



Mitsubishi Programmable Controller

MELSEC iQ-R
series



MELSEC iQ-R C Intelligent Function Module
Programming Manual

SAFETY PRECAUTIONS

(Read these precautions before using this product.)

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

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 intelligent function 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 intelligent function module.

For the problems or specifications of the Wind River Systems product, refer to the corresponding manual or consult Wind River Systems, Inc.

Contact information is available on the following website.

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

INTRODUCTION

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

This manual describes the functions required for programming.

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

Please make sure that the end users read this manual.

CONTENTS

SAFETY PRECAUTIONS	1
CONDITIONS OF USE FOR THE PRODUCT	1
CONSIDERATIONS FOR USE	2
INTRODUCTION	2
RELEVANT MANUALS	4
TERMS	5
CHAPTER 1 COMMON ITEMS	6
1.1 Header Files	6
1.2 C Intelligent Function Module Dedicated Functions	7
Program processing	7
Considerations	7
1.3 MELSEC iQ-R Series Data Link Functions	8
Program processing	8
Considerations	9
Accessible range	10
Argument specification	15
1.4 Considerations on Interrupt Service Routine (ISR)	18
CHAPTER 2 FUNCTION LIST	19
2.1 C Intelligent Function Module Dedicated Functions	19
C intelligent function module dedicated functions	19
C intelligent function module dedicated functions for ISR	21
2.2 MELSEC iQ-R Series Data Link Functions	22
CHAPTER 3 DETAILS OF FUNCTION	23
3.1 C Intelligent Function Module Dedicated Functions	23
C intelligent function module dedicated functions	23
C intelligent function module dedicated functions for ISR	63
3.2 MELSEC iQ-R Series Data Link Functions	74
CHAPTER 4 ERROR CODE LIST	99
4.1 Common Error Codes	99
4.2 C Intelligent Function Module Dedicated Functions	102
4.3 MELSEC iQ-R Series Data Link Functions	104
INDEX	108
FUNCTION INDEX	109
REVISIONS	110
WARRANTY	111
TRADEMARKS	112

RELEVANT MANUALS

Manual name [manual number]	Description	Available form
MELSEC iQ-R C Intelligent Function Module Programming Manual [SH-081568ENG] (this manual)	Explains the programming specifications and dedicated function library of the C intelligent function module.	e-Manual PDF
MELSEC iQ-R C Intelligent Function Module User's Manual (Startup) [SH-081566]	Explains the specifications, procedure before operation, wiring, and operation examples of the C intelligent function module.	Print book e-Manual PDF
MELSEC iQ-R C Intelligent Function Module User's Manual (Application) [SH-081567]	Explains the functions, input/output signals, buffer memory, parameter setting, and troubleshooting of the C intelligent function module.	Print book e-Manual PDF
CW Workbench/CW-Sim Operating Manual [SH-081373ENG]	Explains the system configuration, specifications, functions, and troubleshooting of CW Workbench/CW-Sim.	e-Manual PDF

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
C intelligent function module	An abbreviation for MELSEC iQ-R series C intelligent function module
C intelligent function module dedicated function	A dedicated function library offered by the C intelligent function module It is used to control the C intelligent function module.
CW Workbench	An engineering tool for C Controller modules and C intelligent function modules.
CW-Sim	An abbreviation for VxWorks simulator that can operate and debug the C Controller module and C intelligent function module programs on a personal computer on which CW Workbench installed, without connecting to the actual machine (target)
Dedicated function library	A generic term for C intelligent function module dedicated function and MELSEC iQ-R series data link function
Engineering tool	Another term of the software package for the MELSEC programmable controllers This manual explains the GX Works3.
GX Works3	A generic product name for SWnDND-GXW3-E. ('n' indicates version.)
MELSEC iQ-R series data link function	A dedicated function library offered by the C intelligent function module It is used to access an own station or the modules on the network.
RD55UP06-V	An abbreviation for RD55UP06-V C intelligent function module.
VxWorks	A product name for the real-time operating system manufactured by Wind River Systems, Inc.

1 COMMON ITEMS

A user program is created by using the VxWorks standard API functions^{*1} and dedicated function library provided by the C intelligent function module in accordance with the specification of VxWorks, the operating system of C intelligent function module.

*1 For details on the VxWorks corresponding to the C intelligent function module, refer to the following guide.


 VxWorks"KERNEL PROGRAMMER'S GUIDE"

Dedicated function library offered by C intelligent function module are as follows:

- C intelligent function module dedicated function
- MELSEC iQ-R series data link function

Point

For the execution procedure of user programs for the C intelligent function module, refer to the following manual.

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

1.1 Header Files


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

Dedicated function library	Header file
C intelligent function module dedicated function	CITLFunc.h
MELSEC iQ-R series data link function	MDRFunc.h

Point

The header files are stored in the C intelligent function module.

For details on storage destination, refer to the following manual.

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

1.2 C Intelligent Function Module Dedicated Functions

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

Program processing

The following procedure shows the processing flow of the user program using C intelligent function module.

1. Start a task.
2. Using C intelligent function module dedicated function, read the status of C intelligent function module, and access the resources such as LED control.
3. Complete the task.

Considerations

The following shows the considerations when using the C intelligent function module dedicated function.

Considerations for user WDT (User watchdog timer)

■A user WDT error occurrence

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

In this case, take the following corrective actions.

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

Reset the C intelligent function module once the above corrective actions have been taken.

Point

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

■User WDT setting range

The user WDT period can be set within the range from 100 ms to 10000 ms.

1.3 MELSEC iQ-R Series Data Link Functions

MELSEC iQ-R series data link functions are the integrated communication function libraries which are independent of the communication protocols.

A program can be created regardless of the target hardware or communication protocols by using MELSEC iQ-R series data link functions while creating program to communicate to a CPU module.

The communication functions supported by the MELSEC iQ-R series data link function are as follows:

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

Program processing

The following procedure shows the processing flow of the user program using MELSEC iQ-R series data link function.

When accessing by the device name

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

When accessing by the label name

1. Start a task.
2. Open a communication line. (mdrOpen function)
3. Get the device name assigned to the label name by the target CPU module. (mdrGetLabelInfo function)
4. Access the target CPU module by specifying the obtained device name. (mdrRandRLabel/mdrRandWLabel function)
5. Make sure that label assignment information of the target CPU module does not change.
If it changes, go back to the procedure 3.
6. To stop writing/reading the data to/from the device, go to procedure 7.
To write/read data to/from another device, go back to the procedure 4.
7. Close a communication line. (mdrClose function)
8. Complete the task.

Considerations

The following shows the considerations when using the MELSEC iQ-R series data link function.

1

Considerations for programming

■Open/close processing of communication line (mdrOpen/mdrClose function)

Perform the open/close processing of communication line (the mdrOpen/mdrClose 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.

■At the first execution of the function after using the mdrOpen function

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

■Number of accessing other stations on the same task

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

■mdrGetLabelInfo function call

The mdrGetLabelInfo function does not need to be called each time to access the target CPU module.

Only if the error occurs (Error code: -81) when accessing by the mdrRandRLabel/mdrRandWLabel function, call the mdrGetLabelInfo function again.

■taskDelete execution

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

■Error by access concentration in a CPU module

When using the MELSEC iQ-R series data link function to access a CPU module from multiple modules or a built-in Ethernet port of the CPU module in the system where communication processing such as device access to a CPU module is performed frequently, an error may occur in communication processing on other modules due to the concentration of processing to a CPU module.

When verifying the operation at the system construction and if an error occurs in communication processing, take a following measure before running the system.

- When executing the MELSEC iQ-R series data link function in multiple tasks, do not execute the function at the same time by exclusion control, or execute the function in one task.
- Lengthen the execution interval of the MELSEC iQ-R series data link function to avoid errors in communication processing.

Accessible range

This section shows the accessible CPU module , device, and route by using the C intelligent function module.

Accessible CPU modules

■MELSEC iQ-R series

Module		Model
RCPU	Programmable controller CPU	R04CPU, R08CPU, R16CPU, R32CPU, and R120CPU
	Process CPU	R08PCPU, R16PCPU, R32PCPU, and R120PCPU
	CC-Link IE built-in CPU	R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, and R120ENCPU
C Controller module		R12CCPU-V

■MELSEC-Q series

Module		Model
QCPU (Q mode)	Basic model QCPU	Q00JCPU, Q00CPU, Q01CPU
	High Performance model QCPU	Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, and Q25HCPU
	Universal model QCPU	Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UD(E)CPU, Q04UD(E)HCPU, Q06UD(E)HCPU, Q10UD(E)HCPU, Q13UD(E)HCPU, Q20UD(E)HCPU, Q26UD(E)HCPU, Q50UDEHCPU, Q100UDEHCPU, Q03UDVCPU, Q04UD(P)VCPU, Q06UD(P)VCPU, Q13UD(P)VCPU, and Q26UD(P)VCPU
	Process CPU	Q02PHCPU, Q06PHCPU, Q12PHCPU, and Q25PHCPU
C Controller module		Q12DCCPU-V*1, Q24DHCCPU-V, Q24DHCCPU-VG, and Q24DHCCPU-LS

*1 Only a serial number of which the first 5 digits are "12042" or later

■MELSEC-L series

Module	Model
LCPU	L02CPU, L02CPU-P, L02SCPU, L06CPU, L26CPU, L26CPU-BT, and L26CPU-PBT

Accessible routes

The access target CPU modules are as follows:

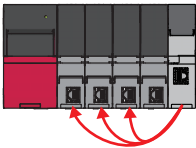
Access target	Accessible CPU modules
(1)	RCPU
(2)	MELSEC iQ-R series C Controller module*1
(3)	QCPU (Q mode)
(4)	LCPU
(5)	MELSEC-Q series C Controller module*1

*1 The module cannot be used as a relay station.

■ Accessing own station (control CPU, other multiple CPU)

Access the CPU module of the station on which C intelligent function module is mounted.

Own station



○: Accessible, ×: Not accessible, —: Not applicable

Access route	Access target				
	(1)	(2)	(3)	(4)	(5)
Control CPU	○	×	—	—	—
The other CPU of a multiple CPU system	○ (CPU No.1 to 4)	○ (CPU No.1 to 4)	—	—	—

■ The access via single network (specify a network number and station number.)

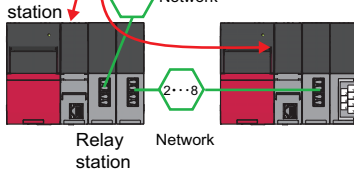
Access by specifying the network number and station number of the target station.

When the access target CPU module can be specified by the network number and station number (CPU number) from the station on which C intelligent function module is mounted in the status where the access target CPU module is connected to the network, the access to the CPU module that is mounted on the eighth network is available via a relay station of RCPU or QCPU (Q mode).

Own station



Other station

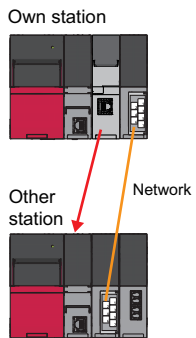


○: Accessible, ×: Not accessible, —: Not applicable

Access route	Access target				
	(1)	(2)	(3)	(4)	(5)
CC-Link IE Controller Network	○ (CPU No.1 to 4)	○ (CPU No.1 to 4)	○ (CPU No.1 to 4)	—	○ (CPU No.1 to 4)
CC-Link IE Field Network				○	
MELSECNET/H				—	

■The access via single network (specify a start I/O number and a station number of the target station.)

Access by specifying a start I/O number of module to access the target station and a station number of the target station.
The following shows the route when the access target CPU module and C intelligent function module of the mounting side are directly connected.



○: Accessible, ×: Not accessible, —: Not applicable

Access route	Access target				
	(1)	(2)	(3)	(4)	(5)
CC-Link	○ (CPU No.1 to 4)	○ (CPU No.1 to 4)	○ (CPU No.1 to 4)	○	○ (CPU No.1 to 4)

Accessible devices

The access target CPU modules are as follows:

Access target	Accessible CPU modules
(1)	RCPU
(2)	MELSEC iQ-R series C Controller module
(3)	QCPU (Q mode)
(4)	LCPU
(5)	MELSEC-Q series C Controller module

○: Accessible, ×: Not accessible

Device (Device name) ^{*1}	Access target					
	(1)	(2)	(3)	(4)	(5)	
Function input (FX)	×	×	×	×	×	
Function output (FY)	×	×	×	×	×	
Function register (FD)	×	×	×	×	×	
Special relay (SM)	○	○	○	○	○	
Special register (SD)	○	○	○	○	○	
Input relay (X)	○	○	○	○	○	
Output relay (Y)	○	○	○	○	○	
Internal relay (M)	○	○	○	○	○	
Latch relay (L)	○	×	○	○	×	
Annunciator (F)	○	×	○	○	×	
Edge relay (V)	○	×	○	○	×	
Link relay (B)	○	○	○	○	○ ^{*2}	
Data register (D)	○	○	○	○	○ ^{*3}	
Link register (W)	○	○	○	○	○ ^{*2}	
Extended internal relay (M)	×	×	○	○	○ ^{*3}	
Extended data register (D) ^{*4}	×	×	○	○	○	
Extended link register (W) ^{*4}	×	×	○	○	○	
Timer	Contact (TS)	○	×	○	○	×
	Coil (TC)	○	×	○	○	×
	Current value (T/TN) ^{*5}	○	×	○	○	×
Long timer	Contact (LTS)	○	×	×	×	×
	Coil (LTC)	○	×	×	×	×
	Current value (LT/LTN) ^{*5}	○	×	×	×	×
Counter	Contact (CS)	○	×	○	○	×
	Coil (CC)	○	×	○	○	×
	Current value (C/CN) ^{*5}	○	×	○	○	×
Long counter	Contact (LCS)	○	×	×	×	×
	Coil (LCC)	○	×	×	×	×
	Current value (LC/LCN) ^{*5}	○	×	×	×	×
Retentive timer	Contact (STS, SS) ^{*6}	○	×	○	○	×
	Coil (STC, SC) ^{*6}	○	×	○	○	×
	Current value (ST/STN, ST/SN) ^{*6} ^{*5}	○	×	○	○	×
Long retentive timer	Contact (LSTS)	○	×	×	×	×
	Coil (LSTC)	○	×	×	×	×
	Current value (LST/LSTN) ^{*5}	○	×	×	×	×
Link special relay (SB)	○	×	○	○	×	
Link special register (SW)	○	×	○	○	×	
Step relay (S)	×	×	×	×	×	
Direct input (DX)	×	×	×	×	×	
Direct output (DY)	×	×	×	×	×	
Accumulator (A)	×	×	×	×	×	

Device (Device name) ^{*1}		Access target				
		(1)	(2)	(3)	(4)	(5)
Index register (Z)		○	×	○	○	×
Long index register (LZ)		○	×	×	×	×
File register	(R)	○	×	○ ^{*7}	○	×
	(ZR) ^{*8}	○	○	○ ^{*7}	○	×
	(ER□\R) ^{*9}	×	×	×	×	×
Link direct device ^{*10}	Link input (J□\X)	○	○	○	○	○
	Link output (J□\Y)	○	○	○	○	○
	Link relay (J□\B)	○	○	○	○	○
	Link special relay (J□\SB)	○	○	○	○	○
	Link register (J□\W)	○	○	○	○	○
	Link special register (J□\SW)	○	○	○	○	○
Refresh data register (RD)		○	×	×	×	×
Module access device	Module access device/Intelligent function module device (U□\G) ^{*11}	○	○	○	○	○
	Multiple CPU shared device (U3E□\G) ^{*12}	×	○	○	○	○
CPU buffer memory access device ^{*12}	CPU buffer memory access device (U3E□\G)	○	×	×	×	×
	CPU buffer memory access device (Fixed cycle communication area) (U3E□\HG)	○	×	×	×	×
Global label (GV) ^{*13} (No device assigned)		○	×	×	×	×

*1 The file registers for each local device and program in which the program name is specified cannot be accessed.

*2 For Q12DCCPU-V, only the extended mode can be accessed.

*3 For Q12DCCPU-V, specify "Use device function".

*4 The extended data register (D) and extended link register (W) can be accessed by the following two methods.

(1): Access by directly specifying the device name of the extended data register (D) and extended link register (W)

(2): Access to the file register (ZR) area assigned to the extended data register (D) and extended link register (W)

*5 Any device name can be specified.

*6 This is the device name in QCPU (Q mode), LCPU, and MELSEC-Q series C Controller module.

*7 It is not accessible when using Q00JCPU or Q00UJCPU.

*8 When accessing out of the range of the file register (ZR) area, the value of -1(FFFFH) is sampled.

*9 "□": Specify the block number.

*10 "□": Specify the network number.

*11 "□": Specify the start I/O number÷ 10H.

*12 "□": Specify the CPU number (CPU No.1: 0, CPU No.2: 1, CPU No.3: 2, CPU No.4: 3)

*13 Only the mdrRandRLabel/mdrRandWLabel function can be used.

Argument specification

This section shows the argument specification used by the communication function.

Channel

A channel shows the connection configuration to media and a target communicating with the C intelligent function module. A channel number is set for each module in a user program.

The channels to be used by MELSEC iQ-R series are as follows:

Channel number	Channel name	Description
12	Bus interface	Used for communication via bus.

CPU number, network number, start I/O number, station number

CPU numbers, Network numbers, start I/O numbers and station numbers to be specified to MELSEC iQ-R series data link functions are as follows:

Access route		CPU number	Network number	Start I/O number	Station number
Bus interface	Own station	<ul style="list-style-type: none"> • 0: Control CPU setting • 1 to 4: Multiple CPU setting 	—*1	—*1	—*1
CC-Link IE Controller Network	Via single network		1 to 239		1 to 120, 0 ^{*2} , 125 ^{*2}
CC-Link IE Field Network			0 to 120		
MELSECNET/H			1 to 64, 0 ^{*2} , 125 ^{*2}		
CC-Link			—*1		0000H to 00FEH

*1 No error will occur even if the value is set.

*2 A specified control station of the network, which is specified to the network number, is accessed. To access a station that is actually operating as the control station, specify the station number.

Device type

This section shows the device type to be specified for MELSEC iQ-R series data link function.

Devices are defined in the header file (MDRFunc.h).



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

Device (Device name)	Device type			
	Code specification		Device name specification	
	Decimal	Hexadecimal		
Input relay (X)	1	1H	DevX	
Output relay (Y)	2	2H	DevY	
Latch relay (L)	3	3H	DevL	
Internal relay (M)	4	4H	DevM	
Special relay (SM)	5	5H	DevSM	
CPU buffer memory ^{*1*2}	CPU No.1 area (U3E0\G)	501	1F5H	DevSPB1
	CPU No.2 area (U3E1\G)	502	1F6H	DevSPB2
	CPU No.3 area (U3E2\G)	503	1F7H	DevSPB3
	CPU No.4 area (U3E3\G)	504	1F8H	DevSPB4
Fixed cycle communication area ^{*1*2}	CPU No.1 area (U3E0\HG)	511	1FFH	DevHSPB1
	CPU No.2 area (U3E1\HG)	512	200H	DevHSPB2
	CPU No.3 area (U3E2\HG)	513	201H	DevHSPB3
	CPU No.4 area (U3E3\HG)	514	202H	DevHSPB4
Annunciator (F)	6	6H	DevF	

Device (Device name)		Device type		
		Code specification		Device name specification
		Decimal	Hexadecimal	
Timer	Contact (TS)	7	7H	DevTT
	Coil (TC)	8	8H	DevTC
	Current value (T/TN)	11	BH	DevTN
Long timer	Contact (LTS)	41	29H	DevLTT
	Coil (LTC)	42	2AH	DevLTC
	Current value (LT/LTN)	43	2BH	DevLTN
Counter	Contact (CS)	9	9H	DevCT
	Coil (CC)	10	AH	DevCC
	Current value (C/CN)	12	CH	DevCN
Long counter	Contact (LCS)	44	2CH	DevLCT
	Coil (LCC)	45	2DH	DevLCC
	Current value (LC/LCN)	46	2EH	DevLCN
Retentive timer	Contact (STS, SS)	26	1AH	DevSTT
	Coil (STC, SC)	27	1BH	DevSTC
	Current value (ST/STN, ST/SN)	35	23H	DevSTN
Long retentive timer	Contact (LSTS)	47	2FH	DevLSTT
	Coil (LSTC)	48	30H	DevLSTC
	Current value (LST/LSTN)	49	31H	DevLSTN
Data register (D)		13	DH	DevD
Special register (SD)		14	EH	DevSD
Index register (Z) ^{*3}		20	14H	DevZ
Long index register (LZ) ^{*3}		38	26H	DevLZ
File register (R) ^{*3}		22	16H	DevR
File register (ZR) ^{*3}		220	DCH	DevZR
Link relay (B)		23	17H	DevB
Link register (W)		24	18H	DevW
Link special relay (SB) ^{*3}		25	19H	DevQSB
Link special register (SW) ^{*3}		28	1CH	DevQSW
Edge relay (V)		30	1EH	DevQV
Module refresh register (RD)		39	27H	DevRD
Global label (GV) ^{*5}	For word, double word, and quad word size	600	258H	DevGV
	For bit 0	601	259H	DevGV_0
	For bit 1	602	25AH	DevGV_1
	For bit 2	603	25BH	DevGV_2
	For bit 3	604	25CH	DevGV_3
	For bit 4	605	25DH	DevGV_4
	For bit 5	606	25EH	DevGV_5
	For bit 6	607	25FH	DevGV_6
	For bit 7	608	260H	DevGV_7
	For bit 8	609	261H	DevGV_8
	For bit 9	610	262H	DevGV_9
	For bit A	611	263H	DevGV_A
	For bit B	612	264H	DevGV_B
	For bit C	613	265H	DevGV_C
	For bit D	614	266H	DevGV_D
	For bit E	615	267H	DevGV_C
For bit F	616	268H	DevGV_F	

Device (Device name)		Device type		
		Code specification		Device name specification
		Decimal	Hexadecimal	
Link direct device ^{*3*4} Argument value of device name (1 to 255): Network number	Link input (J□\X)	1001 to 1255	3E9H to 4E7H	DevLX(1) to DevLX(255)
	Link output (J□\Y)	2001 to 2255	7D1H to 8CFH	DevLY(1) to DevLY(255)
	Link relay (J□\B)	23001 to 23255	59D9H to 5AD7H	DevLB(1) to DevLB(255)
	Link register (J□\W)	24001 to 24255	5DC1H to 5EBFH	DevLW(1) to DevLW(255)
	Link special relay (J□\SB)	25001 to 25255	61A9H to 62A7H	DevLSB(1) to DevLSB(255)
	Link special register (J□\SW)	28001 to 28255	6D61H to 6E5FH	DevLSW(1) to DevLSW(255)
Intelligent function module device, module access device ^{*3} Argument value of device name (0 to 255): Start I/O number ÷ 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.

*2 The devices cannot be used for the mdrDevRst/mdrDevSet/mdrRandR/mdrRandW functions.

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

*4 "□": Shows a network number.

*5 Only the mdrRandRLabel/mdrRandWLabel function can be used.

1.4 Considerations on Interrupt Service Routine (ISR)

Fully understand the restrictions of VxWorks, the operating system for the C intelligent function module, before creating routines to be executed by an interrupt service routine (ISR: InterruptServiceRoutine). To use the other dedicated function with synchronously with an interrupt, implement notification processing in a user program and perform the processing in a task.

Point

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

2 FUNCTION LIST






The functions used for the C intelligent function module are shown below.

2.1 C Intelligent Function Module Dedicated Functions

The C intelligent function module dedicated functions are as listed below.

C intelligent function module dedicated functions

Function name	Description	Reference
CITL_ChangeFileSecurity	Changes the access restriction status for a file of C intelligent function module.	Page 23 CITL_ChangeFileSecurity
CITL_ClearError	Clears errors of C intelligent function module.	Page 24 CITL_ClearError
CITL_DisableYInt	Disables the routine registered with the CITL_EntryYInt function.	Page 25 CITL_DisableYInt
CITL_EnableYInt	Enables the routine registered with the CITL_EntryYInt function.	Page 26 CITL_EnableYInt
CITL_EntryDedicatedInstFunc	Registers a routine to be executed using dedicated instruction (G(P).CEXECUTE).	Page 27 CITL_EntryDedicatedInstFunc
CITL_EntryTimerEvent	Registers a timer event.	Page 28 CITL_EntryTimerEvent
CITL_EntryWDTInt	Registers a routine to be called when a user WDT error interrupt occurs.	Page 30 CITL_EntryWDTInt
CITL_EntryYInt	Registers a routine to be called when a output signal (Y) interrupt occurs.	Page 31 CITL_EntryYInt
CITL_FromBuf	Reads data from the buffer memory of C intelligent function module.	Page 32 CITL_FromBuf
CITL_GetCounterMicros	Obtains a 1 μ s counter value of C intelligent function module.	Page 33 CITL_GetCounterMicros
CITL_GetCounterMillis	Obtains a 1 ms counter value of C intelligent function module.	Page 34 CITL_GetCounterMillis
CITL_GetErrInfo	Obtains the error information of C intelligent function module.	Page 35 CITL_GetErrInfo
CITL_GetFileSecurity	Obtains the file access mode.	Page 36 CITL_GetFileSecurity
CITL_GetIDInfo	Obtains the individual identification information of C intelligent function module.	Page 37 CITL_GetIDInfo
CITL_GetLEDStatus	Obtains the LED status of C intelligent function module.	Page 38 CITL_GetLEDStatus
CITL_GetSerialNo	Obtains the serial number of C intelligent function module.	Page 39 CITL_GetSerialNo
CITL_GetSwitchStatus	Obtains the switch status of C intelligent function module.	Page 40 CITL_GetSwitchStatus
CITL_GetTime	Obtains the clock data (local time) of C intelligent function module.	Page 41 CITL_GetTime
CITL_GetUnitStatus	Obtains the operating status of C intelligent function module.	Page 42 CITL_GetUnitStatus
CITL_MountMemoryCard	Mounts the SD memory card inserted to C intelligent function module.	Page 43 CITL_MountMemoryCard
CITL_RegistEventLog	Registers event logs in the event history of control CPU module.	Page 44 CITL_RegistEventLog
CITL_ResetWDT	Resets the user WDT of C intelligent function module.	Page 45 CITL_ResetWDT
CITL_SetLEDStatus	Sets the LED status of C intelligent function module.	Page 46 CITL_SetLEDStatus
CITL_ShutdownRom	Shuts down the standard ROM of C intelligent function module.	Page 47 CITL_ShutdownRom
CITL_StartWDT	Sets and starts the user WDT of C intelligent function module.	Page 48 CITL_StartWDT
CITL_StopWDT	Stops the user WDT of C intelligent function module.	Page 49 CITL_StopWDT
CITL_SysClkRateGet	Reads the system clock rate specified with the CITL_SysClkRateSet function from the flash ROM.	Page 50 CITL_SysClkRateGet
CITL_SysClkRateSet	Stores the specified system clock rate into the flash ROM.	Page 51 CITL_SysClkRateSet
CITL_ToBuf	Writes data to the buffer memory of C intelligent function module.	Page 52 CITL_ToBuf
CITL_UnmountMemoryCard	Unmounts the SD memory card inserted to C intelligent function module.	Page 53 CITL_UnmountMemoryCard
CITL_WaitSwitchEvent	Waits for a switch interrupt event of C intelligent function module to occur.	Page 54 CITL_WaitSwitchEvent
CITL_WaitTimerEvent	Waits for a timer event to occur.	Page 55 CITL_WaitTimerEvent
CITL_WaitYEvent	Waits for the output signal (Y) interrupt event notification.	Page 56 CITL_WaitYEvent
CITL_X_In_Bit	Reads the input signal (X) in bit (1-point) units.	Page 57 CITL_X_In_Bit















Function name	Description	Reference
CITL_X_In_Word	Reads the input signal (X) in word (16-point) units.	 Page 58 CITL_X_In_Word
CITL_X_Out_Bit	Writes to the input signal (X) in bit (1-point) units.	 Page 59 CITL_X_Out_Bit
CITL_X_Out_Word	Writes to the input signal (X) in word (16-point) units.	 Page 60 CITL_X_Out_Word
CITL_Y_In_Bit	Reads the output signal (Y) in bit (1-point) units.	 Page 61 CITL_Y_In_Bit
CITL_Y_In_Word	Reads the output signal (Y) in word (16-point) units.	 Page 62 CITL_Y_In_Word

C intelligent function module dedicated functions for ISR

Function name	Description	Reference
CITL_DisableYInt_ISR	Disables the routine registered with the CITL_EntryYInt function.	☞ Page 63 CITL_DisableYInt_ISR
CITL_EnableYInt_ISR	Enables the routine registered with the CITL_EntryYInt function.	☞ Page 64 CITL_EnableYInt_ISR
CITL_FromBuf_ISR	Reads data from the buffer memory of C intelligent function module.	☞ Page 65 CITL_FromBuf_ISR
CITL_GetCounterMicros_ISR	Obtains a 1 μ s counter value of C intelligent function module.	☞ Page 66 CITL_GetCounterMicros_ISR
CITL_GetCounterMillis_ISR	Obtains a 1 ms counter value of C intelligent function module.	☞ Page 67 CITL_GetCounterMillis_ISR
CITL_RegistEventLog_ISR	Registers event logs in the event history of control CPU module.	☞ Page 68 CITL_RegistEventLog_ISR
CITL_SetLEDStatus_ISR	Sets the LED status of C intelligent function module.	☞ Page 69 CITL_SetLEDStatus_ISR
CITL_ToBuf_ISR	Writes data to the buffer memory of C intelligent function module.	☞ Page 70 CITL_ToBuf_ISR
CITL_X_In_Word_ISR	Reads the input signal (X) in word (16-point) units.	☞ Page 71 CITL_X_In_Word_ISR
CITL_X_Out_Word_ISR	Writes to the input signal (X) in word (16-point) units.	☞ Page 72 CITL_X_Out_Word_ISR
CITL_Y_In_Word_ISR	Reads the output signal (Y) in word (16-point) units.	☞ Page 73 CITL_Y_In_Word_ISR

2.2 MELSEC iQ-R Series Data Link Functions

The MELSEC iQ-R series data link functions are as listed below.

Function name	Description	Reference
mdrClose	Closes a communication line (channel).	 Page 74 mdrClose
mdrControl	Performs remote operations (RUN/STOP/PAUSE) for the CPU module.	 Page 75 mdrControl
mdrDevRst	Resets (turns OFF) bit device(s).	 Page 76 mdrDevRst
mdrDevSet	Sets (turns ON) bit device(s).	 Page 77 mdrDevSet
mdrGetLabelInfo	Obtains device information corresponding to label names.	 Page 78 mdrGetLabelInfo
mdrInit	Initializes communication route information.	 Page 81 mdrInit
mdrOpen	Opens a communication line (channel).	 Page 82 mdrOpen
mdrRandR	Reads device(s) randomly.	 Page 83 mdrRandR
mdrRandRLabel	Reads the device(s) randomly corresponding to the labels.	 Page 86 mdrRandRLabel
mdrRandW	Writes device(s) randomly.	 Page 89 mdrRandW
mdrRandWLabel	Writes the device(s) randomly corresponding to the labels.	 Page 91 mdrRandWLabel
mdrReceive	Reads devices in batch.	 Page 94 mdrReceive
mdrSend	Writes devices in batch.	 Page 95 mdrSend
mdrTypeRead	Reads the model code of a CPU module.	 Page 96 mdrTypeRead

3 DETAILS OF FUNCTION

This chapter shows the details of the C intelligent function module dedicated function and the MELSEC iQ-R series data link function.

3.1 C Intelligent Function Module Dedicated Functions

This section shows the details of the C intelligent function module dedicated function.

C intelligent function module dedicated functions

CITL_ChangeFileSecurity

Changes the access restriction status for a file of the C intelligent function module.

■Format

short CITL_ChangeFileSecurity(short sMode, char* pcPass)


■Argument

Argument	Name	Description	IN/OUT
sMode	File access mode	Specify the file access mode. (When 'Reserved' is specified, this function ends normally without processing.) <ul style="list-style-type: none">• 0: Access restriction clear mode• 1: Access restriction mode• Others: Reserved	IN
pcPass	Password	Specify the password (the security password) for restricting access to a file.	IN

■Description

Specify the file access mode (sMode) for restricting access to a file. To change the file access mode (sMode), use the security password set in the parameter.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 36 CITL_GetFileSecurity

CITL_ClearError

Clears errors of the C intelligent function module.

■Format

short CITL_ClearError(long* pErrorInfo)


■Argument

Argument	Name	Description	IN/OUT
pErrorInfo	Error information	Not required	IN

■Description

- This function clears errors of the C intelligent function module.
- This function ends normally when the CITL_ClearError function is executed without an error in the C intelligent function module.
- When major or moderate unit error occurs in the C intelligent function module, the error cannot be deleted even if CITL_ClearError function is executed. (CITL_ClearError function ends normally.)

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 35 CITL_GetErrInfo

CITL_DisableYInt

Disables the routine registered with the CITL_EntryYInt function.

■Format

short CITL_DisableYInt (short sYNo)


■Argument

Argument	Name	Description	IN/OUT
sYNo	Output signal (Y) number	Specify the output signal (Y) number. (If -1 is specified, disable all the registered routines.)	IN

■Description

- This function disables the routine registered by using the CITL_EntryYInt function. (The registered routine is not executed when output signal (Y) interrupt occurs.)
- Specify the output signal (Y) number (sYNo) specified in the CITL_EntryYInt function in the output signal (Y) number (sYNo).
- The output signal (Y) interrupt event notification wait function (CITL_WaitYEvent function) and the function executing interrupt routine when output signal (Y) interrupts (Defined by CITL_EntryYInt/CITL_EnableYInt/CITL_DisableYInt function) operate independently. These functions operate independently even if interrupt occurs by the same output signals (Y).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 26 CITL_EnableYInt

 Page 31 CITL_EntryYInt

CITL_EnableYInt

Enables the routine registered with the CITL_EntryYInt function.

■Format

short CITL_EnableYInt (short sYNo)


■Argument

Argument	Name	Description	IN/OUT
sYNo	Output signal (Y) number	Specify the output signal (Y) number. (If -1 is specified, enable all the registered routines.)	IN

■Description

- This function enables the routine registered by using the CITL_EntryYInt function. (The registered routine is executed when the output signal (Y) interrupt occurs.)
- Specify the output signal (Y) number (sYNo) specified in the CITL_EntryYInt function in the output signal (Y) number (sYNo).
- The output signal (Y) interrupt event notification wait function (CITL_WaitYEvent function) and the function executing interrupt routine when output signal (Y) interrupts (Defined by CITL_EntryYInt/CITL_EnableYInt/CITL_DisableYInt function) operate independently. These functions operate independently even if interrupt occurs by the same output signals (Y).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 25 CITL_DisableYInt

 Page 31 CITL_EntryYInt

CITL_EntryDedicatedInstFunc

Registers a routine to be executed using dedicated instruction (G(P).CEXECUTE).

■Format

short CITL_EntryDedicatedInstFunc (CITL_CEXECUTEFUNCPTR pCEXECUTEFuncPtr)

■Argument

Argument	Name	Description	IN/OUT
pCEXECUTEFuncPtr	Registered routine	Specify the routine to be registered. (The routine is deregistered by specifying NULL.)	IN

The data type of registered routine (pCEXECUTEFuncPtr) is defined by the header file (CITLFunc.h) as follows:

- void (*CITL_CEXECUTEFUNCPTR) (unsigned short* pusReqData, unsigned short* pusReqSize, unsigned short* pusAnsData, unsigned short* pusAnsSize)

Argument of data type	Name	Description	IN/OUT
pusReqData	Request data	Receives the requested data specified by the dedicated instruction.	IN
pusReqSize	Request data size	Receives the size of the requested data specified by the dedicated instruction.	IN
pusAnsData	Response data	Returns the response data to the dedicated instruction.	OUT
pusAnsSize	Response data size	Returns the size of the response data to the dedicated instruction.	OUT

■Description

- Registers a routine to be executed using the dedicated instruction in a register routine (pCEXECUTEFuncPtr).
- The registered routine is operated on a task with the following settings.
 - Task priority: 100
 - Stack size: 40000 byte
 - Task option: VX_FP_TASK
- When NULL is specified to the registered routine (pCEXECUTEFuncPtr), the routine is deregistered.
- When CITL_EntryDedicatedInstFunc function is executed several times, the last registered routine will be in effect.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. Page 99 ERROR CODE LIST

CITL_EntryTimerEvent

Registers a timer event.

■Format

short CITL_EntryTimerEvent (long* pEvent)

■Argument

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

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

Stored information position	Stored information	
pEvent[0]	Number of timer event settings (1 to 16)	
pEvent[1]	First timer event number (1 to 16)	First timer event setting
pEvent[2]	Cycle of the first timer event (Clear: 0, Cycle: 1 to 60,000 [ms])	
pEvent[3]	Synchronization type of the first timer event (Batch synchronization: 0, Individual synchronization: 1)	
pEvent[4]	Second timer event number (1 to 16)	Second timer event setting
pEvent[5]	Cycle of the second timer event (Clear: 0, Cycle: 1 to 60,000 [ms])	
pEvent[6]	Synchronization type of the second timer event (Batch synchronization: 0, Individual synchronization: 1)	
pEvent[7]	Third timer event number (1 to 16)	Third timer event setting
pEvent[8]	Cycle of the third timer event (Clear: 0, Cycle: 1 to 60,000 [ms])	
pEvent[9]	Synchronization type of the third timer event (Batch synchronization: 0, Individual synchronization: 1)	
.	.	.
.	.	.
pEvent[46]	16th timer event number (1 to 16)	16th timer event setting
pEvent[47]	Cycle of the 16th timer event (Clear: 0, Cycle: 1 to 60,000 [ms])	
pEvent[48]	Synchronization type of the 16th timer event (Batch synchronization: 0, Individual synchronization: 1)	


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

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

■Description

- The CITL_EntryTimerEvent function sets the cycle and synchronization type for the timer event registration.
- When '0' is specified to the cycle of 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 to 60,000[ms]) and synchronization type (batch synchronization or individual synchronization) can be specified for each event. For the synchronization type, refer to the description of the CITL_WaitTimerEvent function.
- Specify the timer event number without duplication. Otherwise, an error will be returned.
- To change the cycle of a timer event number that the cycle is already set, clear 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 CITL_WaitTimerEvent function.
- All the timer events are cleared at the initial status.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■ Relevant function

👉 Page 55 CRTL_WaitTimerEvent

CITL_EntryWDTInt

Registers a routine to be called when a user WDT error interrupt occurs.

■Format

short CITL_EntryWDTInt (short sType, CITL_FUNCPTR pFuncPtr)

■Argument

Argument	Name	Description	IN/OUT
sType	WDT type	Specify the WDT type. (When 'Reserved' is specified, an error is returned.) <ul style="list-style-type: none">• 0: User WDT• Others: Reserved	IN
pFuncPtr	Registered routine	Specify the routine to be registered. (The routine is deregistered by specifying NULL.)	IN

The data type of the registered routine (pFuncPtr) is defined as void type in the header file (CITLFunc.h).

■Description

- This function registers a routine to call when a user WDT error interrupt of the C intelligent function module occurs.
- Specify the routine to be registered to the registered routine (pFuncPtr).
- When CITL_EntryWDTInt function is executed several times, the last registered routine will be in effect.
- The routine registered with CITL_EntryWDTInt function is executed as an interrupt service routine (ISR) when a user WDT error occurs. (If the CITL_ResetWDT function is not executed within the time interval specified in the CITL_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.
- For processing of a routine to be registered in the registered routine (pFuncPtr), the following points must be considered.
 - The routine to be registered must not have an argument. (Prevent an argument being passed from an interrupt.)
 - When registering a routine, observe the considerations on the interrupt service routine (ISR).
 - Register minimal processing of routine so that the processing time is as short as possible.
 - The C intelligent function module dedicated functions executable from an interrupt service routine can be used for the routine to be registered. Do not use any other function.
 - (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.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 45 CITL_ResetWDT

 Page 48 CITL_StartWDT

 Page 49 CITL_StopWDT

CITL_EntryYInt

Registers a routine to be called when a output signal (Y) interrupt occurs.

■Format

short CITL_EntryYInt (short sYNo, CITL_FUNCPTR pFuncPtr)

■Argument

Argument	Name	Description	IN/OUT
sYNo	Output signal (Y) number	Specify the output signal (Y) number.	IN
pFuncPtr	Registered routine	Specify the routine to be registered. (The routine is deregistered by specifying NULL.)	IN

- The data type of the registered routine (pFuncPtr) is defined as void type in the header file (CITLFunc.h).
- Specify the output signal (Y) number in the following format.
Output signal (Y) number: 0x10 to 0x1F


■Description

- This function registers a routine specified to the registered routine (pFuncPtr) in the interrupt specified with the output signal (Y) number (sYNo).
- When NULL is specified to the registered routine (pFuncPtr), the routine is deregistered.
- Use the CITL_EnableYInt function and enable the registered routine with the CITL_EntryYInt function. If not, routine is not called.

Precautions

- When the operating system is in an interrupt disabled state, the registered routine is not executed.
- For processing of a routine to be registered in the registered routine (pFuncPtr), the following points must be considered.
The routine to be registered must not have an argument. (Prevent an argument being passed from an interrupt.)
When registering a routine, observe the considerations on the interrupt service routine (ISR).
Register minimal processing of a routine so that the processing time is as short as possible.
The C intelligent function module dedicated functions executable from an interrupt service routine can be used for the routine to be registered. Do not use any other function.
(A function to be registered is not checked for an error.)
- When the CITL_EntryYInt function is executed more than once with the same output signal (Y) number (sYNo) specified, the routine, which was specified by the registered routine (pFuncPtr) at last, will be registered. (Multiple routines cannot be registered.)
- The routine is disabled after the registration is done by the CITL_EntryYInt function.
- When the routine registered by the CITL_EntryYInt function is running, calling the routine registered in WDT error interrupt is delayed.
- The output signal (Y) interrupt event notification wait function (CITL_WaitYEvent function) and the function executing interrupt routine when output signal (Y) interrupts (Defined by CITL_EntryYInt/CITL_EnableYInt/CITL_DisableYInt function) operate independently. These functions operate independently even if interrupt occurs by the same output signals (Y).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 25 CITL_DisableYInt

 Page 26 CITL_EnableYInt

CITL_FromBuf

Reads data from the buffer memory of the C intelligent function module.

■Format

short CITL_FromBuf (unsigned long ulOffset, unsigned long ulSize, unsigned short* pusDataBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
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 data size (ulSize) from the buffer memory of the C intelligent function module, and stores the read data in the data storage destination (pusDataBuf). Data is read by specifying offset (ulOffset) address from the start of the buffer memory of the C intelligent function module.


Precautions

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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 52 CITL_ToBuf

CITL_GetCounterMicros

Obtains a 1 μ s counter value of the C intelligent function module.

■Format

short CITL_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 the C intelligent function module and stores the value in the 1 μ s counter value storage destination (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 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 34 CITL_GetCounterMillis

CITL_GetCounterMillis

Obtains a 1 ms counter value of the C intelligent function module.

■Format

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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 33 CITL_GetCounterMicros

CITL_GetErrInfo

Obtains the error information of the C intelligent function module.

■Format

short CITL_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 the C intelligent function module and stores it in the error information storage destination (pusErrorInfo).

- This function obtains the information for the size specified to the error information storage destination size (ulBufSize).
- The information to be stored in the error information storage destination (pusErrorInfo) is as follows.


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.

Stored information position	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 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 24 CITL_ClearError

CITL_GetFileSecurity

Obtains the file access mode.

■Format

short CITL_GetFileSecurity(short* psMode)

■Argument

Argument	Name	Description	IN/OUT
psMode	File access mode	Stores the file access mode. <ul style="list-style-type: none">• 0: Access restriction clear mode• 1: Access restriction mode	OUT

■Description

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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. Page 99 ERROR CODE LIST

■Relevant function

[Page 23 CITL_ChangeFileSecurity](#)

CITL_GetIDInfo

Obtains the individual identification information of the C intelligent function module.

■Format

short CITL_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 the C intelligent function module, and stores it in the individual identification information storage destination (pucGetData).
- This function obtains the information for the size specified to the individual identification information storage destination size (ulBufSize).
- The individual identification information is stored in the individual identification information storage destination (pucGetData) as shown below.

Stored information position	Stored information
pucGetData[0]	Individual identification information
pucGetData[1]	
pucGetData[2]	
pucGetData[3]	
pucGetData[4]	
pucGetData[5]	

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. Page 99 ERROR CODE LIST

■Relevant function

[Page 39 CITL_GetSerialNo](#)

CITL_GetLEDStatus

Obtains the LED status of the C intelligent function module.

■Format

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

■Argument

Argument	Name	Description	IN/OUT
lLed	Target LED	Specify the target LED. (When 'Reserved' is specified, this function ends normally without processing.) <ul style="list-style-type: none">• 0: RUN LED• 1: ERR LED• 2: CARD RDY LED• 3: USER LED• 4 to 6: Reserved• -1: All of the LEDs above• Others: Reserved	IN
pusLedInfo	LED status storage destination	Specify the storage destination of the LED status.	OUT
ulBufSize	LED status storage destination size	Specify the LED status storage destination size in word units. (When '0' is specified, this function ends normally without processing.)	IN

■Description

- This function obtains the LED information on the C intelligent function module specified by the target LED (lLed), and stores it in LED status storage destination (pusLedInfo).
- This function obtains the information for the size specified to the LED status storage destination size (ulBufSize).
- The LED status to be stored in the LED status storage destination (pusLedInfo) is as follows.

Stored information	LED status
0	OFF
1	ON (Red)
2	Flashing at low speed (Red)
3	Flashing at high speed (Red)
4	ON (Green)
5	Flashing at low speed (Green)
6	Flashing at high speed (Green)

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

Stored information position	LED arrangement
pusLedInfo[0]	RUN LED status
pusLedInfo[1]	ERR LED status
pusLedInfo[2]	CARD RDY LED status
pusLedInfo[3]	USER LED status
pusLedInfo[4]	Reserved
pusLedInfo[5]	
pusLedInfo[6]	

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. Page 99 ERROR CODE LIST

■Relevant function

[Page 35 CITL_GetErrInfo](#)

CITL_GetSerialNo

Obtains the serial number of the C intelligent function module.

■Format

short CITL_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 the C intelligent function module and stores it in the serial number storage destination (pcGetData).
- This function obtains the information for the size specified to the serial number storage destination size (ulDataSize).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

CITL_GetSwitchStatus

Obtains the switch status of the C intelligent function module.

■Format

short CITL_GetSwitchStatus(long* plStatusBuf, unsigned long ulBufSize)

■Argument


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

■Description

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

Stored information position	Stored information	Status
plStatusBuf[0]	bit31 to 6	Reserved
	bit5 to 3	MODE/SELECT switch status
	bit2 to 0	Reserved

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

CITL_GetTime

Obtains the clock data (local time) of the C intelligent function module.

■Format

short CITL_GetTime(short* psGetData, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
psGetData	Clock data storage destination	Specify the storage destination of the clock data.	OUT
ulBufSize	Clock data storage destination size	Specify the clock data 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 the C intelligent function module and stores it in the clock data storage destination (psGetData).
- This function obtains the information for the size specified to the clock data storage destination size (ulBufSize).
- The information to be stored in the clock data storage destination (psGetData) is as follows.

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

Stored information position	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 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

CITL_GetUnitStatus

Obtains the operating status of C intelligent function module.

■Format

short CITL_GetUnitStatus(long* plStatusBuf, unsigned long ulBufSize)

■Argument


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

■Description


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

Stored information position	Stored information	Status	
plStatusBuf[0]	bit31 to 8	Reserved	—
	bit7 to 4		
	bit3 to 0		
plStatusBuf[1]	bit31 to 16	Reserved	—
	bit15 to 7		
	bit6, 5		
	bit4, 3	SD memory card status	• 0: Inserted (mounted) • 1: Inserted (unmounted) • 2: Not inserted
	bit2	Reserved	—
	bit1		
	bit0	Standard ROM shutdown status	• 0: Shutdown not performed • 1: Shutdown completed
plStatusBuf[2]	bit31 to 0	Index value for number of the standard ROM write cycle	—

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 35 CITL_GetErrInfo

CITL_MountMemoryCard

Mounts the SD memory card inserted to the C intelligent function module.

■Format

short CITL_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.) <ul style="list-style-type: none">• 1: SD memory card• Others: Reserved	IN


■Description

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

Point


Use the CITL_MountMemoryCard function to access the SD memory card again without removing it after unmounting the SD memory card by using the CITL_UnmountMemoryCard function while the power is ON. In case the SD memory card is replaced, it is mounted automatically. Therefore, it is unnecessary to use this function.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 42 CITL_GetUnitStatus

 Page 53 CITL_UnmountMemoryCard

CITL_RegistEventLog

Registers event logs in the event history of control CPU module.

■Format

short CITL_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 the control CPU module.

The contents to be registered on the event history screen of the engineering tool are as follows:

Item	Description
Occurrence date	Event registered date and time
Event type	Operation (Fixed)
Status	Information (Fixed)
Event code	25000 (Fixed)
Overview	Registration from the user program (Fixed)
Source	RD55UP06-V (Fixed)
Start I/O number	Input/output number of the C intelligent function module that executed the CITL_RegistEventLog function.
Detailed event code information	Detailed code specified to IEventCode (hexadecimal)
Detailed event log information	Detailed information specified to pcEventMsg
Cause	The event history was registered from the C intelligent function module dedicated function. (Fixed)

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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

CITL_ResetWDT

Resets the user WDT of the C intelligent function module.

■Format

short CITL_ResetWDT (short sType)

■Argument

Argument	Name	Description	IN/OUT
sType	WDT type	Specify the WDT type. (When 'Reserved' is specified, an error is returned.) <ul style="list-style-type: none">• 0: User WDT• Others: Reserved	IN

■Description

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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. Page 99 ERROR CODE LIST

■Relevant function

- [Page 30 CITL_EntryWDTInt](#)
- [Page 48 CITL_StartWDT](#)
- [Page 49 CITL_StopWDT](#)

CITL_SetLEDStatus

Sets the LED status of the C intelligent function module.

■Format

short CITL_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.) <ul style="list-style-type: none">• 0: USER LED• Others: Reserved	IN
usLedInfo	LED status information	Specify the LED status information.	IN

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

Stored information	LED status
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 the C intelligent function module to the status specified by the LED status information (usLedInfo).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. Page 99 ERROR CODE LIST

■Relevant function

[Page 38 CITL_GetLEDStatus](#)

CITL_ShutdownRom

Shuts down the standard ROM of the C intelligent function module.

■Format

short CITL_ShutdownRom (void)

■Argument

None

■Description

- This function shuts down the standard ROM of the C intelligent function module.
(The shutdown status can be checked with the function, CITL_GetUnitStatus.)
- The CITL_ShutdownRom function is used to shut down the standard ROM before turning the power OFF of the C intelligent function module. After the shut down, file operations (creating, deleting, and overwriting a file) to the standard ROM will be disabled. Reference to the standard ROM is possible.
- Before calling CITL_ShutdownRom function, it is necessary to ensure that the access to (reading from/writing to) the standard ROM is stopped and all files are closed. Otherwise, data in the standard ROM may be corrupted or a file system error may occur.
- Always turn the power OFF or reset the CPU module after checking that the module is shut down. When operation is continued without turning the power OFF or resetting the CPU module, an error occurs at the time of writing files to the standard ROM. Also, an error occurs when configuring the settings on the "Service and Account Settings" screen.
- If the standard ROM is already shut down, this function ends normally without processing.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. ☞ Page 99 ERROR CODE LIST

■Relevant function

- ☞ Page 42 CITL_GetUnitStatus
- ☞ Page 43 CITL_MountMemoryCard
- ☞ Page 53 CITL_UnmountMemoryCard

CITL_StartWDT

Sets and starts the user WDT of the C intelligent function module.

■Format

short CITL_StartWDT(short sType, short sInterval)


■Argument

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


■Description

- The user WDT is the timer for detecting a hardware failure or program error.
- This function sets an interval of the WDT to $sInterval \times 10$ ms and starts the user WDT.
- When the WDT is not reset periodically within the set time (by execution of the CITL_ResetWDT function), the user WDT error will occur. When the user WDT error occurs, the C intelligent function module will be in the moderate error status. (The RUN LED turns ON, and the ERR LED starts flashing.)
- When CITL_StartWDT function is executed while the WDT is running, an error will be returned.


■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 30 CITL_EntryWDTInt

 Page 45 CITL_ResetWDT

 Page 49 CITL_StopWDT

CITL_StopWDT

Stops the user WDT of the C intelligent function module.

■Format

short CITL_StopWDT(short sType)

■Argument

Argument	Name	Description	IN/OUT
sType	WDT type	Specify the WDT type. (When 'Reserved' is specified, an error is returned.) <ul style="list-style-type: none">• 0: User WDT• Others: Reserved	IN

■Description

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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. Page 99 ERROR CODE LIST

■Relevant function

[Page 30 CITL_EntryWDTInt](#)

[Page 45 CITL_ResetWDT](#)

[Page 48 CITL_StartWDT](#)

CITL_SysClkRateGet

Reads the system clock rate specified with the CITL_SysClkRateSet function from the flash ROM.

■Format

short CITL_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. <ul style="list-style-type: none">• 0: Default value (60 Hz)• 60 to 1000: Specified clock rate value	OUT


■Description

This function reads the system clock rate specified with the CITL_SysClkRateSet function from the flash ROM.


Precautions

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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 51 CITL_SysClkRateSet

CITL_SysClkRateSet

Stores the specified system clock rate into the flash ROM.

■Format

short CITL_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. <ul style="list-style-type: none">• 0: Default value (60 Hz)• 60 to 1000: Specified clock rate value	IN
psRestart	Restart necessity flag	Stores the necessity to restart the C intelligent function module after the execution of this function. Specify 'NULL' when storing the 'restart necessity' is not required. <ul style="list-style-type: none">• 0: Restart is not required. (The C intelligent function module has already been running at the specified clock rate.)• 1: Restart is required. (The C intelligent function module operates at the specified clock rate after restarting it.)	OUT

■Description

- This function stores the specified system clock rate into the flash ROM. The specified system clock rate will be enabled after starting the next time.
- When the output to the restart necessity flag (psRestart) is "0" (restart is not required), continue the application processing.
- When the output to the restart necessity flag (psRestart) is "1" (restart is required), restart the C intelligent function module by stopping the application processing and resetting the CPU module or turning the power OFF→ON.
- Execute the CITL_SysClkRateSet function only once after the C intelligent function module is started. If the function is executed with the same clock rate for the first time, the restart necessity flag (psRestart) will be '0' (restart is not required) regardless of the system clock rate value currently in operation.
- For more details on system clock rate, refer to the manual for VxWorks.

Precautions

Use the CITL_SysClkRateSet 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 (0000H)	Error For details on the error, refer to the following chapter. ☞ Page 99 ERROR CODE LIST

■Relevant function

☞ Page 50 CITL_SysClkRateGet

CITL_ToBuf

Writes data to the buffer memory of the C intelligent function module.

■Format

short CITL_ToBuf (unsigned long ulOffset, unsigned long ulSize, unsigned short* pusDataBuf, unsigned long ulBufSize)

■Argument

Argument	Name	Description	IN/OUT
ulOffset	Offset	Specify the offset in word units.	IN
ulSize	Data size	Specify the write data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of write data.	IN
ulBufSize	Data storage destination size	Specify '0'.	IN

■Description

This function writes the data in the storage destination (pusDataBuf) equivalent to the data size (ulSize) from the buffer memory of the C intelligent function module. The data is written by specifying the offset (ulOffset) address from the start of the buffer memory of the C intelligent function module.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. Page 99 ERROR CODE LIST

■Relevant function

[Page 32 CITL_FromBuf](#)

CITL_UnmountMemoryCard

Unmounts the SD memory card inserted to the C intelligent function module.

■Format

short CITL_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.) <ul style="list-style-type: none">• 1: SD memory card• Others: Reserved	IN

■Description

- This function unmounts the drive specified to the target drive (sDrive).
- The CARD RDY LED is flashing, which indicates that process of unmounting the memory card is in progress, and later upon successful completion, the CARD RDY LED turns OFF.
- The CITL_UnmountMemoryCard function is available when the status of the SD memory card is "Inserted (mounted)". (The status of the SD memory card can be confirmed by the CITL_GetUnitStatus function.)
- When the status of the SD memory card has been already unmounted, this function ends normally without processing.

Precautions

Before calling the CITL_UnmountMemoryCard function, create a program so that accessing to (reading from/writing to) an SD memory card is stopped and all files are closed. Otherwise, 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 (0000H)	Error For details on the error, refer to the following chapter. ☞ Page 99 ERROR CODE LIST

■Relevant function

☞ Page 42 CITL_GetUnitStatus

☞ Page 43 CITL_MountMemoryCard

CITL_WaitSwitchEvent

Waits for a switch interrupt event of the C intelligent function module to occur.

■Format

short CITL_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

The specification method of the switch interrupt event type (sSwitch) is as follows:

Stored information	Event type
0	Reserved
1	
2	MODE switch interrupt event


■Description

- This function waits for a switch interrupt event specified to the switch interrupt event type (sSwitch).
- Returns immediately if an interrupt event has already been notified after calling the CITL_WaitSwitchEvent function.
- If the same switch interrupt event has been notified several times at a time when the CITL_WaitSwitchEvent 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

For the MODE switch interrupt event, an event issuance status cannot be judged from the appearance. To check the issued status of MODE switch interrupt event, implement the processing such as receiving a switch interrupt event using the CITL_WaitSwitchEvent function and making the USER LED turn ON.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 38 CITL_GetLEDStatus

CITL_WaitTimerEvent

Waits for a timer event to occur.

■Format

short CITL_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 specified to a timer event number (IEventNo) to occur.
- The occurrence cycle of the timer event number (1 to 16) can be set, changed, or cleared by the CITL_EntryTimerEvent function.
- When reset operation is performed, any event that has occurred prior to reset is discarded.
- Using the CITL_WaitTimerEvent 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 the CITL_WaitTimerEvent 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 CITL_EntryTimerEvent function.

Precautions

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

- If the synchronization type is batch synchronization, this function is called later, cancel the waiting status of all the tasks waiting for an event. However, if there is no task in the waiting status at the time of event occurrence, the waiting status is not canceled even if the CITL_WaitTimerEvent function is called later.
- If the synchronization type is individual synchronization, cancel the waiting status of one task from the tasks waiting for an event. If multiple tasks are waiting for the same event, cancel the waiting status in the order of the task priority (or in the order the task starts waiting in the case of the same priority). However, if there is no task in the waiting status at the time of event occurrence, the waiting status is not canceled even if the CITL_WaitTimerEvent function is called later.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 28 CITL_EntryTimerEvent

CITL_WaitYEvent

Waits for the output signal (Y) interrupt event notification.

■Format

short CITL_WaitYEvent (short* psYNo, unsigned long ulTimeout, unsigned short* pusSetEventNo)

■Argument

Argument	Name	Description	IN/OUT
psYNo	Output signal (Y) number	Specify the output signal (Y) number.	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
pusSetEventNo	Occurred output signal (Y) event	Stores the occurred event. (Stores the output signal (Y) number of the notified interrupt event.)	OUT

- The specification method of the output signal (Y) number is as follows:

Stored information position	Stored information
psYNo[0]	Number of interrupt event settings (1 to 16)
psYNo[1]	Output signal (Y) number of the first interrupt event (0x10 to 0x1F)
psYNo[2]	Output signal (Y) number of the second interrupt event (0x10 to 0x1F)
.	.
psYNo[8]	Output signal (Y) number of the 8th interrupt event (0x10 to 0x1F)

- The occurred output signal (Y) event (pusSetEventNo) will be output as follows:

Stored information position	Stored information
pusSetEventNo[0]	Output signal (Y) number of the notified interrupt event

■Description

- This function waits only for the timeout value (ulTimeout) for the interrupt event notification of the output signal (Y) number specified in the output signal (Y) number (psYNo).
- When multiple interrupt events occur, the interrupt events are notified in ascending order of the output signal (Y) number.
- Returns immediately if an interrupt event has already been notified after calling the CITL_WaitYEvent function. 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 interrupt event number (the output signal (Y) number) at a time when the CITL_WaitYEvent function is called, the user program performs processing as a single interrupt event notification.
- Set the output signal (Y) 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.
- The output signal (Y) interrupt event notification wait function (CITL_WaitYEvent function) and the function executing interrupt routine when output signal (Y) interrupts (Defined by CITL_EntryYInt/CITL_EnableYInt/CITL_DisableYInt function) operate independently. These functions operate independently even if interrupt occurs by the same output signals (Y).

Precautions

Design the program so that the CITL_WaitYEvent function is not called simultaneously specified by the same interrupt event number (output signal (Y) number) from the multiple tasks. Otherwise, the notification of the interrupt event to any task is unpredictable.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. ☞ Page 99 ERROR CODE LIST

CITL_X_In_Bit

Reads the input signal (X) in bit (1-point) units.

■Format

short CITL_X_In_Bit (unsigned short usXNo, unsigned short* pusData)

■Argument

Argument	Name	Description	IN/OUT
usXNo	Input signal	Specify the input signal (X). (0 to 31)	IN
pusData	Data storage destination	Specify the storage destination of read data.	OUT

■Description

- This function reads the specified input signal (usXNo) in bit (1-point) units.
- The read data (0: OFF, 1: ON) are stored in the data storage destination (pusData).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. ☞ Page 99 ERROR CODE LIST

■Relevant function

- [☞ Page 58 CITL_X_In_Word](#)
- [☞ Page 59 CITL_X_Out_Bit](#)
- [☞ Page 60 CITL_X_Out_Word](#)
- [☞ Page 61 CITL_Y_In_Bit](#)
- [☞ Page 62 CITL_Y_In_Word](#)

CITL_X_In_Word

Reads the input signal (X) in word (16-point) units.

■Format

short CITL_X_In_Word (unsigned short usXNo, unsigned short usSize, unsigned short* pusDataBuf, unsigned short usBufSize)

■Argument

Argument	Name	Description	IN/OUT
usXNo	Start input signal	Specify a start input signal (X). (Only 0x00 and 0x10 can be specified.)	IN
usSize	Read data size	Specify the read data size in word units. • When start input signal is 0x00: Only 1 and 2 can be specified. • When start input signal is 0x10: Only 1 can be specified.	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 the read data size (usSize) from the start input signal (usXNo), and stores the read data in the data storage destination (pusDataBuf).
- Specify the reserved pusDataBuf area size (the value which is equal to or bigger than the one specified to usSize) to the data storage destination size (usBufSize).
- When 0x10 is specified in the start input signal (usXNo) and two words are specified in the read data size (usSize), CITL_X_In_Word function returns the 'I/O access size error' (-204) and the data cannot be read.
- Read data are stored in the data storage destination (pusDataBuf) in the ascending order of data number from the lower bit as shown below.

Stored information position	Stored information
pusDataBuf[0]	Data of usXNo+FH to usXNo
pusDataBuf[1]	Data of usXNo+1FH to usXNo+10H

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. ☞ Page 99 ERROR CODE LIST

■Relevant function

- ☞ Page 57 CITL_X_In_Bit
- ☞ Page 59 CITL_X_Out_Bit
- ☞ Page 60 CITL_X_Out_Word
- ☞ Page 61 CITL_Y_In_Bit
- ☞ Page 62 CITL_Y_In_Word

CITL_X_Out_Bit

Writes to the input signal (X) in bit (1-point) units.

■Format

short CITL_X_Out_Bit (unsigned short usXNo, unsigned short usData)

■Argument

Argument	Name	Description	IN/OUT
usXNo	Input signal	Specify the input signal (X). (0 to 31)	IN
usData	Write data	Specify the written data. (Specify the value of bit 0.) • 0: OFF • 1: ON	IN

■Description

- This function writes in the input signal (X) corresponding to the specified input signal (usXNo) in bit (1-point) units. (Turn ON/OFF.)
- OFF/ON is output according to the value specified to bit 0 in the data storage destination (usData). (Values of bit 1 to 7 are ignored).
- Only the area available for user can be written. Even though the data is written by the CITL_X_Out_Bit function in the area other than the area available for user, no error will occur, but the value will not be written.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. Page 99 ERROR CODE LIST

■Relevant function

- [Page 57 CITL_X_In_Bit](#)
- [Page 58 CITL_X_In_Word](#)
- [Page 60 CITL_X_Out_Word](#)
- [Page 61 CITL_Y_In_Bit](#)
- [Page 62 CITL_Y_In_Word](#)

CITL_X_Out_Word

Writes to the input signal (X) in word (16-point) units.

■Format

short CITL_X_Out_Word (unsigned short usXNo, unsigned short usSize, unsigned short* pusDataBuf, unsigned short usBufSize)

■Argument

Argument	Name	Description	IN/OUT
usXNo	Start input signal	Specify a start input signal (X). (Only 0x00 and 0x10 can be specified.)	IN
usSize	Write data size	Specify the write data size in word units. • When start input signal is 0x00: Only 1 and 2 can be specified. • When start input signal is 0x10: Only 1 can be specified.	IN
pusDataBuf	Data storage destination	Specify the storage destination of write data.	IN
usBufSize	Data storage destination size	Specify '0'.	IN

■Description

- This function writes to the input signal (X) with a specific write data size (usSize) from the start input signal (usXNo) depending on the data storage destination (pusDataBuf). (Turns ON/OFF.)
- When 0x10 is specified in the start input signal (usXNo) and two words are specified in the write data size (usSize), CITL_X_In_Word function returns the 'I/O access size error' (-204) and the value cannot be written.
- Only the area available for user can be written. Even though the data is written by the CITL_X_Out_Word function in the area other than the area available for user, no error will occur, but the value will not be written.
- Store the written data in the data storage destination (pusDataBuf) in an ascending order of data number from the lower bit.

Stored information position	Stored information
pusDataBuf[0]	Data of usXNo+FH to usXNo
pusDataBuf[1]	Data of usXNo+1FH to usXNo+10H

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. ☞ Page 99 ERROR CODE LIST

■Relevant function

- ☞ Page 57 CITL_X_In_Bit
- ☞ Page 58 CITL_X_In_Word
- ☞ Page 59 CITL_X_Out_Bit
- ☞ Page 61 CITL_Y_In_Bit
- ☞ Page 62 CITL_Y_In_Word

CITL_Y_In_Bit

Reads the output signal (Y) in bit (1-point) units.

■Format

short CITL_Y_In_Bit (unsigned short usYNo, unsigned short* pusData)


■Argument

Argument	Name	Description	IN/OUT
usYNo	Output signal	Specify the output signal (Y). (0 to 31)	IN
pusData	Data storage destination	Specify the storage destination of read data.	OUT






■Description

- This function reads the specified output signal (usYNo) in bit (1-point) units.
- The read data (0: OFF, 1: ON) are stored in the data storage destination (pusData).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

-  Page 57 CITL_X_In_Bit
-  Page 58 CITL_X_In_Word
-  Page 59 CITL_X_Out_Bit
-  Page 60 CITL_X_Out_Word
-  Page 62 CITL_Y_In_Word

CITL_Y_In_Word

Reads the output signal (Y) in word (16-point) units.

■Format

short CITL_Y_In_Word (unsigned short usYNo, unsigned short usSize, unsigned short* pusDataBuf, unsigned short usBufSize)

■Argument

Argument	Name	Description	IN/OUT
usYNo	Start output signal	Specify a start output signal (Y). (Only 0x00 and 0x10 can be specified.)	IN
usSize	Read data size	Specify the read data size in word units. • When start output signal is 0x00: Only 1 and 2 can be specified. • When start output signal is 0x10: Only 1 can be specified.	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 the read data size (usSize) from the start output signal (usYNo), and stores the read data in the data storage destination (pusDataBuf).
- Specify the reserved pusDataBuf area size (the value which is equal to or bigger than the one specified to usSize) to the data storage destination size (usBufSize).
- When 0x10 is specified in the start output signal (usYNo) and two words are specified in the read data size (usSize), CITL_Y_In_Word function returns the 'I/O access size error' (-204) and the data cannot be read.
- Read data are stored in the data storage destination (pusDataBuf) in the ascending order of data number from the lower bit as shown below.

Stored information position	Stored information
pusDataBuf[0]	Data of usYNo+FH to usYNo
pusDataBuf[1]	Data of usYNo+1FH to usYNo+10H

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. ☞ Page 99 ERROR CODE LIST

■Relevant function

- ☞ Page 57 CITL_X_In_Bit
- ☞ Page 58 CITL_X_In_Word
- ☞ Page 59 CITL_X_Out_Bit
- ☞ Page 62 CITL_Y_In_Word
- ☞ Page 61 CITL_Y_In_Bit

C intelligent function module dedicated functions for ISR

CITL_DisableYInt_ISR

Disables the routine registered with the CITL_EntryYInt function.

■Format

short CITL_DisableYInt_ISR(short sYNo)

■Argument

Argument	Name	Description	IN/OUT
sYNo	Output signal (Y) number	Specify the output signal (Y) number. (If -1 is specified, disable all the registered routines.)	IN

■Description

- This function disables the routine registered by using the CITL_EntryYInt function. (The registered routine is not executed when output signal (Y) interrupt occurs.)
- Specify the output signal (Y) number (sYNo) specified in the CITL_EntryYInt function in the output signal (Y) number (sYNo).

■⚠️ WARNING

CITL_DisableYInt_ISR function does not check the specified argument.

Create a program with the following conditions in mind.

- Do not specify an unregistered output signal (Y).

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

☞ Page 31 CITL_EntryYInt

☞ Page 64 CITL_EnableYInt_ISR

CITL_EnableYInt_ISR

Enables the routine registered with the CITL_EntryYInt function.

■Format

short CITL_EnableYInt_ISR (short sYNo)

■Argument

Argument	Name	Description	IN/OUT
sYNo	Output signal (Y) number	Specify the output signal (Y) number. (If -1 is specified, enable all the registered routines.)	IN

■Description

- This function enables the routine registered by using the CITL_EntryYInt function. (The registered routine is executed when the output signal (Y) interrupt occurs.)
- Specify the output signal (Y) number (sYNo) specified in the CITL_EntryYInt function in the output signal (Y) number (sYNo).

■ WARNING

CITL_EnableYInt_ISR function does not check the specified argument.


Create a program with the following conditions in mind.

- Do not specify an unregistered output signal (Y).

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

 Page 31 CITL_EntryYInt

 Page 63 CITL_DisableYInt_ISR

CITL_FromBuf_ISR

Reads data from the buffer memory of the C intelligent function module.

■Format

short CITL_FromBuf_ISR (unsigned long ulOffset, unsigned long ulSize, unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
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 data size (ulSize) from the buffer memory of the C intelligent function module, and stores the read data in the data storage destination (pusDataBuf). Data is read by specifying offset (ulOffset) address from the start of the buffer memory of the C intelligent function module.

Restriction

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


■ WARNING

CITL_FromBuf_ISR function does not check the specified argument.

Create a program with the following conditions in mind.

- The offset (ulOffset) is a multiple of 2.
- The data area for the size (words) of the read data is reserved.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 70 CITL_ToBuf_ISR

CITL_GetCounterMicros_ISR

Obtains a 1 μ s counter value of the C intelligent function module.

■Format

short CITL_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 the C intelligent function module and stores the value in the 1 μ s counter value storage destination (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 CITL_GetCounterMicros_ISR function in a routine other than the one registered in the interrupt.

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

 Page 67 CITL_GetCounterMillis_ISR

CITL_GetCounterMillis_ISR

Obtains a 1 ms counter value of the C intelligent function module.

■Format

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

Restriction

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

■Return value

Return value	Description
0 (0000H)	Normal

■Relevant function

 Page 66 CITL_GetCounterMicros_ISR

CITL_RegistEventLog_ISR

Registers event logs in the event history of control CPU module.

■Format

short CITL_RegistEventLog_ISR (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 the control CPU module.

The contents to be registered on the event history screen of the engineering tool are as follows:

Item	Description
Occurrence date	Event registered date and time
Event type	Operation (Fixed)
Status	Information (Fixed)
Event code	25000 (Fixed)
Overview	Registration from the user program (Fixed)
Source	RD55UP06-V (Fixed)
Start I/O number	Input/output number of the C intelligent function module that executed the CITL_RegistEventLog_ISR function.
Detailed event code information	Detailed code specified to IEventCode (hexadecimal)
Detailed event log information	Detailed information specified to pcEventMsg
Cause	The event history was registered from the C intelligent function module dedicated function. (Fixed)

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

Restriction

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

■ WARNING

CITL_RegistEventLog_ISR function does not check the specified argument.

Create a program with the following conditions in mind.

- The detailed information within the range is specified.

■Return value

Return value	Description
0 (0000H)	Normal

CITL_SetLEDStatus_ISR

Sets the LED status of the C intelligent function module.

■Format

short CITL_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

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

Stored information	LED status
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 the C intelligent function module to the status specified by the LED status information (usLedInfo).

Restriction

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

■Return value

Return value	Description
0 (0000H)	Normal

CITL_ToBuf_ISR

Writes data to the buffer memory of the C intelligent function module.

■Format

short CITL_ToBuf_ISR (unsigned long ulOffset, unsigned long ulSize, unsigned short* pusDataBuf)

■Argument

Argument	Name	Description	IN/OUT
ulOffset	Offset	Specify the offset in word units.	IN
ulSize	Data size	Specify the write data size in word units.	IN
pusDataBuf	Data storage destination	Specify the storage destination of write data.	IN

■Description

This function writes data in the data storage destination (pusDataBuf) equivalent to the data size (ulSize) to the CPU buffer memory of the C intelligent function module. Data is written by specifying offset (ulOffset) address from the start of the buffer memory of the C intelligent function module.

Restriction

- Do not execute CITL_ToBuf_ISR function in a routine other than the one registered in the interrupt.
- When data is written to the same buffer memory 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 buffer memory.


■ WARNING

CITL_ToBuf_ISR function does not check the specified argument.

Create a program with the following conditions in mind.

- The offset (ulOffset) is a multiple of 2.
- Do not specify outside the buffer memory or system area.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 65 CITL_FromBuf_ISR

CITL_X_In_Word_ISR

Reads the input signal (X) in word (16-point) units.

■Format

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

■Argument

Argument	Name	Description	IN/OUT
usXNo	Start input signal	Specify a start input signal (X). (Only 0x00 and 0x10 can be specified.)	IN
usSize	Read data size	Specify the read data size in word units. • When start input signal is 0x00: Only 1 and 2 can be specified. • When start input signal is 0x10: Only 1 can be specified.	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 the read data size (usSize) from the start input signal (usXNo), and stores the read data in the data storage destination (pusDataBuf).
- Read data are stored in the data storage destination (pusDataBuf) in the ascending order of data number from the lower bit as shown below.

Stored information position	Stored information
pusDataBuf[0]	Data of usXNo+FH to usXNo
pusDataBuf[1]	Data of usXNo+1FH to usXNo+10H

Restriction

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

■! WARNING

CITL_X_In_Word_ISR function does not check the specified argument.

Create a program with the following conditions in mind.

- The data area for the size (words) of the read data is reserved.
- The input signal (X) within the range (0H to 1FH) is specified.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. ☞ Page 99 ERROR CODE LIST

■Relevant function

☞ Page 72 CITL_X_Out_Word_ISR

☞ Page 73 CITL_Y_In_Word_ISR

CITL_X_Out_Word_ISR

Writes to the input signal (X) in word (16-point) units.

■Format

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

■Argument

Argument	Name	Description	IN/OUT
usXNo	Start input signal	Specify a start input signal (X). (Only 0x00 and 0x10 can be specified.)	IN
usSize	Write data size	Specify the write data size in word units. • When start input signal is 0x00: Only 1 and 2 can be specified. • When start input signal is 0x10: Only 1 can be specified.	IN
pusDataBuf	Data storage destination	Specify the storage destination of write data.	IN

■Description

- This function writes to the input signal (X) with a specific write data size (usSize) from the start input signal (usXNo) depending on the data storage destination (pusDataBuf). (Turns ON/OFF.)
- Only the area available for user can be written. Even though the data is written by the CITL_X_Out_Word function in the area other than the area available for user, no error will occur, but the value will not be written.
- Store the written data in the data storage destination (pusDataBuf) in an ascending order of data number from the lower bit.

Stored information position	Stored information
pusDataBuf[0]	Data of usXNo+FH to usXNo
pusDataBuf[1]	Data of usXNo+1FH to usXNo+10H

Restriction

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


■ WARNING

CITL_X_Out_Word_ISR function does not check the specified argument.

Create a program with the following conditions in mind.

- The input signal (X) within the range (0H to 1FH) is specified.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 71 CITL_X_In_Word_ISR

 Page 73 CITL_Y_In_Word_ISR

CITL_Y_In_Word_ISR

Reads the output signal (Y) in word (16-point) units.

■Format

short CITL_Y_In_Word_ISR (unsigned short usYNo, unsigned short usSize, unsigned short* pusDataBuf, unsigned short usBufSize)

■Argument

Argument	Name	Description	IN/OUT
usYNo	Start output signal	Specify a start output signal (Y). (Only 0x00 and 0x10 can be specified.)	IN
usSize	Read data size	Specify the read data size in word units. • When start output signal is 0x00: Only 1 and 2 can be specified. • When start output signal is 0x10: Only 1 can be specified.	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 the read data size (usSize) from the start output signal (usYNo), and stores the read data in the data storage destination (pusDataBuf).
- Read data are stored in the data storage destination (pusDataBuf) in the ascending order of data number from the lower bit as shown below.

Stored information position	Stored information
pusDataBuf[0]	Data of usYNo+FH to usYNo
pusDataBuf[1]	Data of usYNo+1FH to usYNo+10H

Restriction

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


■ WARNING

CITL_Y_In_Word_ISR function does not check the specified argument.

Create a program with the following conditions in mind.


- The data area for the size (words) of the read data is reserved.
- The output signal (Y) within the range (0H to 1FH) is specified.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 71 CITL_X_In_Word_ISR

 Page 72 CITL_X_Out_Word_ISR

3.2 MELSEC iQ-R Series Data Link Functions

This section shows the details of the MELSEC iQ-R series data link function.

mdrClose

Closes a communication line (channel).

■Format

short mdrClose(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 channel opened by the mdrOpen function.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 82 mdrOpen




mdrControl

Performs remote operations (RUN/STOP/PAUSE) for the CPU module.

■Format

short mdrControl(long IPath, short sRoute, long INetNo, long IloNo, long IStNo, short sCPU, short sCode)

■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
sRoute	Access route	Specify the access route to target module. <ul style="list-style-type: none">• 0: CC-Link IE Controller Network• 1: CC-Link IE Field Network• 2: MELSECNET/H• 3: CC-Link• 4: Bus interface	IN
INetNo	Network number	Specify the network number of target module.  Page 15 Argument specification	IN
IloNo	Start I/O number	Specify the start I/O number divided by 16 of the target module.  Page 15 Argument specification	IN
IStNo	Station number	Specify the station number of target module.  Page 15 Argument specification	IN
sCPU	CPU number	Specify the CPU number of the target CPU module. <ul style="list-style-type: none">• 0: Specification of control CPU• 1 to 4: Multiple CPU specification	IN
sCode	Instruction code	Specify the contents of the remote operation in numerical value.	IN

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

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


■Description

This function changes the status of the CPU module with the station number specified to IStNo to the status specified to the instruction code (sCode).


Restriction


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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 74 mdrClose

 Page 82 mdrOpen

mdrDevRst

Resets (turns OFF) bit device(s).

■Format

short mdrDevRst(long IPath, short sRoute, long INetNo, long IloNo, long IStNo, short sCPU, long IDevType, long IDevNo)

■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
sRoute	Access route	Specify the access route to target module. <ul style="list-style-type: none">• 0: CC-Link IE Controller Network• 1: CC-Link IE Field Network• 2: MELSECNET/H• 3: CC-Link• 4: Bus interface	IN
INetNo	Network number	Specify the network number of target module. ☞ Page 15 Argument specification	IN
IloNo	Start I/O number	Specify the start I/O number divided by 16 of the target module. ☞ Page 15 Argument specification	IN
IStNo	Station number	Specify the station number of target module. ☞ Page 15 Argument specification	IN
sCPU	CPU number	Specify the CPU number of the target CPU module. <ul style="list-style-type: none">• 0: Specification of control CPU• 1 to 4: Multiple CPU specification	IN
IDevType	Device type	Specify the device type of bit device. ☞ Page 15 Argument specification	IN
IDevNo	Device number	Specify the device number of bit device.	IN

■Description

- This function resets (turns OFF) the bit device of the module specified to the network number (INetNo),the start I/O number (IloNo),the station number (IStNo), the CPU number (sCPU),the device type (IDevType),and the device number (IDevNo).
- The mdrDevRst function is dedicated function for bit devices such as link relay (B) and internal relay (M).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. ☞ Page 99 ERROR CODE LIST

■Relevant function

☞ Page 74 mdrClose

☞ Page 77 mdrDevSet

☞ Page 82 mdrOpen





mdrDevSet

Sets (turns ON) bit device(s).

■Format

short mdrDevSet (long IPath, short sRoute, long INetNo, long IloNo, long IStNo, short sCPU, long IDevType, long IDevNo)


■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
sRoute	Access route	Specify the access route to target module. <ul style="list-style-type: none">• 0: CC-Link IE Controller Network• 1: CC-Link IE Field Network• 2: MELSECNET/H• 3: CC-Link• 4: Bus interface	IN
INetNo	Network number	Specify the network number of target module.  Page 15 Argument specification	IN
IloNo	Start I/O number	Specify the start I/O number divided by 16 of the target module.  Page 15 Argument specification	IN
IStNo	Station number	Specify the station number of target module.  Page 15 Argument specification	IN
sCPU	CPU number	Specify the CPU number of the target CPU module. <ul style="list-style-type: none">• 0: Specification of control CPU• 1 to 4: Multiple CPU specification	IN
IDevType	Device type	Specify the device type of bit device.  Page 15 Argument specification	IN
IDevNo	Device number	Specify the device number of bit device.	IN

■Description


- This function sets (turns ON) the bit device of the module specified to the network number (INetNo),the start I/O number (IloNo),the station number (IStNo), the CPU number (sCPU),the device type (IDevType),and the device number (IDevNo).
- The mdrDevSet function is dedicated function for bit devices such as link relay (B) and internal relay (M).


■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 74 mdrClose

 Page 76 mdrDevRst

 Page 82 mdrOpen




mdrGetLabelInfo

Obtains device information corresponding to label names.

■Format

short mdrGetLabelInfo (long IPath, short sRoute, long INetNo, long lIoNo, long lStNo, short sCPU, long lLbCnt, void* pLbLst, long* pDevLst, unsigned long* pullLbCode)

■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
sRoute	Access route	Specify the access route to target module. <ul style="list-style-type: none"> • 0: CC-Link IE Controller Network • 1: CC-Link IE Field Network • 2: MELSECNET/H • 3: CC-Link • 4: Bus interface 	IN
INetNo	Network number	Specify the network number of target module.  Page 15 Argument specification	IN
lIoNo	Start I/O number	Specify the start I/O number divided by 16 of the target module.  Page 15 Argument specification	IN
lStNo	Station number	Specify the station number of target module.  Page 15 Argument specification	IN
sCPU	CPU number	Specify the CPU number of the target CPU module. <ul style="list-style-type: none"> • 0: Specification of control CPU • 1 to 4: Multiple CPU specification 	IN
lLbCnt	Number of labels	Specify the number of labels. (Up to 10240)	IN
pLbLst	Label name array	Specify the storage address of label name for each label.	IN
pDevLst	Device name array	Specify the device that stores the obtained device information. (The information relating to the device assigned to the label specified using the label name array (pLbLst) is stored in a randomly selected device format.)	OUT
pullLbCode	Label code	A value to identify whether the label of the CPU module is changed or not is stored. (Whether the label setting is changed or not can be checked by whether this value is changed or not.)	OUT

The device information assigned to the label specified by the label name array (pLbLst) is stored in the device specified by the device name array (pDevLst) in the following randomly selected device format.

Stored information position	Stored information	
pDevLst[0]	Number of blocks	
pDevLst[1]	Device type	Block 1
pDevLst[2]	Start device number	
pDevLst[3]	Number of read points	
pDevLst[4]	Device type	Block 2
pDevLst[5]	Start device number	
pDevLst[6]	Number of read points	
.	.	.
.	.	.
pDevLst[3n+1]	Device type	Block n
pDevLst[3n+2]	Start device number	
pDevLst[3n+3]	Number of read points	

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

■Description

- The label is read for the module specified in the network number (INetNo), the start I/O number (IloNo), the station number (IStNo) and the CPU number (sCPU).
- Reserve the area for the label name array (pIDevLst) in the call source.
- Reserve the area equivalent to (ILbCnt×3+1) for the size of device name array (pIDevLst) area.
- If any of the labels of which the label information cannot be obtained exists in the label name specified to the label name array (pLbLst), this function returns any of the following errors. For the device type, start device number and number of read points of the label, '0' is stored.

Error number	Description
-82(FFB2H)	When a non-existent label was specified
	When the device assigned to the label does not support random read/write
-84(FFB4H)	The device specification method is incorrect

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

○: Possible, ×: Impossible

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

Precautions

- In CW Workbench, Unicode character strings cannot be entered and source codes including Unicode character strings cannot be compiled. Create a text file with Unicode (UTF-16) character strings entered in the applications (such as Notepad) on Windows.
- When a device is specified such as the bit specification of word device or the digit specification of label, the label information cannot be obtained.
- When a label to which a device is not assigned using engineering tool is specified, DevGV is stored to the device type of the device name array (pIDevLst).
- The DevGV can be specified only by using the mdrRandRLabel/mdrRandWLabel functions.

■Example

The following table shows the examples of values set to the label name array (pLbLst) and data to be read to the device name array (pLDevLst) (for five labels to be read: Label1 to 5).

Preparation for the label name (Unicode character string)

1. Describe the label name to be used in the text file, and save it by specifying Unicode (UTF-16).
2. Read the label name in binary from the saved text file with the user program and store the address of the label name passed to pLbLst in the memory.

- Values set to the label name array (pLbLst)

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

- Data to be read to the device name array (pLDevLst)

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

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. ☞ Page 99 ERROR CODE LIST

■Relevant function

[☞ Page 74 mdrClose](#)

[☞ Page 82 mdrOpen](#)

[☞ Page 86 mdrRandRLabel](#)

[☞ Page 91 mdrRandWLabel](#)

mdrlnit

Initializes communication route information.

■Format

short mdrlnit (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 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 74 mdrClose

 Page 82 mdrOpen


mdrOpen

Opens a communication line (channel).

■Format

short mdrOpen (short sChan, short sMode, long* plPath, long lTimeout)


■Argument

Argument	Name	Description	IN/OUT
sChan	Channel	Specify a communication line (channel).  Page 15 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
lTimeout	Timeout value	Specify the timeout value of MELSEC iQ-R series data link function for MELSEC iQ-R series bus interface (Channel No.12) • Setting range: 1 to 360 sec (1 sec unit)	IN

■Description

- The path of the channel opened by the mdrOpen function is used when MELSEC iQ-R series data link functions are executed.
- To end the program, close the path of the opened channel using the mdrClose function.

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

 Page 74 mdrClose




mdrRandR

Reads device(s) randomly.

■Format

short mdrRandR(long lPath, short sRoute, long lNetNo, long lIoNo, long lStNo, short sCPU, long* plDev, short* psBuf, long lBufSize)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN
sRoute	Access route	Specify the access route to target module. <ul style="list-style-type: none"> • 0: CC-Link IE Controller Network • 1: CC-Link IE Field Network • 2: MELSECNET/H • 3: CC-Link • 4: Bus interface 	IN
lNetNo	Network number	Specify the network number of target module.  Page 15 Argument specification	IN
lIoNo	Start I/O number	Specify the start I/O number divided by 16 of the target module.  Page 15 Argument specification	IN
lStNo	Station number	Specify the station number of target module.  Page 15 Argument specification	IN
sCPU	CPU number	Specify the target CPU number. <ul style="list-style-type: none"> • 0: Specification of control CPU • 1 to 4: Multiple CPU specification 	IN
plDev	Randomly selected device	Specify the number of blocks, device type, start device number, and points of devices to be read.	IN
psBuf	Read data storage destination	Specify the storage destination (address) of read data.	OUT
lBufSize	Read data storage destination size	Specify the size of area allocated in the read data storage destination in byte units.	IN

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

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

■Description

- This function reads devices specified to the randomly selected device (plDev) from the module specified to the network number (lNetNo), the start I/O number (lIoNo), the station number (lStNo), and the CPU number (sCPU).
- The read data is stored in the read data storage destination (psBuf) in word units in order of the specification to the randomly selected device (plDev). A bit device is stored per 16 points, a word device is stored per 1 point, and a double-word device is stored in word units.
- Specify so that the total number of read points specified by each block is 10240 points or less. If specified more than the maximum number, size error (-5) will occur.
- Communication time can vary significantly depending on the contents specified to the randomly selected device (plDev). To reduce communication time, use the mdrReceive 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 the randomly selected device (plDev), data to be read to the read data storage destination (psBuf), and the number of read data bytes.

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

The values specified for the randomly selected device (plDev)

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

Data to be read to the read data storage destination (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	T10=10
psBuf[7]	LCN100	0x1	Lower bit of LCN100 = 0x0001
psBuf[8]			Upper bit of LCN100 = 0x0000
psBuf[9]	LCN101	0x10000	Lower bit of LCN101 = 0x0000
psBuf[10]			Upper bit of LCN101 = 0x0001

Number of bytes of read data set to the read data storage destination size (lBufSize)

(psBuf[0] to psBuf[10] = 11) × 2 = 22

Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. Page 99 ERROR CODE LIST

Relevant function

- [Page 74 mdrClose](#)
- [Page 82 mdrOpen](#)
- [Page 89 mdrRandW](#)




mdrRandRLabel

Reads the device(s) randomly corresponding to the labels.

■Format

short mdrRandRLabel (long lPath, short sRoute, long lNetNo, long lIoNo, long lStNo, short sCPU, long* pIDev, short* psBuf, long lBufSize, unsigned long long ullbCode)

■Argument

Argument	Name	Description	IN/OUT
lPath	Path of channel	Specify the path of the opened channel.	IN
sRoute	Access route	Specify the access route to target module. <ul style="list-style-type: none"> • 0: CC-Link IE Controller Network • 1: CC-Link IE Field Network • 2: MELSECNET/H • 3: CC-Link • 4: Bus interface 	IN
lNetNo	Network number	Specify the network number of target module.  Page 15 Argument specification	IN
lIoNo	Start I/O number	Specify the start I/O number divided by 16 of the target module.  Page 15 Argument specification	IN
lStNo	Station number	Specify the station number of target module.  Page 15 Argument specification	IN
sCPU	CPU number	Specify the target CPU number. <ul style="list-style-type: none"> • 0: Specification of control CPU • 1 to 4: Multiple CPU specification 	IN
pIDev	Randomly selected device	Specify the number of blocks, device type, start device number, and points of devices to be read. (Specify the value obtained by using the mdrGetLabelInfo function.)	IN
psBuf	Read data storage destination	Specify the storage destination (address) of read data.	OUT
lBufSize	Read data storage destination size	Specify the area size reserved in the read data storage destination in byte units.	IN
ullbCode	Label code	Specify the label code obtained by using the mdrGetLabelInfo function.	IN

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

Stored information position	Stored information	
pIDev[0]	Number of blocks	
pIDev[1]	Device type	Block 1
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	
.	.	.
pIDevLst[3n+1]	Device type	Block n
pIDevLst[3n+2]	Start device number	
pIDevLst[3n+3]	Number of read points	

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

■Description

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

■Example

- The following table shows the examples of values set to the randomly selected device (plDev), data to be read to the read data storage destination (psBuf), and the number of read data bytes.

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

The values specified for the randomly selected device (plDev)

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

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

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

Number of bytes of read data set to the read data storage destination size (lBufSize)

(psBuf[0] to psBuf[10]) × 2 = 22

Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. *1 Page 99 ERROR CODE LIST

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

Relevant function

- [Page 74 mdrClose](#)
- [Page 78 mdrGetLabelInfo](#)
- [Page 82 mdrOpen](#)
- [Page 91 mdrRandWLabel](#)




mdrRandW

Writes device(s) randomly.

■Format

short mdrRandW(long IPath, short sRoute, long INetNo, long IloNo, long IStNo, short sCPU, long* plDev, short* psBuf, long lBufSize)

■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
sRoute	Access route	Specify the access route to target module. <ul style="list-style-type: none"> • 0: CC-Link IE Controller Network • 1: CC-Link IE Field Network • 2: MELSECNET/H • 3: CC-Link • 4: Bus interface 	IN
INetNo	Network number	Specify the network number of target module.  Page 15 Argument specification	IN
IloNo	Start I/O number	Specify the start I/O number divided by 16 of the target module.  Page 15 Argument specification	IN
IStNo	Station number	Specify the station number of target module.  Page 15 Argument specification	IN
sCPU	CPU number	Specify the target CPU number. <ul style="list-style-type: none"> • 0: Specification of control CPU • 1 to 4: Multiple CPU 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.	IN
lBufSize	Write data storage destination size	Specify the size of area allocated in the write data storage destination in byte units.	IN

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

Stored information position	Stored information	
plDev[0]	Number of blocks	
plDev[1]	Device type	Block 1
plDev[2]	Start device number	
plDev[3]	Number of write points	
plDev[4]	Device type	Block 2
plDev[5]	Start device number	
plDev[6]	Number of write points	
.	.	.
.	.	.
plDevLst[3n+1]	Device type	Block n
plDevLst[3n+2]	Start device number	
plDevLst[3n+3]	Number of write points	

■Description

- This function writes data to the device, which is specified to the randomly selected device (plDev), of the module specified to the network number (INetNo), the start I/O number (IloNo), the station number (IStNo) and the CPU number (sCPU).
- The data to be written is stored to the write data storage destination (psBuf) in word units. A bit device is stored per 16 points, a word device is stored per 1 point, and a double-word device is stored in word units.
- Specify so that the total number of write points specified by each block is 10240 points or less. If specified more than the maximum number, size error (-5) will occur.
- Note that the extension comment information will be deleted when the data is written to the block to which an extension comment is assigned (extension file register).
- Also, note that sub 2 or sub 3 program will be deleted when data is written to a block (extension file register) overlapping with the program setting area for sub 2 or sub 3.

■Example

- The following table shows the examples of values set to the randomly selected device (plDev) and the data specified in the write data storage destination (psBuf), and the number of write data bytes.

Device to be written randomly	Current value
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.

The values specified for the randomly selected device (plDev)

Setting target	Specified value	Description	
plDev[0]	3	Number of blocks = 3	—
plDev[1]	DevM	Device type = M	Block 1: M100 to M115
plDev[2]	100	Start device number = 100	
plDev[3]	16	Number of write points = 16	
plDev[4]	DevD	Device type = D	Block 2: D10 to D13
plDev[5]	10	Start device number = 10	
plDev[6]	4	Number of write points = 4	
plDev[7]	DevLCN	Device type = LCN	Block 5: LCN100 to LCN101
plDev[8]	100	Start device number = 100	
plDev[9]	2	Number of write points = 2	

Data specified for the write data storage destination (psBuf)

Write position	Value	Description
psBuf[0]	0	Turns all bit devices from M100 to M115 OFF.
psBuf[1]	10	D10=10
psBuf[2]	200	D11=200
psBuf[3]	300	D12=300
psBuf[4]	400	D13=400
psBuf[5]	0x0001	Lower bit of LCN100 = 0x0001
psBuf[6]	0x0000	Upper bit of LCN100 = 0x0000
psBuf[7]	0x0000	Lower bit of LCN101 = 0x0000
psBuf[8]	0x0001	Upper bit of LCN101 = 0x0001

Number of bytes of write data set to the write data storage destination size (lBufSize)

$$(\text{psBuf}[0] \text{ to } \text{psBuf}[8] = 9) \times 2 = 18$$

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. Page 99 ERROR CODE LIST

■Relevant function

[Page 74 mdrClose](#)

[Page 82 mdrOpen](#)

[Page 83 mdrRandR](#)




mdrRandWLabel

Writes the device(s) randomly corresponding to the labels.

■Format

short mdrRandWLabel (long IPath, short sRoute, long INetNo, long IloNo, long IStNo, short sCPU, long* plDev, short* psBuf, long lBufSize, unsigned long long ullbCode)

■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
sRoute	Access route	Specify the access route to target module. <ul style="list-style-type: none"> • 0: CC-Link IE Controller Network • 1: CC-Link IE Field Network • 2: MELSECNET/H • 3: CC-Link • 4: Bus interface 	IN
INetNo	Network number	Specify the network number of target module.  Page 15 Argument specification	IN
IloNo	Start I/O number	Specify the start I/O number divided by 16 of the target module.  Page 15 Argument specification	IN
IStNo	Station number	Specify the station number of target module.  Page 15 Argument specification	IN
sCPU	CPU number	Specify the target CPU number. <ul style="list-style-type: none"> • 0: Specification of control CPU • 1 to 4: Multiple CPU 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.	IN
lBufSize	Write data storage destination size	Unused (Even if a value is set, the operation is not affected.)	IN
ullbCode	Label code	Specify the label code obtained by using the mdrGetLabelInfo function.	IN

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

Stored information position	Stored information	
plDev[0]	Number of blocks	
plDev[1]	Device type	Block 1
plDev[2]	Start device number	
plDev[3]	Number of write points	
plDev[4]	Device type	Block 2
plDev[5]	Start device number	
plDev[6]	Number of write points	
.	.	.
.	.	.
plDevLst[3n+1]	Device type	Block n
plDevLst[3n+2]	Start device number	
plDevLst[3n+3]	Number of write points	

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

■Description

- This function writes data to the device, which is specified to the randomly selected device (plDev), of the module specified to the network number (INetNo), the start I/O number (IloNo), the station number (IStNo) and the CPU number (sCPU).
- The data to be written is stored to the write data storage destination (psBuf) in word units. A bit device and a word device are stored per 1 point, and a double-word device is stored in word units.
- Specify so that the total number of write points specified by each block is 10240 points or less. If specified more than the maximum number, size error (-5) will occur.
- Note that the extension comment information will be deleted when the data is written to the block to which an extension comment is assigned (extension file register).
- Also, note that sub 2 or sub 3 program will be deleted when data is written to a block (extension file register) overlapping with the program setting area for sub 2 or sub 3.
- When '0' is specified to the label code (uIlLbCode), the device is written without checking the label code.

■Example

- The following table shows the examples of values set to the randomly selected device (plDev) and the data specified in the write data storage destination (psBuf), and the number of write data bytes.

Device to be written randomly	Current value
M100	Turns the bit OFF.
D10 to D13	Stores 10 in D10, 200 in D11, 300 in D12, and 400 in D13.
LCN100 to LCN101	Stores 0x1 to LCN100, and 0x10000 to LCN101.

The values specified for the randomly selected device (plDev)

Setting target	Specified value	Description	
plDev[0]	3	Number of blocks = 3	—
plDev[1]	DevM	Device type = M	Block 1: M100
plDev[2]	100	Start device number = 100	
plDev[3]	1	Number of write points = 1	
plDev[4]	DevD	Device type = D	Block 2: D10 to D13
plDev[5]	10	Start device number = 10	
plDev[6]	4	Number of write points = 4	
plDev[7]	DevLCN	Device type = LCN	Block 5: LCN100 to LCN101
plDev[8]	100	Start device number = 100	
plDev[9]	2	Number of write points = 2	

Data specified for the write data storage destination (psBuf)

Write position	Value	Description
psBuf[0]	0	Turns the bit device for M100 OFF.
psBuf[1]	10	D10=10
psBuf[2]	200	D11=200
psBuf[3]	300	D12=300
psBuf[4]	400	D13=400
psBuf[5]	0x0001	Lower bit of LCN100 = 0x0001
psBuf[6]	0x0000	Upper bit of LCN100 = 0x0000
psBuf[7]	0x0000	Lower bit of LCN101 = 0x0000
psBuf[8]	0x0001	Upper bit of LCN101 = 0x0001

Number of bytes of write data set to the write data storage destination size (lBufSize)

(psBuf[0] to psBuf[8] = 9) × 2 = 18

Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter. *1 Page 99 ERROR CODE LIST

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

Relevant function

- [Page 74 mdrClose](#)
- [Page 78 mdrGetLabelInfo](#)
- [Page 82 mdrOpen](#)
- [Page 86 mdrRandRLabel](#)




mdrReceive

Reads devices in batch.

■Format

short mdrReceive(long IPath, short sRoute, long INetNo, long IloNo, long IStNo, short sCPU, long IDevType, long IDevNo, long* plSize, short* psData)


■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
sRoute	Access route	Specify the access route to target module. <ul style="list-style-type: none">• 0: CC-Link IE Controller Network• 1: CC-Link IE Field Network• 2: MELSECNET/H• 3: CC-Link• 4: Bus interface	IN
INetNo	Network number	Specify the network number of target module.  Page 15 Argument specification	IN
IloNo	Start I/O number	Specify the start I/O number divided by 16 of the target module.  Page 15 Argument specification	IN
IStNo	Station number	Specify the station number of target module.  Page 15 Argument specification	IN
sCPU	CPU number	Specify the target CPU number. <ul style="list-style-type: none">• 0: Specification of control CPU• 1 to 4: Multiple CPU 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).	IN
plSize	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 word device or bit device is specified. If the value other than that is specified, the size error (-5) will occur.)	IN/OUT
psData	Read data storage destination	Specify the storage destination (address) of read data.	OUT




■Description

- This function reads data from the module specified to the network number (INetNo), the start I/O number (IloNo), the station number (IStNo) and the CPU number (sCPU).
- This function reads data with the size specified to the read data size (plSize) from the devices specified to the device type (IDevType) and the start device number (IDevNo).
- When the specified read data size exceeds the device range, a readable size is returned to the read data size (plSize).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

-  Page 74 mdrClose
-  Page 82 mdrOpen
-  Page 95 mdrSend




mdrSend

Writes devices in batch.

■Format

short mdrSend(long IPath, short sRoute, long INetNo, long IloNo, long IStNo, short sCPU, long IDevType, long IDevNo, long* plSize, short* psData)


■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
sRoute	Access route	Specify the access route to target module. <ul style="list-style-type: none"> • 0: CC-Link IE Controller Network • 1: CC-Link IE Field Network • 2: MELSECNET/H • 3: CC-Link • 4: Bus interface 	IN
INetNo	Network number	Specify the network number of target module.  Page 15 Argument specification	IN
IloNo	Start I/O number	Specify the start I/O number divided by 16 of the target module.  Page 15 Argument specification	IN
IStNo	Station number	Specify the station number of target module.  Page 15 Argument specification	IN
sCPU	CPU number	Specify the target CPU number. <ul style="list-style-type: none"> • 0: Specification of control CPU • 1 to 4: Multiple CPU 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. (Specify the start device number of bit device in multiples of 8.)	IN
plSize	Write data size	Specify the write data size in byte units. (Specify the value in multiples of 4 when double-word device (LZ, LTN, LCN, LSTN) is specified, or specify the value in multiples of 2 when a word device or bit device is specified. If the value other than that is specified, the size error (-5) will occur.)	IN/OUT
psData	Write data storage destination	Specify the storage destination (address) of write data. (Allocate successive area to the write data storage destination.)	IN




■Description

- This function writes data to the module, which is specified by the network number (INetNo), start I/O number (IloNo), station number (IStNo), and CPU number (sCPU).
- It writes data equivalent to the size specified to the write data size (plSize) by handling the device specified to the device type (IDevType) and the start device number (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, a writable size is returned to the write data size (plSize).
- Note that the extension comment information will be deleted when the data is written to the block to which an extension comment is assigned (extension file register).

■Return value

Return value	Description
0 (0000H)	Normal
Other than 0 (0000H)	Error For details on the error, refer to the following chapter.  Page 99 ERROR CODE LIST

■Relevant function

-  Page 74 mdrClose
-  Page 82 mdrOpen
-  Page 94 mdrReceive




mdrTypeRead

Reads the model code of a CPU module.

■Format

short mdrTypeRead (long IPath, short sRoute, long INetNo, long IloNo, long IStNo, short sCPU, short* psCode)

■Argument

Argument	Name	Description	IN/OUT
IPath	Path of channel	Specify the path of the opened channel.	IN
sRoute	Access route	Specify the access route to target module. <ul style="list-style-type: none">• 0: CC-Link IE Controller Network• 1: CC-Link IE Field Network• 2: MELSECNET/H• 3: CC-Link• 4: Bus interface	IN
INetNo	Network number	Specify the network number of target module.  Page 15 Argument specification	IN
IloNo	Start I/O number	Specify the start I/O number divided by 16 of the target module.  Page 15 Argument specification	IN
IStNo	Station number	Specify the station number of target module.  Page 15 Argument specification	IN
sCPU	CPU number	Specify the target CPU number. <ul style="list-style-type: none">• 0: Specification of control CPU• 1 to 4: Multiple CPU 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 code of the CPU module with the specified station number to IStNo.

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

Model code (hexadecimal)	CPU module model name
02E9H	Q04UDEHCPU
02EAH	Q06UDEHCPU
02EBH	Q13UDEHCPU
02ECH	Q26UDEHCPU
02EDH	Q50UDEHCPU
02EEH	Q100UDEHCPU
0365H	Q26UDVPCPU
0366H	Q03UDVPCPU
0367H	Q04UDVPCPU
0368H	Q06UDVPCPU
036AH	Q13UDVPCPU
036CH	Q26UDVPCPU
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
4805H	R04ENCPU
4806H	R08ENCPU
4807H	R16ENCPU
4808H	R32ENCPU
4809H	R120ENCPU
4820H	R12CCPU-V
4C00H	R16MTCPU
4C01H	R32MTCPU
4C02H	R64MTCPU
4841H	R08PCPU
4842H	R16PCPU
4843H	R32PCPU
4844H	R120PCPU

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 (0000H)	Error For details on the error, refer to the following chapter. Page 99 ERROR CODE LIST

Relevant function

[Page 74 mdrClose](#)

[Page 82 mdrOpen](#)

4 ERROR CODE LIST

This chapter shows the error codes generated by the dedicated function library and the corrective actions.

4.1 Common Error Codes

The error codes common to C intelligent function module dedicated functions and MELSEC iQ-R series data link functions are as shown below.

Error code		Description	Corrective action
Decimal	Hexadecimal		
1	0001H	■Driver not started The driver has not been started.	<ul style="list-style-type: none"> • Check the channel number. • Correct the error that occurred when the driver is started. • Check the status of the system drive of the C intelligent function module. • Check if the operating system is running normally.
2	0002H	■Timeout error <ul style="list-style-type: none"> • A timeout occurred while waiting for response. • During CC-Link communication, the request was issued to other stations even when the own station number is 64. • The module specified as the communication target is not supported. 	<ul style="list-style-type: none"> • Review the operating status and mounting condition of the accessed station. • Retry on the user program. • Increase the timeout value of MELSEC iQ-R series data link function. • When requesting to other stations during CC-Link communication, set the own station number to the number other than '64'. • Check that the module specified as the communication target is supported.
66	0042H	■Already opened error The specified channel has already been opened.	Open once. (Even if the channel OPEN error occurs, the path of the correct channel will be returned to the argument.)
67	0043H	■Already closed error The specified channel has already been closed.	Close once.
69	0045H	■Unsupported function execution error An unsupported function in the target station has been executed.	<ul style="list-style-type: none"> • Check the path of the channel, network number, and station number. • Check that the function used in the target station is supported.
70	0046H	■Station number error <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) . 	Correct the network number and station number of the user program.
77	004DH	■Memory allocation error ■Resource shortage error ■Task over error Sufficient memory could not be allocated. Or, there are too many tasks that are using the C intelligent function module dedicated functions/MELSEC iQ-R series data link functions.	<ul style="list-style-type: none"> • The memory may be insufficient. Terminate other tasks currently running. Or decrease the access size. • Check if the C intelligent function module is running normally. • Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module. • Retry after decreasing the number of tasks that are using the C intelligent function module dedicated functions/MELSEC iQ-R series data link functions. • Review the size or number specified to the arguments of user program.
102	0066H	■Transmission error ■Restart error Data send has failed. Or, data send has been executed during restart.	<ul style="list-style-type: none"> • Retry. • Check if the C intelligent function module is running normally. • Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module. • Retry after completion of the restart.
103	0067H	■Reception error Data reception has failed.	<ul style="list-style-type: none"> • Retry. • Check if the C intelligent function module is running normally. • Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module.
130	0082H	■Device number error <ul style="list-style-type: none"> • The specified device number is out of range. • The specified bit device number is not multiple of 8. 	Check the device number.

Error code		Description	Corrective action
Decimal	Hexadecimal		
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. 	Check the device points.
16384 to 20479*1	4000H to 4FFFH	<p>■Errors detected in the access target CPU module.</p>	Refer to the user's manual of the access target CPU module.
-25056	9E20H	<p>■Processing code error</p> <p>The request target has received a request which cannot be processed by it.</p>	Check the network number and station number of the request target.
-26336	9920H	<p>■Routing request error for unsupported station</p> <p>Routing to another loop was requested to a station which does not support the routing function.</p>	Check the settings of routing parameter.
-28150	920AH	<p>■Device access error during data link stop</p> <p>Own station devices (RX, RY, RWw, RWr) have been accessed when the data link was not performed.</p>	<ul style="list-style-type: none"> • Check the specified device start number and size, or the device range of the parameter for the master station. • Restart the data link. (The data can be written/read despite this error, however; the data will not be guaranteed.)
-28151	9209H	<p>■Abnormal data reception error</p> <p>Abnormal response data has been received.</p>	Check if there is an error in the request target CPU module or link module. If the status is normal, try again.
-28158	9202H	<p>■WDT error</p> <p>WDT (system/user) error occurred.</p>	Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module.
-28410	9106H	<p>■Target CPU busy error</p> <p>The target CPU is busy.</p>	<ul style="list-style-type: none"> • 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	<p>■Target CPU unsupported error</p> <p>An unsupported request has been issued to the target CPU.</p>	Change the target CPU number specified in the user program.
-28413	9103H	<p>■Target CPU down error</p> <p>The target CPU is down.</p>	Check the operating status of the target CPU module and troubleshoot the error.
-28414	9102H	<p>■Target CPU abnormal start error</p> <p>A request has been issued to the CPU module which was not operating normally.</p>	Check the operating status of the target CPU module and troubleshoot the error.
-28415	9101H	<p>■Target CPU critical error</p> <p>A request has been issued to the CPU module in which a critical error occurred.</p>	Check the operating status of the target CPU module and troubleshoot the error.
-28416	9100H	<p>■Target CPU mounting error</p> <p>A request has been issued to the CPU number on which no CPU module is mounted.</p>	<ul style="list-style-type: none"> • Check if the target CPU module is mounted properly. • Change the target CPU number specified in the user program.
-28624	9030H	<p>■Function unsupported error</p> <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. 	<ul style="list-style-type: none"> • Check if the target module is the CC-Link module that 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	<p>■Intelligent function module offline error</p> <p>An attempt was made to access the Intelligent function module while it is offline.</p>	Check the status of the Intelligent function module and access the module while it is online.
-28626	902EH	<p>■Control data setting value out of range error</p> <p>The specified control data is out of range.</p>	Review the user program and correct the value set to the control data.
-28627	902DH	<p>■Transient unsupported error</p> <p>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.)</p>	<ul style="list-style-type: none"> • Check the communication route and target which support the transient request. • Change the station number of the own station.
-28628	902CH	<p>■Pointer address specification error</p> <p>An incorrect address has been specified to the argument pointer.</p>	Check the address of the specified pointer.
-28629	902BH	<p>■WDT not running error</p> <p>WDT was reset when WDT is not started up.</p>	Reset WDT after starting WDT.
-28630	902AH	<p>■WDT startup error</p> <p>An attempt was made to start up WDT while it is running.</p>	Start WDT after stopping WDT.

Error code		Description	Corrective action
Decimal	Hexadecimal		
-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. 	<ul style="list-style-type: none"> • Check the specified offset. • Check the specified buffer size. • Check the offset and its size.
-28632	9028H	■I/O number error The specified I/O number is out of range. No accessible module is mounted on the specified I/O number.	Check the specified I/O number.
-28634	9026H	■Intelligent function module down error There is a problem in the Intelligent function module.	<ul style="list-style-type: none"> • 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.	<ul style="list-style-type: none"> • 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.	<ul style="list-style-type: none"> • 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 operation status of the CPU module to 'RUN'.
-28653	9013H	■I/O assignment error <ul style="list-style-type: none"> • An attempt was made to read the value of input signal (X) from an output module. • An attempt was made to write the value of output signal (Y) to an input module. • An attempt was made to read the value of output signal (Y) from an input module. 	<ul style="list-style-type: none"> • Review the specified input number (X). • Review the specified output number (Y).
-28660	900CH	■Access size error The specified size is out of range.	Review the specified offset and size.
-28661	900BH	■Inaccessible error Inaccessible area was specified.	Review the specified offset and size.
-28662	900AH	■CPU number specification error The specified CPU number is out of range or unavailable.	<ul style="list-style-type: none"> • Review the specified CPU number. • Check the operating status of the specified CPU module.
-28663	9009H	■Base unit number specification error The specified base unit number is out of range.	Review the specified base unit number.
-28664	9008H	■Data send area occupied	Retry.
-28665	9007H	■No registration data error	Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module.
-28666	9006H	■Data length error	Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module.
-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	<ul style="list-style-type: none"> • Review the parameters. • Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module.
-28672	9000H	■Processing code error	Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module.

*1 When the access route specifies a wrong I/O number in the configuration of CC-Link, errors in the following range may occur. Check if the correct I/O number is specified.

- Specify other than network module: 4000H to 4FFFH
- Specify the serial communication module: 7000H to 7FFFH
- Specify the CC-Link IE Field Network module: D000H to DFFFH
- Specify the CC-Link IE Controller Network module: E000H to EFFFH
- Specify the MELSECNET/H module: F000H to FFFFH

4.2 C Intelligent Function Module Dedicated Functions

The following shows the error code of the C intelligent function module dedicated function.

Error code		Description	Corrective action
Decimal	Hexadecimal		
-203	FF35H	■I/O signal error The specified I/O signal is out of range.	Check the specified I/O signal.
-204	FF34H	■I/O access size error The specified I/O access size is out of range.	Check the specified I/O access size (I/O number and read/write size in words) .
-208	FF30H	■Offset error <ul style="list-style-type: none"> • The specified offset is out of range. • An AnS series module (buffer memory) has been accessed. 	<ul style="list-style-type: none"> • Check the specified offset. • Check the specified I/O number.
-209	FF2FH	■Buffer memory size error <ul style="list-style-type: none"> • The specified offset and its size is out of range. • The address of data storage buffer pointer is 0. • The specified size is 0. 	<ul style="list-style-type: none"> • 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.	<ul style="list-style-type: none"> • 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.
-220	FF24H	■WDT type error The specified WDT type is out of range.	Check the specified WDT type.
-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.
-231	FF19H	■Event timeout error A timeout occurred while waiting for an event.	<ul style="list-style-type: none"> • Increase the timeout time. • Check that the event number (interrupt pointer number) is set correctly.
-234	FF16H	■Event wait error An error other than timeout occurred while the function waits for the event.	<ul style="list-style-type: none"> • Check that the operating system is not forcibly terminating the program. • Check if the C intelligent function module is running normally. • Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module.
-235	FF15H	■Number of event settings specification error The specified number of event settings is out of range.	Check the number of specified event settings.
-237	FF13H	■Detailed information character string specification error The length of the specified character string was out of range or characters which cannot be specified was specified.	Correct the length of the specified character string or character string data.
		■Application code specification error Five or more digits of the hexadecimal number is specified in the specified application code.	Change the specified application code.
-239	FF11H	■Memory card mounting error The following function has been executed with no specified memory card inserted. <ul style="list-style-type: none"> • CITL_MountMemoryCard • CITL_UnmountMemoryCard 	Check that the memory card is inserted.
-240	FF10H	■Clock data incorrect error The clock data to be set or the read clock data is incorrect.	<ul style="list-style-type: none"> • Check the clock data to be set. • If this error occurs at reading the clock data, set the clock data.
-241	FF0FH	■Cycle specification error <ul style="list-style-type: none"> • The specified cycle is out of range. • The cycle was set even when it had already been set. 	<ul style="list-style-type: none"> • Check the specified cycle. • Check if the cycle has been already set.
-242	FF0EH	■Synchronization type specification error The specified synchronization type is out of range.	Check the specified synchronization type.
-246	FF0AH	■Timer event registration error Timer event registration has failed.	<ul style="list-style-type: none"> • Retry. • Check if the C intelligent function module is running normally. • Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module.

Error code		Description	Corrective action
Decimal	Hexadecimal		
-257	FEFFH	<p>■Interrupt event type specification error The value specified as the interrupt number is out of range.</p>	Review the specified value, and specify a value within the range.
-258	FEFEH	<p>■Output signal (Y) number specification error The value specified as the output signal (Y) 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 The processing has not been registered when enabling a processing which corresponds to an event (interrupt) .</p>	Register the processing for the event (interrupt) and execute the operation once again.
-260	FEFCH	<p>■Memory card mount/unmount error Mount process or unmount process of the specified memory card has failed.</p>	<ul style="list-style-type: none"> • Retry. • Check if the specified memory card is damaged. • Replace the memory card.
-264	FEF8H	<p>■Pointer error The address of the specified pointer is incorrect.</p>	Check the specified pointer.
-267	FEF5H	<p>■Authentication error The specified password is incorrect.</p>	Check the specified password.
-288	FEE0H	<p>■Individual identification information read error Reading the individual identification information has failed.</p>	<ul style="list-style-type: none"> • Check if the C intelligent function module is running normally. • Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module.
-292	FEDCH	<p>■Standard ROM shutdown error Shutdown processing of the standard ROM has failed.</p>	<ul style="list-style-type: none"> • Check if files in the standard ROM are being accessed. • Check whether all files in the standard ROM have been closed.
-320	FEC0H	<p>■Clock rate specification error The specified clock rate is out of range.</p>	Check the specified clock rate.

4.3 MELSEC iQ-R Series Data Link Functions

The following shows the error code of MELSEC iQ-R series data link function.

Error code		Description	Corrective action
Decimal	Hexadecimal		
-1	FFFFH	■Path error The specified path is unavailable.	Use a path pointer returned by the mdrOpen 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. 	<ul style="list-style-type: none"> 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.	<ul style="list-style-type: none"> 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 mdrRandR function, or the mdrRandW function exceeds 10240. 	<ul style="list-style-type: none"> Check the specified device size. Check the device number and the size. Specify an even-number byte. Set the total number of points specified for each block number assigned for mdrRandR/mdrRandW 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 mdrOpen function is unavailable.	Check the specified channel number.
-11	FFF5H	■Insufficient buffer area error The read data storage destination 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.	<ul style="list-style-type: none"> Check the block number (device type) of the specified device. Check if the specified device and block number are available in the target station.
-13	FFF3H	■Write protect error The specified block number of the extension file register is overlapping with the write protect area of the memory card.	<ul style="list-style-type: none"> Check the block number (device type) of the extension file register. Check the write protect switch of the access target memory card.
-16	FFF0H	■Station number/network number error <ul style="list-style-type: none"> The specified station number or network number is out of range. A device which is not accessible by the target station has been specified. 	<ul style="list-style-type: none"> Check the specified station number and network number. Check the devices which can be accessed by the target station.
-17	FFEFH	■All stations/group number specification error 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 specifying all stations/group number. Specify the device type to "No arrival confirmation" when "All stations" or "Group number" is specified for the station number.
-18	FFEEH	■Remote specification error The code specified with the mdrControl function is invalid.	Check the specified code.
-31	FFE1H	■Module load error Loading modules required for executing functions has failed.	<ul style="list-style-type: none"> 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 intelligent function module.
-32	FFE0H	■Resource timeout error The resource is being used by other tasks/threads and is not released within 30 seconds.	<ul style="list-style-type: none"> Retry. The memory may be insufficient. Terminate other tasks currently running. Check if the C intelligent function module is running normally. Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module.

Error code		Description	Corrective action
Decimal	Hexadecimal		
-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.
-36	FFDCH	■Registry write error Writing parameter files to the registry has failed.	<ul style="list-style-type: none"> • Check if the standard ROM has already been shutdown. • Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module.
-42	FFD6H	■Close error Communications cannot be closed.	<ul style="list-style-type: none"> • Retry. • Check if the C intelligent function module is running normally. • Reset the CPU module or turn the power OFF→ON, and reset C intelligent function module.
-43	FFD5H	■ROM operation error Writing a TC setting value was attempted to the programmable controller CPU that was running the ROM.	Change the TC setting value during RAM operation.
-52	FFCCH	■MELSEC iQ-R series data link function service error MELSEC iQ-R series data link function service is disabled.	Enable the MELSEC iQ-R series data link function service with an engineering tool.
-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 module.
-57	FFC7H	■Access route error The specified access route is out of range.	Check the specified access route.
-80	FFB0H	■Connection destination CPU error The connection destination CPU is other than RCPU.	Connect RCPU.
-81	FFB1H	■Label code mismatch error The label assignment information of the CPU module is changed.	Obtain label information by using the mdrGetLabelInfo function again.
-82	FFB2H	■Label incorrect value error <ul style="list-style-type: none"> • An incorrect label name was specified. • Non-existent label name • A label name assigned to a device that does not support random read/write. • A label name assigned to a device for which the method (index modification or indirect specification) that can not be handled was specified. 	Check the specified label name or the device specification method.
-83	FFB3H	■Size error The number of labels exceeded the range.	Check the number of labels.
-84	FFB4H	■Device specification method error The device was specified with the method (Bit specification or digit specification) that can not be handled.	Check the device specification method.
-4096 to -1	F000H to FFFFH ^{*1}	📖 Q Corresponding MELSECNET/H Network System Reference Manual (PLC to PLC network)	
-4097 to -8192	EFFFH to E000H ^{*1}	Refer to the following manual. 📖 MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application) 📖 MELSEC-Q CC-Link IE Controller Network Reference Manual	
-8193 to -12288	DFFFFH to D000H ^{*1}	Refer to the following manual. 📖 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	
-16384 to -12289	C000H to CFFFFH ^{*1}	Refer to the following manuals. 📖 MELSEC iQ-R Ethernet User's Manual (Application)	
-16385 to -20480	BFFFFH to B000H ^{*1}	Refer to the following manual. 📖 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	
28672 to 32767	7000H to 7FFFH ^{*1}	Refer to the following manuals. 📖 MELSEC iQ-R Serial Communication Module User's Manual (Application)	

- *1 When the access route specifies a wrong I/O number in the configuration of CC-Link, errors in the following range may occur. Check if the correct I/O number is specified.
- Specify other than network module: 4000H to 4FFFH
 - Specify the serial communication module: 7000H to 7FFFH
 - Specify the CC-Link IE Field Network module: D000H to DFFFH
 - Specify the CC-Link IE Controller Network module: E000H to EFFFH
 - Specify the MELSECNET/H module: F000H to FFFFH

INDEX

A

Accessing own station 11

B

Bus interface communication 8

C

C intelligent function module 5
C intelligent function module dedicated function . . . 5
C intelligent function module for ISR 18
CC-Link communication 8
CC-Link IE Controller Network communication . . . 8
CC-Link IE Field Network communication 8
Channel 15
CW Workbench 5
CW-Sim 5

D

Dedicated function library 5,6
Device type 15
Dummy access 9

E

Engineering tool 5

F

File access mode 23

H

Header file 6,15,27,30,31

M

MELSEC iQ-R series data link function 5
MELSECNET/H communication 8

S

Single network 11,12

T

Task 9

U

User watchdog timer 7

V

VxWorks 5,18

FUNCTION INDEX

C

CITL_ChangeFileSecurity	23	mdrRandR	83
CITL_ClearError	24	mdrRandRLabel	86
CITL_DisableYInt	25	mdrRandW	89
CITL_DisableYInt_ISR	63	mdrRandWLabel	91
CITL_EnableYInt	26	mdrReceive	94
CITL_EnableYInt_ISR	64	mdrSend	95
CITL_EntryDedicatedInstFunc	27	mdrTypeRead	96
CITL_EntryTimerEvent	28		
CITL_EntryWDTInt	30		
CITL_EntryYInt	31		
CITL_FromBuf	32		
CITL_FromBuf_ISR	65		
CITL_GetCounterMicros	33		
CITL_GetCounterMicros_ISR	66		
CITL_GetCounterMillis	34		
CITL_GetCounterMillis_ISR	67		
CITL_GetErrInfo	35		
CITL_GetFileSecurity	36		
CITL_GetIDInfo	37		
CITL_GetLEDStatus	38		
CITL_GetSerialNo	39		
CITL_GetSwitchStatus	40		
CITL_GetTime	41		
CITL_GetUnitStatus	42		
CITL_MountMemoryCard	43		
CITL_RegistEventLog	44		
CITL_RegistEventLog_ISR	68		
CITL_ResetWDT	45		
CITL_SetLEDStatus	46		
CITL_SetLEDStatus_ISR	69		
CITL_ShutdownRom	47		
CITL_StartWDT	48		
CITL_StopWDT	49		
CITL_SysClkRateGet	50		
CITL_SysClkRateSet	51		
CITL_ToBuf	52		
CITL_ToBuf_ISR	70		
CITL_UnmountMemoryCard	53		
CITL_WaitSwitchEvent	54		
CITL_WaitTimerEvent	55		
CITL_WaitYEvent	56		
CITL_X_In_Bit	57		
CITL_X_In_Word	58		
CITL_X_In_Word_ISR	71		
CITL_X_Out_Bit	59		
CITL_X_Out_Word	60		
CITL_X_Out_Word_ISR	72		
CITL_Y_In_Bit	61		
CITL_Y_In_Word	62		
CITL_Y_In_Word_ISR	73		

M

mdrClose	74
mdrControl	75
mdrDevRst	76
mdrDevSet	77
mdrGetLabelInfo	78
mdrInit	81
mdrOpen	82

REVISIONS

*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
December 2015	SH(NA)-081568ENG-A	First edition

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.

Japanese manual number: SH-081565-A

© 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 to:

- (1) Damages caused by any cause found not to be the responsibility of Mitsubishi.
- (2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.
- (3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.
- (4) Replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

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

TRADEMARKS

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)-081568ENG-A(1512)KWIX

MODEL:RD55UP06-V-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.