStatus function.)
- When the SD memory card has been already unmounted, this function ends normally without processing.

Precautions

- Design the program so that accessing (reading/writing data from/to) the SD memory card is stopped to close all files before this function is called. When this function is executed while the files on the SD memory card are open, data in the SD memory card may be corrupted or a file system error may occur.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 76 CCPU_MountMemoryCard
- Page 61 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

- Specify psEvent 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)
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)
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)
psEvent[6]	Event number (interrupt pointer number) of the third interrupt event (0 to 15)
.	.
.	.

- The following value is stored in 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 notification from other CPU (CPU module/C Controller module) with the event number (interrupt pointer number) specified to psEvent for the amount of time of ulTimeout.
- When multiple interrupt events occur, the interrupt events are notified in order of the event number (ascending order).
- If an interrupt event has already been notified at a time when this function is called, the execution of this function is immediately completed. 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, the user program performs processing as a single interrupt event notification.
- Design the program so that this function is not called simultaneously due to the same event number (interrupt pointer number) specified by multiple tasks. Otherwise, the execution of the interrupt event notified task is unpredictable.
- Specify the module number of CPU module/C Controller module to the CPU number. Otherwise, an error will be returned.
- 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.
- [Example] 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 '0' are returned to psSetEventNo[0] and psSetEventNo[1], respectively.

Precautions

Do not set the clock data of C controller module while executing this function. Otherwise, this function does not operate properly. (This function is not completed in some cases.)

Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

3

Relevant function

- Page 102 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.	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

- Specify the switch interrupt event type of sSwitch as follows:

Value	Interrupt event type
0	RUN switch interrupt event
1	STOP switch interrupt event
2	SELECT switch interrupt event

If any of the value other than above is specified, an error will be returned.

■Description

- This function waits for a switch interrupt event specified to sSwitch.
- If an interrupt event has already been notified at a time when this function is called, the execution of this function is immediately completed.
- 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. For the method for selecting the operation mode, refer to the following manual.

 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- None

CCPU_WaitTimerEvent

This function waits for a timer event to occur.

■Format

short CCPU_WaitTimerEvent (long IEventNo)

■Argument

Argument	Name	Description	IN/OUT
IEventNo	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 number specified to IEventNo 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.

Synchronization type	Operation
0 (Batch synchronization)	<p>Clears the wait status of all tasks waiting for an event with this function.</p> <p>Even if this function is called later, the wait status is not cleared without any waiting tasks at the time of event occurrence.</p>
1 (Individual synchronization)	<p>Clears the wait status of one task among tasks waiting for an event with this function.</p> <p>When multiple tasks wait for the same event, the wait status of the tasks is cleared in the order of the task priority (or in the order which the task starts waiting in the case of the same priority).</p> <p>In case no tasks are waiting at the time of event occurrence, the wait status will be cleared immediately after this function is called.</p>

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	<p>Error</p> <p>For details, refer to the following chapter.</p> <p>📖 Page 152 ERROR CODE LIST</p>

■Relevant function

- Page 54 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). FFFFFFFFH indicates waiting infinitely.	IN
psSetEventNo	Generated event	Stores the generated event. Stores the event number (interrupt pointer number) of the notified interrupt event.	OUT

- Specify psEvent as follows:

psEvent[0] Number of interrupt event settings (1 to 64)

psEvent[1] Event number (interrupt pointer number) of the first interrupt event (interrupt pointer number) (0 to 15, 50 to 1023)

psEvent[2] Event number (interrupt pointer number) of the second interrupt event (interrupt pointer number) (0 to 15, 50 to 1023)

psEvent[3] Event number (interrupt pointer number) of the third interrupt event (interrupt pointer number) (0 to 15, 50 to 1023)

:

■Description

- This function waits for an interrupt event notification from the module with the event number (interrupt pointer number) specified to psEvent for the amount of time of ulTimeout.
- When multiple interrupt events occur, the interrupt events are notified in order of the event number (ascending order).
- If an interrupt event has already been notified at a time when this function is called, the execution of this function is immediately completed. 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, the user program performs processing as a single interrupt event notification.
- Design the program so that this function is not called simultaneously due to the same event number (interrupt pointer number) specified by multiple tasks. Otherwise, the execution of the interrupt event notified task is unpredictable.
- Set the event number (interrupt pointer number) without duplication. Otherwise, the event number error (-225) will be returned.
- When an interrupt event is notified (return value of this function is normal), the event number of the notified interrupt event is returned to psSetEventNo.

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

- The specified timeout value is rounded to tick unit^{*1}. Specify a timeout value of one tick or more.


^{*1} 'tick' indicates a unit of VxWorks system clock. For details on 'tick', refer to the manual of VxWorks.

- An interrupt event will not be notified at the time of the stop error.

Restriction

Do not set the clock data of C controller module while executing this function. Otherwise, this function does not operate properly. (This function is not completed in some cases.)

Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

Relevant function

- Page 98 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. For the specification method, refer to the following section.  Page 8 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	Fixed to 0.	IN

■Description

- This function writes data equivalent to the size specified to ulSize (words) in pusDataBuf to the specified device (sDevType, ulDevNo) and subsequent devices of C Controller module.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 77 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, or Ethernet module (when CC-Link IE Field Network is selected).

■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. For the specification method, refer to the following section.  Page 8 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	Fixed to 0.	IN

■Description

- This function writes data equivalent to the size specified to ulSize (words) in pusDataBuf to the specified link device (sDevType, ulDevNo) and subsequent devices of CC-Link IE Controller Network module, CC-Link IE Field Network module, or Ethernet module (when CC-Link IE Field Network is selected) that is specified by the start I/O number divided by 16.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 78 CCPU_ReadLinkDevice

CCPU_X_In_BitEx

This function reads the 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, other than 0: Reserved)	IN
usXNo	Input number	Specify an input number (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 the input signal (X) corresponding to the specified usXNo in bit (1-point) units.
- The read data (0: OFF/1: ON) are stored in pusData.
- This function operates to the mounted module corresponding to the specified 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 the specified area is "output module", I/O assignment error occurs.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 107 CCPU_X_In_WordEx
- Page 110 CCPU_Y_Out_BitEx
- Page 111 CCPU_Y_Out_WordEx
- Page 108 CCPU_Y_In_BitEx
- Page 109 CCPU_Y_In_WordEx

CCPU_X_In_WordEx

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

■Format

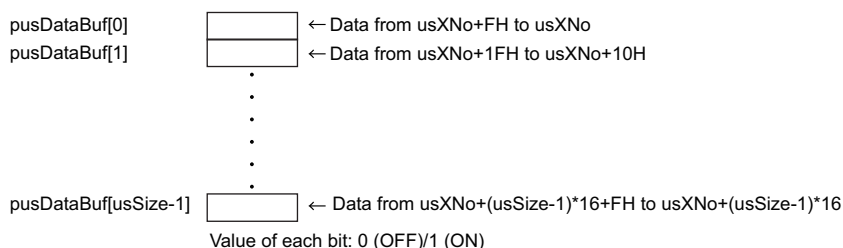
short CCPU_X_In_WordEx (short sFlg, unsigned short usXNo, unsigned short usSize, unsigned short* pusDataBuf, unsigned short usBufSize)

■Argument

Argument	Name	Description	IN/OUT
sFlg	Access flag	Specify an access flag. (0: Normal access, other than 0: Reserved)	IN
usXNo	Start input number	Specify a start input number (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 the input signal (X) equivalent to the size specified to usSize (words) from the start input number (usXNo), and stores the read data in pusDataBuf.
- Specify the reserved pusDataBuf area size (the value which is equal to or bigger than the one specified to usSize) to usBufSize.
- Read data are stored in pusDataBuf in the ascending order of data number from the lower bit as shown below.



- This function operates to the mounted module corresponding to the specified usXNo and usSize 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 the specified area is "output module", I/O assignment error occurs.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter. Page 152 ERROR CODE LIST

■Relevant function

- Page 106 CCPU_X_In_BitEx
- Page 110 CCPU_Y_Out_BitEx
- Page 111 CCPU_Y_Out_WordEx
- Page 108 CCPU_Y_In_BitEx
- Page 109 CCPU_Y_In_WordEx

CCPU_Y_In_BitEx

This function reads the 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, other than 0: Reserved)	IN
usYNo	Output number	Specify an output number (Y).	IN
pusData	Data storage destination	Specify the storage destination of read data. (0: OFF, 1: ON)	OUT

■Description

- This function reads the output signal (Y) corresponding to the specified usYNo in bit (1-point) units.
- The read data (0: OFF/1: ON) are stored in pusData.
- This function operates to the mounted module corresponding to the specified 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 the specified area is "input module", I/O assignment error occurs.
- No error will occur even if this function is executed when the CPU operating status is STOP or PAUSE. The output signal (Y) is read at execution of this function.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 106 CCPU_X_In_BitEx
- Page 107 CCPU_X_In_WordEx
- Page 110 CCPU_Y_Out_BitEx
- Page 111 CCPU_Y_Out_WordEx
- Page 109 CCPU_Y_In_WordEx

CCPU_Y_In_WordEx

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

■Format

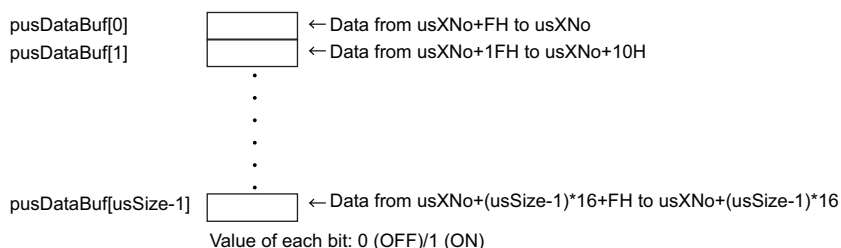
short CCPU_Y_In_WordEx (short sFlg, unsigned short usYNo, unsigned short usSize, unsigned short* pusDataBuf, unsigned short usBufSize)

■Argument

Argument	Name	Description	IN/OUT
sFlg	Access flag	Specify an access flag. (0: Normal access, other than 0: Reserved)	IN
usYNo	Start output number	Specify a start output number (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 the output signal (Y) equivalent to the size specified to usSize (words) from the start output number (usYNo), and stores the read data in pusDataBuf.
- Specify the reserved pusDataBuf area size (the value which is equal to or bigger than the one specified to usSize) to usBufSize.
- Read data are stored in pusDataBuf in the ascending order of data number from the lower bit as shown below.



- This function operates to the mounted module corresponding to the specified usYNo and usSize 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 the specified area is "input module", I/O assignment error occurs.
- No error will occur even if this function is executed when the CPU operating status is STOP or PAUSE. The output signal (Y) is read at execution of this function.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter. 📖 Page 152 ERROR CODE LIST

■Relevant function

- Page 106 CCPU_X_In_BitEx
- Page 107 CCPU_X_In_WordEx
- Page 110 CCPU_Y_Out_BitEx
- Page 111 CCPU_Y_Out_WordEx
- Page 108 CCPU_Y_In_BitEx

CCPU_Y_Out_BitEx

This function outputs the 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, other than 0: Reserved)	IN
usYNo	Output number	Specify an output number (Y).	IN
usData	Data storage destination	Specify the storage destination of output data. (The value of bit 0 is 0: OFF, 1: ON)	IN

■Description

- This function outputs (turns ON/OFF) the output signal (Y) corresponding to the specified usYNo in bit (1-point) units.
- '0' (OFF)'1' (ON) is output according to the value specified to bit 0 in usData (values of bit 1 to 7 are ignored).
- When executing this function while the CPU operating status is not RUN, the STOP/PAUSE error (-28640) occurs.
- When this function is used to "input module", I/O assignment error occurs.
- Do not specify the usYNo corresponding to an output module controlled by the other CPUs. Even when 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 106 CCPU_X_In_BitEx
- Page 107 CCPU_X_In_WordEx
- Page 111 CCPU_Y_Out_WordEx
- Page 108 CCPU_Y_In_BitEx
- Page 109 CCPU_Y_In_WordEx

CCPU_Y_Out_WordEx

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

■Format

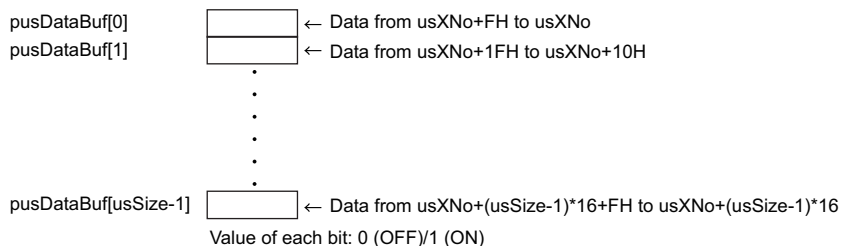
short CCPU_Y_Out_WordEx (short sFlg, unsigned short usYNo, unsigned short usSize, unsigned short* pusDataBuf, unsigned short usBufSize)

■Argument

Argument	Name	Description	IN/OUT
sFlg	Access flag	Specify an access flag. (0: Normal access, other than 0: Reserved)	IN
usYNo	Start output number	Specify a start output number (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	Fixed to 0.	IN

■Description

- This function outputs (turns ON/OFF) the output signal (Y) equivalent to the size specified to usSize from the start output number (usYNo) to pusDataBuf.
- Store the read data in pusDataBuf in the ascending order of data number from the lower bit as shown below.



- When this function is used to "input module", I/O assignment error occurs.
- Do not specify the usYNo corresponding to an output module controlled by the other CPUs. Even when it is specified, no operation is performed to the output module.
- When executing this function while the CPU operating status is not RUN, the STOP/PAUSE error (-28640) occurs.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter. 📖 Page 152 ERROR CODE LIST

■Relevant function

- Page 106 CCPU_X_In_BitEx
- Page 107 CCPU_X_In_WordEx
- Page 110 CCPU_Y_Out_BitEx
- Page 108 CCPU_Y_In_BitEx
- Page 109 CCPU_Y_In_WordEx

C Controller module dedicated functions for ISR

CCPU_DisableInt_ISR

This function disables a registered routine for each interrupt.

■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 registered routine using the CCPU_EntryInt function. (The routine is not executed when interrupt occurs.)
- Specify the interrupt pointer number specified in the CCPU_EntryInt function to 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 52 CCPU_EntryInt
- Page 113 CCPU_EnableInt_ISR

CCPU_EnableInt_ISR

This function enables a registered routine for each interrupt.

■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 registered routine using the CCPU_EntryInt function. (The routine is executed when interrupt occurs.)
- Specify the interrupt pointer number specified in the CCPU_EntryInt function to sSINo.
- Since an interrupt does not occur at the time of the stop error, the registered routine will not be executed even if the routine 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 52 CCPU_EntryInt
- Page 112 CCPU_DisableInt_ISR

CCPU_FromBuf_ISR

This function reads data from the CPU buffer memory and the intelligent function module buffer memory of the module 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 ulSize (words) from the CPU buffer memory of the CPU number and the buffer memory of the Intelligent function module specified by the start I/O number divided by 16, and stores the read data in pusDataBuf. Data is read from the specified offset by offset address (ulOffset) to the start address of the CPU buffer memory and the intelligent function module buffer memory.
- 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 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 the one registered in the interrupt.


WARNING

■This function does not check the argument specified for reading. Create a program with the following conditions in mind.

- The address specified for the read data is a multiple of 2.
- The 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.

■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
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 127 CCPU_ToBuf_ISR

CCPU_FromBufHG_ISR

This function reads data from the fixed cycle communication area of the module 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 ulSize (words) from the fixed cycle communication area of the CPU number specified by the start I/O number divided by 16, and stored the read data to pusDataBuf. Data is read from the specified offset by offset address (ulOffset) to 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 the one registered in the interrupt.


! WARNING

■This function does not check the argument specified for reading. Create a program with the following conditions in mind.

- The address specified for the read data is a multiple of 2.
- The data area for the size (words) of the read data is reserved.
- A non-existent fixed cycle communication area is not specified.

■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
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 114 CCPU_FromBuf_ISR
- Page 127 CCPU_ToBuf_ISR
- Page 128 CCPU_ToBufHG_ISR

CCPU_GetCounterMicros_ISR

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

■Format

short CCPU_GetCounterMicros_ISR(unsigned long* pulMicros)

■Argument

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

■Description

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

Restriction

- Do not execute this function in a routine other than the one registered in the interrupt.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

- Page 117 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 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 the one registered in the interrupt

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

- Page 116 CCPU_GetCounterMicros_ISR

CCPU_GetDotMatrixLED_ISR

This function obtains the value displayed on the dot matrix LED of C Controller module.

■Format

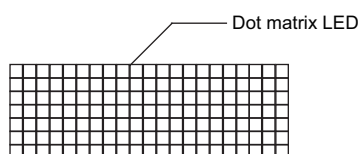
short CCPU_GetDotMatrixLED_ISR (char* pcData, unsigned long ulDataSize)

■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 pcData.
- It obtains the information for the size specified in ulDataSize.
- The LED data are stored in pcData as shown below.



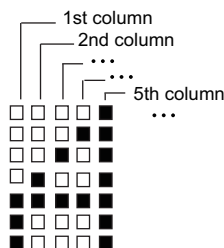
pcData[0] to pcData[19]: Data of the dot matrix LED (7 × 20)

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 the one registered in the interrupt.

⚠ 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 124 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. For the specification method, refer to the following section. ☞ Page 8 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 ulSize (words) from the specified device (sDevType, ulDevNo) and subsequent device of C Controller module, and the read data is stored in pusDataBuf.

Restriction

- Do not execute this function in a routine other than the one registered in the interrupt.

! WARNING

■This function does not check the argument specified for reading. Create a program with the following conditions in mind.

- The data area for the size (words) of the read data is reserved.
- The device within the range is specified.

■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 129 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
IEventCode	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 are as follows.

- ➊ Occurrence Date: Event registered date and time
- ➋ Event Type: Operation (Fixed)
- ➌ Status: Information (Fixed)
- ➍ Event Code: 25000 (Fixed)
- ➎ Overview: Registration from the user program (Fixed)
- ➏ Source: R12CCPU-V (Fixed)
- ➐ Start I/O No.: Input/output number of the C Controller module that executed the CCPU_RegistEventLog_ISR function.
- ➑ Detailed event code information: Detailed code specified to IEventCode (hexadecimal)
- ➒ Detailed event log information: Detailed information specified to pcEventMsg
- ➓ Cause: The event history was registered from the C Controller module detailed function. (Fixed)

No.	Occurrence Date ①	Event Type ②	Status ③	Event Code ④	Overview ⑤
00001	2015/02/28 15:11:42.090	Operation		25000	Registration from the user program
00002	2015/02/28 15:10:33.435	Operation		24100	Operating status change (RUN)
00003	2015/02/28 15:10:32.020	System		00400	Power-on and reset
00004	2015/02/28 15:10:17.117	System		01000	Power shutoff
00005	2015/02/28 15:10:06.534	Operation		20200	Event history clear

Legend

Major
 Moderate
 Minor
 Warning
 Information

Clear All

Detailed Information	Detailed event code information	Detailed event log information	-
	Detail code :12345678	Detail information :Detail information of event history.	-
Cause	The event history was registered from the C controller module dedicated function.		
Corrective Action	-		

	Event Type	Status	Event Code	Overview	Source ⁶	Start I/O No. ⁷
090	Operation		25000	Registration from the user program	R12CCPU-V	3E00
435	Operation		24100	Operating status change (RUN)	R12CCPU-V	3E00
020	System		00400	Power-on and reset	R12CCPU-V	3E00
117	System		01000	Power shutoff	R12CCPU-V	3E00
534	Operation		20200	Event history clear	R12CCPU-V	3E00

Legend
 Major
 Moderate
 Minor
 Warning
 Information

Clear All

Detailed Information	Detailed event code information ⁸	Detailed event log information ⁹	-
	Detail code :12345678	Detail information :Detail information of event history.	-
Cause ¹⁰	The event history was registered from the C controller module dedicated function.		
Corrective Action	-		

- Note that the event history can be stored for the size of the event history file specified with CW Configurator. (Old data are deleted if the event history exceeds the event history file size.)

Restriction

- Do not execute this function in a routine other than the one registered in the interrupt.

WARNING

- This function does not check the specified argument. Create a program with the following conditions in mind.
 - The detailed information within the range is specified.
- 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
Other than 0	Error For details, refer to the following chapter. Page 152 ERROR CODE LIST

Relevant functions

None


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. For the specification method, refer to the following section.  Page 8 Argument specification	IN
ulDevNo	Start device number	Specify the start device number.	IN

■Description

- This function resets (turns OFF) the specified bit device (sDevType, ulDevNo) of C Controller module.

Restriction

- Do not execute this function in a routine other than the one registered in the interrupt.



WARNING

■This function does not check the specified argument. Create a program with the following condition in mind.

- The device within the range is specified.

■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 123 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. For the specification method, refer to the following section.  Page 8 Argument specification	IN
ulDevNo	Device number	Specify the device number.	IN

■Description

- This function sets (turns ON) the specified bit device (sDevType, ulDevNo) of C Controller module.

Restriction

- Do not execute this function in a routine other than the one registered in the interrupt.

WARNING

■This function does not check the specified argument. Create a program with the following condition in mind.

- The device within the range is specified.

■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 122 CCPU_ResetDevice_ISR

CCPU_SetDotMatrixLED_ISR

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

■Format

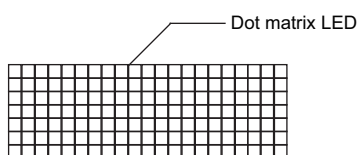
short CCPU_SetDotMatrixLED_ISR (unsigned short usLedMode, char* pcData)

■Argument

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

■Description

- This function displays the value specified to pcData on the dot matrix LED.
- Specify pcData as follows.



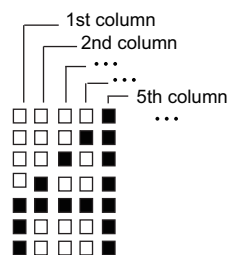
pcData[0] to pcData[19]: Data of the dot matrix LED (7 × 20)

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

- This function can be executed from a routine registered in an interrupt (ISR).
- When this function is used, the screen displayed by selecting [Diagnostics] ⇒ [Module Diagnostics (CPU diagnostics)] of CW Configurator may display an image of the dot matrix LED in which data are being written.

- Do not execute this function in a routine other than the one registered in the interrupt.
- Do not execute this function when the mode other than "USER" is selected in the operation selection mode.
Unintended value may be displayed on the dot matrix LED.
- Do not execute this function while performing the following operation in the operation selection mode.
Unintended value may be displayed on the dot matrix LED.
Selecting an operation
Checking the selected operation



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 118 CCPU_GetDotMatrixLED_ISR

CCPU_SetLEDStatus_ISR

This function sets the indicator LED status of 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 set, the operation is not affected.)	IN
usLedInfo	LED status information	Specify the LED status information.	IN

- Specify usLedInfo as follows.

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 usLedInfo.

Restriction

- Do not execute this function in a routine other than the one registered in the interrupt.

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 87 CCPU_SetLEDStatus

CCPU_ToBuf_ISR

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

■Format

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

■Argument

Argument	Name	Description	IN/OUT
usloNo	Module position	Specify the module position as follows. Start I/O number divided by 16 (0H to FFH, 3E0H to 3E3H) Only the host CPU can be accessed.	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 write data equivalent to the size specified to ulSize (words) in pusDataBuf to the CPU buffer memory (host CPU) of the CPU number and the buffer memory of the Intelligent function module specified by the start I/O number divided by 16. Data is written to the specified offset by offset address (ulOffset) from the start of the CPU buffer memory (host CPU) and the intelligent function module buffer memory
- To access the CPU buffer memory (host CPU) of the module in a multiple CPU configuration (CPU No.1 to No.4), specify 3E0H (CPU No.1) to 3E3H (CPU No.4) to usloNo. However, the CPU buffer memory can be accessed only when the multiple CPU setting is configured.
- When executing this function while the CPU operating status 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 routine other than the ISR, the output value may be overlapped, resulting in an invalid value. Manage the resource such that data is not written to the same CPU buffer memory (host CPU).


! WARNING

■This function does not check the argument specified for writing. Create a program with the following conditions in mind.

- The address specified for 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.

■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
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 114 CCPU_FromBuf_ISR

CCPU_ToBufHG_ISR

This function writes data to the fixed cycle communication area of the module 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 write data equivalent to the size specified to ulSize (words) in pusDataBuf to the fixed cycle communication area of the CPU number specified by the start I/O number divided by 16. Data is written to the specified offset by 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 CPU operating status 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 fixed cycle communication area from routine other than ISR, the output value may be overlapped, resulting in an invalid value. Manage the resource such that data is not written to the same fixed cycle communication area.


WARNING

■This function does not check the argument specified for writing. Create a program with the following conditions in mind.

- The address specified for the write data is a multiple of 2.
- A non-existent fixed cycle communication area is not specified.

■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
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 114 CCPU_FromBuf_ISR
- Page 127 CCPU_ToBuf_ISR
- Page 115 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. For the specification method, refer to the following section.  Page 8 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 equivalent to the size specified to ulSize (words) in pusDataBuf to the specified device (sDevType, ulDevNo) and subsequent devices of C Controller module.

Restriction

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

WARNING

■This function does not check the argument specified for writing. Create a program with the following conditions in mind.

- The address specified for the write data is a multiple of 2.
- The device within the range is specified.

■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 119 CCPU_ReadDevice_ISR

CCPU_X_In_Word_ISR

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

■Format

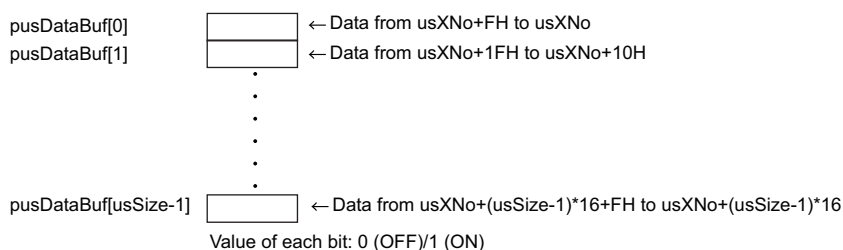
short CCPU_X_In_Word_ISR (unsigned short usXNo, unsigned short usSize, unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
usXNo	Start input number	Specify a start input number (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 reads the input signal (X) equivalent to the size specified to usSize (words) from the start input number (usXNo), and stores the read data in pusDataBuf.
- Specify usXNo in multiples of 16. (The remainder divided by 16 is discarded. Example: 0x05 → 0x00, 0x123 → 0x120)
- Specify the reserved pusDataBuf area size (the value bigger than the one specified to usSize) to usBufSize.
- Read data are stored in pusDataBuf in the ascending order of data number from the lower bit as shown below.



- This function operates to the mounted module corresponding to the specified usXNo and usSize 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 (that is, the read data is '0'). (The setting to import the out-group input status for the multiple CPU setting is ignored.)

Restriction

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


WARNING

■This function does not check the argument specified for reading. Create a program with the following conditions in mind.

- The address specified for the read data is a multiple of 2.
- The data area for the size (words) of the read data is reserved.
- The input signal (X) within the range (0H to FFFH) is specified.

■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
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 131 CCPU_Y_In_Word_ISR
- Page 132 CCPU_Y_Out_Word_ISR

CCPU_Y_In_Word_ISR

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

■Format

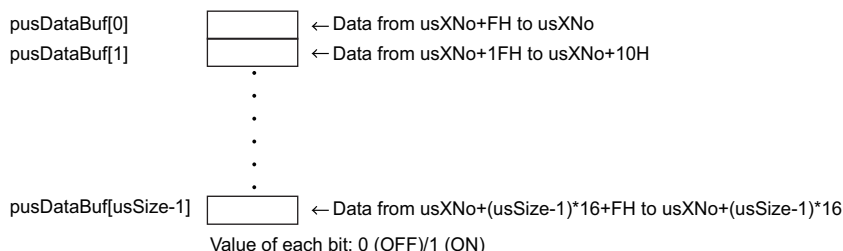
short CCPU_Y_In_Word_ISR (unsigned short usYNo, unsigned short usSize, unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
usYNo	Start output number	Specify a start output number (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 reads the output signal (Y) equivalent to the size specified to usSize (words) from the start output number (usYNo), and stores the read data in pusDataBuf.
- Specify usYNo in multiples of 16. (The remainder divided by 16 is discarded. Example: 0x05 → 0x00, 0x123 → 0x120)
- Read data are stored in pusDataBuf in the ascending order of data number from the lower bit as shown below.



- This function operates to the mounted module corresponding to the specified usYNo and usSize 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 output status controlled by other CPUs is not imported (that is, the read data is '0'). (The setting to import the out-group input status for the multiple CPU setting is ignored.)

Restriction

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

! WARNING

■This function does not check the argument specified for reading. Create a program with the following conditions in mind.

- The address specified for the read data is a multiple of 2.
- The data area for the size (words) of the read data is reserved.
- The output signal (Y) within the range (0H to FFFH) is specified.

■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
Other than 0	Error For details, refer to the following chapter. Page 152 ERROR CODE LIST

■Relevant function

- Page 130 CCPU_X_In_Word_ISR
- Page 132 CCPU_Y_Out_Word_ISR

CCPU_Y_Out_Word_ISR

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

■Format

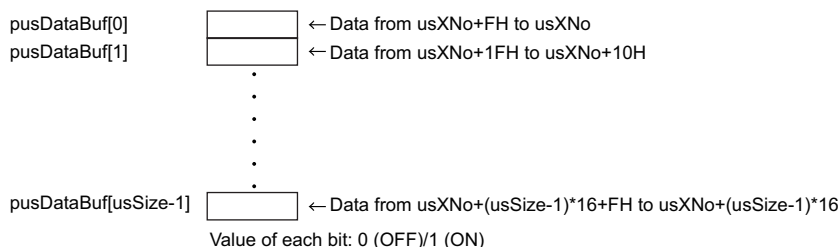
short CCPU_Y_Out_Word_ISR (unsigned short usYNo, unsigned short usSize, unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
usYNo	Start output number	Specify a start output number (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) the output signal (Y) equivalent to the size specified to usSize from the start output number (usYNo) to pusDataBuf.
- Specify usYNo in multiples of 16. (The remainder divided by 16 is discarded. Example: 0x05 → 0x00, 0x123→ 0x120)
- Store the read data in pusDataBuf in the ascending order of data number from the lower bit as shown below.



- Do not specify the usYNo corresponding to an output module controlled by the other CPUs. Even when it is specified, no operation is performed to the output module.

Restriction

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


! WARNING

■This function does not check the argument specified for writing. Create a program with the following conditions in mind.

- The address specified for the write data is a multiple of 2.
- The output signal (Y) within the range (0H to FFFH) is specified.

■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
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant function

- Page 130 CCPU_X_In_Word_ISR
- Page 131 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 IPath)


■Argument

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

■Description

- This function closes the path of 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 138 mdOpen


mdControl

This function performs remote operations (RUN/STOP/PAUSE) for programmable controller CPU.

■Format

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

■Argument

Argument	Name	Description	IN/OUT
IPath	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. For the specification method, refer to the following section.  Page 8 Argument specification	IN
sCode	Instruction code	Specify the contents of the remote operation in numerical value.	IN


- The remote operations that can be specified to sCode are as follows:

Instruction code (decimal)	Remote operation
0	Remote RUN
1	Remote STOP
2	Remote PAUSE

■Description

- This function changes the status of the programmable controller CPU with the station number specified to sStNo to the status specified to sCode.
- This function cannot be executed for C Controller module/PC CPU module/WinCPU module.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 138 mdOpen
- Page 133 mdClose



mdDevRstEx

This function resets bit device(s).

■Format

long mdDevRstEx(long IPath, long INetNo, long IStNo, long IDevType, long IDevNo)


■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
INetNo	Network number	Specify a network number of the target module.	IN
IStNo	Station number	Specify a station number of the target module. For the specification method, refer to the following section.  Page 8 Argument specification	IN
IDevType	Device type	Specify the device type of the bit device. For the specification method, refer to the following section.  Page 8 Argument specification	IN
IDevNo	Device number	Specify the device number of the bit device.	IN

■Description

- This function resets (turns OFF) the bit device of the module specified to INetNo, IStNo, IDevType, and IDevNo.
- 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 138 mdOpen
- Page 133 mdClose
- Page 136 mdDevSetEx



mdDevSetEx

This function sets bit device(s).

■Format

long mdDevSetEx(long IPath, long INetNo, long IStNo, long IDevType, long IDevNo)


■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
INetNo	Network number	Specify a network number of the target module.	IN
IStNo	Station number	Specify a station number of the target module. For the specification method, refer to the following section.  Page 8 Argument specification	IN
IDevType	Device type	Specify the device type of the bit device. For the specification method, refer to the following section.  Page 8 Argument specification	IN
IDevNo	Device number	Specify the device number of the bit device.	IN

■Description

- This function sets (turns ON) the bit device of the module specified to INetNo, IStNo, IDevType, and IDevNo.
- 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 138 mdOpen
- Page 133 mdClose
- Page 135 mdDevRstEx

mdInit

This function initializes communication route information.

■Format

short mdInit(long IPath)


■Argument

Argument	Name	Description	IN/OUT
IPath	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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 138 mdOpen
- Page 133 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). For the specification method, refer to the following section.  Page 8 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

- The path of the channel opened by this function is used when other MELSEC data link functions are executed.
- To end the program, close the path of the opened channel 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 133 mdClose

mdRandREx

This function reads device(s) randomly.

■Format

long mdRandREx(long IPath, long INetNo, long IStNo, long* pIDev, short* psBuf, long lBufSize)

■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
INetNo	Network number	Specify a network number of the target module. For the specification method, refer to the following section. ☞ Page 8 Argument specification	IN
IStNo	Station number	Specify a station number of the target module. For the specification method, refer to the following section. ☞ Page 8 Argument specification	IN
pIDev	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 word units.	IN

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

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

■Description

- This function reads devices specified to pIDev from the module specified to INetNo and IStNo.
- The read data is stored in psBuf in word units in order of the specification to the randomly selected device. 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 points specified by each block number 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. To reduce communication time, use the mdReceiveEx (Device batch read function) 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 table shows the examples of values set to plDev, data to be read to 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	The current value of T10 is 1 second (10).
LCN100 to LCN101	0x1 is stored to LCN100 and 0x10000 is stored to LCN101.

Values set to plDev

Setting target	Setting value	Description
plDev[0]	5	Number of blocks = 5
plDev[1]	DevM	Device type = M
plDev[2]	100	Start device number = 100
plDev[3]	16	Number of read points = 16
plDev[4]	DevD	Device type = D
plDev[5]	10	Start device number = 10
plDev[6]	4	Number of read points = 4
plDev[7]	DevM	Device type = M
plDev[8]	0	Start device number = 0
plDev[9]	14	Number of read points = 14
plDev[10]	DevTN	Device type = T
plDev[11]	10	Start device number = 10
plDev[12]	1	Number of read points = 1
plDev[13]	DevLCN	Device type = LCN
plDev[14]	100	Start device number = 100
plDev[15]	2	Number of read points = 2

Data to be read to psBuf

Read position	Read device	Value	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	The current value of T10 is 10 (1 second).
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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 138 mdOpen
- Page 133 mdClose
- Page 142 mdRandWEx

mdRandWEx

This function writes device(s) randomly.

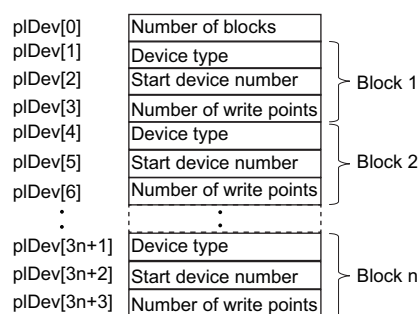
■Format

long mdRandWEx(long IPath, long INetNo, long IStNo, long* plDev, short* psBuf, long lBufSize)

■Argument

Argument	Name	Description	IN/OUT
IPath	Channel	Specify the path of the channel.	IN
INetNo	Network number	Specify a network number of the target module. For the specification method, refer to the following section. ☞ Page 8 Argument specification	IN
IStNo	Station number	Specify a station number of the target module. For the specification method, refer to the following section. ☞ Page 8 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 set, the operation is not affected.)	IN

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



■Description

- This function writes data to the device, which is specified to `plDev`, of the module specified to `INetNo` and `IStNo`.
- The data to be written is stored to `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 points specified by each block number 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 table shows the examples of values set to plDev and 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 set to plDev

Setting target	Setting data	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 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]	DevLCN	Device type = LCN	Block 3: LCN100 to LCN101
plDev[8]	100	Start device number = 100	
plDev[9]	2	Number of read points = 2	

Value set to psBuf

Setting target	Setting data	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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 138 mdOpen
- Page 133 mdClose
- Page 139 mdRandREx



mdReceiveEx

This function reads devices in batch.

■Format

long mdReceiveEx(long IPath, long INetNo, long IStNo, long IDevType, long IDevNo, long* pSize, short* psData)


■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
INetNo	Network number	Specify a network number of the target module. For the specification method, refer to the following section.  Page 8 Argument specification	IN
IStNo	Station number	Specify a station number of the target module. For the specification method, refer to the following section.  Page 8 Argument specification	IN
IDevType	Device type	Specify the device type for device to be read in batch.	IN
IDevNo	Start device number	Specify the start device number for device to be read in batch. For bit devices, set the device number in multiples of 8 (0. 8. 16...).	IN
pSize	Read data size	Specify the read data size in byte units. Specify the value in multiples of 4 when double-word device (LZ, LTN, LCN, LSTN) is specified, or specify the value in multiples of 2 when a device other than those 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 from the module specified to INetNo and IStNo.
- This function reads data with the size specified to pSize from the devices specified to IDevType and IDevNo.
- When the specified read data size exceeds the device range (-5: size error), a readable size is returned to pSize.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 138 mdOpen
- Page 133 mdClose
- Page 147 mdSendEx (Device batch write function)

mdReceiveEx

This function receives messages. (RECV function)

■Format

long mdReceiveEx(long IPath, long INetNo, long IStNo, long IDevType, long IDevNo, long* pSize, short* psData)

■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
INetNo	Network number	Specify '0' (0H).	IN
IStNo	Station number	Specify own station 255 (FFH).	IN
IDevType	Device type	Specify the device type for device to be read in batch. Only "RECV function: 101 (65H and DevMAIL)" is available.	IN
IDevNo	Channel number	Specify the channel number. CC-Link IE Controller Network: 1 to 8 CC-Link IE Field Network: 1 to 2	IN
pSize	Receive message size	Specify the receive message size in byte units. (2 to 1920) Specify an even number. If an odd number is specified, 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 pSize (bytes) are specified.	OUT

■Description

- This function supports the RECV instruction, the dedicated instruction for CC-Link IE Controller Network module when the following conditions are met: 1, the "IPath returned by the mdOpen function by specifying CC-Link IE Controller Network module (channel number 151 to 158) or CC-Link IE Field Network module (channel number 181 to 188)" is specified, and 2, the "RECV function: 101" is specified for the device type.
- It receives the message data to the channel specified to IDevNo from among the messages sent to CC-Link IE Controller Network module or CC-Link IE Field Network module.
- For more advanced RECV function, use the C Controller module dedicated functions. This mdReceiveEx function reads the message data to the specified channel number in the order it was received.
- When the actual size of the received message is smaller than the size specified to pSize, data for the actual size is stored in psData[3] or higher, and the data size of the received message is returned to pSize.
- When the actual size of the received message is bigger than the size specified to pSize, data up to the specified size is stored in psData[3] or higher.
- The received data is stored in the receive buffer as follows.

Receive buffer	Receive data
psData[0]	Send station network number
psData[1]	Send station number
psData[2]	Channel used by send station
psData[3] or higher	<ul style="list-style-type: none"> • Received message (actual data) • (2 to 1920 bytes)


Information on the send source (network number, station number, and used channel of the send station) is stored in psData[0] to psData[2]. Therefore, the storage size of receive data, psData[] is 6 + a value specified to pSize (bytes).

- Confirm that the RECV execution request flag of Network module is ON before executing this function.

- 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	IDevNo
+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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 138 mdOpen
- Page 133 mdClose
- Page 148 mdSendEx (Message send function)
- Page 44 CCPU_DedicatedGInst
- Page 46 CCPU_DedicatedJInst



mdSendEx

This function writes devices in batch.

■Format

long mdSendEx(long IPath, long INetNo, long IStNo, long IDevType, long IDevNo, long* pISize, short* psData)


■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
INetNo	Network number	Specify a network number of the target module. For the specification method, refer to the following section.  Page 8 Argument specification	IN
IStNo	Station number	Specify a station number of the target module. For the specification method, refer to the following section.  Page 8 Argument specification	IN
IDevType	Device type	Specify the device type for device to be written in batch.	IN
IDevNo	Start device number	Specify the start device number to be written in batch. For bit devices, set the device number in multiples of 8 (0, 8, 16 ...).	IN
pISize	Write data size	Specify the write data size in byte units. Specify the value in multiples of 4 when double-word device (LZ, LTN, LCN, LSTN) is specified, or specify the value in multiples of 2 when a device other than those 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 to the module specified to INetNo and IStNo.
- It writes data equivalent to the size specified to pISize by handling the device specified to IDevType and IDevNo as the start device.
- It checks the arguments and verifies whether the address + size determined by the arguments is within the device memory range.
- When the specified write data size exceeds the device range (-5: size error), a writable size is returned to pISize.
- 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, refer to the following chapter.  Page 152 ERROR CODE LIST

■Relevant functions

- Page 138 mdOpen
- Page 133 mdClose
- Page 144 mdReceiveEx (Device batch read function)

mdSendEx

This function sends messages. (SEND function)

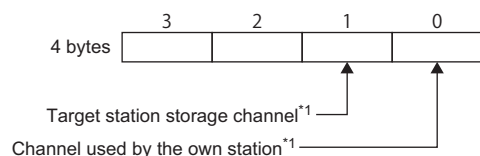
■Format

long mdSendEx(long IPath, long INetNo, long IStNo, long IDevType, long IDevNo, long* pSize, short* psData)

■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
INetNo	Network number	Specify a network number of the target module. A logical station number cannot be specified. For the specification method, refer to the following section. ☞ Page 8 Argument specification	IN
IStNo	Station number	Specify a station number of the target module. A logical station number cannot be specified. For the specification method, refer to the following section. ☞ Page 8 Argument specification	IN
IDevType	Device type	Specify the device type for device to be written in batch. With arrival confirmation: 101 (65H, DevMAIL) No arrival confirmation: 102 (66H, DevMAILNC) When "Group number" or "All stations" is specified for the station number, only "No arrival confirmation" is valid.	IN
IDevNo	Channel number	Specify the channel number.	IN
pSize	Send data size	Specify the send data size in byte units. (2 to 1920) Only an even-number byte can be specified.	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.



*1 CC-Link IE Controller Network: 1 to 8
CC-Link IE Field Network: 1 to 2


■Description

- This function supports the SEND instruction, the dedicated instruction for CC-Link IE Controller Network module and CC-Link IE Field Network module when the following conditions are met: 1, the "IPath returned by the mdOpen function by specifying CC-Link IE Controller Network module (channel number 151 to 158) or CC-Link IE Field Network module (channel number 181 to 188)" is specified, and 2, the "With arrival confirmation: 101" or "No arrival confirmation: 102" is specified for the device type.
- This function sends message data from CC-Link IE Controller Network or CC-Link IE Field Network module to the target (network number/station/channel) specified to IStNo or IDevNo.
- 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	IDevType
+1	Completion status	sRet
+2	Own station channel	IDevNo
+3	Target station storage channel	IDevNo
+4	Target station network number	INetNo
+5	Target station number	IStNo
+6	Not used	—
+7	Number of retransmission (retry)	—
+8	Arrival monitoring time	—
+9	Send data length	pSize
+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, refer to the following chapter.  Page 152 ERROR CODE LIST

Relevant functions

- Page 138 mdOpen
- Page 133 mdClose
- Page 145 mdReceiveEx (Message receive function)
- Page 44 CCPU_DedicatedGInst
- Page 46 CCPU_DedicatedJInst


mdTypeRead

This function reads the model code of a CPU module.

■Format

short mdTypeRead(long IPath, short sStNo, short* psCode)

■Argument

Argument	Name	Description	IN/OUT
IPath	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. For the specification method, refer to the following section.  Page 8 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 sStNo.

Model code (hexadecimal)	CPU module model name
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
036AH	Q13UDVCPU

Model code (hexadecimal)	CPU module model name
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
4800H	R04CPU
4801H	R08CPU
4802H	R16CPU
4803H	R32CPU
4804H	R120CPU
4820H	R12CCPU-V
4C00H	R16MTCPU
4C01H	R32MTCPU
4C02H	R64MTCPU

For the CPU module model names other than the above, the model code will be undefined.

Return value

Return value	Description
0 (0000H)	Normal
Other than 0	Error For details, refer to the following chapter.  Page 152 ERROR CODE LIST

Relevant functions

- Page 138 mdOpen
- Page 133 mdClose

4 ERROR CODE LIST

4.1 Common Error Codes

The error codes common to C Controller module dedicated functions and MELSEC data link functions are as shown below.

Error code (HEX) ^{*1}	Description	Measures
1 (0001H)	<p>■Driver not started</p> <p>The driver has not been started.</p>	<p>Check the channel number.</p> <p>Correct the error that occurred when the driver is started.</p> <p>Check the status of the system drive of the C Controller module.</p> <p>Check if the operating system is running normally.</p>
2 (0002H)	<p>■Timeout error</p> <p>A timeout occurred while waiting for response.</p> <p>During CC-Link communication, the request was issued to other stations even when the own station number is 64.</p> <p>The module specified as the communication target is not supported.</p>	<p>Review the operating status and mounting condition of the accessed station.</p> <p>Retry on the user program.</p> <p>Increase the timeout value of MELSEC data link function.</p> <p>When requesting to other stations during CC-Link communication, set the own station number to the number other than '64'.</p> <p>Check that the module specified as the communication target is supported.</p>
66 (0042H)	<p>■Already opened error</p> <p>The specified channel has already been opened.</p>	<p>Open once.</p> <p>If this error occurred, the path of the right bus will be returned to the argument.</p>
67 (0043H)	<p>■Already closed error</p> <p>The specified channel has already been closed.</p>	<p>Close once.</p>
69 (0045H)	<p>■Unsupported function execution error</p> <p>An unsupported function in the target station has been executed.</p>	<p>Check the path of the channel, network number, and station number.</p> <p>Check that the function used in the target station is supported.</p>
70 (0046H)	<p>■Station number error</p> <ul style="list-style-type: none"> • The specified station number is incorrect. • The process that should have been requested to other station was requested to own station. Or, the network number was not '0' although the station number was own station (FFH). 	<p>Correct the network number and station number of the user program.</p>
71 (0047H)	<p>■Data reception error (When a RECV is requested.)</p> <p>Data has not been received.</p>	<p>Check the channel number.</p> <p>Check whether data is sent from the send station for the SEND function.</p> <p>Retry.</p>
77 (004DH)	<p>■Memory allocation error/ resource shortage error/ task over error</p> <p>Sufficient memory could not be allocated. Or, there are too many tasks using the C Controller module dedicated functions/ MELSEC data link functions.</p>	<p>The memory may be insufficient. Terminate other tasks currently running. Or decrease the access size.</p> <p>Check if the C Controller module is running normally.</p> <p>Reset the C Controller system.</p> <p>Retry after decreasing the number of tasks using the C Controller module dedicated functions/MELSEC data link functions.</p> <p>Review the size or number specified to the arguments of user program.</p>
85 (0055H)	<p>Network channel number error (When a SEND/RECV is requested.)</p>	<p>Check the specified channel number when a SEND/RECV is requested.</p>
102 (0066H)	<p>■Transmission error/restart error</p> <p>Data send has failed. Or, data send has been executed during restart.</p>	<p>Retry.</p> <p>Check if the C Controller module is running normally.</p> <p>Reset the C Controller system.</p> <p>Retry after completion of the restart.</p>
103 (0067H)	<p>■Reception error</p> <p>Data reception has failed.</p>	<p>Retry.</p> <p>Check if the C Controller module is running normally.</p> <p>Reset the C Controller system.</p>
130 (0082H)	<p>■Device number error</p> <ul style="list-style-type: none"> • The specified device number is out of range. • The specified bit device number is not multiple of 8. 	<p>Check the device number.</p>
131 (0083H)	<p>■Number of device points error</p> <ul style="list-style-type: none"> • The specified number of device points is out of range. • The specified bit device points is not multiple of 8. 	<p>Check the device points.</p>
16384 (4000H) to 20479 (4FFFH)	<p>Errors detected in the access target CPU</p>	<p>For the error codes which are not written in this manual, refer to the user's manual of the access target CPU.</p>
-25056 (9E20H)	<p>■Processing code error</p> <p>The request target has received a request which cannot be processed by it.</p>	<p>Check the network number and station number of the request target.</p>

Error code (HEX)*1	Description	Measures
-26334 (9922H)	■Reset error <ul style="list-style-type: none"> Another task using the same channel executed reset operation while accessing other station. Reset operation has been executed during monitoring with CW Configurator. 	Retry. Monitor again.
-26336 (9920H)	■Routing request error for unsupported station Routing to another loop was requested to a station which does not support the routing function.	Check the settings of routing parameter.
-28150 (920AH)	■Device access error during data link stop Own station devices (RX/Ry/RWw/RWr) have been 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 data link. Note that data can be written/read despite this error, however; the data will not be guaranteed.
-28151 (9209H)	■Abnormal data reception error Abnormal response data has been received.	Check if there is an error in the target CPU or link module. If the status is normal, try again.
-28158 (9202H)	■WDT error WDT (system/user) error occurred.	Reset the C Controller system.
-28410 (9106H)	■Target CPU busy error The target CPU is busy.	Add a processing to wait until the completion of a processing or a retry processing to the user program. Increase the timeout time specified to the argument in the user program.
-28412 (9104H)	■Target CPU unsupported error An unsupported request has been issued to the target CPU.	Change the target CPU number specified in the user program.
-28413 (9103H)	■Target CPU down error The target CPU is down.	Check the operating status of the target CPU and troubleshoot the error.
-28414 (9102H)	■Target CPU abnormal start error A request has been issued to the CPU which was not operating normally.	Check the operating status of the target CPU and troubleshoot the error.
-28415 (9101H)	■Target CPU critical error A request has been issued to the CPU in which a critical error occurred.	Check the operating status of the target CPU and troubleshoot the error.
-28416 (9100H)	■Target CPU mounting error A request has been issued to the CPU number on which no CPU is mounted.	Check if the target CPU is mounted properly. Change the target CPU number specified in the user program.
-28622 (9032H)	■Target module busy error <ul style="list-style-type: none"> The target module is busy. The own station channel or the target station storage channel has been used for other instruction, or multiple identical instructions are being executed. 	Add a processing to wait until the completion of a processing or a retry processing to the user program.
-28624 (9030H)	■Function unsupported error <ul style="list-style-type: none"> Process was executed for the module that does not support the cyclic data station-based block data assurance setting. Process was executed for the module that does not support the cyclic data station-based block data assurance setting. An attempt was made to access a module not controlled by the host CPU. 	Check if the version of the target CC-Link module supports the cyclic data station-based block data assurance setting. Check if the cyclic data station-based block data assurance is set for the target module. Check whether the control CPU of the target module is the host CPU.
-28625 (902FH)	■Intelligent function module offline error An attempt was made to access the Intelligent function module while it is offline.	Check the status (mode) 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.	Review the user program and correct the value set to the control data.
-28627 (902DH)	■Transient unsupported error A transient request cannot be executed via the specified communications route and target. (With CC-Link communication, when the station number of the own station was "64", other station was specified.)	Check the communication route and target which support the transient request. Change the station number of the own station.
-28628 (902CH)	■Pointer address specification error An incorrect address has been specified to the argument pointer.	Check the address of the specified pointer.
-28629 (902BH)	■WDT not running error WDT was reset when WDT is not started up.	Reset WDT after starting WDT.
-28630 (902AH)	■WDT startup error An attempt was made to start up WDT while it is running.	Start WDT after stopping WDT.

Error code (HEX)*1	Description	Measures
-28631 (9029H)	■Buffer access range error <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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 non-controlled module when reading from a non-controlled module is not allowed.	Check if the control CPU of the specified module is the host CPU (C Controller module/PC CPU module/WinCPU module).
-28634 (9026H)	■Intelligent function module down error There is a problem in the Intelligent function module.	Check if the Intelligent function module is mounted properly. Replace the Intelligent function module or base unit.
-28635 (9025H)	■Intelligent function module error No Intelligent function module is mounted on the accessed slot.	Check the specified I/O number. Check if the Intelligent function module is mounted properly.
-28636 (9024H)	■Control bus error There is a problem in the control bus to the Intelligent function module.	Check if there is an error in CPU No.1 in the multiple CPU system. Check if the Intelligent function module is mounted properly. Replace the Intelligent function module or base unit.
-28638 (9022H)	Multiple CPU unsupported operation error	Reset CPU No.1.
-28640 (9020H)	■STOP/PAUSE error Output request or buffer memory write request was issued when the CPU status is STOP/PAUSE.	Change the CPU operating status to RUN.
-28653 (9013H)	■I/O assignment error <ul style="list-style-type: none"> An attempt was made to read the value of X from an output module. An attempt was made to output the value of Y to an input module. An Attempt was made to read the value of Y from an input module. 	Review the specified input number (X). Review the specified 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 (C Controller module/PC CPU module/WinCPU module).	Check if the control CPU of the specified module is the host CPU (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 are was specified.	Review the specified offset and size.
-28662 (900AH)	■CPU number specification error The CPU number is out of range or unavailable.	Review the specified CPU number. Check the operating status of the specified CPU.
-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 the C Controller system.
-28666 (9006H)	Data length error	Reset the C Controller system.
-28668 (9004H)	Reply data stored error	Resend the request.
-28669 (9003H)	■Area number error The specified area number, offset address, or mode is out of range.	Review the area number, offset address, and mode.
-28671 (9001H)	Module identification error	Review the parameters. Check the specified module. Reset the C Controller system.
-28672 (9000H)	Processing code error	Reset the C Controller system.

*1 When the function of which return value is a long-type, the value will be eight-digit in hexadecimal form.

4.2 C Controller Module Dedicated Functions

Error code (HEX)	Description	Measures
-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 number is out of range.	Check the specified I/O number.
-204 (FF34H)	■I/O access size error The specified I/O access size is out of range.	Check the specified I/O access size (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 program execution type is out of range.	Check the execution type of the specified program.
-208 (FF30H)	■Offset error • The specified offset is out of range. • An AnS series module (buffer memory) has been 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 time setting is out of range.	Check the specified time.
-214 (FF2AH)	■Intelligent function module error No Intelligent function module is mounted on the accessed slot.	Check the specified I/O number. Check if the Intelligent function module is mounted properly.
-217 (FF27H)	■Driver not started The driver has not been started.	Check if the driver has been started.
-219 (FF25H)	■Program name error The specified program name is unavailable. (The program name does not exist in the CPU module or is not registered in a parameter).	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) has failed.	Enable the setting to allow remote reset (set "Remote Reset" to "Enable") for the bus master CPU (CPU No.1). Change the 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 allocation error Memory allocation has failed.	Check if there are sufficient available memory.
-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 has 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 that the event number (interrupt pointer number) is set correctly.
-232 (FF18H)	■CPU number specification error • The specified CPU number is incorrect. • The specified CPU cannot execute the request. • The host CPU has been specified as the target CPU with 1 to 4 (other CPUs) for the remote operation.	Specify the correct CPU number. Do not issue a request that generated an error to the specified CPU. Specify '0' (host CPU) to perform the remote operation for the host CPU.
-234 (FF16H)	■Event wait error An error other than timeout occurred while the function waits for the event.	Check that the operating system is not forcibly terminating the program. Check if the C Controller module is running normally. Reset the C Controller system. Specify '0' (host CPU) to perform the remote operation for the host CPU.
-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.

Error code (HEX)	Description	Measures
-236 (FF14H)	<p>■Remote operation specification code error The remote operation specification code is out of range.</p>	Check the specified remote operation specification code.
-237 (FF13H)	<p>■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.</p>	Correct the length of the specified character string or character string data.
	<p>■Application code specification error Five or more digits of the hexadecimal number is specified in the specified application code.</p>	Change the specified application code.
-238 (FF12H)	<p>■Event log registration error Event log registration has failed.</p>	Reset the C Controller system.
-239 (FF11H)	<p>■Memory card insertion error The processing has been executed with no SD memory card inserted.</p>	Check if an SD memory card is inserted.
-240 (FF10H)	<p>■Clock data incorrect error The clock data to be set or the read clock data is incorrect.</p>	Check the clock data to be set. If this error occurs at reading the clock data, set the clock data.
-241 (FF0FH)	<p>■Cycle specification error</p> <ul style="list-style-type: none"> • 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)	<p>■Synchronization type specification error The specified synchronization type is out of range.</p>	Check the specified synchronization type.
-245 (FF0BH)	<p>■Not executable during interrupt service routine Function was executed from interrupt service routine with specifying other than 1 (ISR) to the call source flag.</p>	Correct the processing of routine to be registered so that the function is executed by specifying '1' (ISR) to the call source flag.
-246 (FF0AH)	<p>■Timer event registration error Timer event registration has failed.</p>	Retry. Check if the C Controller module is running normally. Reset the C Controller system.
-247 (FF09H)	<p>■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)</p>	Correct the program number specified in the user program.
-248 (FF08H)	<p>■Number of starting axes specification error 9 or more starting axes have been specified.</p>	Correct the number of starting axes specified in the user program.
-249 (FF07H)	<p>■Axis type specification error The specified axis type is other than servo axis/encoder axis/cam axis.</p>	Correct the axis type specified in the user program.
-250 (FF06H)	<p>■Axis number specification error The specified axis number is out of the available range.</p>	Correct the axis number specified in the user program.
-252 (FF04H)	<p>■Torque limit value specification error The specified torque limit value is out of the available range.</p>	Correct the torque limit value specified in the user program.
-253 (FF03H)	<p>■Device number specification error The specified device number is out of range. The specified bit device number is not multiple of 16.</p>	Correct the start device number of the specified device.
-254 (FF02H)	<p>■Device type specification error The specified device type is unavailable.</p>	Correct the device type specified in the user program.
-255 (FF01H)	<p>■Size specification error</p> <ul style="list-style-type: none"> • The specified number of words is out of range. • The specified size is 0. 	Correct the start device number and the number of words specified in the user program.
	<p>■Response completion wait timeout error A timeout occurred while waiting for the completion of a response for a processing requested to other CPU.</p>	Increase the timeout time specified to the argument in the user program. Check and correct the user program (including other tasks which execute motion CPU interaction functions). Check and correct the program of the request target CPU to enable it to process a request from other CPU, such as by adding the WAIT instruction.
-257 (FEFFH)	<p>■Interrupt type specification error The value specified to the interrupt type is out of range.</p>	Review the specified value, and specify a value within the range.
-258 (FEFEH)	<p>The value specified as the interrupt number is out of range.</p>	Review the specified value, and specify a value within the range.
-259 (FEFDH)	<p>■Interrupt service routine unregistered error</p> <ul style="list-style-type: none"> • The processing has not been registered when enabling a processing which corresponds to an event (interrupt). • The specified CPU number is incorrect. 	<ul style="list-style-type: none"> • Register the processing for the event (interrupt) and execute the operation once again. • Check the specified CPU number.

Error code (HEX)	Description	Measures
-260 (FEFCH)	■Memory card mount/unmount error Mount process or unmount process of the SD memory card has failed.	Retry. Check if the SD memory card is damaged. Replace the SD memory card.
-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 specified pointer.
-265 (FEF7H)	■Target system specification error The value specified in the target system is out of range.	Review the specified value, and specify a value within the range.
-266 (FEF6H)	■WDT start error The user WDT is activated.	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 an identical name exist. (Overwrite is "Disabled.") • A file cannot be created in the specified path or the specified path does not exist.	Check the specified file. Check whether overwriting is enabled. Check whether 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 the individual identification information has failed.	Check if the C Controller module is running normally. Reset the C Controller system.
-289 (FEDFH)	■Dot matrix LED mode selection error The mode other than to display the specified content on the dot matrix LED was set by using the CCPU_SetOpSelectMode function or with the MODE/SELECT switch.	Set the mode to display the specified content on the dot matrix LED 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 CCPU_SetOpSelectMode function or the CCPU_SetDotMatrixLED function after finishing the operation selection using the MODE/SELECT switch.
-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. Moreover, confirm that 1K word or more has been set in the fixed cycle communication area setting.
-292 (FEDCH)	■Program memory shutdown error Shutdown processing of the program memory has failed.	Check if files in the program memory are being accessed. Check whether all user files in the program memory have been closed.
-293 (FEDBH)	■Data memory shutdown error Shutdown processing of the data memory has failed.	Check if files in the data memory are being accessed. Check whether all user files in the data memory have been closed.
-295 (FED9H)	■Selected operation is being checked The CCPU_SetDotMatrixLED function was executed while the operation selected in the operation selection mode was being checked.	Execute the CCPU_SetDotMatrixLED function after the check of the selected operation has been done.
-296 (FED8H)	■Setting data size error The setting data size is out of range.	Review the setting data size.

Error code (HEX)	Description	Measures
-297 (FED7H)	■Input/output number, network number incorrect specification An input/output number outside the range (other than 000H to FFFH or 3E0H to 3E3H) has been specified.	Correct the argument for the function.
-298 (FED6H)	■Input/output number, network number incorrect specification An input/output number to which no module corresponds has been specified.	
-299 (FED5H)	■Input/output number, network number incorrect specification • An input/output number of the module which does not support the function has been specified. • The dedicated instruction specified by the program cannot be executed 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 module that cannot be specified with the function has been specified.	Correct the argument for the function.
-301 (FED3H)	■Input/output number, network number incorrect specification A network number outside the range (other than 1 to 239) has been specified.	
-302 (FED2H)	■Input/output number, network number incorrect specification A network number which does not exist has been specified.	
-303 (FED1H)	■Input/output number, network number incorrect specification The I/O module or Intelligent function module controlled by other CPU module has been specified.	<ul style="list-style-type: none"> • Correct the argument for the function. • Delete the Network module controlled by other CPU, which has been specified with a link direct device, from the program. • Specify the Network module controlled by own station with a link direct device.
-304 (FED0H)	■Input/output number, network number incorrect specification The target module cannot be identified in a function where I/O module or Intelligent function module is specified. (The character strings to specify the target module are incorrect.)	Correct the argument for the function.
-305 (FECFH)	■Input/output number, network number incorrect specification The I/O module or Intelligent function module specified with the function 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 with the function. Please consult your local Mitsubishi representative.
-306 (FEC EH)	■Device, buffer memory incorrect specification The device specified with the function exceeded the usable range.	Correct the argument for the function.
-307 (FEC DH)	■Device, buffer memory incorrect specification A device that cannot be specified with a function has been specified.	
-308 (FEC CH)	■Program fault The argument structure of the function specified by the program is incorrect.	
-309 (FEC BH)	■Program fault The number of devices of the function specified by the program is incorrect.	
-310 (FEC AH)	■Operation error Character strings that cannot be handled by a function has been specified.	
-311 (FEC 9H)	■Operation error Data outside the specifiable range have been entered.	
-312 (FEC 8H)	■Operation error In a multiple CPU system, the CCPU_DedicatedDInst function has been 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 of system parameter to "Use".
-313 (FEC 7H)	■Operation error In a multiple CPU system, the number of data points which exceeds the system area size usable for each CPU has been specified.	Correct the number of data points for the CCPU_DedicatedDInst function.
-314(FEC6H)	■Module major error	The possible cause is a hardware failure of the module where the error has been detected. Please consult your local Mitsubishi representative.
-315(FEC5H)	An error has been detected in the Intelligent function module when a function was executed.	

Error code (HEX)	Description	Measures
-316(FEC4H)	■Other CPU module major error	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 module where the error has been detected. Please consult your local Mitsubishi representative.
-317(FEC3H)	An error has been detected in other CPU module when a function was executed.	
-318(FEC2H)	■System bus error An error has been 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 cables. Please consult your local Mitsubishi representative.
-319(FEC1H)	■Hardware failure A hardware failure has been 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 has been 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 has been 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 has been 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 has been 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 has been 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 has been 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

Error code (HEX)*1	Description	Measures
-1 (FFFFH)	■Path error The specified path is unavailable.	Use a path pointer returned by the mdOpen function.
-2 (FFFEH)	■Device number error <ul style="list-style-type: none"> 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 that 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 whether a device type listed in the device type list is used. Check if the specified device is available in the target station.
-5 (FFFBH)	■Size error <ul style="list-style-type: none"> 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 points that are specified for each block number in respect to 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. Adjust the total points that are specified for each block number in respect to the mdRandREx function or the mdRandWEx function to 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 with the mdOpen function is unavailable.	Check the specified channel number.
-11 (FFF5H)	■Insufficient buffer area Read data storage area size is smaller than the read data size.	Check the read data size and the read data storage destination 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 is overlapping with the write protect area of the memory cassette.	Check the block number (device type) of the extension file register. Check the write protect switch on the access target memory cassette.
-16 (FFF0H)	■Station number error, network number error The specified station number or network number is out of range. A device which is not accessible by the target station has been specified.	Check the specified station number and network number. Check the devices which can be accessed by the target station.
-17 (FFE FH)	■All stations/group number specification error The 'all stations'/group number' was specified to the function that does not support it.	<ul style="list-style-type: none"> Check if the function allows the all stations/group number specification. Specify the device type to "No arrival confirmation" when "All stations" or "Group number" is specified for the station number.
-18 (FFEEH)	■Remote operation error The instruction code specified with the mdControl function is unavailable.	Check the specified code.
-19 (FFEDH)	■SEND/RCV channel number error The channel number specified in the SEND/RCV 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 has failed.	The memory may be insufficient. Terminate other tasks currently running. Or decrease the access size. Check the status of the system drive of the C Controller module.
-32 (FFE0H)	■Resource timeout error The resource is being used by other tasks/threads and is not released within 30 seconds.	Retry. The memory may be insufficient. Terminate other task(s) currently running. Check if the C Controller module is running normally. Reset the C Controller system.
-33 (FFDFH)	■Communication target unsupported error The module specified as the communication target by network number and station number is not supported.	Check that the module specified as the communication target by 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 has failed.	Check if the settings of the communication target are correct with CW Configurator.

Error code (HEX)*1	Description	Measures
-35 (FFDDH)	■Registry read error Reading parameter files from the registry has failed.	Check if the settings of the communication target are correct with CW Configurator. Check that the setting for the channel number is valid. Reset the C Controller system after checking the parameters with CW Configurator again and writing parameters.
-36 (FFDCH)	■Registry write error Writing parameter files to the registry has failed.	Check if the standard ROM has already been shutdown. Reset the C Controller system.
-37 (FFDBH)	■Communications initialization error Initialization for communication has failed.	Retry. The memory may be insufficient. Terminate other tasks currently running. Check the available memory capacity. Check if the C Controller module is running normally. Reset the C Controller system.
-42 (FFD6H)	■Close error Communications cannot be closed.	Retry. Check if the C Controller module is running normally. Reset the C Controller system.
-43 (FFD5H)	■ROM operation error TC setting value was written to the CPU 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.
-4097 (EFFFH) to -8192 (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 (DFFFH) to -12288 (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 (BFFFH) to -20480 (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 return value is a long-type, the value will be eight-digit in hexadecimal form.

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 ( ERROR CODE LIST) to perform the troubleshooting.

APPENDIX

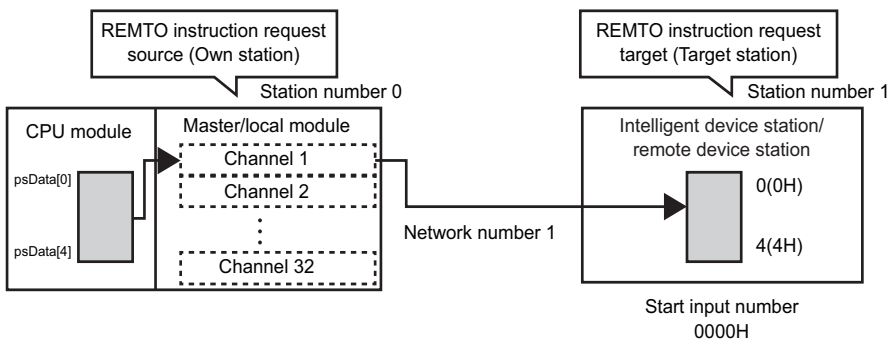
Appendix 1 Example for Replacing Ladder by C Language

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

(0)	SM400	JP.REMTO	J1	K1	K1	K0	K0	D0	K5	M0
(15)										{END}

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]="REMT0"; /* 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 168 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 167 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	—	QBFDDev_LXBuf	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with the following method.  Page 167 Refreshing device, The access target is Network module on the own station.
Link output internal buffer	—	QBFDDev_LYBuf	
Link relay internal buffer	—	QBFDDev_LBBuf	
Link register internal buffer	—	QBFDDev_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 167 Refreshing device, The access target is Network module on the own station. ☞ Page 167 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 167 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	
Other station random access buffer	—	DevRAB	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with the following method. ☞ Page 167 Module access device, The access target is Network module on the other station.
Other station remote input	—	DevRX	
Other station remote output	—	DevRY	The area (device) corresponding to the device type deleted from R12CCPU-V can be accessed with either of the methods below. ☞ Page 167 Refreshing device, The access target is Network module on the other station.
Other station link register	—	DevRW	
Other station link special relay	—	DevSB	☞ Page 167 Module access device, The access target is Network module on the other station.
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 167 Refreshing device, The access target is Network module on the own station. ☞ Page 167 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 167 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 167 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 167 Refreshing device, The access target is Network module on the own station. ☞ Page 167 Module access device, The access target is Network module on the own station. ☞ Page 167 CCPU_ReadLinkDevice/CCPU_WriteLinkDevice
Other 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 167 Refreshing device, The access target is Network module on the own station. ☞ Page 167 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 167 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 78 CCPU_ReadLinkDevice, Page 105 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_CommunicateMCPProtocol	Extended mode	×	—
CCPU_SetOpenNoMCPProtocol	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

A

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)

FUNCTION INDEX

C

CCPU_ChangeFileSecurity	38
CCPU_ClearError	39
CCPU_Control	40
CCPU_DedicatedDInst	41
CCPU_DedicatedGInst	44
CCPU_DedicatedJInst	46
CCPU_DedicatedMInst	48
CCPU_DisableInt	50
CCPU_DisableInt_ISR	112
CCPU_EnableInt	51
CCPU_EnableInt_ISR	113
CCPU_EntryInt	52
CCPU_EntryTimerEvent	54
CCPU_EntryWDTInt	55
CCPU_FromBuf	56
CCPU_FromBufHG	57
CCPU_FromBufHG_ISR	115
CCPU_FromBuf_ISR	114
CCPU_GetConstantProcessStatus	58
CCPU_GetCounterMicros	59
CCPU_GetCounterMicros_ISR	116
CCPU_GetCounterMillis	60
CCPU_GetCounterMillis_ISR	117
CCPU_GetCpuStatus	61
CCPU_GetDotMatrixLED	62
CCPU_GetDotMatrixLED_ISR	118
CCPU_GetErrInfo	63
CCPU_GetFileSecurity	64
CCPU_GetIDInfo	65
CCPU_GetLEDStatus	66
CCPU_GetOpSelectMode	68
CCPU_GetPowerStatus	69
CCPU_GetRTC	70
CCPU_GetSerialNo	71
CCPU_GetSwitchStatus	72
CCPU_GetUnitInfo	73
CCPU_MountMemoryCard	76
CCPU_ReadDevice	77
CCPU_ReadDevice_ISR	119
CCPU_ReadLinkDevice	78
CCPU_RegistEventLog	79
CCPU_RegistEventLog_ISR	120
CCPU_Reset	81
CCPU_ResetDevice	82
CCPU_ResetDevice_ISR	122
CCPU_ResetWDT	83
CCPU_SetDevice	84
CCPU_SetDevice_ISR	123
CCPU_SetDotMatrixLED	85
CCPU_SetDotMatrixLED_ISR	124
CCPU_SetLEDStatus	87
CCPU_SetLEDStatus_ISR	126
CCPU_SetOpSelectMode	88
CCPU_SetRTC	89
CCPU_ShutdownRom	90
CCPU_StartWDT	91
CCPU_StopWDT	92
CCPU_SysClkRateGet	93
CCPU_SysClkRateSet	94
CCPU_ToBuf	95

CCPU_ToBufHG	96
CCPU_ToBufHG_ISR	128
CCPU_ToBuf_ISR	127
CCPU_UnmountMemoryCard	97
CCPU_WaitEvent	98
CCPU_WaitSwitchEvent	100
CCPU_WaitTimerEvent	101
CCPU_WaitUnitEvent	102
CCPU_WriteDevice	104
CCPU_WriteDevice_ISR	129
CCPU_WriteLinkDevice	105
CCPU_X_In_BitEx	106
CCPU_X_In_WordEx	107
CCPU_X_In_Word_ISR	130
CCPU_Y_In_BitEx	108
CCPU_Y_In_WordEx	109
CCPU_Y_In_Word_ISR	131
CCPU_Y_Out_BitEx	110
CCPU_Y_Out_WordEx	111
CCPU_Y_Out_Word_ISR	132

M

mdClose	133
mdControl	134
mdDevRstEx	135
mdDevSetEx	136
mdInit	137
mdOpen	138
mdRandREx	139
mdRandWEx	142
mdReceiveEx	144,145
mdSendEx	147,148
mdTypeRead	150

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

Japanese manual number: SH-081370-B

This manual confers no industrial property rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2015 MITSUBISHI ELECTRIC CORPORATION

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

- (1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
- (2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
 1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
 2. Failure caused by unapproved modifications, etc., to the product by the user.
 3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
 4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
 5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
 6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
 7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

- (1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.
- (2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation of damages caused by any cause found not to be the responsibility of Mitsubishi, loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products, special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products, replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

TRADEMARKS

Microsoft, Windows, Windows Vista, Windows NT, Windows XP, Windows Server, Visio, Excel, PowerPoint, Visual Basic, Visual C++, and Access are either registered trademarks or trademarks of Microsoft Corporation in the United States, Japan, and other countries.

Intel, Pentium, and Celeron are either registered trademarks or trademarks of Intel Corporation in the United States and other countries.

Ethernet is a trademark of Xerox Corp.

The SD and SDHC logos are either registered trademarks or trademarks of SD-3C, LLC.

VxWorks is either a trademark or a registered trademark of Wind River Systems, Inc. in the United States.

All other company names and product names used in this manual are either trademarks or registered trademarks of their respective companies.



SH(NA)-081371ENG-B(1504)

MODEL:R-CCPU-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

When exported from Japan, this manual does not require application to the
Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.