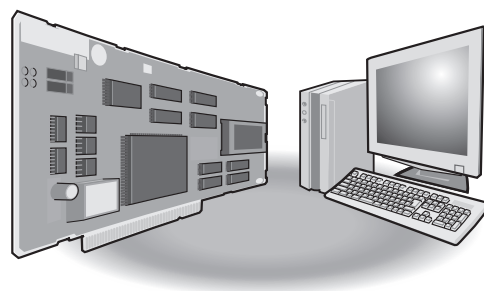




Network Interface Board

MELSEC Data Link Library Reference Manual

-Q80BD-J61BT11N
-Q81BD-J61BT11
-Q80BD-J71LP21-25
-Q80BD-J71LP21S-25
-Q81BD-J71LP21-25
-Q80BD-J71LP21G
-Q80BD-J71BR11
-Q80BD-J71GP21-SX
-Q80BD-J71GP21S-SX
-Q81BD-J71GP21-SX
-Q81BD-J71GP21S-SX
-Q80BD-J71GF11-T2
-Q81BD-J71GF11-T2
-NZ81GN11-SX
-NZ81GN11-T2



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.

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

INTRODUCTION

Thank you for purchasing the Mitsubishi Electric network interface boards.

This manual describes the programming procedure and function specifications of the MELSEC data link library.

Before using this product, please read this manual and the relevant manuals carefully and develop familiarity with the functions and performance of the MELSEC data link library to handle the product correctly.

Note that the menu names and operating procedures may differ depending on an operating system in use and its version.

When reading this manual, replace the names and procedures with the applicable ones as necessary.

Please make sure that the end users read this manual.

Relevant products

Item	Model
CC-Link IE TSN board	NZ81GN11-SX, NZ81GN11-T2
CC-Link IE Controller Network board	Q80BD-J71GP21-SX, Q80BD-J71GP21S-SX, Q81BD-J71GP21-SX, Q81BD-J71GP21S-SX
CC-Link IE Field Network board	Q80BD-J71GF11-T2, Q81BD-J71GF11-T2
CC-Link Ver.2 board	Q80BD-J61BT11N, Q81BD-J61BT11
MELSECNET/H board	Q80BD-J71LP21-25, Q81BD-J71LP21-25, Q80BD-J71LP21S-25, Q80BD-J71LP21G, Q80BD-J71LP21GE, Q80BD-J71BR11

CONTENTS

SAFETY PRECAUTIONS	1
INTRODUCTION	1
RELATED MANUALS	5
GENERIC TERMS AND ABBREVIATIONS	6
DISCONTINUED MODELS	6
CHAPTER 1 OVERVIEW	7
CHAPTER 2 PROGRAMMING	9
2.1 Programming Procedure	9
2.2 Precautions when Using MELSEC Data Link Library	10
Precautions when programming	10
Precautions when accessing own station link devices and other station's programmable controller devices. . .	13
Transient timeout monitoring time	15
2.3 Settings for Using Functions	15
Using Visual Basic 5.0 or Visual Basic 6.0	15
Using Visual C++ 5.0 or Visual C++ 6.0	16
Using Visual Basic.NET	17
Using Visual C++ .NET 2003, Visual C++ 2005, or Visual C++ 2008	18
Using Visual C++ in Visual Studio 2010 or Visual Studio 2012	20
Using Visual C++ in Visual Studio 2013 or later	24
Using Visual C#	30
CHAPTER 3 ACCESSIBLE DEVICES AND RANGES	31
3.1 Access Target	31
3.2 Accessible Ranges	33
Access target on own network	33
Access target when connected via network	34
3.3 Accessible Devices	36
Access to link devices and buffer memory of the own station	36
Access to devices of MELSEC iQ-R series module	37
Access to devices of MELSEC-Q/L/QnA series module	39
Access to devices of MELSEC-A series module	40
Access to an Ethernet adapter module and head module	41
Access to a remote station and remote device station	41
Access to other station buffer memory of CC-Link	41
Access using the SEND function or the RECV function	42
CHAPTER 4 FUNCTIONS	43
4.1 Function List	43
4.2 Common Specifications of Functions	45
Specifying channel number	45
Specifying station numbers	46
Specifying network numbers and station numbers for extended functions	48
Specifying device types	50
4.3 Function Details	53
mdOpen (opening communication lines)	54
mdClose (closing communication lines)	55

mdSendEx (batch writing extended devices)	56
mdSendEx (SEND function)	58
mdReceiveEx (batch reading extended devices)	61
mdReceiveEx (RECV function)	63
mdRandWEx (writing extended devices randomly)	66
mdRandREx (reading extended devices randomly)	69
mdDevSetEx (setting extended bit devices)	73
mdDevRstEx (resetting extended bit devices)	74
mdRemBufWriteEx (writing data to the buffer memory of a remote device station/remote station)	75
mdRemBufReadEx (reading data from the buffer memory of a remote device station/remote station)	77
mdRemBufWriteIPEX (writing data to the buffer memory of a remote station, target station IP address specified)	79
mdRemBufReadIPEX (reading data from the buffer memory of a remote station, target station IP address specified)	81
mdTypeRead (reading model names of CPU)	83
mdControl (remote RUN/STOP/PAUSE)	87
mdWaitBdEvent (waiting for event occurrence)	88
mdBdRst (resetting board)	91
mdBdModSet (setting modes of board)	92
mdBdModRead (reading modes of board)	94
mdBdLedRead (reading LED information of the board)	96
mdBdSwRead (reading switch status of the board)	103
mdBdVerRead (reading version information of the board)	104
mdInit (initializing programmable controller information table)	106
mdSend (batch writing devices)	107
mdSend (SEND function)	109
mdReceive (batch reading devices)	111
mdReceive (RECV function)	113
mdRandW (writing devices randomly)	115
mdRandR (reading devices randomly)	117
mdDevSet (setting bit devices)	119
mdDevRst (resetting bit devices)	120

CHAPTER 5 SAMPLE PROGRAMS 121

5.1 Using Visual Basic .NET	123
MTEST, MTEST2	124
5.2 Using Visual Basic 5.0 or Visual Basic 6.0	124
5.3 Using Visual C++ or Visual C#	125

CHAPTER 6 DEVICE MONITOR UTILITY 126

6.1 Operation Flow	127
6.2 Menu List	128
6.3 Screen Configuration and Basic Operations	129
Start and end	129
Screen configuration and basic operations	130
Help function	131
6.4 Selecting Monitoring Method	132
Monitoring devices in a batch	132
Monitoring registered 16-point devices	132
6.5 Setting Monitoring Target	133

6.6	Setting Devices to be Monitored	134
6.7	Changing Current Values	135
	Changing word device/double-word device values	135
	Turning bit devices ON and OFF	137
6.8	Switching Display Format	138
6.9	Using Numerical Pad	139

CHAPTER 7	ERROR CODES	140
------------------	--------------------	------------

APPENDIX	148
-----------------	------------

Appendix 1	Method for Increasing Minimum Working Set Size of Personal Computer	148
	Processing overview of sample program	148
	Sample programs	149

FUNCTION INDEX	151
-----------------------	------------

REVISIONS	153
CONDITIONS OF USE FOR THE PRODUCT	154
WARRANTY	155
INFORMATION AND SERVICES	156
TRADEMARKS	156
COPYRIGHTS	156

RELATED MANUALS

Manual name [manual number]	Description	Available form
MELSEC Data Link Library Reference Manual [SH-081035ENG] (this manual)	Programming, accessible devices and their ranges, functions, sample programs, and error codes for the MELSEC data link library	Print book e-Manual PDF
CC-Link IE TSN Interface Board User's Manual [SH-082614ENG]	System configuration, specifications, functions, handling, wiring, and troubleshooting for CC-Link IE TSN interface board	Print book e-Manual PDF
CC-Link IE Controller Network Interface Board User's Manual (For SW1DNC-MNETG-B) [SH-080691ENG]	System configuration, specifications, functions, handling, wiring, and troubleshooting for CC-Link IE Controller Network interface board	Print book PDF
CC-Link IE Field Network Interface Board User's Manual (For SW1DNC-CCIEF-B) [SH-080980ENG]	System configuration, specifications, functions, handling, wiring, and troubleshooting for CC-Link IE Field Network interface board	Print book e-Manual PDF
Type Q80BD-J61BT11N/Q81BD-J61BT11 CC-Link System Master/Local Interface Board User's Manual (For SW1DNC-CCBD2-B) [SH-080527ENG]	System configuration, specifications, functions, handling, wiring, and troubleshooting for type Q80BD-J61BT11N/Q81BD-J61BT11 CC-Link system master/local interface board	Print book PDF
MELSECNET/H Interface Board User's Manual (For SW0DNC-MNETH-B) [SH-080128]	System configuration, specifications, functions, handling, wiring, and troubleshooting for MELSECNET/H interface board	Print book PDF



e-Manual refers to the Mitsubishi Electric 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.

GENERIC TERMS AND ABBREVIATIONS

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

Generic term/ abbreviation	Description
CC-Link IE TSN board	NZ81GN11-SX, NZ81GN11-T2 CC-Link IE TSN interface board
CC-Link IE Controller Network board	Q80BD-J71GP21-SX, Q80BD-J71GP21S-SX, Q81BD-J71GP21-SX, Q81BD-J71GP21S-SX CC-Link IE Controller Network interface board
CC-Link IE Field Network board	Q80BD-J71GF11-T2, Q81BD-J71GF11-T2 CC-Link IE Field Network interface board
CC-Link Ver.2 board	Q80BD-J61BT11N, Q81BD-J61BT11 CC-Link system master/local interface board
MELSECNET/H board	Q80BD-J71LP21-25, Q81BD-J71LP21-25, Q80BD-J71LP21S-25, Q80BD-J71LP21G, Q80BD-J71LP21GE, Q80BD-J71BR11 MELSECNET/H interface board
Board	PC interface boards supported by MELSEC data link library
Utility	Utilities of PC interface board supported by MELSEC data link library

DISCONTINUED MODELS

The following models are described in this manual, but have no longer been produced.

For the onerous repair term after discontinuation of production, refer to "WARRANTY" in this manual.

Model	Production discontinuation
Q80BD-J71LP21GE	October 2010
Q80BD-J71LP21S-25	June 2022

1 OVERVIEW

This chapter explains the overview of the MELSEC data link library.

MELSEC data link library

MELSEC data link library is a library used to access own station link devices of the board and device memory of other station's programmable controller CPU which uses the board.

With the MELSEC data link library, programs to access devices or device memories can be created easily without concern for communication routes.

MEMO

2 PROGRAMMING

This chapter explains how to use the MELSEC data link library.

An overview of the programming languages supported by each board is as follows.

○: Supported, —: Not supported

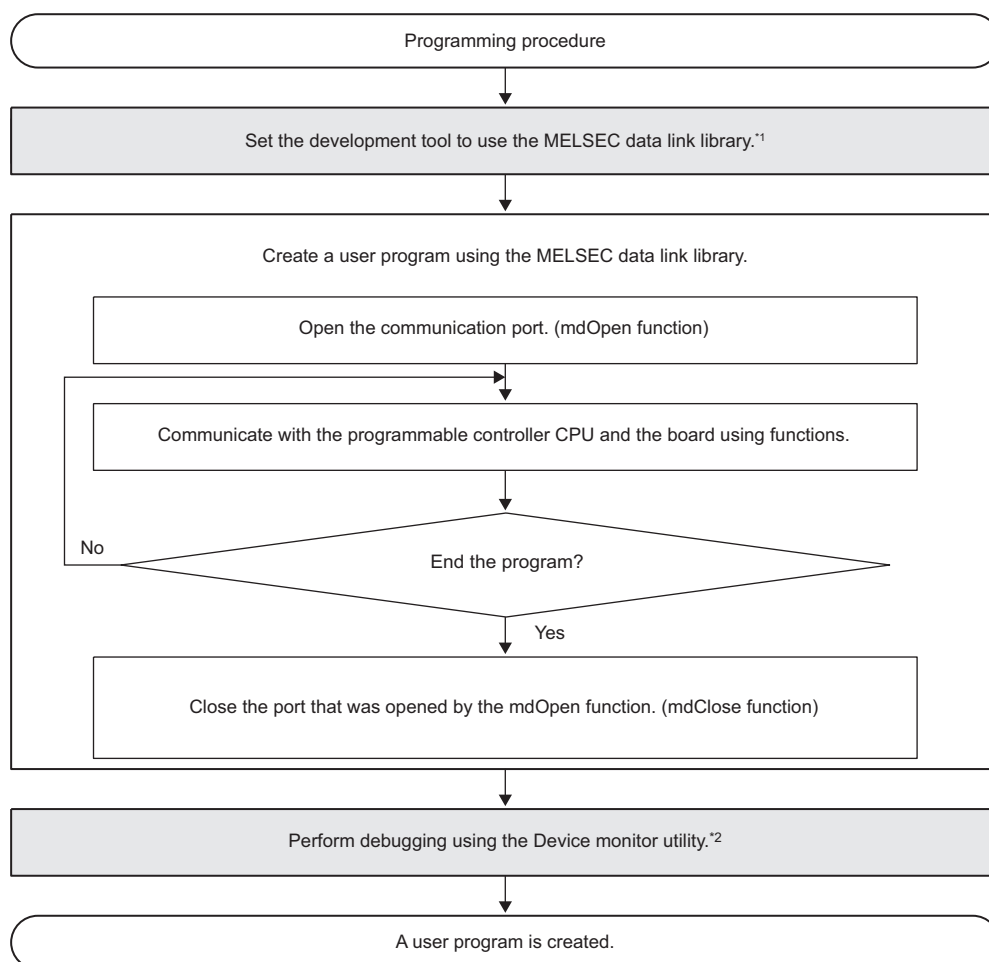
Type	Visual Basic 5.0	Visual Basic 6.0	Visual Basic .NET ^{*1}	Visual C++	Visual C#
CC-Link IE TSN board	—	—	○	○	○
CC-Link IE Controller Network board	—	○	○	○	—
CC-Link IE Field Network board	—	—	○	○	—
CC-Link Ver.2 board	○	○	○	○	—
MELSECNET/H board	○	○	○	○	—

^{*1} Visual Basic other than Visual Basic 5.0 and Visual Basic 6.0

For details, refer to the manual for each board.

2.1 Programming Procedure

The following flow chart shows the procedure to create a user program using the MELSEC data link library on the personal computer to which the software package is installed.



*1 Page 15 Settings for Using Functions

*2 Page 126 DEVICE MONITOR UTILITY

2.2 Precautions when Using MELSEC Data Link Library

Precautions when programming

Header file

When creating programs in C or C++, include the Windows header file (windows.h).

Multi-thread communications

■Unsupported boards

For the following boards, the MELSEC data link library cannot be used for multiple threads in a process.

- CC-Link IE Controller Network board
- CC-Link IE Field Network board
- CC-Link Ver.2 board
- MELSECNET/H board

Access the MELSEC data link library with a single thread.

■Supported board

CC-Link IE TSN boards can be used for multithreading.

When using the MELSEC data link library for multiple threads in a process, note the following:

- Execute mdOpen at the start of a process, and use the path obtained with mdOpen for each thread.
- After executing functions in all threads, execute mdClose. If executing mdClose while executing the functions, a MELSEC data link library error may occur in other threads.
- Do not execute mdInit in a thread. If executed, a MELSEC data link library error may occur in other threads.

Opening and closing of a communication line

Perform the opening and closing processes of a communication line (mdOpen, mdClose) only once at the beginning and the end of a user program.

Repeating opening and closing processes for each communication causes deterioration of communication performance.

Number of stations for accessing other stations

When accessing other stations with the user program, limit the total number of access stations to 256 or less.

The communication performance will be deteriorated if the total number of access stations is 257 or more.

Forcible termination of a user program

When the user program in which the MELSEC data link library operation is currently running is forcibly terminated, the following symptoms may occur.

- The application that is forcibly terminated cannot be ended.
- A MELSEC data link library error occurs in other applications.
- The forcible termination affects other Mitsubishi Electric software packages (such as MX Component, GX Works2).

Execution speed

■Function execution time at the initial access

The MELSEC data link library obtains detailed information of the programmable controller at the initial access to the programmable controller CPU.

Therefore, a longer function execution time is required for the initial function.

■Effects of Windows processes or other applications

The execution speed and the execution interval of the MELSEC data link library function may be extended temporarily by Windows® processes or other applications.

Create programs considering these conditions.

■When executing multiple MELSEC data link library functions simultaneously

Some functions may take long to complete.

Static type variables

Do not specify any variables which are declared in static for output arguments of the MELSEC data link library functions.

Service applications

The MELSEC data link library cannot be accessed from Windows Service applications.

Access the MELSEC data link library from a user application.

Board reset

■Opening and closing of a communication line

If a board is reset due to parameter writing, etc., an error occurs in a user application for which a communication line is already open.

Close the communication line once, then reopen it.

■Board reset processing with functions

When executing the mdBdRst or mdBdModSet function in a user program, board reset processing with each function is completed at the time when the value is returned.

Create a program which checks the returned value of the function.

64-bit version user program

■Accessing CPU modules other than QCPU (Q mode) or MELSEC iQ-R series CPU modules

64-bit version user program cannot access CPU modules other than QCPU (Q mode) or MELSEC iQ-R series CPU modules.

Use a 32-bit version user program.

■Accessing boards other than CC-Link IE TSN boards

64-bit version user program cannot access boards other than CC-Link IE TSN boards.

Use a 32-bit version user program.

■Creating 64-bit version user program

To create 64-bit version user program, a project needs to be configured to set the 64-bit platform as a target platform.

For configuring a project and setting a target platform, refer to Help (How to: Configure Projects to Target Platforms) in Visual Studio®.

■Creating 64-bit version user program using Visual Basic

.NET Framework 4.0 or .NET Framework compatible with .NET Framework 4.0 is required.

Use Visual Studio 2010 or later.

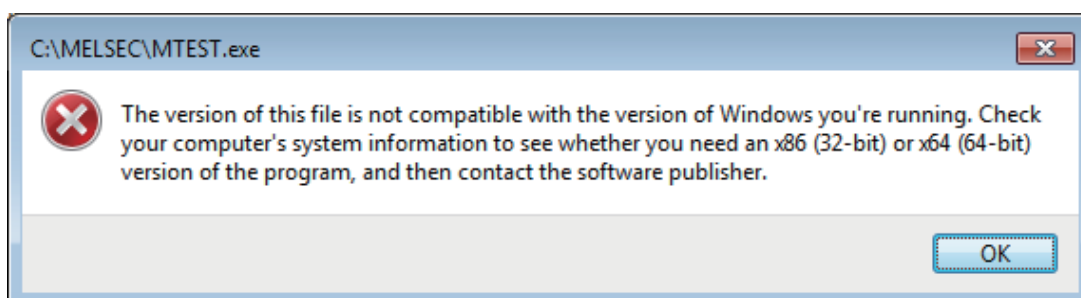
■Restriction when creating 64-bit version user programs

64-bit version user programs can be created on a 32-bit version operating system.

However, the following screen appears and the programs cannot be executed.

Ex.

<When using Windows 7 (32-bit version)>



/SAFESEH (Image has Safe Exception Handlers)

Do not use /SAFESEH (Image has Safe Exception Handlers) option.
The project cannot be built normally.

Influence of operating system and other applications

When the system resource of the operating system is insufficient due to the automatic start of the update program of the operating system or other applications, or the devices are accessed from other applications, "Board Driver I/F error 102 (0066H)" may occur during executing a MELSEC data link library function.

Take the following measures as necessary.

- Retry process of a MELSEC data link library function
- Disable the automatic update of the operating system and other applications
- Stop other applications

Device access when the cyclic data assurance is enabled

Use the batch write/batch read function (mdSendEx, mdSend, mdReceiveEx, mdReceive) to access devices when enabling the cyclic data assurance (32-bit data integrity assurance and block data assurance per station).

The cyclic data assurance (32-bit data integrity assurance and block data assurance per station) is not enabled while accessing the device by the random write/random read function (mdRandWEx, mdRandW, mdRandREx, mdRandR).

Random access

When using the random write/random read function (mdRandWEx, mdRandW, mdRandREx, mdRandR) with too many points specified, the operation of a user application may become unstable due to the insufficient system resources of an operating system.

In this case, use the batch write/batch read function (mdSendEx, mdSend, mdReceiveEx, mdReceive).

Programming when using Visual C#

■Namespace and class name

Use the following namespace and class name for the MELSEC data link library functions.

- Namespace: MitsubishiElectric.MELSEC.Board
- Class name: mdFunctions

Precautions when accessing own station link devices and other station's programmable controller devices

It is necessary to establish an interlock depending on a link status between the own station and other station.
Data are validated only when the following conditions are satisfied.

CC-Link IE TSN

■Access to own station link devices (RX, RY, RW, LB, LW)

Writing/reading data to/from the own station link devices are validated only when the bit of the own station data link error status (SB49) is OFF (normal communication).

However, even if the above condition is not satisfied, the processing of writing/reading data to/from the CC-Link IE TSN board ends normally.

■Other station transient access (remote operation and device access of other station's programmable controller CPU)

While the access is validated for the devices which check the own station link device accesses, the other station transient access can be performed when the bits of the each station data link status (the bits in the accessed station correspond to SW0B0 to 0B7 read from the own station) is OFF (normal communication).

CC-Link IE Controller Network

■Access to own station link devices (LX, LY, LB, LW)

Writing/reading data to/from the own station link devices are validated only when the bits of the own station handshaking status (SB47) and own station data link status (SB49) are OFF (normal communication), and the bit of the own station module status (SB20) is OFF (normal communication).

However, even if the above conditions are not satisfied, the processing of writing/reading data to/from the CC-Link IE Controller Network board ends normally.

■Other station transient access (remote operation and device access of other station's programmable controller CPU)

While the access is validated for the devices which check the link device accesses, the other station transient access can be performed when the bits of the handshake status in accessed station (the bits in the accessed station correspond to SWA0 to A7 read from the own station) and the bits of the data link status (the bits in the accessed station correspond to SWB0 to B7 read from the own station) are OFF (normal communication).

CC-Link IE Field Network

■Access to own station link devices (RX, RY, RW)

Writing/reading data to/from the own station link devices are validated only when the bits of the own station handshaking status (SB47) and own station data link status (SB49) are OFF (normal communication).

However, even if the above conditions are not satisfied, the processing of writing/reading data to/from the CC-Link IE Field Network board ends normally.

■Other station transient access (remote operation and device access of other station's programmable controller CPU)

While the access is validated for the devices which check the link device accesses, the other station transient access can be performed when the bits of the handshake status in accessed station (the bits in the accessed station correspond to SWA0 to A7 read from the own station) and the bits of the data link status (the bits in the accessed station correspond to SWB0 to B7 read from the own station) are OFF (normal communication).

MELSECNET/H

■Access to own station link devices (LX, LY, LB, LW)

Writing/reading data to/from the own station link devices are validated only when the bits of the own station handshaking status (SB47) and own station data link status (SB49) are OFF (normal communication), and the bit of the own station module status (SB20) is OFF (normal communication).

However, even if the above conditions are not satisfied, the processing of writing/reading data to/from the MELSECNET/H board ends normally.

■Other station transient access (remote operation and device access of other station's programmable controller CPU)

While the access is validated for the devices which check the link device accesses, the other station transient access can be performed when the bits of the handshake status in accessed station (the bits in the accessed station correspond to SW70 to 73 read from the own station) and OFF (normal communication) and the bits of the cyclic transmission status (the bits in the accessed station correspond to SW74 to 77 read from the own station) are OFF (cyclic transmission is being processed).

Transient timeout monitoring time

The transient timeout monitoring time for the MELSEC data link library refers to the timeout time for processing of one communication with an access target. Communication processing is performed multiple times for the first access^{*1} or for some functions used; therefore, the actual timeout detection time may be longer than a set transient timeout monitoring time.

- *1 Regardless of the functions used, communication processing is performed multiple times for the first access to perform the following processing:
- Determining an access target
 - Acquiring information on the access target programmable controller CPU or board

2.3 Settings for Using Functions

This section explains how to set the development tool to use the MELSEC data link library functions.
For the programming languages supported by each board, refer to the manual for each board.

Point

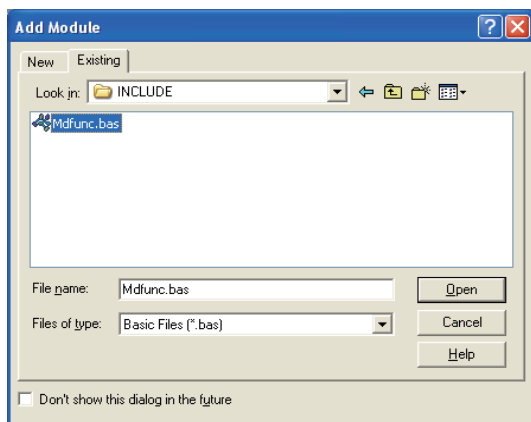
When creating 64-bit version user application

- To create a 64-bit version user application, a project needs to be configured to set the 64-bit platform as a target platform.
- For configuring a project and setting a target platform, refer to Help (How to: Configure Projects to Target Platforms) in Visual Studio.

When creating 64-bit version user programs using Visual Basic

- .NET Framework 4.0 or .NET Framework compatible with .NET Framework 4.0 is required.
- Use Visual Studio 2010 or later as a development environment.

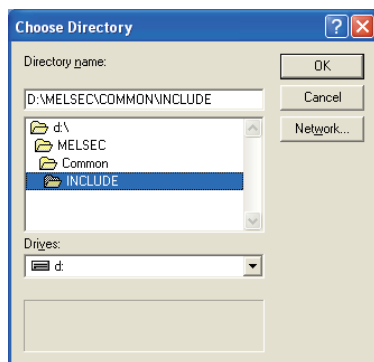
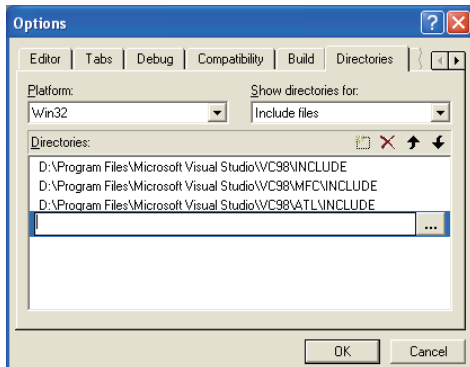
Using Visual Basic 5.0 or Visual Basic 6.0



1. Start Visual Basic® and select [Project] ⇒ [Add Module].
2. Select the [Existing] tab and select "Mdfunc.bas."
 - "Mdfunc.bas" is stored in 'COMMON\INCLUDE' in the folder where the software package has been installed.

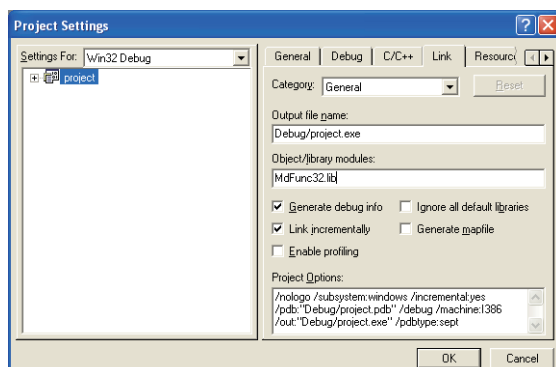
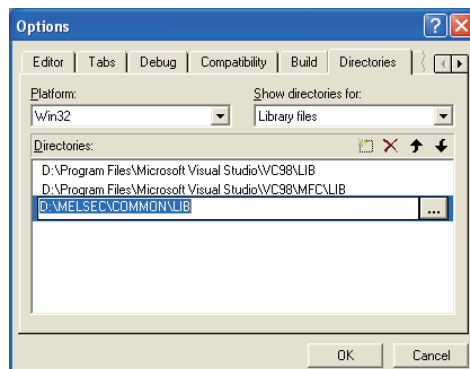
Using Visual C++ 5.0 or Visual C++ 6.0

Setting include files



1. Start Visual C++® and select [Tools] ⇒ [Options].
2. Select the [Directories] tab and specify "Include files" for "Show directories for."
3. Double-click the item to be set and browse the folder containing include files.
 - "Mdfunc.h" is stored in 'COMMON\INCLUDE' in the folder where the software package has been installed.
4. Add #include<Mdfunc.h> at the beginning of the program.

Setting library files



1. Start Visual C++ and select [Tools] ⇒ [Options].
2. Select the [Directories] tab and specify "Library files" for "Show directories for."
3. Double-click the item to be set and browse the folder containing library files.
 - "MdFunc32.lib" is stored in 'COMMON\LIB' in the folder where the software package has been installed.
4. Open the workspace to create a user application and select [Project] ⇒ [Settings].
5. Select the [Link] tab and specify "General" for the category, and enter "MdFunc32.lib" in the "Object/library modules" field.

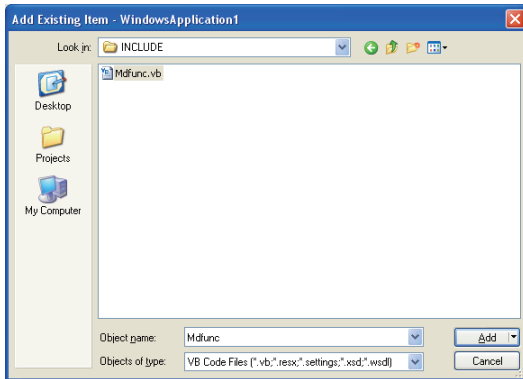
Using Visual Basic.NET

The setting operation when using Visual Basic in Visual Studio or Visual Studio .NET 2003 is shown below.



The screens of Visual Studio 2008 Visual Basic are used for the explanation in this section.
These screens are slightly different from other Visual Basic.

2



1. Start Visual Basic and select the [Add Existing Item] menu.
 - For Visual Basic other than that in Visual Studio .NET 2003: Select [Project] ⇒ [Add Existing Item].
 - For Visual Basic in Visual Studio .NET 2003: Select [File] ⇒ [Add Existing Item].
2. Select "Mdfunc.vb" on the "Add Existing Item" screen.
 - "Mdfunc.vb" is stored in 'COMMON\INCLUDE' or 'BD\Common\INCLUDE' in the folder where the software package has been installed.



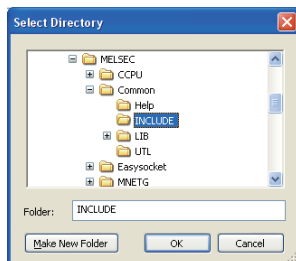
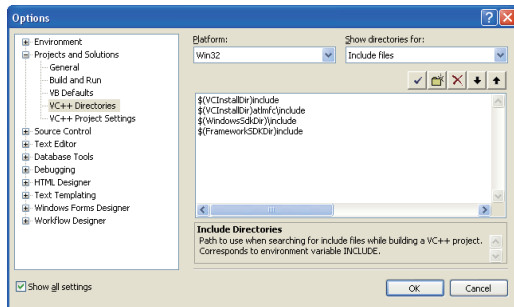
When selecting "Add As Link" from the sub menu of the [Add] button in the "Add Existing Item" screen, this procedure is not required again even if the added file is updated.

Using Visual C++ .NET 2003, Visual C++ 2005, or Visual C++ 2008



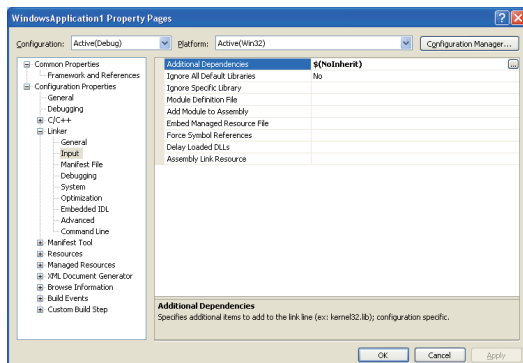
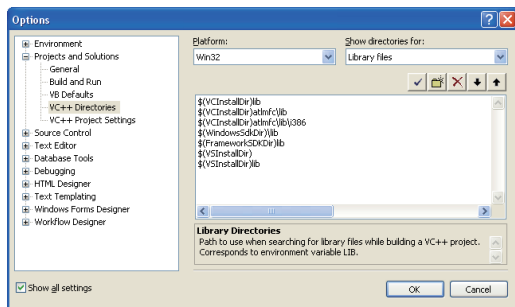
The screens of Visual C++ in Visual Studio 2008 are used for the explanation in this section.
These screens are slightly different from other Visual C++.

Setting include files



1. Start Visual C++ and select [Tools] ⇒ [Options].
2. Select "VC++ Directories" in the folder area of the "Options" screen.
 - For other than Visual C++ .NET 2003: Select "Projects and Solutions" ⇒ "VC++ Directories."
 - For Visual C++ .NET 2003: Select "Projects" ⇒ "VC++ Directories."
3. Specify "Include Files" for "Show directories for" and click the  button.
4. Browse the folder containing include files.
 - "Mdfunc.h" is stored in 'COMMON\INCLUDE' in the folder where the software package has been installed.
5. Add #include<Mdfunc.h> at the beginning of the program.

Setting library files



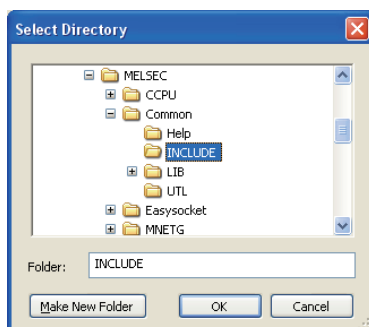
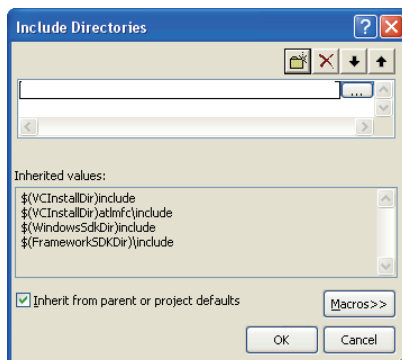
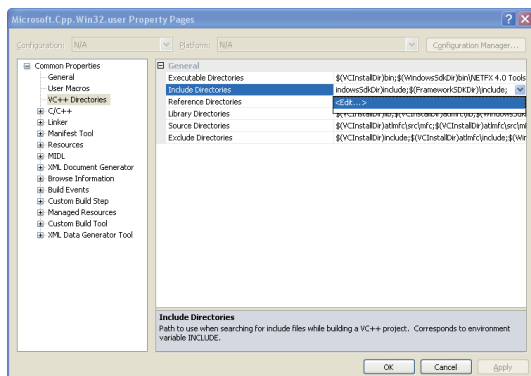
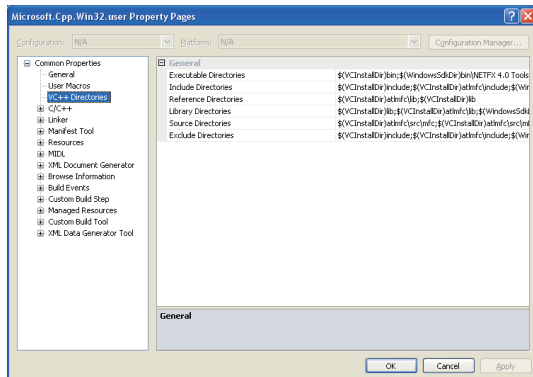
1. Start Visual C++ and select [Tools] ⇒ [Options].
2. Select "VC++ Directories" in the folder area of the "Options" screen.
 - For other than Visual C++ .NET 2003: Select "Projects and Solutions" ⇒ "VC++ Directories."
 - For Visual C++ .NET 2003: Select "Projects" ⇒ "VC++ Directories."
3. Specify "Library files" for "Show directories for" and click the  button.
4. Browse the folder containing library files.
 - For creating a 32-bit version user application: "MdFunc32.lib" is stored in 'COMMON\LIB' in the folder where the software package has been installed.
 - For creating a 64-bit version user application: "MdFunc32.lib" is stored in 'COMMON\LIB\x64' in the folder where the software package has been installed.
5. Open a project to create a user application and select [Project] ⇒ [project Properties].
6. Select "Configuration Properties" ⇒ "Linker" ⇒ "Input" in the folder area of the "Property Pages" screen. Enter "MdFunc32.lib" in the "Additional Dependencies" field.

Using Visual C++ in Visual Studio 2010 or Visual Studio 2012



The screens of Visual C++ in Visual Studio 2010 are used for the explanation in this section.
These screens are slightly different from other Visual C++.

Setting include files



1. Open the project to create a user application and select [View] ⇒ [Property Manager]*¹.

*¹ The menu configuration may differ depending on the development setting or customized setting.

2. Right-click the user property sheet, and select [Properties] from the shortcut menu.

The "Property Pages" screen appears.

3. Select "Common Properties" ⇒ "VC++ Directories."

4. Select "Include Directories." Click the  button and select "<Edit...>."

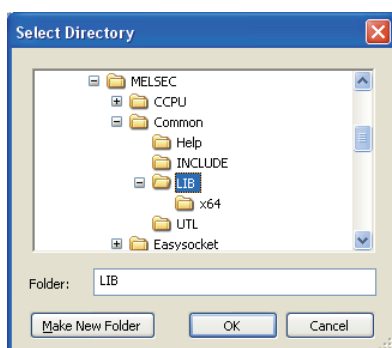
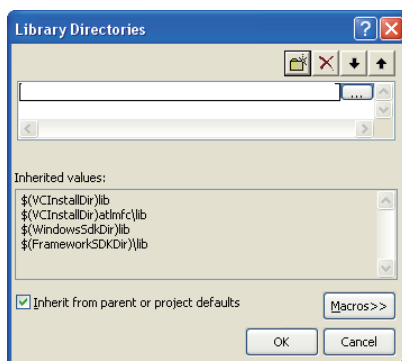
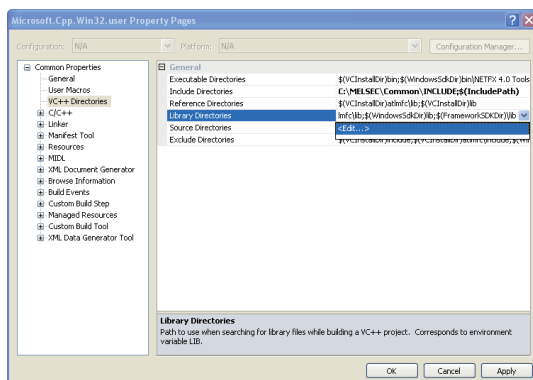
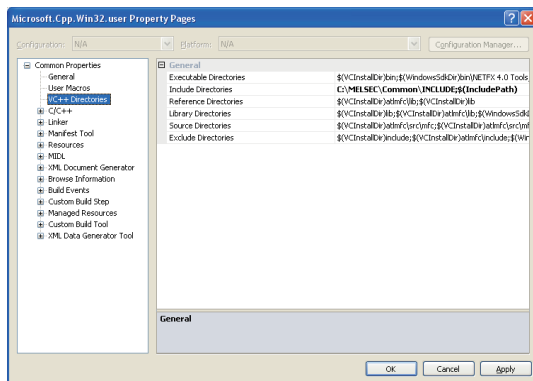
5. On the "Include Directories" screen, click the  button, and then click .

6. On the "Select Directory" screen, select the folder to which the include file is stored.

• "Mdfunc.h" is stored in 'COMMON\INCLUDE' in the folder where the software package has been installed.

7. Add #include<Mdfunc.h> at the beginning of the program.

Setting library files



1. Open the project to create a user application and select [View] ⇒ [Property Manager]^{*1}.

^{*1} The menu configuration may differ depending on the development setting or customized setting.

2. Right-click the user property sheet, and select [Properties] from the shortcut menu.

The "Property Pages" screen appears.

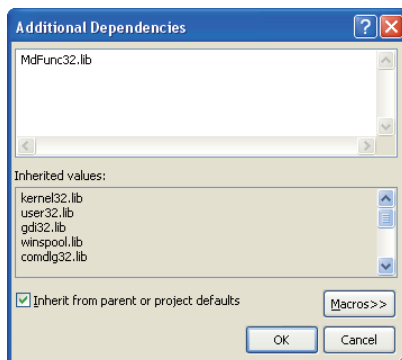
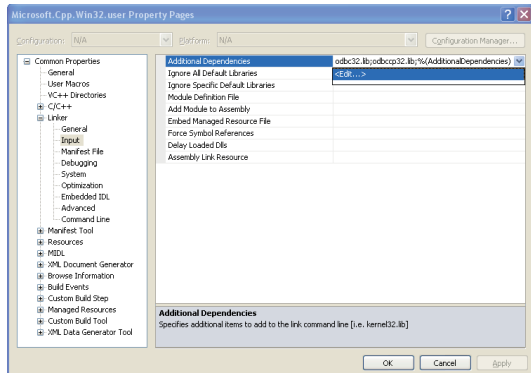
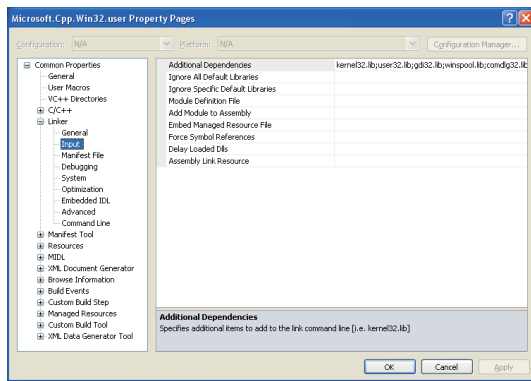
3. Select "Common Properties" ⇒ "VC++ Directories."

4. Select "Library Directories." Click the  button and select "<Edit...>."

5. On the "Library Directories" screen, click the  button, and then click .

6. On the "Select Directory" screen, select the folder to which the library file is stored.

- For creating a 32-bit version user application: "MdFunc32.lib" is stored in 'COMMON\LIB' in the folder where the software package has been installed.
- For creating a 64-bit version user application: "MdFunc32.lib" is stored in 'COMMON\LIB\x64' in the folder where the software package has been installed.



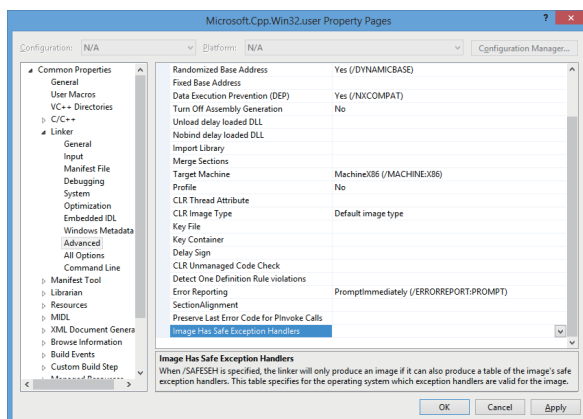
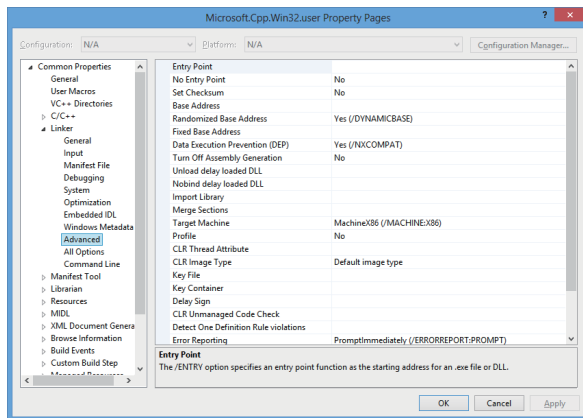
7. Open a project to create a user application and select [Project] ⇒ [project Properties].
8. In the folder area of the "Property Pages" screen, select "Configuration Properties" ⇒ "Linker" ⇒ "Input".

9. Select "Additional Dependencies." Click the  button and select "<Edit...>."

10. Enter "MdFunc32.lib" in the "Additional Dependencies" screen.

Procedure to delete /SAFESEH (Image has Safe Exception Handlers) option

When /SAFESEH (Image has Safe Exception Handlers) option is set to the linker option, the project cannot be built normally. For Visual Studio 2012 Visual C++, delete the /SAFESEH option following the procedure shown below because it is set as a default.



1. Open the project to create a user application and select [View] ⇒ [Property Manager]^{*1}.

^{*1} The menu configuration may differ depending on the development setting or customized setting.

2. Right-click the user property sheet, and select [Properties] from the shortcut menu.

The "Property Pages" screen appears.

3. Select "Common Properties" ⇒ "Linker" ⇒ "Advanced."

4. Select "Image Has Safe Exception Handlers." When the option has been set, delete it.

Using Visual C++ in Visual Studio 2013 or later

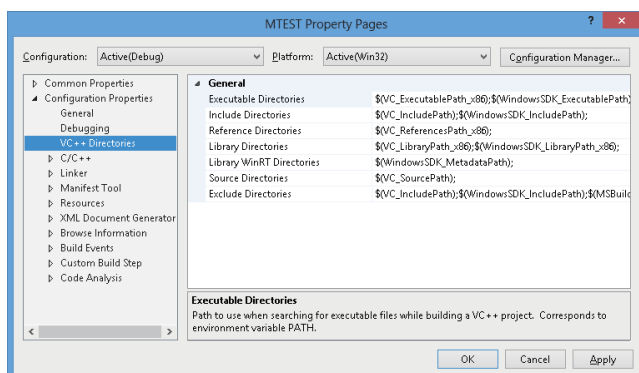
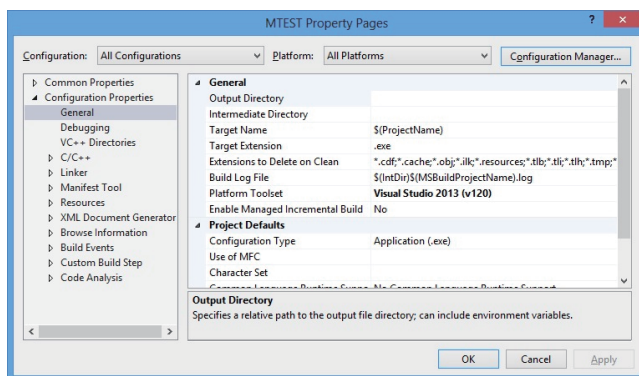
The setting operations when using Visual C++ in the following Visual Studio 2013 or later are shown below.

- Visual Studio 2013
- Visual Studio 2015
- Visual Studio 2017
- Visual Studio 2019
- Visual Studio 2022

Point

- The screens of Visual C++ 2013 opened by converting the sample program "MTEST(VC)" on Windows 8 Professional (x64) are used for the explanation in this section.
- If Visual Studio other than Visual Studio 2013 is installed by default, some necessary functions are not installed; an error may occur in the project conversion of a sample program. Customize the installation by selecting "Microsoft Foundation Classes for C++". For the method to select a function during the installation process or to add a function after the installation, refer to the Microsoft website.

Setting include files



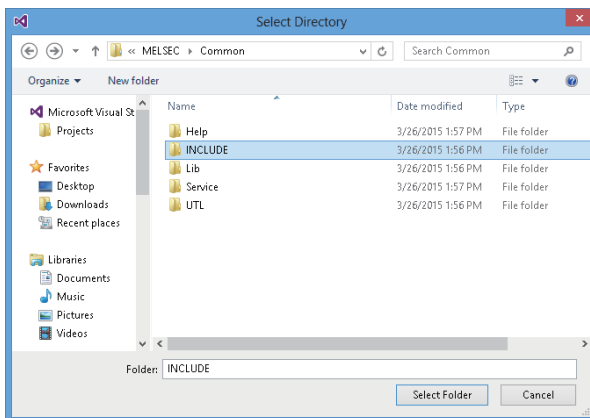
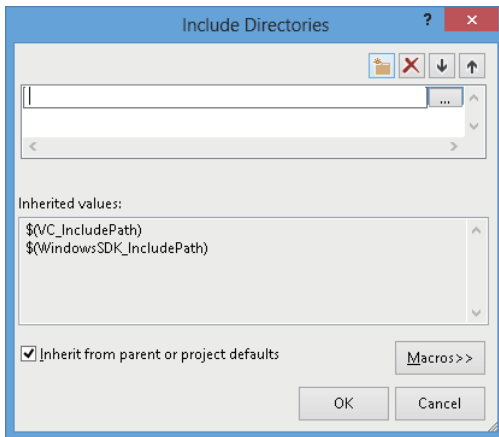
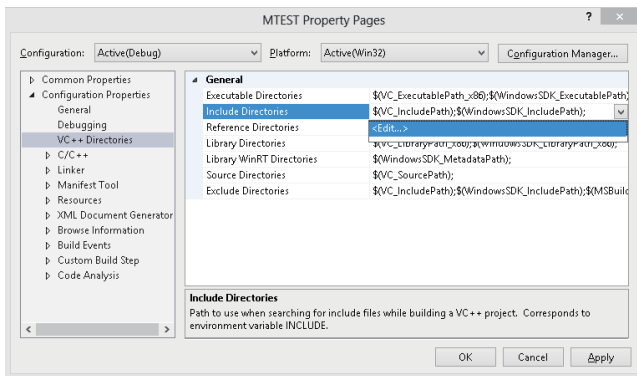
1. Open the project to create a user application and select [VIEW] ⇒ [Solution Explorer].
2. Right-click the project in the Solution Explorer, and select [Properties] from the shortcut menu.

The "Property Pages" screen appears.

3. Select the configuration and the platform to be changed in the Configuration and the Platform.

If there are multiple configurations, select the following option to change settings at a time.

- Configuration: "All Configurations"
 - Platform: "All Platforms"
4. Select [Configuration Properties] ⇒ "VC++ Directories."



5. Select "Include Directories." Click the  button and select "<Edit...>."

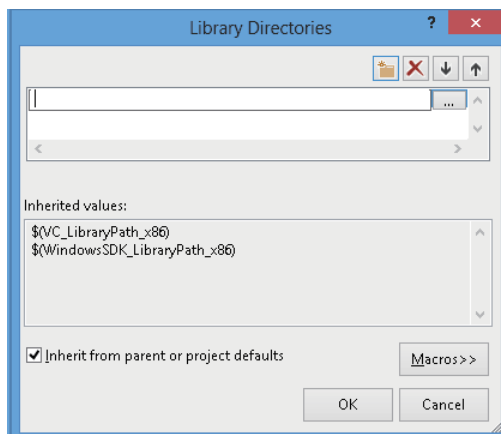
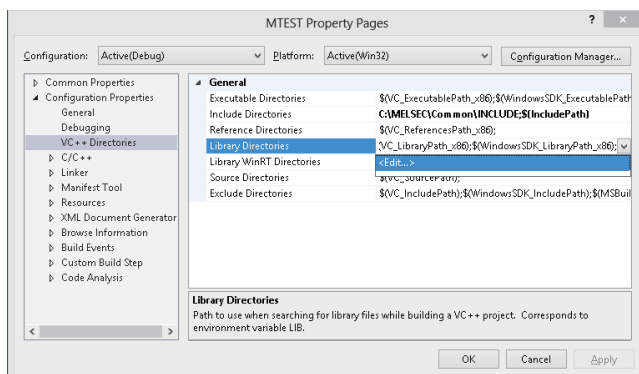
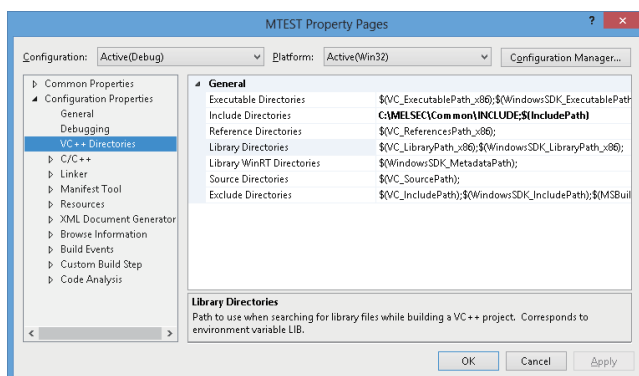
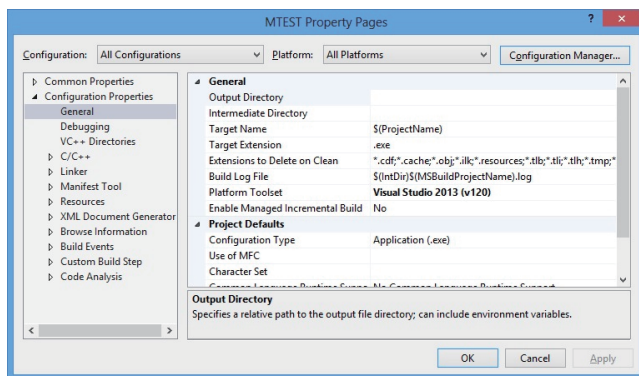
6. On the "Include Directories" screen, click the  button, and then click .

7. On the "Select Directory" screen, select the folder to which the include file is stored.

- "Mdfunc.h" is stored in 'COMMON\INCLUDE' or 'BD\Common\INCLUDE' in the folder where the software package has been installed.

8. Add `#include<Mdfunc.h>` at the beginning of the program.

Setting library files



1. Open the project to create a user application and select [VIEW] ⇒ [Solution Explorer].

2. Right-click the project in the Solution Explorer, and select [Properties] from the shortcut menu.

The "Property Pages" screen appears.

3. Select the configuration and the platform to be changed in the Configuration and the Platform.

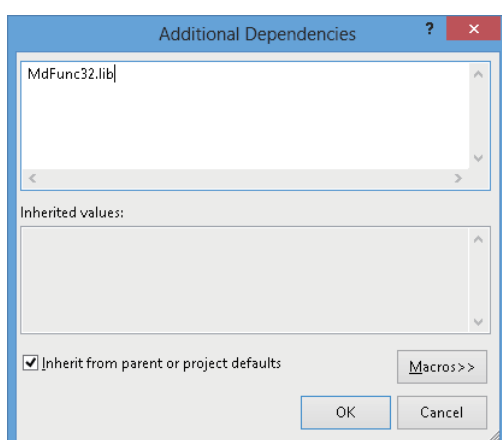
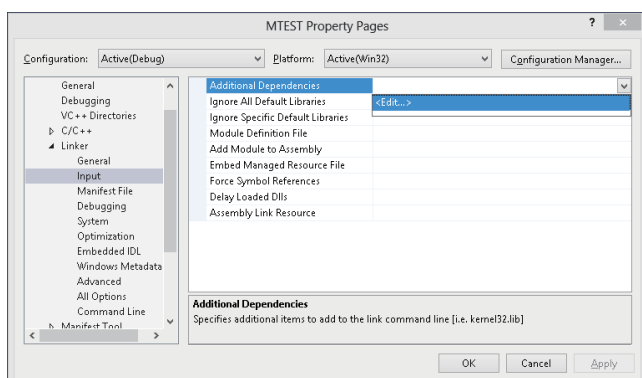
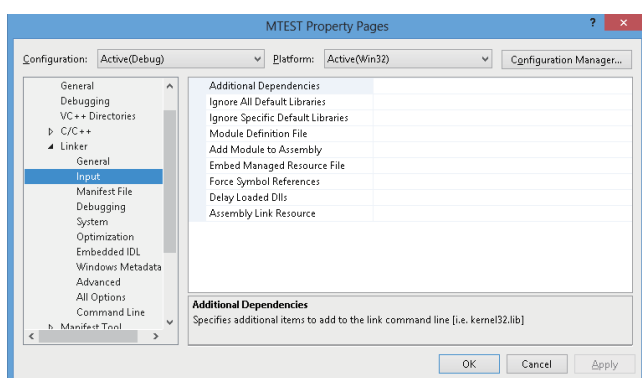
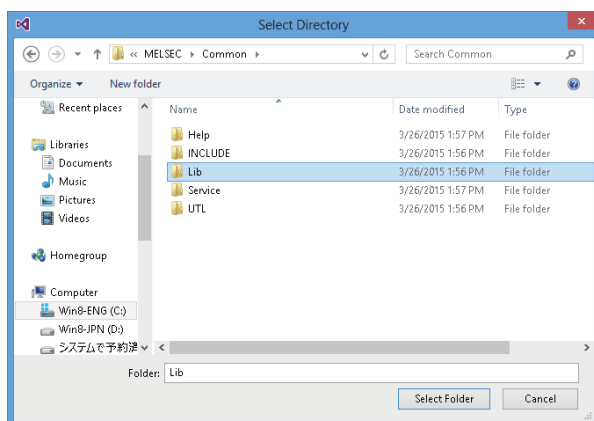
If there are multiple configurations, select the following option to change settings at a time.

- Configuration: "All Configurations"
- Platform: "All Platforms"

4. Select [Configuration Properties] ⇒ "VC++ Directories."

5. Select "Library Directories." Click the  button and select "<Edit...>."

6. On the "Library Directories" screen, click the  button, and then click .



7. On the "Select Directory" screen, select the folder to which the library file is stored.

- For creating a 32-bit version user application: "MdFunc32.lib" is stored in 'COMMON\LIB' or 'BD\Common\LIB' in the folder where the software package has been installed.
- For creating a 64-bit version user application: "MdFunc32.lib" is stored in 'COMMON\LIB\x64' or 'BD\Common\LIB\x64' in the folder where the software package has been installed.

8. Open a project to create a user application and select [Project] ⇒ [project Properties].

The "Property Pages" screen appears.

9. Select "Configuration Properties" ⇒ "Linker" ⇒ "Input."

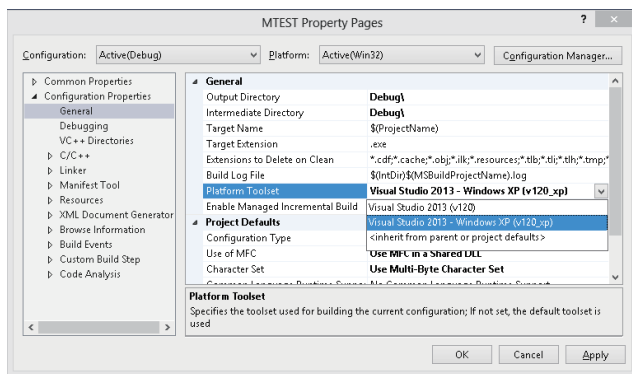
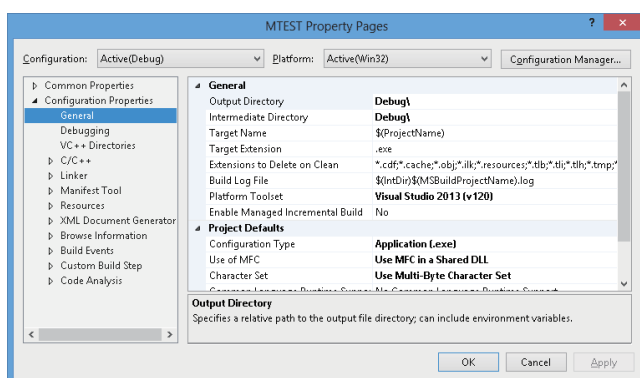
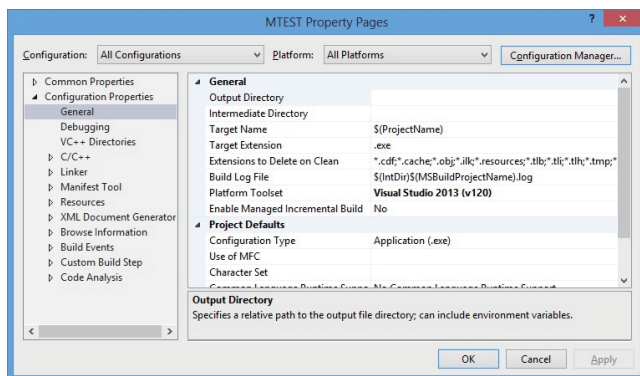
10. Select "Additional Dependencies." Click the  button and select "<Edit...>."

11. Enter "MdFunc32.lib" in the "Additional Dependencies" screen.

Setting platform and deleting option that are unnecessary

Set the platform by the following operations 1) to 5).

Also, delete unnecessary options by the following operations 6) to 9).



1. Open the project to create a user application and select [VIEW] ⇒ [Solution Explorer].

2. Right-click the project in the Solution Explorer, and select [Properties] from the shortcut menu.

The "Property Pages" screen appears.

3. Select the configuration and the platform to be changed in the Configuration and the Platform.

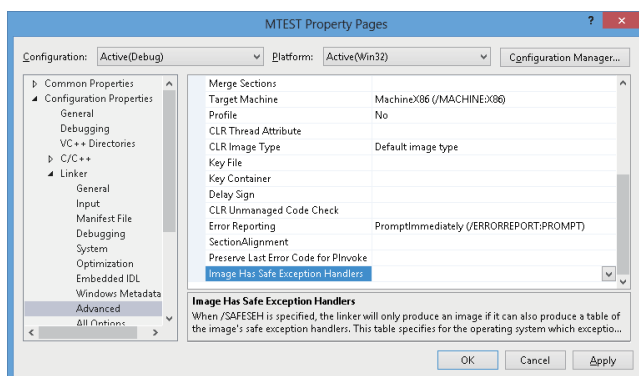
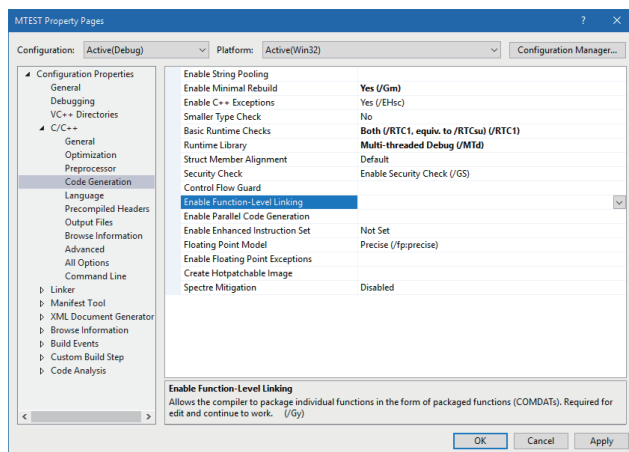
If there are multiple configurations, select the following option to change settings at a time.

- Configuration: "All Configurations"
- Platform: "All Platforms"

4. Select [Configuration Properties] ⇒ [General].

5. When configuring a program for Windows XP, select "Platform Toolset," click the  button, and select any of the following:

- "Visual Studio 2013 - Windows XP (v120_xp)"
- "Visual Studio 2015 - Windows XP (v140_xp)"
- "Visual Studio 2017 - Windows XP (v141_xp)"



6. Select "Configuration Properties" ⇒ "C/C++" ⇒ "Code Generation."^{*1}

^{*1} The screen shown on the left is the screen of Windows Server 2017.

7. Check that the option is not set for "Enable Function-Level Linking." When the option has been set, delete it.

If this option has been set, the command-line error 'D8016' may occur when a sample program is updated.

8. Select "Configuration Properties" ⇒ "Linker" ⇒ "Advanced."

9. Check that no option is set for "Image Has Safe Exception Handlers." When an option has been set, delete it.

When using "Image Has Safe Exception Handlers" option, the project cannot be built normally.

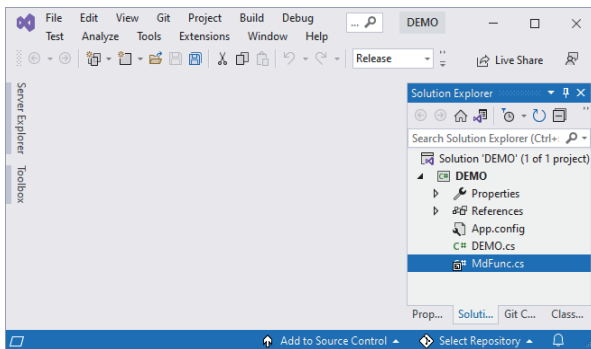
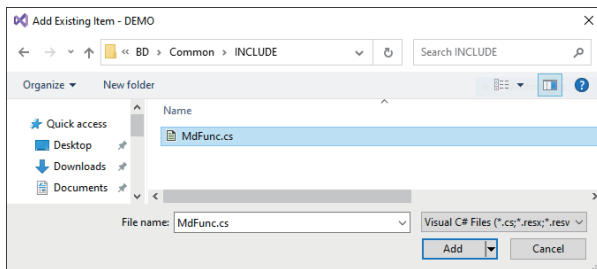
Using Visual C#

The setting operation when using Visual C#® in the following Visual Studio is shown below.

- Visual Studio 2017
- Visual Studio 2019
- Visual Studio 2022

Point

The screen opened by using Visual Studio 2022 and converting the sample program "DEMO" is used for the explanation in this section.



Point

When selecting "Add As Link" from the sub menu of the [Add] button in the "Add Existing Item" screen, this procedure is not required again even if the added file is updated.

1. Open the project to create a user application and select [VIEW] ⇒ [Solution Explorer].
2. Right-click the project in the Solution Explorer, and select [Add] ⇒ [Existing Item] from the shortcut menu. The "Add Existing Item" screen appears.
3. Select "Mdfunc.cs" on the "Add Existing Item" screen.
 - "Mdfunc.cs" is stored in 'COMMON\INCLUDE' or 'BD\Common\INCLUDE' in the folder where the software package has been installed.
4. Check that the selected file is displayed in the "Solution Explorer" screen.

3 ACCESSIBLE DEVICES AND RANGES

This chapter explains the devices and the ranges that can be accessed when communicating with each type of boards.

3.1 Access Target

The following table shows the accessible other stations.

CC-Link IE TSN board

Item			Model name*1
MELSEC iQ-R series	CPU module	RCPU	■Programmable controller CPU R00CPU, R01CPU, R02CPU, R04CPU, R04ENCPU, R08CPU, R08ENCPU, R16CPU, R16ENCPU, R32CPU, R32ENCPU, R120CPU, R120ENCPU ■Process CPU R08PCPU, R16PCPU, R32PCPU, R120PCPU ■SIL2 process CPU R08PSFCPU, R16PSFCPU, R32PSFCPU, R120PSFCPU ■Safety CPU R08SFCPU, R16SFCPU, R32SFCPU, R120SFCPU ■C Controller module R12CCPU-V ■MELSECWinCPU module R102WCPU-W
	Remote head module		RJ72GF15-T2
MELSEC-Q series	CPU module	QCPU (Q mode)	■Universal model QCPU Q00UCPU, Q00UJCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q03UDECPU, Q03UDVCPU, Q04UDHCPU, Q04UDEHCPU, Q04UDVCPU, Q04UDPVCPU, Q06UDHCPU, Q06UDEHCPU, Q06UDVCPU, Q06UDPVCPU, Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDEHCPU, Q13UDVCPU, Q13UDPVCPU, Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDEHCPU, Q26UDVCPU, Q26UDPVCPU, Q50UDEHCPU, Q100UDEHCPU
MELSEC-L series	CPU module	LCPU	L02SCPU, L02SCPU-P, L02CPU, L02CPU-P, L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT
	Head module		LJ72GF15-T2
Ethernet adapter module			NZ2GF-ETB
CC-Link IE TSN remote station manufactured by Mitsubishi Electric			NZ2GN2S1-16D and other CC-Link IE TSN remote stations manufactured by Mitsubishi Electric which have buffer memory
Network interface board	CC-Link IE TSN board		NZ81GN11-SX, NZ81GN11-T2
	CC-Link IE Controller Network board		Q80BD-J71GP21-SX, Q80BD-J71GP21S-SX, Q81BD-J71GP21-SX, Q81BD-J71GP21S-SX
	CC-Link IE Field Network board		Q80BD-J71GF11-T2, Q81BD-J71GF11-T2

*1 If the model name or version of a product is not supported by each network, the product cannot be accessed.
For the supported network of each product, refer to the manual of product.

Point

When accessing MELSEC-L series, Ethernet adapter modules, CC-Link IE Controller Network boards, or CC-Link IE Field Network boards, use a 32-bit version user application.

Other boards

Item			Model name*1
MELSEC iQ-R series	CPU module	RCPU	R04CPU, R04ENCPU, R08CPU, R08ENCPU, R16CPU, R16ENCPU, R32CPU, R32ENCPU, R120CPU, R120ENCPU
MELSEC-Q series	CPU module	QCPU (Q mode)	■Basic model QCPU Q00JCPU, Q00CPU, Q01CPU ■High performance model QCPU Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU ■Process CPU Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU ■Redundant CPU Q12PRHCPU, Q25PRHCPU ■Universal model QCPU Q00UCPU, Q00JUCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q03UDECPU, Q03UDVCPU, Q04UDHCPU, Q04UDEHCPU, Q04UDVCPU, Q04UDPVCPU, Q06UDHCPU, Q06UDEHCPU, Q06UDVCPU, Q06UDPVCPU, Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDEHCPU, Q13UDVCPU, Q13UDPVCPU, Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDEHCPU, Q26UDVCPU, Q26UDPVCPU, Q50UDEHCPU, Q100UDEHCPU
MELSEC-L series	CPU module	LCPU	L02SCPU, L02SCPU-P, L02CPU, L02CPU-P, L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT
	Head module		LJ72GF15-T2
MELSEC-A series	CPU module	ACPU*2	A0J2HCPU, A1SCPU, A1SJCPU, A1SHCPU, A1SJHCPU, A1NCP, A2CCPU, A2CJCPU, A2NCP, A2NCP-S1, A2SCPU, A2SHCPU, A3NCP, A2ACPU, A2ACPU-S1, A3ACPU, A2UCPU, A2UCPU-S1, A2USCPU, A2USCPU-S1, A2USHCPU-S1, A3UCPU, A4UCPU
		QCPU (A mode)	Q02CPU-A, Q02HCPU-A, Q06HCPU-A
MELSEC-QnA series	CPU module	QnACPU	Q2ACPU, Q2ASCPU, Q2ASHCPU, Q2ACPU-S1, Q2ASCPU-S1, Q2ASHCPU-S1, Q3ACPU, Q4ACPU, Q4ARCPU
Intelligent device station			AJ65BT-R2 and other slave stations that can perform transient transmission on a CC-Link system
Ethernet adapter module			NZ2GF-ETB
CC-Link IE Field Network remote device station			NZ2GF2B1-16D and other CC-Link IE Field Network remote device stations manufactured by Mitsubishi Electric which have buffer memory
Network interface board	CC-Link IE Controller Network board		Q80BD-J71GP21-SX, Q80BD-J71GP21S-SX, Q81BD-J71GP21-SX, Q81BD-J71GP21S-SX
	CC-Link IE Field Network board		Q80BD-J71GF11-T2, Q81BD-J71GF11-T2
	CC-Link Ver.2 board		Q80BD-J61BT11N, Q81BD-J61BT11
	MELSECNET/H board		Q80BD-J71LP21-25, Q81BD-J71LP21-25, Q80BD-J71LP21S-25, Q80BD-J71LP21G, Q80BD-J71LP21GE, Q80BD-J71BR11

*1 If the model name or version of a product is not supported by each network, the product cannot be accessed.
For the supported network of each product, refer to the manual of product.

*2 For CC-Link IE Field network board, only A2UCPU, A2UCPU-S1, A2USCPU, A2USCPU-S1, A2USHCPU-S1, A3UCPU, and A4UCPU can be accessed.



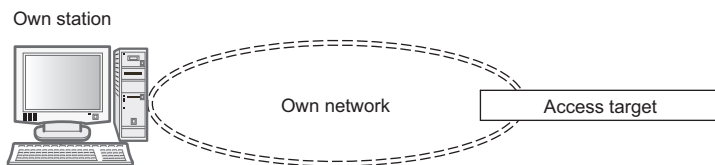
When accessing QCPU (Q mode), RCP, or CC-Link IE Field Network remote device station, a 64-bit version user application can be used. When accessing other than one of these devices, use a 32-bit version user application.

3.2 Accessible Ranges

This section explains accessible ranges when communicating with boards.

Access target on own network

The following module, board, or own station can be accessed on the network connected to each board.



○: Accessible, —: Not accessible

Access target ^{*1}		Own network					
		CC-Link IE TSN	CC-Link IE Controller Network	CC-Link IE Field Network	CC-Link ^{*2}	MELSEC NET/H	MELSEC NET/10
Network interface board	CC-Link IE TSN board	○	—	—	—	—	—
	CC-Link IE Controller Network board	—	○	—	—	—	—
	CC-Link IE Field Network board	—	—	○	—	—	—
	CC-Link Ver.2 board	—	—	—	○	—	—
	MELSECNET/H board	—	—	—	—	○	○
MELSEC iQ-R	RCPU						
	Programmable controller CPU	○	○	○	○	○	—
	Process CPU, SIL2 process CPU, safety CPU	○	—	—	—	—	—
	C Controller module	○	—	—	—	—	—
	MELSECWinCPU module	○	—	—	—	—	—
MELSEC-Q	QCPU (Q mode)	—	○	○	○	○	○
MELSEC-L	LCPU	—	—	○	○	—	—
	Head module (LJ72GF15-T2)	—	—	○	○	—	—
MELSEC-QnA	QnACPU	—	—	—	○	—	○
MELSEC-A	ACPU, QCPU (A mode)	—	—	—	○	—	○
Intelligent device station (AJ65BT-R2, etc.)		—	—	—	○	—	—
Ethernet adapter module (NZ2GF-ETB)		—	—	○	—	—	—
CC-Link IE Field Network remote device station (NZ2GF2B1-16D, etc.)		—	—	○	—	—	—
CC-Link IE TSN remote station manufactured by Mitsubishi Electric (NZ2GN2S1-16D, etc.)		○	—	—	—	—	—

*1 Page 31 Access Target

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

For the accessible devices of each access target, refer to the following:

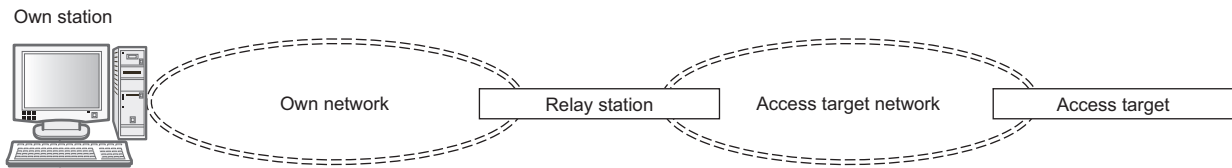
Access target		Reference
Access to link devices and buffer memory of the own station	Network interface board	Page 36 Access to link devices and buffer memory of the own station
Access to devices of other station module	RCPU	Page 37 Access to devices of MELSEC iQ-R series module
	QCPU (Q mode), LCP, QnACPU	Page 39 Access to devices of MELSEC-Q/L/QnA series module
	ACPU, QCPU (A mode)	Page 40 Access to devices of MELSEC-A series module
	Ethernet adapter module, head module	Page 41 Access to an Ethernet adapter module and head module
Access to other station buffer memory of CC-Link IE	CC-Link IE Field Network remote device station, CC-Link IE TSN remote station	Page 41 Access to a remote station and remote device station
Access to other station buffer memory of CC-Link	CC-Link network module, CC-Link Ver.2 board, etc.	Page 41 Access to other station buffer memory of CC-Link
Access using the SEND function or the RECV function	QnACPU, QCPU (Q mode), LCP, RCP, network interface board	Page 42 Access using the SEND function or the RECV function

Access target when connected via network

The following shows the accessibility when accessing via multiple networks.

The combination other than shown in the following table cannot be accessed.

Accessing the access target via multiple networks is not supported by CC-Link network.



Numbers in the tables indicate devices in the following table:

No.	Access target*1		
1	Network interface board		
2	MELSEC iQ-R	RCPU	Programmable controller CPU
3			Process CPU, SIL2 process CPU, safety CPU
4			C Controller module
5			MELSECWinCPU module
6			Remote head module (RJ72GF15-T2)
7	MELSEC-Q	QCPU (Q mode)	Universal model QCPU
8			Basic model QCPU, high performance model QCPU, process CPU, redundant CPU
9	MELSEC-L, etc.	LCPU	
10		Head module (LJ72GF15-T2)	
11		Ethernet adapter module (NZ2GF-ETB)	
12	MELSEC-A	ACPU	
		QCPU (A mode)	
13	MELSEC-QnA	QnACPU	

*1 Page 31 Access Target

○: Accessible, —: Not accessible

Own network	Relay station	Access target network	Access target										
			Board	MELSEC iQ-R					MELSEC-Q		MELSEC-L, etc.	MELSEC-A MELSEC-QnA	
			1	2	3	4	5	6	7	8	9, 10, 11	12, 13	
CC-Link IE TSN	RCPU ^{*1}	CC-Link IE TSN	○	○	○	○	○	—	—	—	—	—	
		CC-Link IE Controller Network	○	○	○	○	○	—	○	—	—	—	
		CC-Link IE Field Network	○	○	○	○	—	○	○	—	○	—	
CC-Link IE Controller Network, CC-Link IE Field Network	RCPU ^{*1}	CC-Link IE TSN	○	○	—	—	—	—	—	—	—	—	
		CC-Link IE Controller Network	○	○	—	—	—	—	○	○	—	—	
		CC-Link IE Field Network	○	○	—	—	—	—	○	○	○	—	
	QCPU (Q mode) ^{*2}	CC-Link IE Controller Network	○	○	—	—	—	—	○	○	—	—	
		CC-Link IE Field Network	○	○	—	—	—	—	○	○	○	—	
		MELSECNET/H	○	—	—	—	—	—	○	○	—	—	
		MELSECNET/10	○	—	—	—	—	—	○	○	—	○ ^{*3}	
CC-Link IE Field Network	LCPU	CC-Link IE Field Network	○	○	—	—	—	—	○	○	○	—	
MELSECNET/H, MELSECNET/10	QCPU (Q mode) ^{*2}	CC-Link IE Controller Network	○	—	—	—	—	—	○	○	—	—	
		CC-Link IE Field Network	○	—	—	—	—	—	○	○	○	—	
		MELSECNET/H	○	—	—	—	—	—	○	○	—	—	
		MELSECNET/10	○	—	—	—	—	—	○	○	—	○	
MELSECNET/10	QnACPU, ACPU, QCPU (A mode)	MELSECNET/10	○	—	—	—	—	—	○	○	—	○	

- *1 C Controller modules and MELSECWinCPU modules cannot be set as relay stations.
- *2 When using a CPU module whose number of mountable network module is one, the CPU module cannot be set as a relay station.
- *3 For CC-Link IE Field network board, A2UCPU, A2UCPU-S1, A2USCPU, A2USCPU-S1, A2USHCPU-S1, A3UCPU, and A4UCPU can be accessed.

For the accessible devices of each access target, refer to the following:

Access target		Reference
Access to devices of other station module	RCPU	Page 37 Access to devices of MELSEC iQ-R series module
	QCPU (Q mode), LCPU, QnACPU	Page 39 Access to devices of MELSEC-Q/L/QnA series module
	ACPU, QCPU (A mode)	Page 40 Access to devices of MELSEC-A series module
	Ethernet adapter module, head module, remote head module	Page 41 Access to an Ethernet adapter module and head module
Access using the SEND function or the RECV function	RCPU, QCPU (Q mode), LCPU, QnACPU, network interface board	Page 42 Access using the SEND function or the RECV function

3.3 Accessible Devices

This section shows the accessible devices of each board.

The table in this section divides the functions used for access into "Batch" and "Random" and indicates accessibility for each of these two categories.

Access type	Description
Batch	Batch write (mdSend, mdSendEx)
	Batch read (mdReceive, mdReceiveEx)
Random	Random write (mdRandW, mdRandWEx)
	Random read (mdRandR, mdRandREx)
	Bit set (mdDevSet, mdDevSetEx)
	Bit reset (mdDevRst, mdDevRstEx)

Devices not listed in the tables in this section and those not included in target modules cannot be accessed.

Access to link devices and buffer memory of the own station

The following table shows the accessible devices when accessing a board (own station).

○: Accessible, —: Not accessible

Device		Access type	Access target	
			CC-Link Ver.2 board	MELSECNET/H board
Link input	LX	Batch/Random	—	○
Link output	LY	Batch/Random	—	○
Link relay	LB	Batch/Random	—	○
Link register	LW	Batch/Random	—	○
Link special relay	SB	Batch/Random	○	○
Link special register	SW	Batch/Random	○	○
Remote input	RX	Batch/Random	○	—
Remote output	RY	Batch/Random	○	—
Remote register (for transmission)	RWw	Batch/Random	○	—
Remote register (for reception)	RWr	Batch/Random	○	—
Buffer memory	—	Batch/Random	○	—
Random access buffer	—	Batch/Random	○	—

Device		Access type	Access target		
			CC-Link IE TSN board	CC-Link IE Controller Network board	CC-Link IE Field Network board
Link input	LX	Batch/Random	—	○	—
Link output	LY	Batch/Random	—	○	—
Link relay	LB	Batch/Random	○	○	—
Link register	LW	Batch/Random	○	○	—
Link special relay	SB	Batch/Random	○	○	○
Link special register	SW	Batch/Random	○	○	○
Remote input	RX	Batch/Random	○	—	○
Remote output	RY	Batch/Random	○	—	○
Remote register (for transmission)	RWw	Batch/Random	○	—	○
Remote register (for reception)	RWr	Batch/Random	○	—	○
Buffer memory	—	Batch/Random	○	○	○
Random access buffer	—	Batch/Random	—	—	—

Access to devices of MELSEC iQ-R series module

The following table shows the accessible devices of MELSEC iQ-R series CPU modules controlling other station module.

Programmable controller CPU, process CPU, SIL2 process CPU, safety CPU

○: Accessible, —: Not accessible

Device			Access type	Access target ^{*1}
				RCPU
Input		X	Batch/Random	○
Output		Y	Batch/Random	○
Internal relay		M	Batch/Random	○
Latch relay		L	Batch/Random	○
Link relay		B	Batch/Random	○
Annunciator		F	Batch/Random	○
Link special relay		SB	Batch/Random	○
Edge relay		V	Batch/Random	○
Timer	contact, coil, current value	T	Batch/Random	○
Retentive timer	contact, coil, current value	ST	Batch/Random	○
Long timer	contact, coil, current value	LT	Batch/Random	○ ^{*2}
Long retentive timer	contact, coil, current value	LST	Batch/Random	○ ^{*2}
Counter	contact, coil, current value	C	Batch/Random	○
Long counter	contact, coil, current value	LC	Batch/Random	○ ^{*2}
Data register		D	Batch/Random	○
Link register		W	Batch/Random	○
Link special register		SW	Batch/Random	○
Special relay		SM	Batch/Random	○
Special register		SD	Batch/Random	○
Link direct device	Link input Link output Link relay Link special relay Link register Link special register	Jn\X Jn\Y Jn\B Jn\SB Jn\W Jn\SW	Batch/Random	○
Module access device		Un\G	Batch/Random	○
Index register		Z	Batch/Random	○
Long index register		LZ	Batch/Random	○ ^{*2}
File register		R ^{*3} , ZR	Batch/Random	○
Refresh data register		RD	Batch/Random	○ ^{*2}

*1  Page 31 Access Target

*2 Cannot be accessed from MELSECNET/H board.

*3 When using the device type DevER0 to 256, a device can be accessed by specifying a block number.

C Controller module, MELSECWinCPU module

○: Accessible, —: Not accessible

Device			Access type	Access target	
				R12CCPU-V	R102WCPU-W
Input		X	Batch/Random	○	○
Output		Y	Batch/Random	○	○
Internal relay		M	Batch/Random	○	○
Link relay		B	Batch/Random	○	○
Data register		D	Batch/Random	○	○
Link register		W	Batch/Random	○	○
Special relay		SM	Batch/Random	○	○
Special register		SD	Batch/Random	○	○
Link direct device	Link input Link output Link relay Link special relay Link register Link special register	Jn\X Jn\Y Jn\B Jn\SB Jn\W Jn\SW	Batch/Random	○	○
Module access device		Un\G	Batch/Random	○	○
File register		ZR	Batch/Random	○	—

Access to devices of MELSEC-Q/L/QnA series module

The following table shows the accessible devices of QCPU (Q mode), LCP, and QnACPU controlling other station module.

○: Accessible, —: Not accessible

Device			Access type	Access target ^{*1}
				QCPU, LCPU, QnACPU
Input		X	Batch/Random	○
Output		Y	Batch/Random	○
Internal relay		M	Batch/Random	○
Latch relay		L	Batch/Random	○
Annunciator		F	Batch/Random	○
Edge relay		V	Batch/Random	○
Link relay		B	Batch/Random	○
Link special relay		SB	Batch/Random	○
Timer	contact, coil, current value	T	Batch/Random	○
Retentive timer	contact, coil, current value	ST	Batch/Random	○
Counter	contact, coil, current value	C	Batch/Random	○
Data register		D	Batch/Random	○
Link register		W	Batch/Random	○
Link special register		SW	Batch/Random	○
Special relay		SM	Batch/Random	○
Special register		SD	Batch/Random	○
Link direct device	Link input Link output Link relay Link register Link special relay Link special register	Jn\X Jn\Y Jn\B Jn\W Jn\SB Jn\SW	Batch/Random	○
Intelligent function module device		Un\G	Batch/Random	○
Index register		Z	Batch/Random	○
File register		R ^{*2} , ZR	Batch/Random	○ ^{*3}

*1  Page 31 Access Target

*2 When using the device type DevER0 to 256, a device can be accessed by specifying a block number.

*3 Cannot be accessed with Q00CPU.

Access to devices of MELSEC-A series module

The following table shows the accessible devices of ACPU or QCPU (A mode) controlling other station module.

○: Accessible, —: Not accessible

Device			Access type	Access target				
				A1NCPU* ¹	A0J2HCPU, A1SCPU, A1SJCPU, A1SHCPU, A1SJHCPU, A2CCPU, A2CJCPU, A2NCPU, A2NCPU-S1, A2SCPU, A2SHCPU, A2ACPU, A2ACPU-S1* ¹	A3NCPU, A3ACPU* ¹	A2UCPU, A2UCPU-S1, A2USCPU, A2USCPU-S1, A2USHCPU-S1, A3UCPU, QCPU (A mode)	A4UCPU
Input	X	Batch/Random	○	○	○	○	○	○
Output	Y	Batch/Random	○	○	○	○	○	○
Latch relay	L	Batch/Random	○	○	○	○	○	○
Internal relay	M	Batch/Random	○	○	○	○	○	○
Special relay	M9000 to	Batch/Random	○	○	○	○	○	○
Data register	D	Batch/Random	○	○	○	○	○	○
Special register	D9000 to	Batch/Random	○	○	○	○	○	○
Annunciator	F	Batch/Random	○	○	○	○	○	○
Timer	contact, coil, current value	T	Batch/Random	○	○	○	○	○
	setting value main		Batch	○	○	○	○	○
			Random	—	—	—	—	—
	setting value sub1		Batch	—	—	○	○	○
			Random	—	—	—	—	—
	setting value sub2, sub3		Batch	—	—	—	—	○
			Random	—	—	—	—	—
Counter	contact, coil, current value	C	Batch/Random	○	○	○	○	○
	setting value main		Batch	○	○	○	○	○
			Random	—	—	—	—	—
	setting value sub1		Batch	—	—	○	○	○
			Random	—	—	—	—	—
	setting value sub2, sub3		Batch	—	—	—	—	○
			Random	—	—	—	—	—
Accumulator	A	Batch/Random	○	○	○	○	○	○
Index register	Z, V	Batch/Random	○	○	○	○	○	○
File register Extended file register* ²	R	Batch/Random	—	○	○	○	○	○
Link relay	B	Batch/Random	○	○	○	○	○	○
Link register	W	Batch/Random	○	○	○	○	○	○

*¹ Cannot be accessed from CC-Link IE Field Network board.

*² When using the device type DevER0 to 64, a device can be accessed by specifying a block number.

Access to an Ethernet adapter module and head module

The following table shows the accessibility to an Ethernet adapter module, head module, or remote head module.

○: Accessible, —: Not accessible

Device		Access target	Access target		
			Ethernet adapter module	Head module	Remote head module
Input	X	Batch/Random	○	○	○
Output	Y	Batch/Random	○	○	○
Special relay	SM	Batch/Random	○	○	○
Special register	SD	Batch/Random	○	○	○
Link register	W	Batch/Random	○	○	○
Link special relay	SB	Batch/Random	○	○	○
Link special register	SW	Batch/Random	○	○	○
Intelligent function module device	Un/G	Batch/Random	—	○	○
Buffer memory					
Intelligent function module access device					
Refresh data register	RD	Batch/Random	—	—	○

Access to a remote station and remote device station

For a CC-Link IE TSN remote station and CC-Link IE Field Network remote device station, the buffer memory can be accessed only with the following functions.

○: Accessible, —: Not accessible

Function name	CC-Link IE TSN remote station	CC-Link IE Field Network remote device station
mdRemBufWriteEx	○	○
mdRemBufReadEx	○	○
mdRemBufWriteIPEX	○	—
mdRemBufReadIPEX	○	—

Access to other station buffer memory of CC-Link

The following table shows the accessible buffer memory of CC-Link network module and CC-Link Ver.2 board.

The multiple CPU system (when the logical station is specified) cannot be accessed.

○: Accessible, —: Not accessible

Device		Access type	Accessibility
Link special relay	SB	Batch	○
Link special register	SW	Batch	○
Remote input	RX	Batch	○
Remote output	RY	Batch	○
Remote register	RW	Batch	○
Buffer memory		Batch	○
Random access buffer		Batch	○



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

Access using the SEND function or the RECV function

The same operation as device access, the SEND function and the RECV function execute Batch write (mdSend, mdSendEx) or Batch read (mdReceive, mdReceiveEx) by specifying a device type for each function.

○: Accessible, —: Not accessible

Device	Access type	Access target	
		RCPU, QCPU (Q mode), LCPU, QnACPU	Network interface board
RECV function	Batch	—	○ (Own station)
SEND function (with arrival acknowledgment)	Batch	○	○
SEND function (without arrival acknowledgment)			

Point

- The SEND function and the RECV function are not supported by CC-Link.
- The SEND function and the RECV function are supported by SW1DNC-MNETG-B Version 1.08J or later.
- The SEND function and the RECV function are not supported by a MELSEC-A series CPU module, Ethernet adapter module, and head module.

4 FUNCTIONS

This chapter explains the MELSEC data link library functions.

4.1 Function List

The following table shows the list of the functions in the MELSEC data link library that is provided with the software package.

—: N/A

Function name	Description	Remarks	Reference
mdOpen	Opens a communication line.	—	Page 54 mdOpen (opening communication lines)
mdClose	Closes a communication line.	—	Page 55 mdClose (closing communication lines)
mdSendEx	Batch writes devices.	Extended function ^{*1}	Page 56 mdSendEx (batch writing extended devices)
	Sends data. (SEND function) ^{*2*3*4}	Extended function ^{*1}	Page 58 mdSendEx (SEND function)
mdReceiveEx	Batch reads devices.	Extended function ^{*1}	Page 61 mdReceiveEx (batch reading extended devices)
	Receives data. (RECV function) ^{*2*3*4}	Extended function ^{*1}	Page 63 mdReceiveEx (RECV function)
mdRandWEx	Writes devices randomly.	Extended function ^{*1}	Page 66 mdRandWEx (writing extended devices randomly)
mdRandREx	Reads devices randomly.	Extended function ^{*1}	Page 69 mdRandREx (reading extended devices randomly)
mdDevSetEx	Sets a bit device.	Extended function ^{*1}	Page 73 mdDevSetEx (setting extended bit devices)
mdDevRstEx	Resets a bit device.	Extended function ^{*1}	Page 74 mdDevRstEx (resetting extended bit devices)
mdRemBufWriteEx	Writes data to the buffer memory of a remote device station or remote station. ^{*3*5}	Extended function ^{*1}	Page 75 mdRemBufWriteEx (writing data to the buffer memory of a remote device station/remote station)
mdRemBufReadEx	Reads data from the buffer memory of a remote device station or remote station. ^{*3*5}	Extended function ^{*1}	Page 77 mdRemBufReadEx (reading data from the buffer memory of a remote device station/remote station)
mdRemBufWriteIPEx	Writes data to the buffer memory of a remote station. ^{*3} (Target station IP address specified)	Extended function ^{*1}	Page 79 mdRemBufWriteIPEx (writing data to the buffer memory of a remote station, target station IP address specified)
mdRemBufReadIPEx	Reads data from the buffer memory of a remote station. ^{*3} (Target station IP address specified)	Extended function ^{*1}	Page 81 mdRemBufReadIPEx (reading data from the buffer memory of a remote station, target station IP address specified)
mdTypeRead	Reads the type of programmable controller CPU.	—	Page 83 mdTypeRead (reading model names of CPU)
mdControl	Remote operation of programmable controller CPU. (RUN/STOP/PAUSE).	—	Page 87 mdControl (remote RUN/STOP/PAUSE)
mdWaitBdEvent	Waits for an event occurrence. ^{*2}	—	Page 88 mdWaitBdEvent (waiting for event occurrence)
mdBdRst	Resets the board.	—	Page 91 mdBdRst (resetting board)
mdBdModSet	Sets the mode of the board.	—	Page 92 mdBdModSet (setting modes of board)
mdBdModRead	Reads the mode of the board.	—	Page 94 mdBdModRead (reading modes of board)
mdBdLedRead	Reads the LED information of the board.	—	Page 96 mdBdLedRead (reading LED information of the board)
mdBdSwRead	Reads the switch status of the board.	—	Page 103 mdBdSwRead (reading switch status of the board)
mdBdVerRead	Reads the version information of the board.	—	Page 104 mdBdVerRead (reading version information of the board)
mdInit	Initializes programmable controller information table.	—	Page 106 mdInit (initializing programmable controller information table)

^{*1} A function in which the access range is extended according to the extension of the device points at the access target. It is accessible to all device numbers.

Use extended functions when creating a new program.

^{*2} Applicable to CC-Link IE Controller Network boards and CC-Link IE Field Network boards.

*3 Applicable to CC-Link IE TSN boards.

*4 Supported by SW1DNC-MNETG-B with version 1.08J or later for CC-Link IE Controller Network boards.

*5 Applicable to CC-Link IE Field Network boards with a serial number the first five digits of which are '15102' or higher, and SW1DNC-CCIEF-B with version 1.06G or later.

For a 64-bit version user application, use SW1DNC-CCIEF-B with version 1.12N or later.

Conventional compatible function list

Conventional compatible functions are functions used for programs created conventionally.

—: N/A

Function name	Description	Remarks	Reference
mdSend	Batch writes devices.	—	Page 107 mdSend (batch writing devices)
	Sends data. (SEND function)*1,*2	—	Page 109 mdSend (SEND function)
mdReceive	Batch reads devices.	—	Page 111 mdReceive (batch reading devices)
	Receives data. (RECV function)*1,*2	—	Page 113 mdReceive (RECV function)
mdRandW	Writes devices randomly.	—	Page 115 mdRandW (writing devices randomly)
mdRandR	Reads devices randomly.	—	Page 117 mdRandR (reading devices randomly)
mdDevSet	Sets a bit device.	—	Page 119 mdDevSet (setting bit devices)
mdDevRst	Resets a bit device.	—	Page 120 mdDevRst (resetting bit devices)

*1 Applicable to MELSECNET/H boards, CC-Link IE Controller Network boards, and CC-Link IE Field Network boards.

*2 Supported by CC-Link IE Controller Network boards with version 1.08J or later.

Point

- When using the conventional compatible functions, the accessible device numbers are from 0 to 32767.
- When using these functions for CC-Link IE TSN boards, a path error (FFFFH) occurs.

4.2 Common Specifications of Functions

This section explains the definitions of arguments commonly used with the MELSEC data link library functions.

Specifying channel number

The following table shows the channels used with the MELSEC data link library.

Channel number	Channel name	Description
51	MELSECNET/H (1 slot)	MELSECNET/H board Channel number is set with MELSECNET/H utility.
52	MELSECNET/H (2 slot)	
53	MELSECNET/H (3 slot)	
54	MELSECNET/H (4 slot)	
81	CC-Link (1 slot)	CC-Link Ver.2 board Channel number is set with the channel number setting switch. The channels are set as follows according to the SW1 and SW2 settings: • 81: OFF, OFF • 82: ON, OFF • 83: OFF, ON • 84: ON, ON
82	CC-Link (2 slot)	
83	CC-Link (3 slot)	
84	CC-Link (4 slot)	
151	CC-Link IE Controller Network (Channel No.151)	CC-Link IE Controller Network board Channel number is set with CC IE Control utility.
152	CC-Link IE Controller Network (Channel No.152)	
153	CC-Link IE Controller Network (Channel No.153)	
154	CC-Link IE Controller Network (Channel No.154)	
181	CC-Link IE Field Network (Channel No.181)	CC-Link IE Field Network board Channel number is set with CC IE Field utility.
182	CC-Link IE Field Network (Channel No.182)	
183	CC-Link IE Field Network (Channel No.183)	
184	CC-Link IE Field Network (Channel No.184)	
281	CC-Link IE TSN (Channel No.281)	CC-Link IE TSN board Channel number is set with CC IE TSN utility.
282	CC-Link IE TSN (Channel No.282)	
283	CC-Link IE TSN (Channel No.283)	
284	CC-Link IE TSN (Channel No.284)	

Specifying station numbers

The following tables show the station numbers specified in the MELSEC data link library.

For specifying network numbers and station numbers for extended functions, refer to the following:

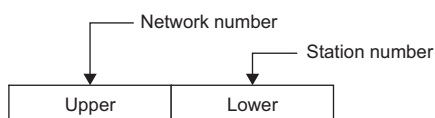
( Page 48 Specifying network numbers and station numbers for extended functions)

CC-Link IE TSN board

Specification		Station number	
		Upper byte	Lower byte
Own station		255 (FFH)	
Other station	Station number	Network number ^{*1} 1 (01H) to 239 (EFH)	0 (00H) ^{*2} to 120 (78H)
			125 (7DH) ^{*2}
The logical station number set with the utility		0 (00H) to 239 (EFH)	

^{*1} For specifying another station, set a network number in the upper byte of the station number.

<Setting a station number when another station is specified>



^{*2} Access the control station or master station (station number 0) specified with the network number on the network.

If the own station is a master station and the own network number is specified for a network number, station numbers 0 (00H) and 125 (7DH) cannot be specified.

When accessing the control station (operating as a control station) and the master station (operating as a master station when using the submaster function), specify the station number.

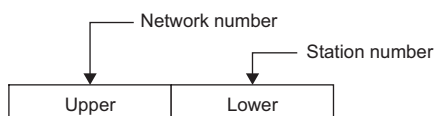
CC-Link IE Controller Network board

Specification		Station number	
		Upper byte	Lower byte
Own station		255 (FFH)	
Other station	Station number	Network number ^{*2} 1 (01H) to 239 (EFH)	0 (00H) ^{*3}
			1 (01H) to 120 (78H)
			125 (7DH) ^{*3}
	Group number 1 to 32 ^{*1}		129 (81H) to 160 (A0H)
	All stations ^{*1}		240 (F0H)
The logical station number set with the utility		0 (00H) to 239 (EFH)	

^{*1} All stations and group numbers can be specified when using the SEND function (mdSend) without arrival acknowledgment.

^{*2} For specifying another station, set a network number in the upper byte of the station number.

<Setting a station number when another station is specified>



^{*3} Access the control station or master station (station number 0) specified with the network number on the network.

When accessing the control station (operating as a control station) and the master station (operating as a master station when using the submaster function), specify the station number.

If the own station is a control station and the own network number is specified for a network number, station numbers 0 (00H) and 125 (7DH) cannot be specified.

CC-Link IE Field Network board

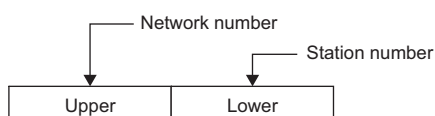
Specification		Station number	
		Upper byte	Lower byte
Own station		255 (FFH)	
Other station	Station number	Network number ^{*3} 1 (01H) to 239 (EFH)	0 (00H) ^{*4} to 120 (78H)
	Group number 1 to 32 ^{*1,*2}		125 (7DH) ^{*4}
	All stations ^{*1}		129 (81H) to 160 (A0H)
The logical station number set with the utility		240 (F0H)	
		0 (00H) to 239 (EFH)	

*1 All stations and group numbers can be specified when using the SEND function (mdSend) without arrival acknowledgment.

*2 Group numbers can be specified when using MELSECNET/H network, CC-Link IE Controller Network, or CC-Link IE TSN.

*3 For specifying another station, set a network number in the upper byte of the station number.

<Setting a station number when another station is specified>



*4 Access the control station or master station (station number 0) specified with the network number on the network. When accessing the control station (operating as a control station) and the master station (operating as a master station when using the submaster function), specify the station number. If the own station is a master station and the own network number is specified for a network number, station numbers 0 (00H) and 125 (7DH) cannot be specified.

CC-Link Ver.2 board

Specification	Station number
Own station	255 (FFH)
Other station	0 (00H) to 63 (3FH) ^{*1}
The logical station number set with the utility	65 (41H) to 239 (EFH)

*1 Station number 64 cannot be specified on the CC-Link Ver.2 board.

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

MELSECNET/H board

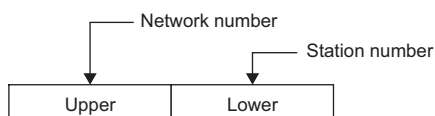
Specification		Station number	
		Upper byte	Lower byte
Own station		255 (FFH)	
Other station	Station number	Network number ^{*3} 1 (01H) to 239 (EFH)	0 (00H) ^{*4}
			1 (01H) to 120 (78H) ^{*5}
			125 (7DH) ^{*4}
	Group number 1 to 32 ^{*1,*2}		129 (81H) to 160 (A0H)
All stations ^{*1}		240 (F0H)	
The logical station number set with the utility		65 (41H) to 239 (EFH)	

*1 All stations and group numbers can be specified when using the SEND function (mdSend) without arrival acknowledgment.

*2 For MELSECNET/10 mode, only group numbers from 1 to 9 (129 (81H) to 137 (89H)) can be specified.

*3 For specifying another station, set a network number in the upper byte of the station number.

<Setting a station number when another station is specified>



*4 Access the control station or master station (station number 0) specified with the network number on the network. When accessing the control station (operating as a control station) and the master station (operating as a master station when using the submaster function), specify the station number. If the own station is a control station and the own network number is specified for a network number, station numbers 0 (00H) and 125 (7DH) cannot be specified.

*5 Station numbers from 65 (41H) to 120 (78H) can be specified when using CC-Link IE Controller Network.

Specifying network numbers and station numbers for extended functions

The following tables show the network numbers and the station numbers used for extended functions in the MELSEC data link library.

For specifying network numbers and station numbers for functions other than the extended functions, refer to the following:

( Page 46 Specifying station numbers)

CC-Link IE TSN board

Specification		Network number	Station number
Own station		0 (00H)	255 (FFH)
Other station	Station number	1 (01H) to 239 (EFH)	0 (00H)* ² to 120 (78H)
			125 (7DH)* ²
	Group number 1 to 32* ¹		129 (81H) to 160 (A0H)
	All stations* ¹		240 (F0H)
The logical station number set with the utility		0 (00H)	0 (00H) to 239 (EFH)

*1 All stations and group numbers can be specified when using the SEND function (mdSendEx) without arrival acknowledgment.

*2 Access the control station or master station (station number 0) specified with the network number on the network.

If the own station is a master station and the own network number is specified for a network number, station numbers 0 (00H) and 125 (7DH) cannot be specified.

When accessing the control station (operating as a control station) and the master station (operating as a master station when using the submaster function), specify the station number.

CC-Link IE Controller Network board

Specification		Network number	Station number
Own station		0 (00H)	255 (FFH)
Other station	Station number	1 (01H) to 239 (EFH)	0 (00H)* ²
			1 (01H) to 120 (78H)
			125 (7DH)* ²
	Group number 1 to 32* ¹		129 (81H) to 160 (A0H)
	All stations* ¹		240 (F0H)
The logical station number set with the utility		0 (00H)	0 (00H) to 239 (EFH)

*1 All stations and group numbers can be specified when using the SEND function (mdSendEx) without arrival acknowledgment.

*2 Access the control station or master station (station number 0) specified with the network number on the network.

When accessing the control station (operating as a control station) and the master station (operating as a master station when using the submaster function), specify the station number.

If the own station is a control station and the own network number is specified for a network number, station numbers 0 (00H) and 125 (7DH) cannot be specified.

CC-Link IE Field Network board

Specification		Network number	Station number
Own station		0 (00H)	255 (FFH)
Other station	Station number	1 (01H) to 239 (EFH)	0 (00H)* ³ to 120 (78H)
			125 (7DH)* ³
	Group number 1 to 32* ^{1,2}		129 (81H) to 160 (A0H)
	All stations* ¹		240 (F0H)
The logical station number set with the utility		0 (00H)	0 (00H) to 239 (EFH)

*1 All stations and group numbers can be specified when using the SEND function (mdSendEx) without arrival acknowledgment.

*2 Group numbers can be specified when using MELSECNET/H network, CC-Link IE Controller Network, or CC-Link IE TSN.

*3 Access the control station or master station (station number 0) specified with the network number on the network.

When accessing the control station (operating as a control station) and the master station (operating as a master station when using the submaster function), specify the station number.

If the own station is a master station and the own network number is specified for a network number, station numbers 0 (00H) and 125 (7DH) cannot be specified.

CC-Link Ver.2 board

Specification	Network number	Station number
Own station	0 (00H)	255 (FFH)
Other station		0 (00H) to 63 (3FH) ^{*1}
The logical station number set with the utility		65 (41H) to 239 (EFH)

*1 Station number 64 cannot be specified on the CC-Link Ver.2 board.

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

MELSECNET/H board

Specification	Network number	Station number
Own station	0 (00H)	255 (FFH)
Other station	1 (01H) to 239 (EFH)	0 (00H) ^{*1}
		1 (01H) to 120 (78H) ^{*2}
		125 (7DH) ^{*1}
The logical station number set with the utility	0 (00H)	65 (41H) to 239 (EFH)

*1 Access the control station or master station (station number 0) specified with the network number on the network.

When accessing the control station (operating as a control station) and the master station (operating as a master station when using the submaster function), specify the station number.

If the own station is a control station and the own network number is specified for a network number, station numbers 0 (00H) and 125 (7DH) cannot be specified.

*2 Station numbers from 65 (41H) to 120 (78H) can be specified when using CC-Link IE Controller Network.

Specifying device types

The following tables show the device types to be used for the MELSEC data link library and Device Monitor Utility. Specify a device type by code specification or device name specification for the MELSEC data link library functions. Set a device type listed in the 'Device type when monitoring devices' column in the following tables for Device Monitor Utility.

Data size per point of device

Devices to be specified for a device type have the following data types: bits, words, and double-words. Data size which can be handled per point of device varies depending on respective data types.

The following shows the data size per point of device for each device data type:

—: N/A

Type	Data size	
	Bit	Byte
Bit	1	—
Word	16	2
Double word	32	4

Link devices and buffer memory of own station

Device			Device type to be specified to a function			Device type when monitoring devices
Device name	Symbol	Type	Code specification		Device name specification	
			Decimal	Hexadecimal		
Link input	LX	Bit	1	0001H	DevX	X
Remote input	RX	Bit				
Link output	LY	Bit	2	0002H	DevY	Y
Remote output	RY	Bit				
Special relay	SM	Bit	5	0005H	DevSM	SM
Link special relay	SB	Bit				
Special register	SD	Word	14	000EH	DevSD	SD
Link special register	SW	Word				
Link relay	LB	Bit	23	0017H	DevB	B
Link register	LW	Word	24	0018H	DevW ^{*1}	W
Remote register	RW	Word				
Remote register (for transmission)	RWw	Word	36	0024H	DevWw	Ww
Remote register (for reception)	RWr	Word	37	0025H	DevWr	Wr
Buffer memory		Word	50	0032H	DevSPB	SPB
Random access buffer		Word	33	0021H	DevMRB	MRB

*1 Can be specified to access a remote register for CC-Link IE Field Network boards only.

Devices of other station module

Device			Device type to be specified to a function			Device type when monitoring devices	
Device name		Symbol	Type	Code specification			Device name specification
				Decimal	Hexadecimal		
Input		X	Bit	1	0001H	DevX	X
Output		Y	Bit	2	0002H	DevY	Y
Latch relay		L	Bit	3	0003H	DevL	L
Internal relay		M	Bit	4	0004H	DevM	M
Special relay		SM	Bit	5	0005H	DevSM	SM
Annunciator		F	Bit	6	0006H	DevF	F
Timer	contact	T	Bit	7	0007H	DevTT	TT
	coil		Bit	8	0008H	DevTC	TC
Counter	contact	C	Bit	9	0009H	DevCT	CT
	coil		Bit	10	000AH	DevCC	CC
Timer	current value	T	Word	11	000BH	DevTN	TN
Counter	current value	C	Word	12	000CH	DevCN	CN
Data register		D	Word	13	000DH	DevD	D
Special register		SD	Word	14	000EH	DevSD	SD
Timer	setting value main	T	Word	15	000FH	DevTM	TM
	setting value sub1		Word	16	0010H	DevTS	TS
	setting value sub2		Word	16002	3E82H	DevTS2	TS2
	setting value sub3		Word	16003	3E83H	DevTS3	TS3
Counter	setting value main	C	Word	17	0011H	DevCM	CM
	setting value sub1		Word	18	0012H	DevCS	CS
	setting value sub2		Word	18002	4652H	DevCS2	CS2
	setting value sub3		Word	18003	4653H	DevCS3	CS3
Accumulator		A	Word	19	0013H	DevA	A
Index register		Z	Word	20	0014H	DevZ	Z
		V	Word	21	0015H	DevV	V
File register Extended file register *1	Block switching method	R	Word	22	0016H	DevR	R
			Word	22000 to 22256*2	55F0H to 56F0H*2	DevER0 to 256*2	ER
	Serial number method	ZR	Word	220	00DCH	DevZR	ZR
Link relay		B	Bit	23	0017H	DevB	B
Link register		W	Word	24	0018H	DevW	W
Link special relay		SB	Bit	25	0019H	DevQSB	QSB
Retentive timer	contact	ST	Bit	26	001AH	DevSTT	STT
	coil		Bit	27	001BH	DevSTC	STC
Link special register		SW	Word	28	001CH	DevQSW	QSW
Edge relay		V	Bit	30	001EH	DevQV	QV
Retentive timer	current value	ST	Word	35	0023H	DevSTN	STN
Long index register		LZ	Double word	38	0026H	DevLZ	LZ
Refresh data register		RD	Word	39	0027H	DevRD	RD
Long Timer	contact	LT	Bit	41	0029H	DevLTT	LTT
	coil		Bit	42	002AH	DevLTC	LTC
	current value		Double word	43	002BH	DevLTN	LTN
Long counter	contact	LC	Bit	44	002CH	DevLCT	LCT
	coil		Bit	45	002DH	DevLCC	LCC
	current value		Double word	46	002EH	DevLCN	LCN
Long retentive timer	contact	LST	Bit	47	002FH	DevLSTT	LSTT
	coil		Bit	48	0030H	DevLSTC	LSTC
	current value		Double word	49	0031H	DevLSTN	LSTN

Device				Device type to be specified to a function			Device type when monitoring devices
Device name		Symbol	Type	Code specification		Device name specification	
				Decimal	Hexadecimal		
Link direct device *1,*3	Link input	Jn\X	Bit	1001 to 1255	03E9H to 04E7H	DevLX1 to 255	LX
	Link output	Jn\Y	Bit	2001 to 2255	07D1H to 08CFH	DevLY1 to 255	LY
	Link relay	Jn\B	Bit	23001 to 23255	59D9H to 5AD7H	DevLB1 to 255	LB
	Link register*4	Jn\W	Word	24001 to 24255	5DC1H to 5EBFH	DevLW1 to 255	LW
	Link special relay	Jn\SB	Bit	25001 to 25255	61A9H to 62A7H	DevLSB1 to 255	LSB
	Link special register	Jn\SW	Word	28001 to 28255	6D61H to 6E5FH	DevLSW1 to 255	LSW
Intelligent function module device Buffer memory Module access device *1,*5		Un\G	Word	29000 to 29255	7148H to 7247H	DevSPG0 to 255	SPG

*1 With the random read function (mdRandR, mdRandREx), the function may complete normally even if the specified devices do not exist. (The read data is error.)

*2 Access the file register of the block specified with the device type.

Specify a block number (0 to 256) for the lower 3 digits of the code specification (decimal) and the numerical value of the device name specification.

*3 Specify a network number (1 to 255) for the lower 3 digits of the code specification (decimal) and the numerical value of the device specification.

*4 For some network modules, multiple link devices can be accessed with one device name. For the specification method for direct access to link devices, refer to the manual for a network module. (□□Manual for a module used)

*5 Specify a value (start I/O number divided by 16) for the lower 3 digits of the code specification (decimal) and the numerical value of the device name specification.

Buffer memory of CC-Link other station

When accessing the buffer memory of CC-Link network module or other CC-Link Ver.2 board from CC-Link Ver.2 board, specify the following device type.

Device			Device type to be specified to a function			Device type when monitoring devices
Device name	Symbol	Type	Code specification		Device name specification	
			Decimal	Hexadecimal		
Buffer memory ^{*1}		Word	-32768	8000H	DevRBM	RBM
Random access buffer ^{*1}		Word	-32736	8020H	DevRAB	RAB
Remote input ^{*1}	RX	Bit	-32735	8021H	DevRX	RX
Remote output ^{*1}	RY	Bit	-32734	8022H	DevRY	RY
Remote register ^{*1}	RWw, RWr	Word	-32732	8024H	DevRW	RW
Link special relay ^{*1}	SB	Bit	-32669	8063H	DevSB	SB
Link special register ^{*1}	SW	Word	-32668	8064H	DevSW	SW

*1 These devices can not be used for random write (mdRandW, mdRandWEx), random read (mdRandR, mdRandREx), bit set (mdDevSet, mdDevSetEx) and bit reset (mdDevRst, mdDevRstEx).

The SEND function and the RECV function

The same operation as device access, the SEND function and the RECV function send data (mdSend, mdSendEx) or read data (mdReceive, mdReceiveEx) by specifying a device type for each function.

Note that this device type cannot be specified in Device Monitor Utility as it is the special device type which can be used only for the SEND function and the RECV function.

Function	Device type to be specified to a function		
	Code specification		Device name specification
	Decimal	Hexadecimal	
RECV function	101	0065H	DevMAIL
SEND function (with arrival acknowledgment)			
SEND function (without arrival acknowledgment)	102	0066H	DevMAILNC

4.3 Function Details

The following shows the detailed specifications of the MELSEC data link library.
This section provides the description of the instructions in the layout as shown below.

Format

Indicates the description format in each programming language.

■Visual C++

A format for Microsoft® Visual C++

■Visual C#

A format for Microsoft Visual C#

For the namespace and class, etc., refer to the following:

 Page 12 Programming when using Visual C#

■Visual Basic .NET

A format for the following Visual Basic

- Visual Studio .NET 2003 Visual Basic
- Visual Studio 2005 Visual Basic
- Visual Studio 2008 Visual Basic
- Visual Studio 2010 Visual Basic
- Visual Studio 2012 Visual Basic
- Visual Studio 2013 Visual Basic
- Visual Studio 2015 Visual Basic
- Visual Studio 2017 Visual Basic
- Visual Studio 2019 Visual Basic
- Visual Studio 2022 Visual Basic

■Visual Basic 5.0, Visual Basic 6.0

A format for the following Visual Basic

- Visual Basic 5.0
- Visual Basic 6.0

Detailed specifications

■Argument

Describes arguments of the function.

■Explanation

Describes features of the function and details of arguments.

■Return value

Indicates the return values.

■Related function

Indicates related functions for programming.

mdOpen (opening communication lines)

Open a communication line by specifying a channel number of communication line.

Format

■Visual C++

Format	ret = mdOpen(chan,mode,path);		
Argument	short	ret;	//Return value OUT
	short	chan;	//Channel number of communication line IN
	short	mode;	//Dummy IN
	long	*path;	//Opened line path pointer OUT

■Visual C#

Format	ret = [Class name].mdOpen(chan,mode,out path);		
Argument	short	ret;	//Return value OUT
	short	chan;	//Channel number of communication line IN
	short	mode;	//Dummy IN
	int	path;	//Opened line path pointer OUT

■Visual Basic .NET

Format	ret = mdOpen(chan,mode,path)		
Argument	Short	ret	;Return value OUT
	Short	chan	;Channel number of communication line IN
	Short	mode	;Dummy IN
	Integer	path	;Opened line path pointer OUT

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdOpen(chan%,mode%,path&)		
Argument	Integer	ret	;Return value OUT
	Integer	chan	;Channel number of communication line IN
	Integer	mode	;Dummy IN
	Long	path	;Opened line path pointer OUT

Detailed specifications

■Argument

Argument	Description
chan	Specify the channel number of communication line. (📖 Page 45 Specifying channel number)
mode	Specify -1.
path	Return the opened line path.

■Explanation

- A path of the channel used for argument of another function is returned to the opened line path pointer.
- When using multiple communication lines, opening for each channel number is necessary.

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. (📖 Page 140 ERROR CODES)

■Related function

mdClose()

mdClose (closing communication lines)

Close a communication line by specifying a communication line path.

Format

■Visual C++

Format	ret = mdClose(path);			
Argument	short	ret;	//Return value	OUT
	long	path;	//Path of channel	IN

■Visual C#

Format	ret = [Class name].mdClose(path);			
Argument	short	ret;	//Return value	OUT
	int	path;	//Path of channel	IN

■Visual Basic .NET

Format	ret = mdClose(path)			
Argument	Short	ret	;Return value	OUT
	Integer	path	;Path of channel	IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdClose(path&)			
Argument	Integer	ret	;Return value	OUT
	Long	path	;Path of channel	IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)

■Explanation

When using multiple communication lines, they need to be closed for each channel number.

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. (📖 Page 140 ERROR CODES)

■Related function

mdOpen()

mdSendEx (batch writing extended devices)

Batch write data to the devices on the target station for the number of written data bytes from the start device number.

Format

■Visual C++

Format	ret = mdSendEx(path,netno,stno,devtyp,devno,size,data);		
Argument	long	ret;	//Return value OUT
	long	path;	//Path of channel IN
	long	netno;	//Network number IN
	long	stno;	//Station number IN
	long	devtyp;	//Device type IN
	long	devno;	//Start device number IN
	long	*size;	//Written byte size IN/OUT
	short	data[];	//Written data (single-precision integer array) IN

■Visual C#

Format	ret = [Class name].mdSendEx(path,netno,stno,devtyp,devno,ref size,data);		
Argument	int	ret;	//Return value OUT
	int	path;	//Path of channel IN
	int	netno;	//Network number IN
	int	stno;	//Station number IN
	int	devtyp;	//Device type IN
	int	devno;	//Start device number IN
	int	size;	//Written byte size IN/OUT
	short	data[];	//Written data (single-precision integer array) IN

■Visual Basic .NET

Format	ret = mdSendEx(path,netno,stno,devtyp,devno,size,data(0))		
Argument	Integer	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Integer	netno	;Network number IN
	Integer	stno	;Station number IN
	Integer	devtyp	;Device type IN
	Integer	devno	Start device number IN
	Integer	size	;Written byte size IN/OUT
	Short	data(n)	;Written data (single-precision integer array) IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdSendEx(path&,netno%,stno%,devtyp%,devno%,size%,data%(0))		
Argument	Long	ret	;Return value OUT
	Long	path	;Path of channel IN
	Long	netno	;Network number IN
	Long	stno	;Station number IN
	Long	devtyp	;Device type IN
	Long	devno	;Start device number IN
	Long	size	;Written byte size IN/OUT
	Any	data(n)	;Written data (single-precision integer array) IN

Detailed specifications

Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Specify the network number of a target station. (➤ Page 48 Specifying network numbers and station numbers for extended functions)
stno	Specify the station number of a target station. (➤ Page 48 Specifying network numbers and station numbers for extended functions)
devtyp	Specify the type of device to which the data is written. (➤ Page 50 Specifying device types)
devno	Specify the start device number of device to which the data is written. - Specify the start device number with a multiple of 8 for the access to a bit device. (However, the start device number can be specified arbitrarily for accessing a contact and coil of the long timer and long retentive timer.) - Specify the start device number with a multiple of 16 (0, 16, 32...) for the access to a bit device (RX, RY, SB) of CC-Link other station link device.
size	Specify the byte size to be written. It must be specified according to devtyp as follows: - For a bit device: A multiple of 1 - For a word device: A multiple of 2 - For a double-word device: A multiple of 4 When the specified byte size to be written exceeds the device range (-5: size error), the applicable size is returned to "size".
data	Specify the data to be written in single-precision integer array.

Explanation

- When the specified written byte size exceeds the transient transmission size, data are divided inside the function and written.
- When accessing another station, the extended comment information will be deleted by writing data to the block (extended file register) to which the extended comment is assigned.
- When accessing another station, the sub2 and sub3 programs will be deleted by writing data to the block (extended file register) which overlaps with the setting areas of the sub2 and sub3 programs.
- When a double word device is specified to "devtyp", store the data to be written to "data" as follows:

Ex.

When "devtyp" is LZ and "size" is 8

Array	Value
data[0]	Lower 1 word of LZ0
data[1]	Upper 1 word of LZ0
data[2]	Lower 1 word of LZ1
data[3]	Upper 1 word of LZ1

Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. (➤ Page 140 ERROR CODES)

Related function

mdOpen(), mdClose(), mdSend(), mdReceive(), mdReceiveEx()

mdSendEx (SEND function)

Send data to the specified channel number of the target station.

Format

■Visual C++

Format	ret = mdSendEx(path,netno,stno,devtyp,devno,size,data);		
Argument	long	ret;	//Return value OUT
	long	path;	//Path of channel IN
	long	netno;	//Network number IN
	long	stno;	//Station number IN
	long	devtyp;	//Device type IN
	long	devno;	//Channel number IN
	long	*size;	//Send byte size IN/OUT
	short	data[];	//Send data (single-precision integer array) IN

■Visual C#

Format	ret = [Class name].mdSendEx(path,netno,stno,devtyp,devno,ref size,data);		
Argument	int	ret;	//Return value OUT
	int	path;	//Path of channel IN
	int	netno;	//Network number IN
	int	stno;	//Station number IN
	int	devtyp;	//Device type IN
	int	devno;	//Channel number IN
	int	size;	//Send byte size IN/OUT
	short	data[];	//Send data (single-precision integer array) IN

■Visual Basic .NET

Format	ret = mdSendEx(path,netno,stno,devtyp,devno,size,data(0))		
Argument	Integer	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Integer	netno	;Network number IN
	Integer	stno	;Station number IN
	Integer	devtyp	;Device type IN
	Integer	devno	;Channel number IN
	Integer	size	;Send byte size IN/OUT
	Short	data(n)	;Send data (single-precision integer array) IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdSendEx(path&,.netno%,stno%,devtyp%,devno%,size%,data%(0))		
Argument	Long	ret	;Return value OUT
	Long	path	;Path of channel IN
	Long	netno	;Network number IN
	Long	stno	;Station number IN
	Long	devtyp	;Device type IN
	Long	devno	;Channel number IN
	Long	size	;Send byte size IN/OUT
	Any	data(n)	;Send data (single-precision integer array) IN

Detailed specifications

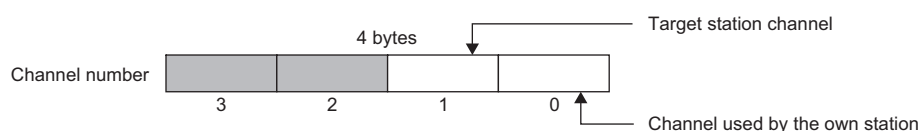
Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Specify the network number of a target station. (☞ Page 48 Specifying network numbers and station numbers for extended functions)
stno	Specify the station number of a target station. (☞ Page 48 Specifying network numbers and station numbers for extended functions) • Logical station numbers cannot be specified.
devtyp	Specify whether to send data with arrival acknowledgment or without arrival acknowledgment. • With arrival acknowledgment*1: Specify 101 (65H) • Without arrival acknowledgment: Specify 102 (66H)
devno	Specify a channel used by the own station and a target station channel. (☞ Page 59 Explanation)
size	Specify the byte size of send data in even number. • Specify the byte size within the range of 2 to 1920. • When accessing via MELSECNET/10 network, specify the byte size within the range of 2 to 960. If a specified size is an odd number or out of the range above, '0' is returned to size.
data	Specify the data to be written in single-precision integer array.

*1 Do not specify all stations or group numbers for the station number when sending data with arrival acknowledgment.

Explanation

- This function supports the SEND instruction of the link dedicated instruction. For details of the functions, refer to the manuals of each board.
- The following explains how to specify the channel numbers.



Item	Description		
Target station channel	Specify a target station channel number.	CC-Link IE TSN board, CC-Link IE Controller Network board, MELSECNET/H board	1 to 8
		CC-Link IE Field Network board	1 to 2
Channel used by the own station	Specify a channel number used by the own station.	CC-Link IE TSN board, CC-Link IE Controller Network board	1 to 8
		CC-Link IE Field Network board	1 to 2

- The arguments of the mdSendEx function correspond to the control data (device) of the dedicated instruction (SEND) as shown below:

—: N/A

Device	Item	Corresponding argument and return value
+0	Execution/error completion type	devtyp
+1	Completion status	ret (return value)
+2	Channel used by the own station	devno
+3	Target station channel	devno
+4	Target station network number	netno
+5	Target station number	stno
+6	Not used	—
+7	Number of retransmissions (retries)	—
+8	Arrival monitoring time	—
+9	Send data length	size
+10	Not used	—
+11	Clock setting flag	—
+12	Clock data	—
+13		—
+14		—
+15		—

Device	Item	Corresponding argument and return value
+16	Error-detected network number	—
+17	Error-detected station number	—

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. (👉 Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdSend(), mdReceive(), mdReceiveEx()

mdReceiveEx (batch reading extended devices)

Batch read data from the devices on the target station for the number of read data bytes from the start device number.

Format

■Visual C++

Format	ret = mdReceiveEx(path,netno,stno,devtyp,devno,size,data);		
Argument	long	ret;	//Return value OUT
	long	path;	//Path of channel IN
	long	netno;	//Network number IN
	long	stno;	//Station number IN
	long	devtyp;	//Device type IN
	long	devno;	//Start device number IN
	long	*size;	//Read byte size IN/OUT
	short	data[];	//Read data (single-precision integer array) OUT

■Visual C#

Format	ret = [Class name].mdReceiveEx(path,netno,stno,devtyp,devno,ref size,data);		
Argument	int	ret;	//Return value OUT
	int	path;	//Path of channel IN
	int	netno;	//Network number IN
	int	stno;	//Station number IN
	int	devtyp;	//Device type IN
	int	devno;	//Start device number IN
	int	size;	//Read byte size IN/OUT
	short	data[];	//Read data (single-precision integer array) OUT

■Visual Basic .NET

Format	ret = mdReceiveEx(path,netno,stno,devtyp,devno,size,data(0))		
Argument	Integer	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Integer	netno	;Network number IN
	Integer	stno	;Station number IN
	Integer	devtyp	;Device type IN
	Integer	devno	;Start device number IN
	Integer	size	;Read byte size IN/OUT
	Short	data(n)	;Read data (single-precision integer array) OUT

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdReceiveEx(path&,netno%,stno%,devtyp%,devno%,size%,data%(0))		
Argument	Long	ret	;Return value OUT
	Long	path	;Path of channel IN
	Long	netno	;Network number IN
	Long	stno	;Station number IN
	Long	devtyp	;Device type IN
	Long	devno	;Start device number IN
	Long	size	;Read byte size IN/OUT
	Any	data(n)	;Read data (single-precision integer array) OUT

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Specify the network number of a target station. ( Page 48 Specifying network numbers and station numbers for extended functions)
stno	Specify the station number of a target station. ( Page 48 Specifying network numbers and station numbers for extended functions)
devtyp	Specify the type of device from which the data is read. ( Page 50 Specifying device types)
devno	Specify the start device number of device range to be read. - Specify the start device number with a multiple of 8 for the access to a bit device. (However, the start device number can be specified arbitrarily for accessing a contact and coil of the long timer and long retentive timer.) - Specify the start device number with a multiple of 16 (0, 16, 32...) for the access to a bit device (RX, RY, SB) of CC-Link other station link device.
size	Specify the byte size to be read. It must be specified according to devtyp as follows: - For a bit device: A multiple of 1 - For a word device: A multiple of 2 - For a double-word device: A multiple of 4 When the specified byte size to be read exceeds the device range (-5: size error), the applicable size is returned to "size".
data	Store the data being read.

■Explanation

- When the specified read byte size exceeds the transient transmission size, data are divided inside the function and read.
- When a double word device is specified to "devtyp", the data is stored to "data" as follows:

Ex.

When "devtyp" is LZ and "size" is 8

Array	Value
data[0]	Lower 1 word of LZ0
data[1]	Upper 1 word of LZ0
data[2]	Lower 1 word of LZ1
data[3]	Upper 1 word of LZ1

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdReceive(), mdSend(), mdSendEx()

mdReceiveEx (RECV function)

Read data of the specified channel number from the data which are received by the own station.

Format

■Visual C++

Format	ret = mdReceiveEx(path,netno,stno,devtyp,devno,size,data);		
Argument	long	ret;	//Return value OUT
	long	path;	//Path of channel IN
	long	netno;	//Network number IN
	long	stno;	//Station number IN
	long	devtyp;	//Device type IN
	long	devno;	//Channel number IN
	long	*size;	//Receive byte size IN/OUT
	short	data[];	//Receive data with send source information (single-precision integer array) OUT

■Visual C#

Format	ret = [Class name].mdReceiveEx(path,netno,stno,devtyp,devno,ref size,data);		
Argument	int	ret;	//Return value OUT
	int	path;	//Path of channel IN
	int	netno;	//Network number IN
	int	stno;	//Station number IN
	int	devtyp;	//Device type IN
	int	devno;	//Channel number IN
	int	size;	//Receive byte size IN/OUT
	short	data[];	//Receive data with send source information (single-precision integer array) OUT

■Visual Basic .NET

Format	ret = mdReceiveEx(path,netno,stno,devtyp,devno,size,data(0))		
Argument	Integer	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Integer	netno	;Network number IN
	Integer	stno	;Station number IN
	Integer	devtyp	;Device type IN
	Integer	devno	;Channel number IN
	Integer	size	;Receive byte size IN/OUT
	Short	data(n)	;Receive data with send source information (single-precision integer array) OUT

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdReceiveEx(path&,netno%,stno%,devtyp%,devno%,size%,data%(0))		
Argument	Long	ret	;Return value OUT
	Long	path	;Path of channel IN
	Long	netno	;Network number IN
	Long	stno	;Station number IN
	Long	devtyp	;Device type IN
	Long	devno	;Channel number IN
	Long	size	;Receive byte size IN/OUT
	Any	data(n)	;Receive data with send source information (single-precision integer array) OUT

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Specify 0 (0H).
stno	Specify the own station (255 (FFH)).
devtyp	Specify the RECV function (101 (65H)).

Argument	Description
devno	Specify the own station channel number on which the received messages are stored. - CC-Link IE TSN board, CC-Link IE Controller Network board: 1 to 8 - CC-Link IE Field Network board: 1 to 2
size	Specify the byte size of received data in even number. - Specify the byte size within the range of 2 to 1920. - When receiving data sent from a MELSECNET/H board in a CC-Link IE Controller Network board or CC-Link IE Field Network board, specify it within the range of 2 to 960 bytes. The size of received actual data is received. If a specified size is an odd number or out of the range above, '0' is returned to size.
data	Received actual data and send source information are stored as single-precision integer array. Reserve an area for the specified receive byte size and send source information (6 bytes).

■Explanation

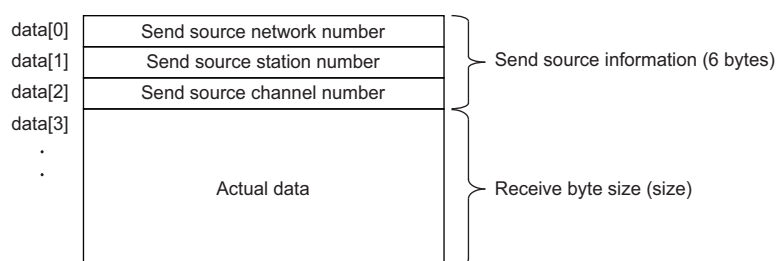
This function supports the RECV instruction of the link dedicated instruction.

For details of the function, refer to the manuals of each board.

< Receive data with send source information (data) >

- Receive byte size and send source information (6 bytes) are stored in "receive data with send source information (data)".
Reserve an area for [receive byte size and send source information (6 bytes)] in "receive data with send source information (data)".
- Data are stored in "receive data with send source information (data)" as shown below.

Receive data with send source information (data)



- Only the actual data with the specified receive byte size (size) is stored when the received actual data size is greater than the specified receive byte size (size).
- 125 (7DH) is stored to the send source station number when the station number of the send source is 0.
- The arguments of the mdReceiveEx function correspond to the control data (device) of the dedicated instruction (RECV) as shown below:

—: N/A

Device	Item	Corresponding argument and return value
+0	Error completion type	—
+1	Completion status	ret (return value)
+2	Own station storage channel	devno
+3	Channel used by send station	data[2]
+4	Send station network number	data[0]
+5	Send station number	data[1]
+6	Not used	—
+7	Not used	—
+8	Arrival monitoring time	—
+9	Receive data length	size
+10	Not used	—
+11	Clock setting flag	—
+12	Clock data (Set only in an abnormal state.)	—
+13		—
+14		—
+15		—
+16	Error-detected network number	—
+17	Error-detected station number	—

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdReceive(), mdSend(), mdSendEx()

mdRandWEx (writing extended devices randomly)

Write data to the devices on the target station specified with the randomly-specified devices.

Format

■Visual C++

Format	ret = mdRandWEx(path,netno,stno,dev,buf,bufsize);		
Argument	long	ret;	//Return value OUT
	long	path;	//Path of channel IN
	long	netno;	//Network number IN
	long	stno;	//Station number IN
	long	dev[];	//Randomly-specified device IN
	short	buf[];	//Written data (single-precision integer array) IN
	long	bufsize;	//Dummy IN

■Visual C#

Format	ret = [Class name].mdRandWEx(path,netno,stno,dev,buf,bufsize);		
Argument	int	ret;	//Return value OUT
	int	path;	//Path of channel IN
	int	netno;	//Network number IN
	int	stno;	//Station number IN
	int	dev[];	//Randomly-specified device IN
	short	buf[];	//Written data (single-precision integer array) IN
	int	bufsize;	//Dummy IN

■Visual Basic .NET

Format	ret = mdRandWEx(path,netno,stno,dev(0),buf(0),bufsize)		
Argument	Integer	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Integer	netno	;Network number IN
	Integer	stno	;Station number IN
	Integer	dev(n)	;Randomly-specified device IN
	Short	buf(n)	;Written data (single-precision integer array) IN
	Integer	bufsize	;Dummy IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdRandWEx(path&,netno%,stno%,dev%(0),buf%(0),bufsize%)		
Argument	Long	ret	;Return value OUT
	Long	path	;Path of channel IN
	Long	netno	;Network number IN
	Long	stno	;Station number IN
	Long	dev(n)	;Randomly-specified device IN
	Any	buf(n)	;Written data (single-precision integer array) IN
	Long	bufsize	;Dummy IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Specify the network number of a target station. ( Page 48 Specifying network numbers and station numbers for extended functions)
stno	Specify the station number of a target station. ( Page 48 Specifying network numbers and station numbers for extended functions)
dev	Specify the number of blocks, the device type, the start device number, and points of device to be written. ( Page 68 How to specify the randomly-specified devices (dev))
buf	Specify the data to be written to the array variable of buf[0] or later. ( Page 68 How to specify the randomly-specified devices (dev))
bufsize	—

■Explanation

- The number of transient transmissions performed inside of the function changes with the specified randomly-specified devices.
- A longer function processing time is required for the random write function as compared with the batch write function. Consider using the batch write function if a shorter function processing time is required.
- When accessing another station, the extended comment information will be deleted by writing data to the block (extended file register) to which the extended comment is assigned.
- When accessing another station, the sub2 and sub3 programs will be deleted by writing data to the block (extended file register) which overlaps with the setting areas of the sub2 and sub3 programs.
- The size of written data varies for each device type depending on the number of points specified for randomly-specified devices.
( Page 50 Data size per point of device)

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdRandREx()

How to specify the randomly-specified devices (dev)

The following table shows how to specify the randomly-specified devices (dev).

Randomly-specified device	Description	
dev[0]	Specify the number of blocks. (1 or more)	If the memory on a personal computer is insufficient for a specified number of blocks, a memory reservation error/resource memory shortage error (error code 77) occurs.
dev[1]	Device type	
dev[2]	Start device number	Specify the device of block 1.
dev[3]	Points	
dev[4]	Device type	
dev[5]	Start device number	Specify the device of block 2.
dev[6]	Points	
:	(Specify the devices for the specified number of blocks.)	

■When writing values to multiple devices

The following tables show an example when writing the values in devices.

Block	Specified device	Device value
Block 1	M100 to M115 (16 points of bit device)	All the bits are OFF. (0000H)
Block 2	D10 to D13 (4 points of word device)	D10: 10, D11: 200, D12: 300, D13: 400

• Randomly-specified device (dev)

Program example	Description
dev[0]=2;	Number of blocks: 2
dev[1]=DevM; dev[2]=100; dev[3]=16;	Block 1 • Start device: M100 • Points: 16 points
dev[4]=DevD; dev[5]=10; dev[6]=4;	Block 2 • Start device: D10 • Points: 4 points

• Written data (buf)

Array	Value	Device
buf[0]	0	M100 to M115
buf[1]	10	D10
buf[2]	200	D11
buf[3]	300	D12
buf[4]	400	D13

■When writing values to double word devices

The following tables show an example when writing the values in devices.

Block	Specified device	Device value
Block 1	LC100 to LC101 (2 points of double word device)	LC100 (current value) = 1, LC101 (current value) = 65536 (10000H)

• Randomly-specified device (dev)

Program example	Description
dev[0]=1;	Number of blocks: 1
dev[1]=DevLCN; dev[2]=100; dev[3]=2;	Block 1 • Start device: LC100 (current value) • Points: 2 points

• Written data (buf)

Array	Value	Device
buf[0]	1	Lower 1 word of LC100 (current value)
buf[1]	0	Upper 1 word of LC100 (current value)
buf[2]	0	Lower 1 word of LC101 (current value)
buf[3]	1	Upper 1 word of LC101 (current value)

mdRandREx (reading extended devices randomly)

Read the device specified with the randomly-specified devices from the target station.

Format

■Visual C++

Format	ret = mdRandREx(path,netno,stno,dev,buf,bufsize);		
Argument	long	ret;	//Return value OUT
	long	path;	//Path of channel IN
	long	netno;	//Network number IN
	long	stno;	//Station number IN
	long	dev[];	//Randomly-specified device IN
	short	buf[];	//Read data (single-precision integer array) OUT
	long	bufsize;	//Number of bytes of read data IN

■Visual C#

Format	ret = [Class name].mdRandREx(path,netno,stno,dev,buf,bufsize);		
Argument	int	ret;	//Return value OUT
	int	path;	//Path of channel IN
	int	netno;	//Network number IN
	int	stno;	//Station number IN
	int	dev[];	//Randomly-specified device IN
	short	buf[];	//Read data (single-precision integer array) OUT
	int	bufsize;	//Number of bytes of read data IN

■Visual Basic .NET

Format	ret = mdRandREx(path,netno,stno,dev(0),buf(0),bufsize)		
Argument	Integer	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Integer	netno	;Network number IN
	Integer	stno	;Station number IN
	Integer	dev(n)	;Randomly-specified device IN
	Short	buf(n)	;Read data (single-precision integer array) OUT
	Integer	bufsize	;Number of bytes of read data IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdRandREx(path&,netno%,stno%,dev%(0),buf%(0),bufsize%)		
Argument	Long	ret	;Return value OUT
	Long	path	;Path of channel IN
	Long	netno	;Network number IN
	Long	stno	;Station number IN
	Long	dev(n)	;Randomly-specified device IN
	Any	buf(n)	;Read data (single-precision integer array) OUT
	Long	bufsize	;Number of bytes of read data IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Specify the network number of a target station. ( Page 48 Specifying network numbers and station numbers for extended functions)
stno	Specify the station number of a target station. ( Page 48 Specifying network numbers and station numbers for extended functions)
dev	Specify the number of blocks, the device type, the start device number, and points of device to be read. ( Page 71 How to specify the randomly-specified devices (dev))
buf	Store the read data to the array variable of buf[0] or later. ( Page 71 How to specify the randomly-specified devices (dev))
bufsize	Specify the number of bytes of read data. ( Page 71 How to specify the randomly-specified devices (dev))

■Explanation

- The number of transient transmissions performed inside of the function changes with the specified randomly-specified devices.
- A longer function processing time is required for the random read function as compared with the batch read function. Consider using the batch read function if a shorter function processing time is required.
- The size of read data varies for each device type depending on the number of points specified for randomly-specified devices.
( Page 50 Data size per point of device)

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdRandWEx()

How to specify the randomly-specified devices (dev)

The following table shows how to specify the randomly-specified devices (dev).

Randomly-specified device	Description	
dev[0]	Specify the number of blocks. (1 or more)	If the memory on a personal computer is insufficient for a specified number of blocks, a memory reservation error/resource memory shortage error (error code 77) occurs.
dev[1]	Device type	Specify the device of block 1.
dev[2]	Start device number	
dev[3]	Points	
dev[4]	Device type	Specify the device of block 2.
dev[5]	Start device number	
dev[6]	Points	
:	(Specify the devices for the specified number of blocks.)	

■When reading values from multiple devices

The following tables show an example when reading the values from devices.

Block	Specified device	Device value
Block 1	M100 to M115 (16 points of bit device)	All the bits are OFF. (0000H)
Block 2	D10 to D13 (4 points of word device)	D10: 10, D11: 200, D12: 300, D13: 400
Block 3	M0 to M13 (14 points of bit device)	All the bits are ON. (3FFFH)
Block 4	T10 (current value) (1 point of word device)	T10 (current value): 10

• Randomly-specified device (dev)

Program example	Description
dev[0]=4;	Number of blocks: 4
dev[1]=DevM; dev[2]=100; dev[3]=16;	Block 1 • Start device: M100 • Points: 16 points
dev[4]=DevD; dev[5]=10; dev[6]=4;	Block 2 • Start device: D10 • Points: 4 points
dev[7]=DevM; dev[8]=0; dev[9]=14;	Block 3 • Start device: M0 • Points: 14 points
dev[10]=DevTN; dev[11]=10; dev[12]=1;	Block 4 • Start device: T10 (current value) • Points: 1 point

• Read data (buf)

Array	Value	Device
buf[0]	0	M100 to M115
buf[1]	10	D10
buf[2]	200	D11
buf[3]	300	D12
buf[4]	400	D13
buf[5]	16383 (3FFFH)	M0 to M13
buf[6]	10	T10 (current value)

• Number of bytes of read data (bufsize)

Specify the number of bytes of array variable buf which stores the read data.

(buf[0] to buf[6] = 7) × 2 = 14 bytes

Enter "14" for this example.

■When reading values from double word devices

The following tables show an example when reading the values from devices.

Block	Specified device	Device value
Block 1	LC100 to LC101 (2 points of double word device)	LC100 (current value) = 1, LC101 (current value) = 65536 (10000H)

- Randomly-specified device (dev)

Program example	Description
dev[0]=1;	Number of blocks: 1
dev[1]=DevLCN; dev[2]=100; dev[3]=2;	Block 1 • Start device: LC100 (current value) • Points: 2 points

- Read data (buf)

Array	Value	Device
buf[0]	1	Lower 1 word of LC100 (current value)
buf[1]	0	Upper 1 word of LC100 (current value)
buf[2]	0	Lower 1 word of LC101 (current value)
buf[3]	1	Upper 1 word of LC101 (current value)

- Number of bytes of read data (bufsize)

Specify the number of bytes of array variable buf which stores the read data.

(buf[0] to buf[3] = 4) × 2 = 8 bytes

Enter "8" for this example.

mdDevSetEx (setting extended bit devices)

Set the specified bit device on the target station (to ON).

Format

■Visual C++

Format	ret = mdDevSetEx(path,netno,stno,devtyp,devno);		
Argument	long	ret;	//Return value OUT
	long	path;	//Path of channel IN
	long	netno;	//Network number IN
	long	stno;	//Station number IN
	long	devtyp;	//Device type IN
	long	devno;	//Specified device number IN

■Visual C#

Format	ret = [Class name].mdDevSetEx(path,netno,stno,devtyp,devno);		
Argument	int	ret;	//Return value OUT
	int	path;	//Path of channel IN
	int	netno;	//Network number IN
	int	stno;	//Station number IN
	int	devtyp;	//Device type IN
	int	devno;	//Specified device number IN

■Visual Basic .NET

Format	ret = mdDevSetEx(path,netno,stno,devtyp,devno)		
Argument	Integer	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Integer	netno	;Network number IN
	Integer	stno	;Station number IN
	Integer	devtyp	;Device type IN
	Integer	devno	;Specified device number IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdDevSetEx(path&,netno%,stno%,devtyp%,devno%)		
Argument	Long	ret	;Return value OUT
	Long	path	;Path of channel IN
	Long	netno	;Network number IN
	Long	stno	;Station number IN
	Long	devtyp	;Device type IN
	Long	devno	;Specified device number IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Specify the network number of a target station. ( Page 48 Specifying network numbers and station numbers for extended functions)
stno	Specify the station number of a target station. ( Page 48 Specifying network numbers and station numbers for extended functions)
devtyp	Specify the type of device to be set (ON). ( Page 50 Specifying device types)
devno	Specify the device number of device to be set (ON).

■Explanation

This function is a dedicated function for bit devices such as link relays (B) and internal relays (M).

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdDevRstEx()

mdDevRstEx (resetting extended bit devices)

Reset the specified bit device on the target station (to OFF).

Format

■Visual C++

Format	ret = mdDevRstEx(path,netno,stno,devtyp,devno);		
Argument	long	ret;	//Return value OUT
	long	path;	//Path of channel IN
	long	netno;	//Network number IN
	long	stno;	//Station number IN
	long	devtyp;	//Device type IN
	long	devno;	//Specified device number IN

■Visual C#

Format	ret = [Class name].mdDevRstEx(path,netno,stno,devtyp,devno);		
Argument	int	ret;	//Return value OUT
	int	path;	//Path of channel IN
	int	netno;	//Network number IN
	int	stno;	//Station number IN
	int	devtyp;	//Device type IN
	int	devno;	//Specified device number IN

■Visual Basic .NET

Format	ret = mdDevRstEx(path,netno,stno,devtyp,devno)		
Argument	Integer	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Integer	netno	;Network number IN
	Integer	stno	;Station number IN
	Integer	devtyp	;Device type IN
	Integer	devno	;Specified device number IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdDevRstEx(path&,netno%,stno%,devtyp%,devno%)		
Argument	Long	ret	;Return value OUT
	Long	path	;Path of channel IN
	Long	netno	;Network number IN
	Long	stno	;Station number IN
	Long	devtyp	;Device type IN
	Long	devno	;Specified device number IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Specify the network number of a target station. (📖 Page 48 Specifying network numbers and station numbers for extended functions)
stno	Specify the station number of a target station. (📖 Page 48 Specifying network numbers and station numbers for extended functions)
devtyp	Specify the type of device to be reset (OFF). (📖 Page 50 Specifying device types)
devno	Specify the device number of device to be reset (OFF).

■Explanation

This function is a dedicated function for bit devices such as link relays (B) and internal relays (M).

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. (📖 Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdDevSetEx()

mdRemBufWriteEx (writing data to the buffer memory of a remote device station/remote station)

Write data to the buffer memory of a target station (remote device station on CC-Link IE Field Network or remote station on CC-Link IE TSN).

Format

■Visual C++

Format	ret = mdRemBufWriteEx(path,netno,stno,offset,size,data);			
Argument	long	ret;	//Return value	OUT
	long	path;	//Path of channel	IN
	long	netno;	//Network number	IN
	long	stno;	//Station number	IN
	long	offset;	//Offset	IN
	long	*size;	//Written byte size	IN/OUT
	short	data[];	//Written data (single-precision integer array)	IN

■Visual C#

Format	ret = [Class name].mdRemBufWriteEx(path,netno,stno,offset,ref size,data);			
Argument	int	ret;	//Return value	OUT
	int	path;	//Path of channel	IN
	int	netno;	//Network number	IN
	int	stno;	//Station number	IN
	int	offset;	//Offset	IN
	int	size;	//Written byte size	IN/OUT
	short	data[];	//Written data (single-precision integer array)	IN

■Visual Basic .NET

Format	ret = mdRemBufWriteEx(path,netno,stno,offset,size,data(0))			
Argument	Integer	ret	:Return value	OUT
	Integer	path	:Path of channel	IN
	Integer	netno	:Network number	IN
	Integer	stno	:Station number	IN
	Integer	offset	:Offset	IN
	Integer	size	:Written byte size	IN/OUT
	Short	data(n)	:Written data (single-precision integer array)	IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Specify the network number of a target station. ( Page 48 Specifying network numbers and station numbers for extended functions)
stno	Specify the station number of a target station. ( Page 48 Specifying network numbers and station numbers for extended functions)
offset	Specify the start address of the buffer memory of a target station to which data is written.
size	Specify the byte size to be written in even number (2 to 480). If a specified byte size is out of the range (-5: size error), '0' is returned to size.
data	Specify the data to be written in single-precision integer array.

■Explanation

- This function writes data only to the buffer memory of a remote device station on CC-Link IE Field Network or a remote station on CC-Link IE TSN.
Data cannot be written to the buffer memory of an intelligent function module controlled by a head module on CC-Link IE Field Network.
When writing data to the buffer memory of an intelligent function module controlled by a head module on CC-Link IE Field Network, use the mdSendEx/mdSend function.
- Make sure that the write byte size from an offset does not exceed the buffer memory range of a target station. If it is exceeded, a timeout error may occur or the target station may not operate normally. For the buffer memory range of a target station, refer to the manual for the station.
- When using a 64-bit version user application to access a remote device station from a CC-Link IE Field Network board, use SW1DNC-CCIEF-B with version 1.12N or later.

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdRemBufReadEx()

mdRemBufReadEx (reading data from the buffer memory of a remote device station/remote station)

Read data from the buffer memory of a target station (remote device station on CC-Link IE Field Network or remote station on CC-Link IE TSN).

Format

■Visual C++

Format	ret = mdRemBufReadEx(path,netno,stno,offset,size,data);		
Argument	long	ret;	//Return value OUT
	long	path;	//Path of channel IN
	long	netno;	//Network number IN
	long	stno;	//Station number IN
	long	offset;	//Offset IN
	long	*size;	//Read byte size IN/OUT
	short	data[];	//Read data (single-precision integer array) OUT

■Visual C#

Format	ret = [Class name].mdRemBufReadEx(path,netno,stno,offset,ref size,data);		
Argument	int	ret;	//Return value OUT
	int	path;	//Path of channel IN
	int	netno;	//Network number IN
	int	stno;	//Station number IN
	int	offset;	//Offset IN
	int	size;	//Read byte size IN/OUT
	short	data[];	//Read data (single-precision integer array) OUT

■Visual Basic .NET

Format	ret = mdRemBufReadEx(path,netno,stno,offset,size,data(0))		
Argument	Integer	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Integer	netno	;Network number IN
	Integer	stno	;Station number IN
	Integer	offset	;Offset IN
	Integer	size	;Read byte size IN/OUT
	Short	data(n)	;Read data (single-precision integer array) OUT

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Specify the network number of a target station. ( Page 48 Specifying network numbers and station numbers for extended functions)
stno	Specify the station number of a target station. ( Page 48 Specifying network numbers and station numbers for extended functions)
offset	Specify the start address of the buffer memory of a target station from which data is read.
size	Specify the byte size to be read in even number (2 to 480). If a specified byte size is out of the range (-5: size error), '0' is returned to size.
data	Read data is stored as a single-precision integer array.

■Explanation

- This function reads data only from the buffer memory of a remote device station on CC-Link IE Field Network or a remote station on CC-Link IE TSN.
Data cannot be read from the buffer memory of an intelligent function module controlled by a head module on CC-Link IE Field Network.
When reading data from the buffer memory of an intelligent function module controlled by a head module on CC-Link IE Field Network, use the mdReceiveEx/mdReceive function.
- Make sure that the read byte size from an offset does not exceed the buffer memory range of a target station. If it is exceeded, a timeout error may occur or an undefined value may be read. For the buffer memory range of a target station, refer to the manual for the station.
- When using a 64-bit version user application to access a remote device station from a CC-Link IE Field Network board, use SW1DNC-CCIEF-B with version 1.12N or later.

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdRemBufWriteEx()

mdRemBufWriteIPEx (writing data to the buffer memory of a remote station, target station IP address specified)

Write data to the buffer memory of a target station (remote station on CC-Link IE TSN).

Format

■Visual C++

Format	ret= mdRemBufWriteIPEx(path,ipaddress,offset,size,data);			
Argument	long	ret;	//Return value	OUT
	long	path;	//Path of channel	IN
	long	ipaddress;	//IP address	IN
	long	offset;	//Offset	IN
	long	*size;	//Written byte size	IN/OUT
	short	data[];	//Written data (single-precision integer array)	IN

■Visual C#

Format	ret = [Class name].mdRemBufWriteIPEx(path,ipaddress,offset,ref size,data);			
Argument	int	ret;	//Return value	OUT
	int	path;	//Path of channel	IN
	int	ipaddress;	//IP address	IN
	int	offset;	//Offset	IN
	int	size;	//Written byte size	IN/OUT
	short	data[];	//Written data (single-precision integer array)	IN

■Visual Basic .NET

Format	ret = mdRemBufWriteIPEx(path,ipaddress,offset,size,data(0))			
Argument	Integer	ret	;Return value	OUT
	Integer	path	;Path of channel	IN
	Integer	ipaddress	;IP address	IN
	Integer	offset	;Offset	IN
	Integer	size	;Written byte size	IN/OUT
	Short	data(n)	;Written data (single-precision integer array)	IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
ipaddress	Specify the IP address of a target station in hexadecimal (0x00000001 to 0xFFFFFFFFFE).
offset	Specify the start address of the buffer memory of a target station to which data is written.
size	Specify the byte size to be written in even number (2 to 480). If a specified byte size is out of the range (-5: size error), '0' is returned to size.
data	Specify the data to be written in single-precision integer array.

■Explanation

- This function writes data only to the buffer memory of a remote station on CC-Link IE TSN.
- Make sure that the write byte size from an offset does not exceed the buffer memory range of a target station. If it is exceeded, a timeout error may occur or the target station may not operate normally. For the buffer memory range of a target station, refer to the manual for the station.

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdRemBufReadIPEX()

IP address specification method

Specify an IP address as follows:

b31 to b24	b23 to b16	b15 to b8	b7 to b0
(1)	(2)	(3)	(4)

(1): First octet

(2): Second octet

(3): Third octet

(4): Fourth octet

Ex.

When the IP address is '192.168.3.4,' it will be '0xC0A80304.'

mdRemBufReadIPEX (reading data from the buffer memory of a remote station, target station IP address specified)

Read data from the buffer memory of a target station (remote station on CC-Link IE TSN).

Format

■Visual C++

Format	ret= mdRemBufReadIPEX(path,ipaddress,offset,size,data);		
Argument	long	ret;	//Return value OUT
	long	path;	//Path of channel IN
	long	ipaddress;	//IP address IN
	long	offset;	//Offset IN
	long	*size;	//Read byte size IN/OUT
	short	data[];	//Read data (single-precision integer array) OUT

■Visual C#

Format	ret = [Class name].mdRemBufReadIPEX(path,ipaddress,offset,ref size,data);		
Argument	int	ret;	//Return value OUT
	int	path;	//Path of channel IN
	int	ipaddress;	//IP address IN
	int	offset;	//Offset IN
	int	size;	//Read byte size IN/OUT
	short	data[];	//Read data (single-precision integer array) OUT

■Visual Basic .NET

Format	ret = mdRemBufReadIPEX(path,ipaddress,offset,size,data(0))		
Argument	Integer	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Integer	ipaddress	;IP address IN
	Integer	offset	;Offset IN
	Integer	size	;Read byte size IN/OUT
	Short	data(n)	;Read data (single-precision integer array) OUT

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
ipaddress	Specify the IP address of a target station in hexadecimal (0x00000001 to 0xDFFFFFFE).
offset	Specify the start address of the buffer memory of a target station from which data is read.
size	Specify the byte size to be read in even number (2 to 480). If a specified byte size is out of the range (-5: size error), '0' is returned to size.
data	Read data is stored as a single-precision integer array.

■Explanation

- This function reads data only from the buffer memory of a remote station on CC-Link IE TSN.
- Make sure that the read byte size from an offset does not exceed the buffer memory range of a target station. If it is exceeded, a timeout error may occur or an undefined value may be read. For the buffer memory range of a target station, refer to the manual for the station.

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdRemBufWriteIPEX()

IP address specification method

Specify an IP address as follows:

b31 to b24	b23 to b16	b15 to b8	b7 to b0
(1)	(2)	(3)	(4)

(1): First octet

(2): Second octet

(3): Third octet

(4): Fourth octet

Ex.

When the IP address is '192.168.3.4,' it will be '0xC0A80304.'

mdTypeRead (reading model names of CPU)

Read a model name code of the CPU on the target station.

Format

■Visual C++

```
Format      ret = mdTypeRead(path,stno,buf);
Argument    short      ret;          //Return value          OUT
            long       path;        //Path of channel      IN
            short      stno;        //Station number      IN
            short      *buf;        //Model name code     OUT
```

■Visual C#

```
Format      ret = [Class name].mdTypeRead(path,stno,out buf);
Argument    short      ret;          //Return value          OUT
            int        path;        //Path of channel      IN
            short      stno;        //Station number      IN
            short      buf;         //Model name code     OUT
```

■Visual Basic .NET

```
Format      ret = mdTypeRead(path,stno,buf)
Argument    Short      ret          ;Return value          OUT
            Integer    path         ;Path of channel      IN
            Short      stno         ;Station number      IN
            Short      buf          ;Model name code     OUT
```

■Visual Basic 5.0, Visual Basic 6.0

```
Format      ret% = mdTypeRead(path&,stno%,buf%)
Argument    Integer    ret          ;Return value          OUT
            Long       path         ;Path of channel      IN
            Integer    stno         ;Station number      IN
            Integer    buf          ;Model name code     OUT
```

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Specify the station number of a target station. ( Page 46 Specifying station numbers)
buf	Return the model name code of CPU being read. ( Page 84 List of Model code and CPU model)

■Explanation

The following table shows the model name codes and corresponding model names. ( Page 84 List of Model code and CPU model)

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose()

List of Model code and CPU model

The following model name codes are stored.

If specifying an unsupported model as a target station, a model name code not in the list may be stored.

■MELSEC iQ-R series

Model name code (Hexadecimal)	Model name
4800H	R04CPU
4801H	R08CPU
4802H	R16CPU
4803H	R32CPU
4804H	R120CPU
4805H	R04ENCPU
4806H	R08ENCPU
4807H	R16ENCPU
4808H	R32ENCPU
4809H	R120ENCPU
4820H	R12CCPU-V
4841H	R08PCPU
4842H	R16PCPU
4843H	R32PCPU
4844H	R120PCPU
4851H	R08PSFCPU
4852H	R16PSFCPU
4853H	R32PSFCPU
4854H	R120PSFCPU
4860H	RJ72GF15-T2
4861H	RJ72GF15-T2 (redundant system (single line))
4862H	RJ72GF15-T2 (redundant system (redundant line))
4891H	R08SFCPU
4892H	R16SFCPU
4893H	R32SFCPU
4894H	R120SFCPU
48A0H	R00CPU
48A1H	R01CPU
48A2H	R02CPU
4C20H	R102WCPU-W

■MELSEC-L series

Model name code (Hexadecimal)	Model name
0541H	L02CPU
0543H	L02SCPU
0544H	L06CPU
0545H	L26CPU
0548H	L26CPU-BT
0549H	L02CPU-P
054AH	L26CPU-PBT
054BH	L26CPU-P
054CH	L02SCPU-P
054DH	L06CPU-P
0641H	LJ72GF15-T2

■MELSEC-Q series

Model name code (Hexadecimal)	Model name
0041H	Q02CPU, Q02HCPU
0042H	Q06HCPU
0043H	Q12HCPU
0044H	Q25HCPU
0049H	Q12PHCPU
004AH	Q25PHCPU
004BH	Q12PRHCPU
004CH	Q25PRHCPU
004DH	Q02PHCPU
004EH	Q06PHCPU
0141H	Q02CPU (A mode), Q02HCPU (A mode)
0142H	Q06HCPU (A mode)
0250H	Q00JCPU
0251H	Q00CPU
0252H	Q01CPU
0260H	Q00UJCPU
0261H	Q00UCPU
0262H	Q01UCPU
0263H	Q02UCPU
0266H	Q10UDHCPU
0267H	Q20UDHCPU
0268H	Q03UDCPU
0269H	Q04UDHCPU
026AH	Q06UDHCPU
026BH	Q13UDHCPU
026CH	Q26UDHCPU
02E6H	Q10UDEHCPU
02E7H	Q20UDEHCPU
02E8H	Q03UDECPU
02E9H	Q04UDEHCPU
02EAH	Q06UDEHCPU
02EBH	Q13UDEHCPU
02ECH	Q26UDEHCPU
02EDH	Q50UDEHCPU
02EEH	Q100UDEHCPU
0362H	Q04UDPVCPU
0363H	Q06UDPVCPU
0364H	Q13UDPVCPU
0365H	Q26UDPVCPU
0366H	Q03UDVCPU
0367H	Q04UDVCPU
0368H	Q06UDVCPU
036AH	Q13UDVCPU
036CH	Q26UDVCPU
2010H	Q172CPU, Q172CPUN, Q172CPUN-T
2012H	Q172HCPU
2011H	Q173CPU, Q173CPUN, Q173CPUN-T
2013H	Q173HCPU
2014H	Q172DCPU
2015H	Q173DCPU
2018H	Q172DSCPU
2019H	Q173DSCPU

■MELSEC-QnA series

Model name code (Hexadecimal)	Model name
0021H	Q2ACPU, Q2AHCPU, Q2ASCPU, Q2ASHCPU
0022H	Q2ACPU-S1, Q2AHCPU-S1, Q2ASCPU-S1, Q2ASHCPU-S1
0023H	Q3ACPU
0024H	Q4ACPU, Q4ARCPU

■MELSEC-A series

Model name code (Hexadecimal)	Model name
0082H	A2UCPU, A2USCPU
0083H	A2UCPU-S1, A2USCPU-S1
0084H	A3UCPU, A2USHCPU-S1
0085H	A4UCPU
0092H	A2ACPU
0093H	A2ACPU-S1
0094H	A3ACPU
0098H	A0J2HCPU, A1SCPU, A1SJCPU
009AH	A2CCPU, A2CJCPU
00A0H	A0J2CPU
00A1H	A1CPU, A1NCP
00A2H	A2CPU, A2NCP, A2SCPU
00A3H	A3CPU, A3NCP, A1SHCPU, A1SJHCPU, A2SHCPU, A2SH1CPU
00A4H	A3HCPU, A3MCP

■Others

Model name code (Hexadecimal)	Model name
0090H	Q80BD-J71GF11-T2, Q81BD-J71GF11-T2, Q80BD-J71GP21-SX, Q80BD-J71GP21S-SX, Q81BD-J71GP21-SX, Q81BD-J71GP21S-SX, Q80BD-J71LP21-25, Q81BD-J71LP21-25, Q80BD-J71LP21S-25, Q80BD-J71LP21G, Q80BD-J71LP21GE, Q80BD-J71BR11, Q80BD-J61BT11N, Q81BD-J61BT11, NZ81GN11-SX, NZ81GN11-T2
0642H	NZ2GF-ETB

mdControl (remote RUN/STOP/PAUSE)

Remotely operate a CPU on the target station. (Remote RUN/STOP/PAUSE)

Format

■Visual C++

Format	ret = mdControl(path,stno,buf);			
Argument	short	ret;	//Return value	OUT
	long	path;	//Path of channel	IN
	short	stno;	//Station number	IN
	short	buf;	//Command code	IN

■Visual C#

Format	ret = [Class name].mdControl(path,stno,buf);			
Argument	short	ret;	//Return value	OUT
	int	path;	//Path of channel	IN
	short	stno;	//Station number	IN
	short	buf;	//Command code	IN

■Visual Basic .NET

Format	ret = mdControl(path,stno,buf)			
Argument	Short	ret	;Return value	OUT
	Integer	path	;Path of channel	IN
	Short	stno	;Station number	IN
	Short	buf	;Command code	IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdControl(path&,stno%,buf%)			
Argument	Integer	ret	;Return value	OUT
	Long	path	;Path of channel	IN
	Integer	stno	;Station number	IN
	Integer	buf	;Command code	IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Specify the station number of a target station. ( Page 46 Specifying station numbers)
buf	Specify the command code (remote RUN/STOP/PAUSE) for remote operation of CPU. ( Page 87 Explanation)

■Explanation

The following table shows the command codes and the corresponding descriptions.

Command code (Hexadecimal)	Description
0	Remote RUN
1	Remote STOP
2	Remote PAUSE

This function cannot be executed for C Controller modules or MELSECWinCPU modules.

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose()

mdWaitBdEvent (waiting for event occurrence)

Wait an occurrence of event until the time out.

Format

■Visual C++

Format	ret = mdWaitBdEvent(path, eventno, timeout, signaledno, details);		
Argument	short	ret;	//Return value OUT
	long	path;	//Path of channel IN
	short	eventno[];	//Waiting event number IN
	long	timeout;	//Timeout value IN
	short	*signaledno;	//Driven event number OUT
	short	details[4];	//Event detail information OUT

■Visual Basic .NET

Format	ret = mdWaitBdEvent(path, eventno(0), timeout, signaledno, details(0))		
Argument	Short	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Short	eventno(n)	;Waiting event number IN
	Integer	timeout	;Timeout value IN
	Short	signaledno	;Driven event number OUT
	Short	details(4)	;Event detail information OUT

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdWaitBdEvent(path&,eventno%(0),timeout&,signaledno%,details%(0))		
Argument	Integer	ret	;Return value OUT
	Long	path	;Path of channel IN
	Integer	eventno(n)	;Waiting event number IN
	Long	timeout	;Timeout value IN
	Integer	signaledno	;Driven event number OUT
	Integer	details(4)	;Event detail information OUT

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
eventno	Specify the event number of event to wait. (Use the event number set in the utility.)
timeout	Specify the time until time out to wait the event.
signaledno	Return the occurred event number.
details	Store the bit pattern or device value when the conditions of device are satisfied. Reserve the variable for 4 words.

■Explanation

This function waits the occurrence of the event specified by the waiting event number on the specified channel for a period of time out.

Timeout value	Description
-1	Indefinitely waiting
0	No wait
1 to 2147483647 (7FFFFFFFH)	Millisecond waiting

Waiting event number (eventno) is specified as follows.

Waiting event (eventno)	Specified data
eventno[0]	Number of waiting events (1 to 64)
eventno[1]	Waiting event number (0 to 63) ... 1st event
eventno[2]	Waiting event number (0 to 63) ... 2nd event
.	.
eventno[64]	Waiting event number (0 to 63) ... 64th event

The following is an example when waiting the multiple events simultaneously.

Ex.

When waiting the event No. 1, No. 5, and No. 12 simultaneously.

Waiting event (eventno)	Specified data
eventno[0]=3;	Number of waiting events (3)
eventno[1]=1;	Waiting event number (1) ... 1st event
eventno[2]=5;	Waiting event number (5) ... 2nd event
eventno[3]=12;	Waiting event number (12) ... 3rd event

When the No. 5 interrupt event occurs, 5 is returned to "signaledno".

Values stored in the event detail information (details) differ depending on whether the specified device of the occurred event condition is a bit device or a word device. For details, refer to the following:

📖 Page 90 Values stored to the event detail information (details)

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. (📖 Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose()

Point

- If the data link is not executed properly, an event does not occur.
- When the device assigned by "Network range assignment" is not used, an event does not occur.
- This function returns immediately if the event has already occurred when it is called.
- Create a program to avoid calling this function from multiple processes with specifying same event number of the same channel number simultaneously. If executed, "Event number registration overlap error -63 (FFC1H)" of return value is returned to the process that called this function later.
- Lengthen the occurrence interval of the event up to extent where the user program can process satisfactorily.
- This function returns "The event is not set error -67 (FFBDH)" of return value if executed without the event setting.
- When the board or the master station (control station) is reset during the execution of this function, the function returns "Event initialization error -66 (FFBEH)" of return value.
- When multiple or more events occur simultaneously, either of events is detected first. When this function is executed again, the other events are detected.
- This function is a dedicated function for CC-Link IE Controller Network board and CC-Link IE Field Network board. If channel path of another board is specified, an error occurs. (📖 Page 140 ERROR CODES)

Values stored to the event detail information (details)

Values stored to the event detail information (details) are as follows depending on whether the specified device of the occurred event condition is a bit device or a word device.

■Bit device

A bit pattern which indicates the condition-satisfied device is stored to the event detail information.

- 1 is set to the bits that correspond to the condition-satisfied device.
- 1 is set to the first bit at the condition satisfaction because device points cannot be specified with the CC-Link IE Field Network board.
- 0 is set to the bits other than the corresponding bits when the device point is less than 64.

Event detail information (details)	Number of points from the start device	Bit pattern to be stored
details[0]	1st to 16th point	bit0: 1st point ... bit15: 16th point
details[1]	17th to 32nd point	bit0: 17th point ... bit15: 32nd point
details[2]	33rd to 48th point	bit0: 33rd point ... bit15: 48th point
details[3]	49th to 64th point	bit0: 49th point ... bit15: 64th point

■Word device

Device values at the condition satisfaction are stored to the event detail information.

Event detail information (details)	Device value to be stored
details[0]	Device value when condition is satisfied.
details[1]	0
details[2]	0
details[3]	0

■RECV function (CC-Link IE Field Network board)

The information on received data is stored to the event detail information.

Event detail information (details)	Device value to be stored
details[0]	Receive channel (1 to 2)
details[1]	0
details[2]	0
details[3]	0

When the event does not occur within the timeout time, a timeout error is returned as a return value.

When the same event number occurs multiple times

When the events whose number is the same occur multiple times, "Execution result of this function" and "Value details[] stored" of "Bit device" and "Word device" are as follows.

■Bit device

- This function ends normally.
- "Logical add in the bit pattern of each driven event" is stored in "details[]".

■Word device

- This function returns a value of "Event overlapped occurrence error -70 (FFBAH)."
- The device value when the condition of the event that occurred first is satisfied is stored in "details[0]".

mdBdRst (resetting board)

Reset a board.

Format

■Visual C++

Format	ret = mdBdRst(path);			
Argument	short	ret;	//Return value	OUT
	long	path;	//Path of channel	IN

■Visual C#

Format	ret = [Class name].mdBdRst(path);			
Argument	short	ret;	//Return value	OUT
	int	path;	//Path of channel	IN

■Visual Basic .NET

Format	ret = mdBdRst(path)			
Argument	Short	ret	;Return value	OUT
	Integer	path	;Path of channel	IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdBdRst(path&)			
Argument	Integer	ret	;Return value	OUT
	Long	path	;Path of channel	IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)

■Explanation

- The board is reset to the current settings with the utility function. (The same process as that of the board reset operation of the utility function.)
- The reset execution error/firmware update execution error (9922H), which notifies that the board is reset, is returned to other applications accessing the reset board.

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. (📖 Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose()

mdBdModSet (setting modes of board)

Change the modes of a board temporarily.

Format

■Visual C++

Format	ret = mdBdModSet(path,mode);			
Argument	short	ret;	//Return value	OUT
	long	path;	//Path of channel	IN
	short	mode;	//Mode	IN

■Visual C#

Format	ret = [Class name].mdBdModSet(path,mode);			
Argument	short	ret;	//Return value	OUT
	int	path;	//Path of channel	IN
	short	mode;	//Mode	IN

■Visual Basic .NET

Format	ret = mdBdModSet(path,mode)			
Argument	Short	ret	;Return value	OUT
	Integer	path	;Path of channel	IN
	Short	mode	;Mode	IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdBdModSet(path&,mode%)			
Argument	Integer	ret	;Return value	OUT
	Long	path	;Path of channel	IN
	Integer	mode	;Mode	IN

Detailed specifications

Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
mode	Specify the code corresponding to the mode to be executed. (☞ Page 93 Explanation)

Explanation

- The board is reset when this function is executed.
- Reset the board or restart the personal computer to recover the mode set in the utility.
- The following table shows the modes and the corresponding codes.

—: N/A

Code	Description				
	CC-Link IE TSN	CC-Link IE Controller Network	CC-Link IE Field Network	CC-Link Ver.2	MELSECNET/H
0000H	Online	Online	Online (normal mode)	Online (with automatic return)	Online (with automatic return)
0001H	—	—	Online (high-speed mode)*1	—	—
0002H	Offline	Offline	Offline	Offline	Offline
0003H	—	—	—	Data link test	Forward loop test*2
0004H	—	—	—	Remote station test	Reverse loop test*2
0005H	—	Station-to-station test*2	—	—	Station-to-station test (executing stations)*2
0006H	—	Loop test*2	Loop test*1,*3	Hardware test	Station-to-station test (target stations)*2
0007H	—	Self-loopback test*2	Self-loopback test*3	Not applicable	Self-loopback test*2
0008H	—	—	—	—	Self-loopback test (internal)*2
0009H	—	Hardware test*2	Hardware test*3	—	Hardware test*2
000BH	Board communication test*2	—	—	—	—
000EH	—	Bus interface test*2	Bus interface test*4	—	—
0010H	—	—	Memory test*4	—	—

*1 Cannot be set when the target board is local station.

*2 For the test method and check method of the test result, refer to the manuals of each board.

*3 For the test method, refer to the manuals of each board. For the test result, check with the values of the link special relays/link special registers of each board.

*4 Check the test result with the return value.

Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. (☞ Page 140 ERROR CODES)

Related function

mdOpen(), mdClose(), mdBdModRead()

mdBdModRead (reading modes of board)

Read the mode of a board set in the utility.

Format

■Visual C++

Format	ret = mdBdModRead(path,mode);			
Argument	short	ret;	//Return value	OUT
	long	path;	//Path of channel	IN
	short	*mode;	//Mode	OUT

■Visual C#

Format	ret = [Class name].mdBdModRead(path,out mode);			
Argument	short	ret;	//Return value	OUT
	int	path;	//Path of channel	IN
	short	mode;	//Mode	OUT

■Visual Basic .NET

Format	ret = mdBdModRead(path,mode)			
Argument	Short	ret	;Return value	OUT
	Integer	path	;Path of channel	IN
	Short	mode	;Mode	OUT

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdBdModRead(path&,mode%)			
Argument	Integer	ret	;Return value	OUT
	Long	path	;Path of channel	IN
	Integer	mode	;Mode	OUT

Detailed specifications

Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
mode	Return the corresponding code to the mode being read. ( Page 95 Explanation)

Explanation

- When the mdBdModSet() function is executed and the mode of the board is changed temporarily, the mode set in the utility is read instead of the currently operating mode.
- The following table shows the modes and the corresponding codes.

—: N/A

Code	Description				
	CC-Link IE TSN	CC-Link IE Controller Network	CC-Link IE Field Network	CC-Link Ver.2	MELSECNET/H
0000H	Online	Online	Online (normal mode)	Online (with automatic return)	Online (with automatic return)
0001H	—	—	Online (high-speed mode)	—	—
0002H	Offline	Offline	Offline	Offline	Offline
0003H	—	—	—	Data link test	Forward loop test
0004H	—	—	—	Remote station test	Reverse loop test
0005H	—	Station-to-station test	—	—	Station-to-station test (executing stations)
0006H	—	Loop test	Loop test	Hardware test	Station-to-station test (target stations)
0007H	—	Self-loopback test	Self-loopback test	Not applicable	Self-loopback test
0008H	—	—	—	—	Self-loopback test (internal)
0009H	—	Hardware test	Hardware test	—	Hardware test
000BH	Board communication test	—	—	—	—
000EH	—	Bus interface test	Bus interface test	—	—
0010H	—	—	Memory test	—	—

Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

Related function

mdOpen(), mdClose(), mdBdModSet()

mdBdLedRead (reading LED information of the board)

Read the LED information of the board.

Format

■Visual C++

Format	ret = mdBdLedRead(path,buf);		
Argument	short	ret;	//Return value OUT
	long	path;	//Path of channel IN
	short	buf[];	//Read data OUT

■Visual C#

Format	ret = [Class name].mdBdLedRead(path,buf);		
Argument	short	ret;	//Return value OUT
	int	path;	//Path of channel IN
	short	buf[];	//Read data OUT

■Visual Basic .NET

Format	ret = mdBdLedRead(path,buf(0))		
Argument	Short	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Short	buf(n)	;Read data OUT

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdBdLedRead(path&,buf%(0))		
Argument	Integer	ret	;Return value OUT
	Long	path	;Path of channel IN
	Integer	buf(n)	;Read data OUT

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
buf	Store the LED information being read. ( Page 97 List of LED information)

■Explanation

- For LED information, refer to the 'List of LED information'. ( Page 97 List of LED information)
- For CC-Link IE Controller Network boards, CC-Link IE Field Network boards, CC-Link Ver.2 boards, and MELSECNET/H boards, the status at read is stored when the LED is flashing.
For details on the LED, refer to the manual for each board.
- For the array variable of buf to which the read data is stored, reserve the total size of data for elements from the start to the end in each table ( Page 97 List of LED information). Include the element even if the end of which is RESERVE.

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose()

List of LED information

■CC-Link IE TSN

Element	Bit	LED	Description	Remarks
buff[0]	b15	P2 L ERR or P2 L ER	Display the port status. 1: Error data received or loopback in progress 0: Normal data received and loopback not performed	ON: 1 OFF: 0
	b14	P1 L ERR or P1 L ER		
	b13-b11	D.LINK	■Online mode Display the data link status (cyclic transmission status). 2: Cyclic transmission stopped 1: Cyclic transmission in progress 0: Disconnected ■Offline mode 0: OFF	Flashing for 1 s: 2 ON: 1 OFF: 0
	b10-b8	MST	Display the station type. 1: Operating as the master station 0: Operating as a local station	ON: 1 OFF: 0
	b7	P2 SD/RD	Display the data sending/receiving status. 1: Data being sent/received 0: Data not being sent/received	ON: 1 OFF: 0
	b6	P1 SD/RD		
	b5-3	ERR	■Online mode Display the error detection status. 4: Error occurring 3: Data link faulty station detected 1: Error occurring, or error detected in all stations 0: Operating normally ■Offline mode 0: OFF	Flashing for 200 s: 4 Flashing for 500 s: 3 ON: 1 OFF: 0
	b2-0	RUN	Display the operating status of board. 1: Operating normally 0: Error occurring	ON: 1 OFF: 0

■CC-Link IE Controller Network

—: No remarks

Element	Bit	LED	Description	Remarks
buff[0]	b15	EXT.PW	Display the external power supply status of CC-Link IE Controller Network module with external power supply function. 1: External power supply being supplied 0: External power supply not supplied	ON: 1 OFF: 0
	b14-b7	RESERVE	—	—
	b6	RD	Display the data reception status. 1: Data being received 0: Data not received	ON: 1 OFF: 0
	b5	D.LNK	Display the data link status (cyclic transmission status). 1: Data link in operation 0: No data link	
	b4	PRM	Display the station type. 1: Operating as a control station 0: Operating as other than a control station	
	b3	ERR.	Display the error detection status. 1: Detected 0: Not detected	
	b2	SD	Display the data sending status. 1: Data being sent 0: Data not sent	
	b1	MODE	Display the operating mode. 1: Operating in online mode 0: Operating in offline mode	
	b0	RUN	Display the operating status of board. 1: Operating normally 0: Hardware error or Board WDT error	
buff[1]	b15-b0	RESERVE	—	—

■CC-Link IE Field Network

—: No remarks

Element	Bit	LED	Description	Remarks
buff[0]	b15-b12	RESERVE	—	—
	b11	LNK2	Display the link status of PORT2. 1: Linkup in progress 0: Linkdown in progress	ON: 1 OFF: 0
	b10	LINK1	Display the link status of PORT1. 1: Linkup in progress 0: Linkdown in progress	
	b9	LER2	Display the frame loss status of PORT2. 1: Frame loss occurring 0: No frame loss	
	b8	LER1	Display the frame loss status of PORT1. 1: Frame loss occurring 0: No frame loss	
	b7	MODE	Display the operating mode. 1: Operating in online mode 0: Operating in offline mode	
	b6	RD	Display the network reception status. 1: Data being received 0: Data not received	
	b5	SD	Display the network sending status. 1: Data being sent 0: Data not sent	
	b4	L.ERR	Display the detection status of communication error. 1: Detected 0: Not detected	
	b3	ERR.	Display the detection status of network board error. 1: Detected 0: Not detected	
	b2	D.LINK	Display the data link status (cyclic transmission status). 1: Data link in operation 0: No data link	
	b1	MST	Display the station type. 1: Operating as the master station 0: Other than master station	
	b0	RUN	Display the operating status of board. 1: Operating normally 0: A board WDT error is occurring or the board is resetting	
buff[1]	b15-b0	RESERVE	—	—

■CC-Link Ver.2 Board

—: No remarks

Element	Bit	LED	Description	Remarks
buff[0]	b15-b9	RESERVE	—	—
	b8	ERR	Display the detection status of communication error for all stations. 1: Detected 0: Not detected	ON: 1 OFF: 0
	b7-b1	RESERVE	—	—
	b0	RUN	Display the operating status of board. 1: Operating normally 0: Board WDT error	ON: 1 OFF: 0
buff[1]	b15-b1	RESERVE	—	—
	b0	SW	Display the detection status of switch setting error. 1: Detected 0: Not detected	ON: 1 OFF: 0
buff[2]	b15-b9	RESERVE	—	—
	b8	PRM	Display the detection status of parameter error. 1: Detected 0: Not detected	ON: 1 OFF: 0
	b7-b1	RESERVE	—	—
	b0	M/S	Display the detection status of master station duplication error. 1: Detected 0: Not detected	ON: 1 OFF: 0
buff[3]	b15-b9	RESERVE	—	—
	b8	LINE	Display the detection status of cable disconnection error. 1: Detected 0: Not detected	ON: 1 OFF: 0
	b7-b1	RESERVE	—	—
	b0	TIME	Display the error detection status when the data link monitoring timer is operated. 1: Detected 0: Not detected	ON: 1 OFF: 0
buff[4]	b15-b9	RESERVE	—	—
	b8	L.ERR	Display the detection status of communication error. 1: Detected 0: Not detected	ON: 1 OFF: 0
	b7-b1	RESERVE	—	—
	b0	L.RUN	Display the data link execution status. 1: Data link in operation 0: No data link	ON: 1 OFF: 0
buff[5]	b15-b9	RESERVE	—	—
	b8	RD	Display the data reception status. 1: Data being received 0: Data not received	ON: 1 OFF: 0
	b7-b1	RESERVE	—	—
	b0	SD	Display the data sending status. 1: Data being sent 0: Data not sent	ON: 1 OFF: 0

Element	Bit	LED	Description	Remarks
buff[0]	b15	R.LOOP OVER	Display the detection status of data import delay error on the reverse loop side. 1: Detected 0: Not detected	ON: 1 OFF: 0
	b14	F.LOOP CRC	Display the detection status of code check error on the forward loop side. 1: Detected 0: Not detected	
	b13	R.LOOP CRC	Display the detection status of code check error on the reverse loop side. 1: Detected 0: Not detected	
	b12	F.LOOP	Display the detection status of loop line error on the forward loop side. 1: Detected 0: Not detected	
	b11	R.LOOP	Display the detection status of loop line error on the reverse loop side. 1: Detected 0: Not detected	
	b10	F.LOOP RD	Display the data reception status on the forward loop side. 1: Data being received 0: Data not received	
	b9	R.LOOP RD	Display the data reception status on the reverse loop side. 1: Data being received 0: Data not received	
	b8	F.LOOP SD	Display the data sending status on the forward loop side. 1: Data being sent 0: Data not sent	
	b7	R.LOOP SD	Display the data sending status on the reverse loop side. 1: Data being sent 0: Data not sent	
	b6	S.MNG	Display the station type. 1: Operating as a sub control station 0: Other than sub control station	
	b5	MNG	Display the station type. 1: Operating as a control station 0: Other than control station	
	b4	D.LINK	Display the data link status (cyclic transmission status). 1: Data link in operation 0: No data link	
	b3	T.PASS	Display the baton pass status. 1: Baton pass in progress 0: No baton pass	
	b2	PRM.E.	Display the detection status of parameter error. 1: Detected 0: Not detected	
	b1	SW.E.	Display the detection status of switch setting error. 1: Detected 0: Not detected	
	b0	RUN	Display the operating status of board. 1: Operating normally 0: Hardware error or Board WDT error	

—: No remarks

Element	Bit	LED	Description	Remarks
buff[1]	b15-b10	RESERVE	—	—
	b9	M/S.E.	Display the detection status of station number or control station duplication error on the same network. 1: Detected 0: Not detected	ON: 1 OFF: 0
	b8	F.LOOP UNDER	Display the error detection status when the internal processing of send data is not performed on the forward loop side at regular intervals. 1: Detected 0: Not detected	
	b7	R.LOOP UNDER	Display the error detection status when the internal processing of send data is not performed on the reverse loop side at regular intervals. 1: Detected 0: Not detected	
	b6	F.LOOP DATA	Display the error detection status when the error data whose size is 2K bytes or more is received on the forward loop side. 1: Detected 0: Not detected	
	b5	R.LOOP DATA	Display the error detection status when the error data whose size is 2K bytes or more is received on the reverse loop side. 1: Detected 0: Not detected	
	b4	F.LOOP TIME	Display the error detection status when the data link monitoring timer is operated on the forward loop side. 1: Detected 0: Not detected	
	b3	R.LOOP TIME	Display the error detection status when the data link monitoring timer is operated on the reverse loop side. 1: Detected 0: Not detected	
	b2	F.LOOP AB.IF	Display the error detection status when receiving "1" more than specified number of times continuously, or the received data length is too short on the forward loop side. 1: Detected 0: Not detected	
	b1	R.LOOP AB.IF	Display the error detection status when receiving "1" more than specified number of times continuously, or the received data length is too short on the reverse loop side. 1: Detected 0: Not detected	
	b0	F.LOOP OVER	Display the detection status of data import delay error on the forward loop side. 1: Detected 0: Not detected	

mdBdSwRead (reading switch status of the board)

Read a board switch status (such as station number setting, board number setting, board identification, and I/O address setting information).

Format

■Visual C++

```
Format      ret = mdBdSwRead(path,buf);
Argument    short      ret;          //Return value          OUT
            long       path;        //Path of channel      IN
            short      buf[];       //Read data            OUT
```

■Visual C#

```
Format      ret = [Class name].mdBdSwRead(path,buf);
Argument    short      ret;          //Return value          OUT
            int         path;        //Path of channel      IN
            short      buf[];       //Read data            OUT
```

■Visual Basic .NET

```
Format      ret = mdBdSwRead(path,buf(0))
Argument    Short      ret          ;Return value          OUT
            Integer     path        ;Path of channel      IN
            Short        buf(n)     ;Read data            OUT
```

■Visual Basic 5.0, Visual Basic 6.0

```
Format      ret% = mdBdSwRead(path&,buf%(0))
Argument    Integer     ret          ;Return value          OUT
            Long         path        ;Path of channel      IN
            Integer     buf(n)       ;Read data            OUT
```

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
buf	Store the board switch status being read. ( Page 103 Explanation)

■Explanation

The board switch status (such as station number setting, board number setting, board identification, I/O address setting information) is read.

Element	Item	Description				
		CC-Link IE TSN	CC-Link IE Controller Network	CC-Link IE Field Network	CC-Link Ver.2	MELSECNET/H
buf[0]	Setting value of station number	0 to 120	1 to 120	0 to 120	0 to 64 (0: master station)	1 to 64
buf[1]	Setting value of group number	0 to 32	0 to 32	0 fixed	0 fixed	MNET/10 mode: 0 to 9 MNET/H mode: 0 to 32
buf[2]	Setting value of network number	1 to 239	1 to 239	1 to 239	0 fixed	1 to 239
buf[3]	RESERVE	0 fixed	0 fixed	0 fixed	0 fixed	0 fixed
buf[4]	RESERVE	0 fixed	0 fixed	0 fixed	0 fixed	0 fixed
buf[5]	RESERVE	0 fixed	0 fixed	0 fixed	0 fixed	0 fixed

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose()

mdBdVerRead (reading version information of the board)

Read the version information of the board.

Format

■Visual C++

Format	ret = mdBdVerRead(path,buf);			
Argument	short	ret;	//Return value	OUT
	long	path;	//Path of channel	IN
	short	buf[];	//Read data	OUT

■Visual C#

Format	ret = [Class name].mdBdVerRead(path,buf);			
Argument	short	ret;	//Return value	OUT
	int	path;	//Path of channel	IN
	short	buf[];	//Read data	OUT

■Visual Basic .NET

Format	ret = mdBdVerRead(path,buf(0))			
Argument	Short	ret	;Return value	OUT
	Integer	path	;Path of channel	IN
	Short	buf(n)	;Read data	OUT

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdBdVerRead(path&,buf%(0))			
Argument	Integer	ret	;Return value	OUT
	Long	path	;Path of channel	IN
	Integer	buf(n)	;Read data	OUT

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
buf	Store the version information of board being read. ( Page 105 Explanation)

■Explanation

The following table shows the details of the version information.

Offset (word)	Item	Description				
		CC-Link IE TSN	CC-Link IE Controller Network	CC-Link IE Field Network	CC-Link Ver.2	MELSECNET/H
00H	Fixed value	'S', 'G' fixed				
01H	Checksum	Sum of 02H to 0FH (calculated in bytes)				
02H	Software version	'01' to '99'	'0A' to '9Z'	'0A' to '9Z'	'A' to 'ZZ'	'0A' to '9Z'
03H to 05H	Date	Date year / month / day Example) 2010/2/1 '1' '0' '0' '2' '0' '1'				
06H to 07H	Reservation area (4 bytes)	000H fixed				
08H to 0FH	Software model name (16 bytes)	'RJ71GN11-SX', 'RJ71GN11-T2'	'QJ71GP21-SX', 'QJ71GP21S-SX'	'QJ71GF11-T2'	'QJ61BT11N'	'QJ71LP21-25', 'QJ71LP21S-2', 'QJ71LP21G', 'QJ71BR11'
10H to 17H	Hardware model name (16 bytes)	'NZ81GN11-SX', 'NZ81GN11-T2'	'Q80BD-J71GP21', 'Q80BD-J71GP21S', 'Q81BD-J71GP21', 'Q81BD-J71GP21S'	'Q80BD-J71GF11-T2', 'Q81BD-J71GF11-T2'	'Q80BD-J61BT11N', 'Q81BD-J61BT11'	'Q80BD-J71LP21-25', 'Q81BD-J71LP21-25', 'Q80BD-J71LP21S-25', 'Q80BD-J71LP21G', 'Q80BD-J71LP21GE', 'Q80BD-J71BR11'
18H	2 port memory possession size	0080H (128 KB)	0180H (384 KB)	0080H (128 KB)	0200H (512 KB)	0080H (128 KB)
19H	2 port attribute	0080H fixed				
1AH	Applicable offset	0000H fixed				
1BH (L)	Machine classification (10 bytes)	0000H fixed	Function version ('A', 'B'...)	Function version ('A', 'B'...)	Function version ('A', 'B'...)	0000H fixed
1BH (H)			0000H fixed	0000H fixed	Major version of CC-Link (0002H)	
1CH (L)					Minor version of CC-Link (0000H)	
1CH (H) to 1FH					0000H fixed	

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. (📖 Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose()

mdInit (initializing programmable controller information table)

Refresh a programmable controller device address table which is the internal data of the MELSEC data link library.

Format

■Visual C++

Format	ret = mdInit(path);			
Argument	short	ret;	//Return value	OUT
	long	path;	//Path of channel	IN

■Visual C#

Format	ret = [Class name].mdInit(path);			
Argument	short	ret;	//Return value	OUT
	int	path;	//Path of channel	IN

■Visual Basic .NET

Format	ret = mdInit(path)			
Argument	Short	ret	;Return value	OUT
	Integer	path	;Path of channel	IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdInit(path&)			
Argument	Integer	ret	;Return value	OUT
	Long	path	;Path of channel	IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)

■Explanation

- Programmable controller device information acquired at the initial access to the programmable controller CPU is discarded.
- After the execution of the mdInit function, programmable controller device information is reacquired at the initial access to the programmable controller CPU. Therefore, after the execution of the mdInit function, a longer function execution time is required at the initial access.

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose()

mdSend (batch writing devices)

Batch write data to the devices on the target station for the number of written data bytes from the start device number.

Format

■Visual C++

Format	ret = mdSend(path,stno,devtyp,devno,size,data);		
Argument	short	ret;	//Return value OUT
	long	path;	//Path of channel IN
	short	stno;	//Station number IN
	short	devtyp;	//Device type IN
	short	devno;	//Start device number IN
	short	*size;	//Written byte size IN/OUT
	short	data[];	//Written data (single-precision integer array) IN

■Visual Basic .NET

Format	ret = mdSend(path,stno,devtyp,devno,size,data(0))		
Argument	Short	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Short	stno	;Station number IN
	Short	devtyp	;Device type IN
	Short	devno	;Start device number IN
	Short	size	;Written byte size IN/OUT
	Short	data(n)	;Written data (single-precision integer array) IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdSend(path&,stno%,devtyp%,devno%,size%,data%(0))		
Argument	Integer	ret	;Return value OUT
	Long	path	;Path of channel IN
	Integer	stno	;Station number IN
	Integer	devtyp	;Device type IN
	Integer	devno	;Start device number IN
	Integer	size	;Written byte size IN/OUT
	Integer	data(n)	;Written data (single-precision integer array) IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Specify the station number of a target station. (☞ Page 46 Specifying station numbers)
devtyp	Specify the type of device to which the data is written. (☞ Page 50 Specifying device types)
devno	Specify the start device number of device to which the data is written. - Specify the start device number with a multiple of 8 for the access to a bit device. (However, the start device number can be specified arbitrarily for accessing a contact and coil of the long timer and long retentive timer.) - Specify the start device number with a multiple of 16 (0, 16, 32...) for the access to a bit device (RX, RY, SB) of CC-Link other station link device.
size	Specify the byte size to be written. It must be specified according to devtyp as follows: - For a bit device: A multiple of 1 - For a word device: A multiple of 2 - For a double-word device: A multiple of 4 When the specified byte size to be written exceeds the device range (-5: size error), the applicable size is returned to "size".
data	Specify the data to be written in single-precision integer array.

■Explanation

- When the specified written byte size exceeds the transient transmission size, data are divided inside the function and written.
- When accessing another station, the extended comment information will be deleted by writing data to the block (extended file register) to which the extended comment is assigned.
- When accessing another station, the sub2 and sub3 programs will be deleted by writing data to the block (extended file register) which overlaps with the setting areas of the sub2 and sub3 programs.
- When a double word device is specified to "devtyp", store the data to be written to "data" as follows:

Ex.

When "devtyp" is LZ and "size" is 8

Array	Value
data(0)	Lower 1 word of LZ0
data(1)	Upper 1 word of LZ0
data(2)	Lower 1 word of LZ1
data(3)	Upper 1 word of LZ1

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. (👉 Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdSendEx(), mdReceive(), mdReceiveEx()

mdSend (SEND function)

Send data to the specified channel number of the target station.

Format

■Visual C++

Format	ret = mdSend(path,stno,devtyp,devno,size,data);		
Argument	short	ret;	//Return value OUT
	long	path;	//Path of channel IN
	short	stno;	//Station number IN
	short	devtyp;	//Device type IN
	short	devno;	//Channel number IN
	short	*size;	//Send byte size IN/OUT
	short	data[];	//Send data (single-precision integer array) IN

■Visual Basic .NET

Format	ret = mdSend(path,stno,devtyp,devno,size,data(0))		
Argument	Short	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Short	stno	;Station number IN
	Short	devtyp	;Device type IN
	Short	devno	;Channel number IN
	Short	size	;Send byte size IN/OUT
	Short	data(n)	;Send data (single-precision integer array) IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdSend(path&,stno%,devtyp%,devno%,size%,data%(0))		
Argument	Integer	ret	;Return value OUT
	Long	path	;Path of channel IN
	Integer	stno	;Station number IN
	Integer	devtyp	;Device type IN
	Integer	devno	;Channel number IN
	Integer	size	;Send byte size IN/OUT
	Any	data(n)	;Send data (single-precision integer array) IN

Detailed specifications

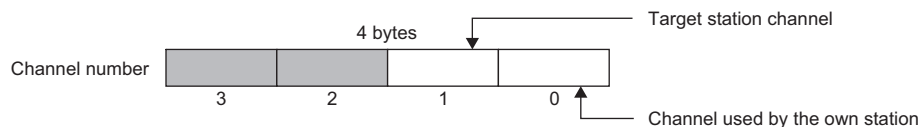
■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Specify the station number of a target station. (☞ Page 46 Specifying station numbers) • Logical station numbers cannot be specified.
devtyp	Specify whether to send data with arrival acknowledgment or without arrival acknowledgment. • With arrival acknowledgment*1: Specify 101 (65H) • Without arrival acknowledgment: Specify 102 (66H)
devno	Specify a channel used by the own station and a target station channel. (☞ Page 110 Explanation)
size	Specify the byte size of send data in even number. • Specify the byte size within the range of 2 to 1920. • When sending data from a MELSECNET/H board to a CC-Link IE Controller Network board or CC-Link IE Field Network board, specify it within the range of 2 to 960 bytes. • When accessing via MELSECNET/10 network, specify the byte size within the range of 2 to 960. If a specified size is an odd number or out of the range above, '0' is returned to size.
data	Specify the data to be written in single-precision integer array.

*1 Do not specify all stations or group numbers for the station number when sending data with arrival acknowledgment.

■Explanation

- This function supports the SEND instruction of the link dedicated instruction. For details of the functions , refer to the manuals of each board.
- The following explains how to specify the channel numbers.



Item	Description		
Target station channel	Specify a target station channel number.	• CC-Link IE Controller Network board, MELSECNET/H board	1 to 8
		• CC-Link IE Field Network board	1 to 2
Channel used by the own station	Specify a channel number used by the own station.	• CC-Link IE Controller Network board, MELSECNET/H board	1 to 8
		• CC-Link IE Field Network board	1 to 2

- The arguments of the mdSend function correspond to the control data (device) of the dedicated instruction (SEND) as shown below:

—: N/A

Device	Item	Corresponding argument and return value
+0	Execution/error completion type	devtyp
+1	Completion status	ret (return value)
+2	Channel used by the own station	devno
+3	Target station channel	devno
+4	Target station network number	stno
+5	Target station number	stno
+6	Not used	—
+7	Number of retransmissions (retries)	—
+8	Arrival monitoring time	—
+9	Send data length	size
+10	Not used	—
+11	Clock setting flag	—
+12	Clock data	—
+13		—
+14		—
+15		—
+16	Error-detected network number	—
+17	Error-detected station number	—

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdSendEx(), mdReceive(), mdReceiveEx()

mdReceive (batch reading devices)

Batch read data from the devices on the target station for the number of read data bytes from the start device number.

Format

■Visual C++

Format	ret = mdReceive(path,stno,devtyp,devno,size,data);		
Argument	short	ret;	//Return value OUT
	long	path;	//Path of channel IN
	short	stno;	//Station number IN
	short	devtyp;	//Device type IN
	short	devno;	//Start device number IN
	short	*size;	//Read byte size IN/OUT
	short	data[];	//Read data (single-precision integer array) OUT

■Visual Basic .NET

Format	ret = mdReceive(path,stno,devtyp,devno,size,data(0))		
Argument	Short	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Short	stno	;Station number IN
	Short	devtyp	;Device type IN
	Short	devno	;Start device number IN
	Short	size	;Read byte size IN/OUT
	Short	data(n)	;Read data (single-precision integer array) OUT

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdReceive(path&,stno%,devtyp%,devno%,size%,data%(0))		
Argument	Integer	ret	;Return value OUT
	Long	path	;Path of channel IN
	Integer	stno	;Station number IN
	Integer	devtyp	;Device type IN
	Integer	devno	;Start device number IN
	Integer	size	;Read byte size IN/OUT
	Integer	data(n)	;Read data (single-precision integer array) OUT

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Specify the station number of a target station. ( Page 46 Specifying station numbers)
devtyp	Specify the type of device from which the data is read. ( Page 50 Specifying device types)
devno	Specify the start device number of device range to be read. - Specify the start device number with a multiple of 8 for the access to a bit device. (However, the start device number can be specified arbitrarily for accessing a contact and coil of the long timer and long retentive timer.) - Specify the start device number with a multiple of 16 (0, 16, 32...) for the access to a bit device (RX, RY, SB) of CC-Link other station link device.
size	Specify the byte size to be read. It must be specified according to devtyp as follows: - For a bit device: A multiple of 1 - For a word device: A multiple of 2 - For a double-word device: A multiple of 4 When the specified byte size to be read exceeds the device range (-5: size error), the applicable size is returned to "size".
data	Store the data being read.

■Explanation

- When the specified read byte size exceeds the transient transmission size, data are divided inside the function and read.
- When a double word device is specified to "devtyp", the data is stored to "data" as follows:

Ex.

When "devtyp" is LZ and "size" is 8

Array	Value
data(0)	Lower 1 word of LZ0
data(1)	Upper 1 word of LZ0
data(2)	Lower 1 word of LZ1
data(3)	Upper 1 word of LZ1

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdReceiveEx(), mdSend(), mdSendEx()

mdReceive (RECV function)

Read data of the specified channel number from the data which are received by the own station.

Format

■Visual C++

Format	ret = mdReceive(path,stno,devtyp,devno,size,data);		
Argument	short	ret;	//Return value OUT
	long	path;	//Path of channel IN
	short	stno;	//Station number IN
	short	devtyp;	//Device type IN
	short	devno;	//Channel number IN
	short	*size;	//Receive byte size IN/OUT
	short	data[];	//Receive data with send source information (single-precision integer array) OUT

■Visual Basic .NET

Format	ret = mdReceive(path,stno,devtyp,devno,size,data(0))		
Argument	Short	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Short	stno	;Station number IN
	Short	devtyp	;Device type IN
	Short	devno	;Channel number IN
	Short	size	;Receive byte size IN/OUT
	Short	data(n)	;Receive data with send source information (single-precision integer array) OUT

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdReceive(path&,stno%,devtyp%,devno%,size%,data%(0))		
Argument	Integer	ret	;Return value OUT
	Long	path	;Path of channel IN
	Integer	stno	;Station number IN
	Integer	devtyp	;Device type IN
	Integer	devno	;Channel number IN
	Integer	size	;Receive byte size IN/OUT
	Integer	data(n)	;Receive data with send source information (single-precision integer array) OUT

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Specify the own station (255 (FFH)).
devtyp	Specify the RECV function (101 (65H)).
devno	Specify the own station channel number on which the received messages are stored. • CC-Link IE Controller Network board, MELSECNET/H board: 1 to 8 • CC-Link IE Field Network board: 1 to 2
size	Specify the byte size of received data in even number. • Specify the byte size within the range of 2 to 1920. • When receiving data sent from a MELSECNET/H board in a CC-Link IE Controller Network board or CC-Link IE Field Network board, specify it within the range of 2 to 960 bytes. • When accessing via MELSECNET/10 network, specify the byte size within the range of 2 to 960. The size of received actual data is received. If a specified size is an odd number or out of the range above, '0' is returned to size.
data	Received actual data and send source information are stored as single-precision integer array. Reserve an area for the specified receive byte size and send source information (6 bytes).

■Explanation

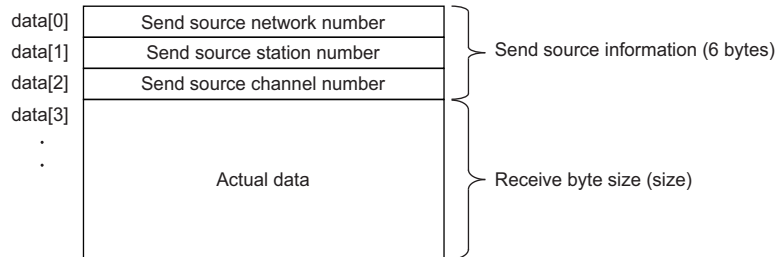
This function supports the RECV instruction of the link dedicated instruction.

For details of the function, refer to the manuals of each board.

< Receive data with send source information (data) >

- Receive byte size and send source information (6 bytes) are stored in "receive data with send source information (data)".
Reserve an area for [receive byte size and send source information (6 bytes)] in "receive data with send source information (data)".
- Data are stored in "receive data with send source information (data)" as shown below.

Receive data with send source information (data)



- Only the actual data with the specified receive byte size (size) is stored when the received actual data size is greater than the specified receive byte size (size).
- 125 (7DH) is stored to the send source station number when the station number of the send source is 0.
- The arguments of the mdReceive function correspond to the control data (device) of the dedicated instruction (RECV) as shown below:

—: N/A

Device	Item	Corresponding argument and return value
+0	Error completion type	—
+1	Completion status	ret (return value)
+2	Own station storage channel	devno
+3	Channel used by send station	data[2]
+4	Send station network number	data[0]
+5	Send station number	data[1]
+6	Not used	—
+7	Not used	—
+8	Arrival monitoring time	—
+9	Receive data length	size
+10	Not used	—
+11	Clock setting flag	—
+12	Clock data (Set only in an abnormal state.)	—
+13		—
+14		—
+15		—
+16	Error-detected network number	—
+17	Error-detected station number	—

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. (📖 Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdReceiveEx(), mdSend(), mdSendEx()

mdRandW (writing devices randomly)

Write data to the devices on the target station specified with the randomly-specified devices.

Format

■Visual C++

Format	ret = mdRandW(path,stno,dev,buf,bufsize);		
Argument	short	ret;	//Return value OUT
	long	path;	//Path of channel IN
	short	stno;	//Station number IN
	short	dev[];	//Randomly-specified device IN
	short	buf[];	//Written data (single-precision integer array) IN
	short	bufsize;	//Dummy IN

■Visual Basic .NET

Format	ret = mdRandW(path,stno,dev(0),buf(0),bufsize)		
Argument	short	ret	;Return value OUT
	Integer	path	;Path of channel IN
	short	stno	;Station number IN
	short	dev(n)	;Randomly-specified device IN
	Short	buf(n)	;Written data (single-precision integer array) IN
	short	bufsize	;Dummy IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdRandW(path&,stno%,dev%(0),buf%(0),bufsize%)		
Argument	Integer	ret	;Return value OUT
	Long	path	;Path of channel IN
	Integer	stno	;Station number IN
	Integer	dev(n)	;Randomly-specified device IN
	Any	buf(n)	;Written data (single-precision integer array) IN
	Integer	bufsize	;Dummy IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Specify the station number of a target station. (☞ Page 46 Specifying station numbers)
dev	Specify the number of blocks, the device type, the start device number, and points of device to be written. (☞ Page 68 How to specify the randomly-specified devices (dev))
buf	Specify the data to be written in single-precision integer array. (☞ Page 68 How to specify the randomly-specified devices (dev))
bufsize	—

■Explanation

- The number of transient transmissions performed inside of the function changes with the specified randomly-specified devices.
- A longer function processing time is required for the random write function as compared with the batch write function. Consider using the batch write function if a shorter function processing time is required.
- When accessing another station, the extended comment information will be deleted by writing data to the block (extended file register) to which the extended comment is assigned.
- When accessing another station, the sub2 and sub3 programs will be deleted by writing data to the block (extended file register) which overlaps with the setting areas of the sub2 and sub3 programs.
- The size of written data varies for each device type depending on the number of points specified for randomly-specified devices. (☞ Page 50 Data size per point of device)
- If the total number of start device number and points exceed 32768, "Start device number error -2 (FFFEH)" is returned. When accessing with the start device number and points the total number of which exceeds 32768, use the mdRandWEx function.

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdRandR()

mdRandR (reading devices randomly)

Read the device specified with the randomly-specified devices from the target station.

Format

■Visual C++

Format	ret = mdRandR(path,stno,dev,buf,bufsize);		
Argument	short	ret;	//Return value OUT
	long	path;	//Path of channel IN
	short	stno;	//Station number IN
	short	dev[];	//Randomly-specified device IN
	short	buf[];	//Read data (single-precision integer array) OUT
	short	bufsize;	//Number of bytes of read data IN

■Visual Basic .NET

Format	ret = mdRandR(path,stno,dev(0),buf(0),bufsize)		
Argument	Short	ret	;Return value OUT
	Integer	path	;Path of channel IN
	Short	stno	;Station number IN
	Short	dev(n)	;Randomly-specified device IN
	Short	buf(n)	;Read data (single-precision integer array) OUT
	Short	bufsize	;Number of bytes of read data IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdRandR(path&,stno%,dev%(0),buf%(0),bufsize%)		
Argument	Integer	ret	;Return value OUT
	Long	path	;Path of channel IN
	Integer	stno	;Station number IN
	Integer	dev(n)	;Randomly-specified device IN
	Any	buf(n)	;Read data (single-precision integer array) OUT
	Integer	bufsize	;Number of bytes of read data IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Specify the station number of a target station. (☞ Page 46 Specifying station numbers)
dev	Specify the number of blocks, the device type, the start device number, and points of device to be read. (☞ Page 71 How to specify the randomly-specified devices (dev))
buf	Store the data being read. (☞ Page 71 How to specify the randomly-specified devices (dev))
bufsize	Specify the number of bytes of read data. (☞ Page 71 How to specify the randomly-specified devices (dev))

■Explanation

- The number of transient transmissions performed inside of the function changes with the specified randomly-specified devices.
- A longer function processing time is required for the random read function as compared with the batch read function. Consider using the batch read function if a shorter function processing time is required.
- The size of read data varies for each device type depending on the number of points specified for randomly-specified devices.
(☞ Page 50 Data size per point of device)
- If the total number of start device number and points exceed 32768, "Start device number error -2 (FFFEH)" is returned. When accessing with the start device number and points the total number of which exceeds 32768, use the mdRandREx function.

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. (☞ Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdRandW()

mdDevSet (setting bit devices)

Set the specified bit device on the target station (to ON).

Format

■Visual C++

```
Format      ret = mdDevSet(path,stno,devtyp,devno);
Argument    short      ret;           //Return value          OUT
            long      path;          //Path of channel       IN
            short     stno;          //Station number        IN
            short     devtyp;        //Device type           IN
            short     devno;         //Specified device number IN
```

■Visual Basic .NET

```
Format      ret = mdDevSet(path,stno,devtyp,devno)
Argument    Short      ret           ;Return value          OUT
            Integer    path          ;Path of channel       IN
            Short      stno          ;Station number        IN
            Short      devtyp        ;Device type           IN
            Short      devno         ;Specified device number IN
```

■Visual Basic 5.0, Visual Basic 6.0

```
Format      ret% = mdDevSet(path&,stno%,devtyp%,devno%)
Argument    Integer    ret           ;Return value          OUT
            Long       path          ;Path of channel       IN
            Integer    stno          ;Station number        IN
            Integer    devtyp        ;Device type           IN
            Integer    devno         ;Specified device number IN
```

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Specify the station number of a target station. ( Page 46 Specifying station numbers)
devtyp	Specify the type of device to be set (ON). ( Page 50 Specifying device types)
devno	Specify the device number of device to be set (ON).

■Explanation

This function is a dedicated function for bit devices such as link relays (B) and internal relays (M).

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdDevRst()

mdDevRst (resetting bit devices)

Reset the specified bit device on the target station (to OFF).

Format

■Visual C++

Format	ret = mdDevRst(path,stno,devtyp,devno);			
Argument	short	ret;	//Return value	OUT
	long	path;	//Path of channel	IN
	short	stno;	//Station number	IN
	short	devtyp;	//Device type	IN
	short	devno;	//Specified device number	IN

■Visual Basic .NET

Format	ret = mdDevRst(path,stno,devtyp,devno)			
Argument	Short	ret	;Return value	OUT
	Integer	path	;Path of channel	IN
	Short	stno	;Station number	IN
	Short	devtyp	;Device type	IN
	Short	devno	;Specified device number	IN

■Visual Basic 5.0, Visual Basic 6.0

Format	ret% = mdDevRst(path&,stno%,devtyp%,devno%)			
Argument	Integer	ret	;Return value	OUT
	Long	path	;Path of channel	IN
	Integer	stno	;Station number	IN
	Integer	devtyp	;Device type	IN
	Integer	devno	;Specified device number	IN

Detailed specifications

■Argument

Argument	Description
path	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Specify the station number of a target station. ( Page 46 Specifying station numbers)
devtyp	Specify the type of device to be reset (OFF). ( Page 50 Specifying device types)
devno	Specify the device number of device to be reset (OFF).

■Explanation

This function is a dedicated function for bit devices such as link relays (B) and internal relays (M).

■Return value

Success: '0' is returned.

Error: A value other than '0' is returned. Refer to the error code list. ( Page 140 ERROR CODES)

■Related function

mdOpen(), mdClose(), mdDevSet()

5 SAMPLE PROGRAMS

This chapter explains the sample programs that are included in the software package.

The sample programs are registered in the folder where the utility is installed.

Type	Folder	Description
CC-Link IE TSN board	\CCIETBDM\SAMPLES\ (each language) \DEMO, MTEST, etc.	A sample program for MELSEC data link library
	\CCIETBDM\SAMPLES\ (each language) \Remote	A sample program for a remote I/O module
CC-Link IE Field Network board	\CCIEF\SAMPLES\ (each language) \DEMO, MTEST, etc.	A sample program for MELSEC data link library
	\CCIEF\SAMPLES\ (each language) \RemoteDevice	A sample program for a remote I/O module
CC-Link IE Controller Network board	\MNETG\SAMPLES\ (each language) \DEMO, MTEST, etc.	A sample program for MELSEC data link library
MELSECNET/H board	\MNETH\SAMPLES\ (each language) \DEMO, MTEST, etc.	A sample program for MELSEC data link library
CC-Link Ver.2 board	\CCBD2\Sample\MDFunction	A sample program for MELSEC data link library
	\CCBD2\Sample\LocalStation	A sample program for data link check of a CC-Link Ver.2 board
	\CCBD2\Sample\MasterStation	

5

Point

Sample programs are provided as a reference for creating user application programs.

Use sample programs on a user's own responsibility.

Sample program for MELSEC data link library

The following sample programs are stored as examples of functions.

Folder	Description		
DEMO	Reading device data	CC-Link IE TSN board CC-Link IE Controller Network board CC-Link IE Field Network board MELSECNET/H board	A sample program for reading device D of network number 1 and station number 1
		CC-Link Ver.2 board	A sample program for reading device D0 of a master station
MTEST MTEST2	Testing functions in general		A sample program for MELSEC data link library in general

Sample program for a remote I/O module

Program examples are stored to configure the initial setting of a remote I/O module (remote device station/remote station) with the following functions:

- CC-Link IE Field Network board: mdRemBufWriteEx, mdRemBufReadEx
- CC-Link IE TSN board: mdRemBufWriteIPEX, mdRemBufReadIPEX

Sample program for data link check of a CC-Link Ver.2 board

The following sample programs are stored as program examples to check whether the data link between the master station and each station is executed properly.

Folder		Description	
LocalStation	Datalink	Communication between a master station and a local station	A sample program for communication between a master station and a local station

Folder		Description	
MasterStation	Datalink	Communication between a master station and a local station	A sample program for communication between a master station and a local station
	PositioningSystem	Communication between a master station and an intelligent device station (AJ65BT-D75P2-S3)	A sample program for initialization, positioning, origin point return, and JOG operation for AJ65BT-D75P2-S3
	R2	Communication between a master station and an intelligent device station (AJ65BT-R2)	A sample program for initialization, transmission and reception for AJ65BT-R2
	RemoteDevice	Communication between a master station and a remote device station	A sample program for converting digital/analog for AJ65BT-64DAV
	RemoteIO	Communication between a master station and a remote I/O station	A sample program for reading/writing remote I/Os for a remote I/O station

For details on the checking method, refer to the following:

 Type Q80BD-J61BT11N/Q81BD-J61BT11 CC-Link System Master/Local Interface Board User's Manual (For SW1DNC-CCBD2-B)

5.1 Using Visual Basic .NET

Sample programs for Visual Basic .NET are stored in the "NETVB" folder.

They are created in the following project formats.

When using a different format, convert the project.

Software package	Project format of a sample program
SW1DNN-CCIETBDM-B	Visual Studio 2017 Visual Basic
SW1DNC-CCIEF-B	Visual Studio.NET2003 Visual Basic
SW1DNC-MNETG-B	Visual Studio.NET2003 Visual Basic
SW0DNC-MNETH-B	Visual Studio.NET2003 Visual Basic
SW1DNC-CCBD2-B	Visual Studio.NET2003 Visual Basic

Considerations for using sample programs

■For all boards

"Mdfunc.vb" is not included in the sample program.

Add "Mdfunc.vb" before using the sample program.

For the setting method, refer to the following:

 Page 17 Using Visual Basic.NET

■For CC-Link IE TSN boards

For application development in Visual Basic by using Visual Studio, a target framework (.NET Framework version) must be specified.

In an environment where a target pack supporting the target framework set in a project of the sample program is not installed, a screen indicating that the target framework is not installed appears when reading the project.

When the screen appears, select an option displayed in the screen and click the [OK] button.

For details on the methods for changing a target framework and installing a target pack, visit the Microsoft website.

MTEST, MTEST2

Sample programs for testing MELSEC data link library functions in general are stored in the following folders for each programming language used.

Folder	MTEST	MTEST2
Programming language	Visual Studio .NET 2003 Visual Studio 2005 Visual Studio 2008 Visual Studio 2010	Visual Studio 2012 Visual Studio 2013 Visual Studio 2015 Visual Studio 2017

Point

Sample programs in MTEST and MTEST2 cannot be used as 64-bit version user applications. To use them on a 64-bit version operating system, select "x86" for the platform in Visual Studio, and build them as 32-bit version user applications.

NETVB\MTEST

MSFlexGrid is used in the sample program. If MSFlexGrid is not installed, a warning occurs, however, the operation of sample program is not affected.

NETVB\MTEST2

A warning occurs because a controller created by the Visual Basic 6.0 is used for this sample program. However, there is no effect for the operation of the sample program.

5.2 Using Visual Basic 5.0 or Visual Basic 6.0

Sample programs for Visual Basic 5.0 and Visual Basic 6.0 are stored in the "VB" folder.

Considerations for using sample programs

"Mdfunc.bas" is not included in the sample program.

Add "Mdfunc.bas" before using the sample program.

For the setting method, refer to the following:

 Page 15 Using Visual Basic 5.0 or Visual Basic 6.0

5.3 Using Visual C++ or Visual C#

Sample programs for Visual C++ are stored in the "VC" folder.

Sample programs for Visual C# are stored in the "VCS" folder.

They are created in the following project formats.

When using a different format, convert the project.

Software package	Project format of a sample program	
	VC	VCS
SW1DNN-CCIETBDM-B	Visual Studio 2017 Visual C++	Visual Studio 2017 Visual C#
SW1DNC-CCIEF-B	Visual Studio.NET2003 Visual C++	—
SW1DNC-MNETG-B	Visual C++ 6.0	—
SW0DNC-MNETH-B	Visual C++ 5.0	—
SW1DNC-CCBD2-B	Visual C++ 5.0	—

Considerations for using sample programs

■For all boards

An include file and a library file are not included in the sample program.

Set an include file and a library file before using the sample program.

For the setting method, refer to the following:

-  Page 16 Using Visual C++ 5.0 or Visual C++ 6.0
-  Page 18 Using Visual C++ .NET 2003, Visual C++ 2005, or Visual C++ 2008
-  Page 20 Using Visual C++ in Visual Studio 2010 or Visual Studio 2012
-  Page 24 Using Visual C++ in Visual Studio 2013 or later
-  Page 30 Using Visual C#

If Visual Studio is installed by default, some necessary functions are not installed; an error may occur in the project conversion of a sample program.

Customize the installation by selecting necessary functions.

Example) For Visual Studio 2017, select the checkbox(es) of the following item(s).

- To use MFC: "Microsoft Foundation Classes for C++"
- To create a program which can be executed on Windows XP: "Windows XP Support for C++" and "Windows Universal CRT SDK"

For the method to select a function during the installation process or to add a function after the installation, refer to the Microsoft website.

■For applications in C# for CC-Link IE TSN boards

For application development in C# by using Visual Studio, a target framework (.NET Framework version) must be specified.

In an environment where a target pack supporting the target framework set in a project of the sample program is not installed, a screen indicating that the target framework is not installed appears when reading the project.

When the screen appears, select an option displayed in the screen and click the [OK] button.

For details on the methods for changing a target framework and installing a target pack, visit the Microsoft website.

■For applications in C++ for CC-Link IE TSN boards

For application development in C++ by using Visual Studio, a Windows SDK version used for a build must be specified.

In an environment where Windows SDK with the version set in a project of the sample program is not installed, an error occurs during a build.

If an error related to Windows SDK versions occurs, take either of the following actions:

- Change the setting of the project in Visual Studio to use an installed version of Windows SDK.
- Install Windows SDK with a version displayed in the error message, then perform a build again.

6 DEVICE MONITOR UTILITY

Device Monitor Utility is a utility to monitor devices and change current values.

Device Monitor Utility monitors devices by using the MELSEC data link library.

For accessible devices of Device Monitor Utility, refer to the following:

 Page 36 Accessible Devices

Installing Device Monitor Utility

It is included in the following software packages. When installing it in the same personal computer, Device Monitor Utility included in a newer software package is valid.

For the operating environment and installation method, refer to the manual of each software package.

Software package	Reference
SW1DNN-CCIBTBDM-B	 CC-Link IE TSN Interface Board User's Manual
SW1DNC-MNETG-B	 CC-Link IE Controller Network Interface Board User's Manual (For SW1DNC-MNETG-B)
SW1DNC-CCIEF-B	 CC-Link IE Field Network Interface Board User's Manual (For SW1DNC-CCIEF-B)
SW1DNC-CCBD2-B	 Type Q80BD-J61BT11N/Q81BD-J61BT11 CC-Link System Master/Local Interface Board User's Manual (For SW1DNC-CCBD2-B)
SW0DNC-MNETH-B	 MELSECNET/H Interface Board User's Manual (For SW0DNC-MNETH-B)

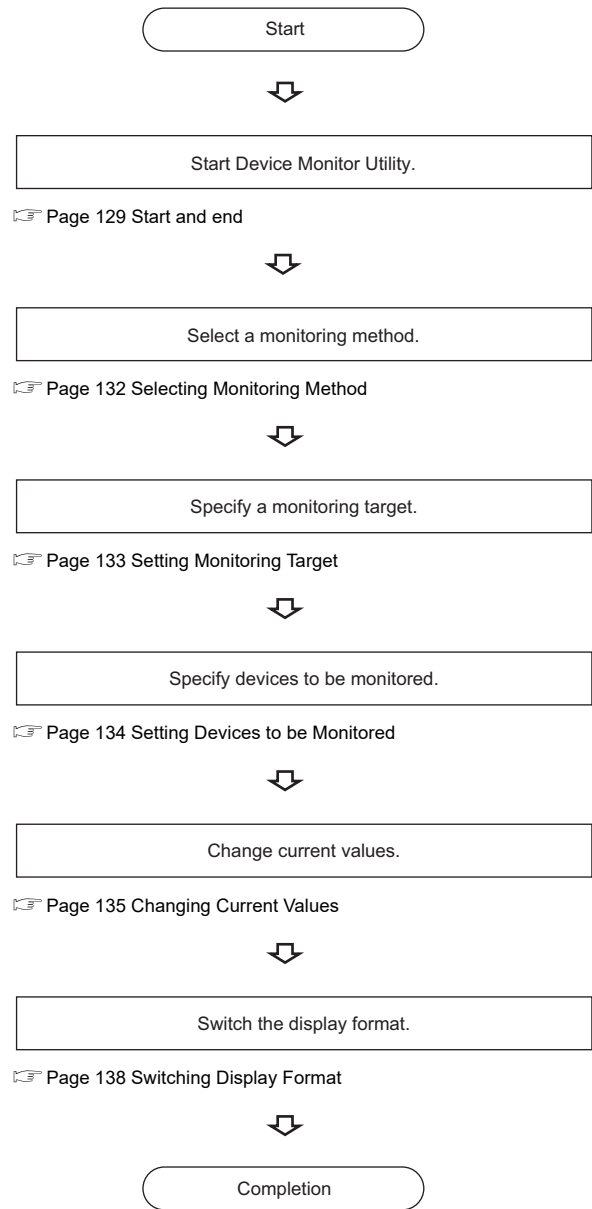
■Considerations for the display language switching function

When switching the display language in the utility of a board supporting the display language switching function, Device Monitor Utility is displayed in the set language only when displayed in the utility of the board.

When starting Device Monitor Utility from the utility of a board without the display language switching function, the display language of Device Monitor follows the setting of the operating system.

6.1 Operation Flow

The following flowchart shows a basic monitoring operation flow.



Point

- The buffer memory of a CC-Link IE Field Network remote device station or CC-Link IE TSN remote station cannot be monitored in Device Monitor Utility.
- If an error occurs in Device Monitor Utility, check the error code in the error code list. (Page 140 ERROR CODES)

6.2 Menu List

This section explains the functions of Device Monitor Utility.

[Menu]		Reference
⇒ [Batch monitor]	Batch monitor one type of devices.	Page 132 Monitoring devices in a batch
⇒ [16-point register monitor]	Monitor multiple devices simultaneously in units of 16 points.	Page 132 Monitoring registered 16-point devices
⇒ [Exit]	End Device Monitor Utility.	Page 129 Start and end
[Setting]		Reference
⇒ [Network setting]	Set a programmable controller CPU or board to be monitored.	Page 133 Setting Monitoring Target
⇒ [Device setting]	Set devices to be monitored.	Page 134 Setting Devices to be Monitored
[Device Write]		Reference
⇒ [Data changing]	Change the word device/double-word device value.	Page 135 Changing word device/double-word device values
⇒ [Continuous change in data]	Change the sequenced word devices/double-word devices to the same value.	
⇒ [Bit device setting]	Turn ON the specified bit devices.	Page 137 Turning bit devices ON and OFF
⇒ [Bit device resetting]	Turn OFF the specified bit devices.	
[Data Format]		Reference
⇒ [Word device] ⇒ [(display format)]	Change the display format of word devices.	Page 138 Switching Display Format
⇒ [Double Word device] ⇒ [(display format)]	Change the display format of double-word devices.	
⇒ [Bit device] ⇒ [(display format)]	Change the display format of bit devices.	
[Option]		Reference
⇒ [Numerical Pad]	Enable/disable the setting to use the numerical pad.	Page 139 Using Numerical Pad
[Help]		Reference
⇒ [Help]	Display Help.	Page 131 Help function
⇒ [About]	Display product information such as a product version.	

6.3 Screen Configuration and Basic Operations

This section explains the screen configuration and basic operations of Device Monitor Utility.

Start and end

The following explains the operating method to start/end the utility.

Start

Start Device Monitor Utility.

It can be started from the menu of the utility for each board or from Windows Start.

■Start from CC IE TSN utility

Operating procedure

1. Select [Tool] ⇒ [Device Monitor utility].

■Start from CC IE Field utility

Operating procedure

1. Select [Online] ⇒ [Device Monitor Utility].

■Start from CC IE Control Utility

Operating procedure

1. Select [Device Monitor Utility] from the system menu.

■Start from CC-Link Ver.2 Utility or MNETH Utility

Operating procedure

1. Click the [Device Monitor] button.

■Start from the Windows Start

Operating procedure

1. Select either of the following:
 - [MELSEC] ⇒ [Device Monitor Utility (Board)] from Windows Start
 - [MELSOFT] ⇒ [Device Monitor Utility (Board)] from Windows Start

End

End Device Monitor Utility.

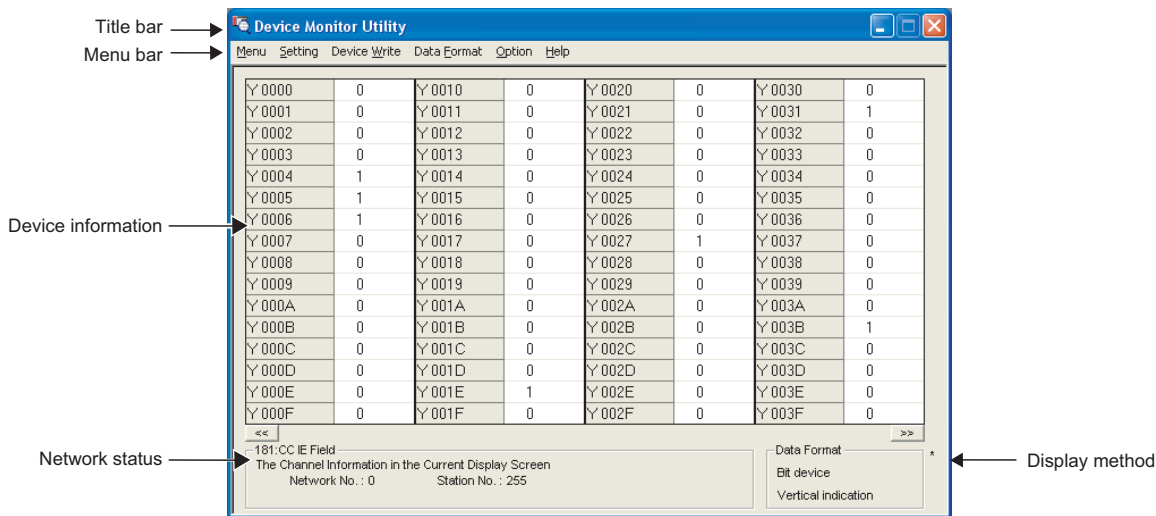
Operating procedure

1. Select [Menu] ⇒ [Exit].

Screen configuration and basic operations

The following explains the screen configuration of Device Monitor Utility.

Window



Displayed items

Item	Description
Title bar	Displays a project name.
Menu bar	Displays menu options for executing each function.
Device information	Display the current device status. For changing the monitoring method, refer to the following: ☞ Page 132 Selecting Monitoring Method For changing the current values, refer to the following: ☞ Page 135 Changing Current Values
Network status	Display the currently set network status. For setting the network, refer to the following: ☞ Page 133 Setting Monitoring Target
Display method	Display the type and display format of devices being displayed. For changing a device type, refer to the following: ☞ Page 134 Setting Devices to be Monitored For changing a display format, refer to the following: ☞ Page 138 Switching Display Format

Point

- When own station is set as a monitoring target
"0" is displayed for the network number and "255" is displayed for the station number as a network status.

Help function

Displaying Help screen

Display the Help screen of Device Monitor Utility.

Operating procedure

1. Select [Help] ⇒ [Help].

Checking product version

Displays information such as, software version of Device Monitor Utility.

Operating procedure

1. Select [Help] ⇒ [About].



6.4 Selecting Monitoring Method

This section explains a method for selecting a monitoring method in Device Monitor Utility.

Monitoring devices in a batch

Specify one type of devices to monitor in a batch.

Window

 [Menu] ⇒ [Batch monitor]

< Bit devices >								< Word devices >							
Y 0000	0	Y 0010	0	Y 0020	0	Y 0030	0	Ww 0000	0	Ww 0010	0	Ww 0020	0	Ww 0030	0
Y 0001	0	Y 0011	0	Y 0021	0	Y 0031	1	Ww 0001	0	Ww 0011	0	Ww 0021	0	Ww 0031	0
Y 0002	0	Y 0012	0	Y 0022	0	Y 0032	0	Ww 0002	0	Ww 0012	0	Ww 0022	0	Ww 0032	0
Y 0003	0	Y 0013	0	Y 0023	0	Y 0033	0	Ww 0003	0	Ww 0013	0	Ww 0023	0	Ww 0033	0
Y 0004	1	Y 0014	0	Y 0024	0	Y 0034	0	Ww 0004	0	Ww 0014	0	Ww 0024	0	Ww 0034	0
Y 0005	1	Y 0015	0	Y 0025	0	Y 0035	0	Ww 0005	10	Ww 0015	0	Ww 0025	0	Ww 0035	0
Y 0006	1	Y 0016	0	Y 0026	0	Y 0036	0	Ww 0006	0	Ww 0016	0	Ww 0026	0	Ww 0036	0
Y 0007	0	Y 0017	0	Y 0027	1	Y 0037	0	Ww 0007	0	Ww 0017	0	Ww 0027	0	Ww 0037	0
Y 0008	0	Y 0018	0	Y 0028	0	Y 0038	0	Ww 0008	0	Ww 0018	0	Ww 0028	0	Ww 0038	0
Y 0009	0	Y 0019	0	Y 0029	0	Y 0039	0	Ww 0009	0	Ww 0019	255	Ww 0029	0	Ww 0039	0
Y 000A	0	Y 001A	0	Y 002A	0	Y 003A	0	Ww 000A	0	Ww 001A	0	Ww 002A	0	Ww 003A	0
Y 000B	0	Y 001B	0	Y 002B	0	Y 003B	1	Ww 000B	0	Ww 001B	0	Ww 002B	0	Ww 003B	0
Y 000C	0	Y 001C	0	Y 002C	0	Y 003C	0	Ww 000C	0	Ww 001C	0	Ww 002C	0	Ww 003C	0
Y 000D	0	Y 001D	0	Y 002D	0	Y 003D	0	Ww 000D	0	Ww 001D	0	Ww 002D	0	Ww 003D	0
Y 000E	0	Y 001E	1	Y 002E	0	Y 003E	0	Ww 000E	0	Ww 001E	0	Ww 002E	0	Ww 003E	0
Y 000F	0	Y 001F	0	Y 002F	0	Y 003F	0	Ww 000F	0	Ww 001F	0	Ww 002F	0	Ww 003F	0

Displayed items

Item	Description
Device information*1	—
	Device name
	Monitoring result
[<< / >>] button	

*1 The number of displayed columns will differ depending on the settings of a connection target CPU and display format.

*2 When the display language is English, the display position of each digit of a value may differ for each row.

Monitoring registered 16-point devices

Monitor devices by specifying different types of devices in units of 16 points.

Up to five types of bit devices and one type of word devices/double-word devices can be monitored simultaneously.

The displayed information of devices is the same as that of the batch monitor function.

Window

 [Menu] ⇒ [16-point register monitor]

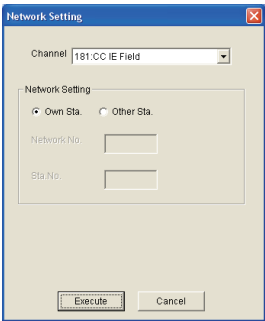
X 0000	0	X 0010	0	Y 0000	0	Y 0010	0	Ww 0000	0
X 0001	0	X 0011	0	Y 0001	0	Y 0011	0	Ww 0001	0
X 0002	0	X 0012	0	Y 0002	0	Y 0012	0	Ww 0002	0
X 0003	0	X 0013	0	Y 0003	0	Y 0013	0	Ww 0003	0
X 0004	0	X 0014	0	Y 0004	1	Y 0014	0	Ww 0004	0
X 0005	0	X 0015	0	Y 0005	1	Y 0015	0	Ww 0005	10
X 0006	0	X 0016	0	Y 0006	1	Y 0016	0	Ww 0006	0
X 0007	0	X 0017	0	Y 0007	0	Y 0017	0	Ww 0007	0
X 0008	0	X 0018	0	Y 0008	0	Y 0018	0	Ww 0008	0
X 0009	0	X 0019	0	Y 0009	0	Y 0019	0	Ww 0009	0
X 000A	0	X 001A	0	Y 000A	0	Y 001A	0	Ww 000A	0
X 000B	0	X 001B	0	Y 000B	0	Y 001B	0	Ww 000B	0
X 000C	0	X 001C	0	Y 000C	0	Y 001C	0	Ww 000C	0
X 000D	0	X 001D	0	Y 000D	0	Y 001D	0	Ww 000D	0
X 000E	0	X 001E	0	Y 000E	0	Y 001E	1	Ww 000E	0
X 000F	0	X 001F	0	Y 000F	0	Y 001F	0	Ww 000F	0

6.5 Setting Monitoring Target

This section explains the method for setting a programmable controller CPU or board to be monitored.
Set a target when starting Device Monitor Utility.

Window

 [Setting] ⇒ [Network setting]



Operating procedure

1. Set the items on the screen.

Item	Description
Channel	Set the access source channel number to be used.
Network Setting	Set the monitoring target. Set the network number and the station number when the access target is at another station.

2. Click the [Execute] button.

Point

- Accessing multiple CPU system
Enter "0" for the network number and a value of "logical station number" which was set in the each utility for the station number.
- Do not specify the remote I/O station and intelligent device station on CC-Link for the monitoring target.
Otherwise, an error occurs.

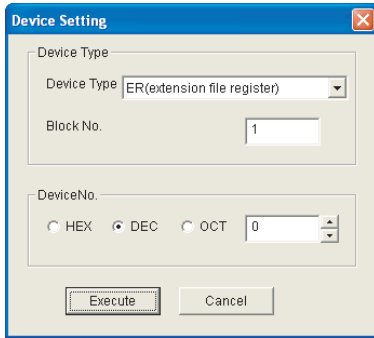
6.6 Setting Devices to be Monitored

This section explains a method for setting devices to be monitored.

Window

 [Setting] ⇒ [Device setting]

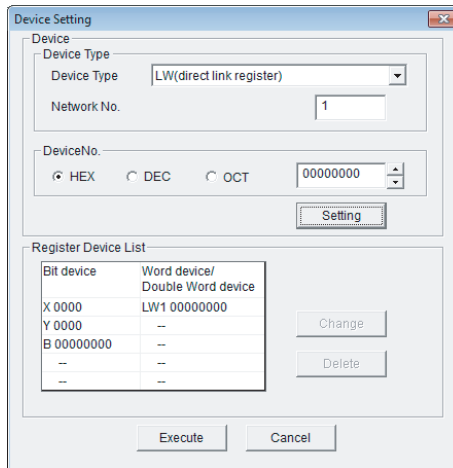
< For batch monitor >



The "Device Setting" dialog box for batch monitor has the following fields and controls:

- Device Type:** A dropdown menu with "ER(extension file register)" selected.
- Block No.:** A text input field containing the value "1".
- DeviceNo.:** A section with three radio buttons: "HEX", "DEC" (selected), and "OCT". Next to them is a numeric input field containing "0".
- Buttons:** "Execute" and "Cancel" buttons at the bottom.

< For 16-point register monitor >



The "Device Setting" dialog box for 16-point register monitor has the following fields and controls:

- Device Type:** A dropdown menu with "LW(direct link register)" selected.
- Network No.:** A text input field containing the value "1".
- DeviceNo.:** A section with three radio buttons: "HEX", "DEC" (selected), and "OCT". Next to them is a numeric input field containing "00000000".
- Setting button:** A button located below the DeviceNo. section.
- Register Device List:** A table with two columns: "Bit device" and "Word device/ Double Word device".

Bit device	Word device/ Double Word device
X 0000	LW1 00000000
Y 0000	--
B 00000000	--
--	--
--	--
- Buttons:** "Change" and "Delete" buttons next to the Register Device List table, and "Execute" and "Cancel" buttons at the bottom.

Operating procedure

1. Set the items on the screen.

Item	Description
Device Type	Set a type of a device to be monitored. Specify a device type of the MELSEC data link library. ( Page 50 Specifying device types) • Enter a block number for ER. • Enter a network number for LX, LY, LB, LW, LSB, and LSW. • Enter a value of start I/O number divided by 16 in decimal for SPG.
Device No.	Set a start number of the device to be monitored. (HEX: Hexadecimal, DEC: Decimal, OCT: Octal)
Register Device List	Display a list of registered devices.
[Setting] button	Adds the items set for "Device Type" and "Device No." to "Register Device List".
[Change] button	Sets the setting of the device selected in "Register Device List" to the settings set for "Device Type" and "Device No.".
[Delete] button	Deletes the setting of the device selected in "Register Device List".

2. Click the [Execute] button.

Point

- Only devices available for random access can be monitored in units of 16 points.
If specifying another device, a device type error occurs.

For the availability of random access for each device, refer to the following:

 Page 36 Accessible Devices

- In monitoring registered devices in units of 16 points, devices cannot be monitored if they are less than 16 points or the total number of start device number and points (16 points) in a specified block exceeds the device range.
- When displaying devices in the range including unavailable device numbers, incorrect data may be displayed in monitoring devices in a batch or monitoring registered ones in units of 16 points. For available device ranges, refer to the manual for each product.

6.7 Changing Current Values

This section explains a method for changing current device values.

Precaution

When changing data during operation, configure an interlock circuit in the program to ensure that the entire system will always operate safely. And determine corrective actions to be taken between the personal computer and CPU module in case of a communication failure.

Changing word device/double-word device values

Change word device/double-word device values. The continuous change in data function changes the values of continuous word devices/double-word devices to the same values.

Window

 [Device Write] ⇒ [Data changing]/[Continuous change in data]

< For Data Changing >

Data Changing

Device Type

Device Type [ER(extension file register)]

Block No.

1

Device No.

HEX

DEC

OCT

0

Setting Data

HEX

DEC

0

Execute

Cancel

< For Continuous Change in Data >

Continuous Change in Data

Device Type

Device Type [SPG(special direct buffer register)]

Start I/O No./16(DEC)

2

Device No.

HEX

DEC

OCT

0

Setting Data

HEX

DEC

1

Points

HEX

DEC

OCT

5

Execute

Cancel

6

Operating procedure

1. Set the type and number of devices whose current values are to be changed.

Item	Description
Device Type	Set a type of devices whose values are to be changed. Specify a device type of the MELSEC data link library. ( Page 50 Specifying device types) • Enter a block number for ER. • Enter a network number for LW and LSW. • Enter a value of start I/O number divided by 16 in decimal for SPG.
Device No.	Set a start number of devices whose values are to be changed. (HEX: Hexadecimal, DEC: Decimal, OCT: Octal)

2. Set a value and points to be changed.

Item	Description
Setting Data	Set a value to be changed. (HEX: Hexadecimal, DEC: Decimal)*1

*1 When the DEC (Decimal) is selected, only signed decimal format can be specified.

3. Set device points to be changed for the continuous change in data.

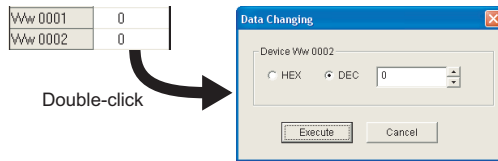
Item	Description
Points	Set a number of devices whose values are to be changed. (HEX: Hexadecimal, DEC: Decimal, OCT: Octal) Setting range: 1 to 64 points

4. Click the [Execute] button.

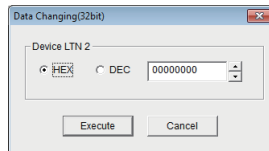
6 DEVICE MONITOR UTILITY
6.7 Changing Current Values

135

Word data/double-word data can also be changed by double-clicking the device column of the device information in the monitoring screen.



When the display format is set to decimal (unsigned 16-bit) or decimal (unsigned 32-bit) on the Data Changing screen displayed by double-clicking the device column, the value same as the display format can be set.

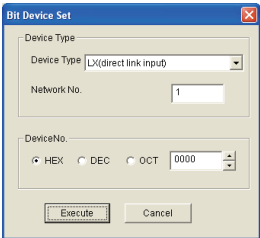


Turning bit devices ON and OFF

Change ON and OFF of bit devices.

Window

[Device Write] ⇒ [Bit device set]/[Bit device reset]



Operating procedure

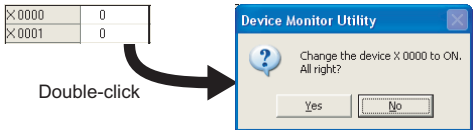
1. Set the type and number of devices whose current values are to be changed.

Item	Description
Device Type	Set a type of devices whose values are to be changed. Specify a device type of the MELSEC data link library. (Page 50 Specifying device types) • Enter a network number for LX, LY, LB, and LSB.
Device No.	Set a start number of devices whose values are to be changed. (HEX: Hexadecimal, DEC: Decimal, OCT: Octal)

2. Click the [Execute] button.

Point

- Devices that can be changed
Only devices available for random access can be changed by using this function.
If specifying another device, a device type error occurs.
For the availability of random access for each device, refer to the following:
Page 36 Accessible Devices
- A bit device can also be turned ON and OFF by double-clicking the device column of the device information on the monitoring screen.



6.8 Switching Display Format

This section explains a method for switching a display format (such as in hexadecimal or in decimal) of device information for monitoring.

The following shows the menus that can be selected for the batch monitor and 16-point register monitor functions.

Operating procedure

1. Select [Data Format] ⇒ [Word device]/[Double Word device]/[Bit device] ⇒ [(display format)].

Word devices

○: Can be displayed, —: Cannot be displayed

Display format	Batch monitor	16-point register monitor
DEC (decimal) (signed 16 bits)	○	○
DEC (decimal) (unsigned 16 bits)	○	○
HEX (hexadecimal) (16 bits)	○	○
OCT (octal) (16 bits)	○	○
BIN (binary) (16 bits)	○	—
DEC (decimal) (signed 32 bits)* ¹	○	○
DEC (decimal) (unsigned 32 bits)* ¹	○	○
HEX (hexadecimal) (32 bits)* ¹	○	○
OCT (octal) (32 bits)* ¹	○	○
BIN (binary) (32 bits)* ¹	○	—

*¹ When the display format is set to 32 bits, 32-bit data (two word devices) is displayed as one point.

Double-word devices

○: Can be displayed, —: Cannot be displayed

Display format	Batch monitor	16-point register monitor
DEC (decimal) (signed 32 bits)	○	○
DEC (decimal) (unsigned 32 bits)	○	○
HEX (hexadecimal) (32 bits)	○	○
OCT (octal) (32 bits)	○	○
BIN (binary) (32 bits)	○	—

Bit devices

○: Can be displayed, —: Cannot be displayed

Display format	Batch monitor	16-point register monitor
Vertical Indication	○	—
Horizontal Indication (F-0) (decimal)	○	—
Horizontal Indication (F-0) (hexadecimal)	○	—
Horizontal Indication (0-F) (decimal)	○	—
Horizontal Indication (0-F) (hexadecimal)	○	—

6.9 Using Numerical Pad

This section explains the option function for entering numerals.

A numerical pad allows numeric values, such as device values, to be entered only with the mouse operation.

Enabling/disabling numerical pad

Operating procedure

1. [Option] ⇒ [Numerical Pad]

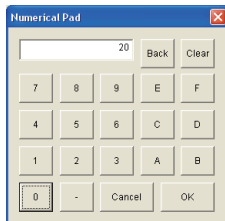
After the setting, a numerical pad is displayed by clicking a numeric value input field.

The setting is disabled by selecting this menu again.

Entering values using numerical pad

Window

 Click a numeric value input field on a screen.



Operating procedure

1. Click the buttons and enter a value.
2. Click the [OK] button.

The value is entered in the numeric value input field.

7 ERROR CODES

The following table shows the errors and the corrective actions that correspond to the error code.

When an error whose error code is not described in the following table occurs, contact your local Mitsubishi Electric representative.

Error Code		Error description	Corrective action
Decimal	Hexadecimal		
1	0001H	■Driver not started The driver is not started.	• Check the channel number. • Check the driver operating condition with Device Manager. • Check the error of event viewer. • Check the board settings. • Reinstall the software package.
2	0002H	■Time-out error • Timeout during waiting for the response of process • The process was requested with a 64-bit version user application to a CPU module other than QCPU (Q mode) or MELSEC iQ-R series CPU modules. • The consistency between the board and the software package cannot be identified. • When accessing the buffer memory of the remote device station or remote station, the offset value or offset + write/read byte size out of the range of the buffer memory of the target station is specified. • When the own station number is 64 on the CC-Link Ver.2 board, a request was made to other station. • The specified IP address is incorrect.	• Check the status of the network, operation status of the access station(s), and mounting condition of the module(s). • Check the mounting condition of the board. • Check the Target Setting of the utility. • Check the Transient Timeout Monitoring Time of the utility. • Retry the operation. • When accessing CPU modules other than QCPU (Q mode) or MELSEC iQ-R series CPU modules, use the 32-bit version user application. • When this error has occurred while executing the RECV function with CC-Link IE Field Network board, use SW1DNC-CCIEF-B version 1.04E or later. • Check if the offset values or offset + write/read byte size is within the range of the buffer memory of the target station. • When requesting to other station by using CC-Link Ver.2 board, set a station number other than 64 to the own station number. • Check if a line error such as CRC error occurs. For the checking method, refer to the manuals of network systems to use. • Check the IP address.
66	0042H	■Channel-opened error Specified channel is already opened.	• Open the channel only once. • Since the correct value is stored to the path, this error can be regarded as normal status.
68	0044H	■Path error • The specified path is invalid. • No board exists at the specified path.	• Check the path. • Check the driver operating condition with Device Manager. • Check the mounting condition of the board.
69	0045H	■Unsupported function execution error • A function which is not supported by the target station was executed. • A function which is not supported by the specified channel is executed.	• Check the path of the channel, the network number, and the station number. • Check if the function is supported by the target station.
70	0046H	■Station number error • The specified station number is incorrect. • A process that should be requested to other station was requested to the own station, or the station number corresponds to the own station (255 (FFH)) but the network number is not 0. • The process was requested with a 64-bit version user application to a CPU module other than QCPU (Q mode) or MELSEC iQ-R series CPU modules.	• Check the network number and the station number. • When accessing CPU modules other than QCPU (Q mode) or MELSEC iQ-R series CPU modules, use the 32-bit version user application.
71	0047H	■No reception data error (for RECV function) • Data is not received. • With the RECV function of CC-Link IE Controller Network board or CC-Link IE Field Network board, the data over 960 bytes has been received from MELSECNET/H.	• Check the channel number. • Check whether the data is sent on the sending station with the SEND function. • Check if the sent data using the SEND function of MELSECNET/H board exceeds 960 bytes. • When the receive station is CC-Link IE Controller Network board, restart a personal computer of the receive station. • Retry the operation. • Check the system log of event viewer, and take a corrective action on the registered error.

Error Code		Error description	Corrective action
Decimal	Hexadecimal		
77	004DH	■Memory reservation error/resource memory shortage error Enough memory could not be reserved.	• Reduce the size of data to be accessed or the number of blocks. • If using the random write/random read function (mdRandWEx, mdRandW, mdRandREx, mdRandR), use the batch write/batch read function (mdSendEx, mdSend, mdReceiveEx, mdReceive) instead. • The memory may be insufficient. End another running application. • End the program and restart the personal computer. • Increase the minimum working set size of the personal computer. (☞ Page 148 Method for Increasing Minimum Working Set Size of Personal Computer)
85	0055H	■SEND/RCV channel number error The channel number specified with the SEND/RCV function is incorrect.	Check the channel number.
100	0064H	■Board H/W resource busy The next processing cannot be executed because of the insufficient resource on the board.	• Retry the operation. • If the error occurs repeatedly, please contact your local Mitsubishi Electric representative.
101	0065H	■Routing parameter error • The routing parameters, network number/station number, or network/station No. <-> IP information setting of the request source or relay CPU module are not set or are incorrect. • The control CPU module for the network module or the CPU module to which data is routed has not started for routing via a multiple CPU system. • The relay CPU module does not support the routing settings.	• Check the routing parameters, network number/station number, or network/station No. <-> IP information setting. • Retry it after a while. Or, check the startup of the system that relays data, and start communication. • Check if the system serving as the relay station supports the routing settings.
102	0066H	■Board Driver I/F error • An attempt to send request data to the board driver is failed. • The system resource of the operating system is insufficient. For details, refer to the following: ☞ Page 12 Influence of operating system and other applications	• Retry the operation. • Check with the function such as event log whether Windows® is operating normally. • End the program and restart the personal computer. • Check the mounting condition of the board.
103	0067H	■Board Driver I/F error • An attempt to receive response data from the board driver is failed. • The consistency between the board and the software package cannot be identified.	• Retry the operation. • Check with the function such as event log whether Windows® is operating normally. • End the program and restart the personal computer. • When this error has occurred while executing the SEND function with CC-Link IE Field Network board, use SW1DNC-CCIEF-B version 1.04E or later.
130	0082H	■Start device number error • The specified start device number is out of the range. • The set of start device number and size is over the device range.	• Check the start device number. • Check the start device number and size.
131	0083H	■Size error The set of start device number and size is over the device range.	Check the start device number and size.
133	0085H	■Parameter error A parameter set on the board is incorrect.	• Check if the installed board and the model name set in the parameter match. • Reset the board. • Correct the parameter.
4096 to 16383	1000H to 3FFFH	■MELSEC data link library internal error	• End the program and restart the personal computer. • Reinstall the software package. • Contact your local Mitsubishi Electric representative.
16384 to 20479	4000H to 4FFFH	■Error detected by the access target CPU	Refer to the user's manual of the access target CPU module.
16385	4001H	The specified target CPU does not exist.	Review the target CPU.
16386	4002H	A request that cannot be processed by the request destination station was received.	• Change the request destination. • Check the request destination station number and the content of request data.

Error Code		Error description	Corrective action
Decimal	Hexadecimal		
16418	4022H	■File related error Creating an event history file failed.	- Reset the board. - End the program and restart the personal computer.
16420	4024H	■File related error Accessing the event history file failed.	End the program and restart the personal computer.
16421	4025H	■File related error The event history file is being used by another board driver.	End the program and restart the personal computer.
16432	4030H	■Device error The specified device type is invalid.	- Check if the device is accessible for the specified network number and station number. (Page 31 ACCESSIBLE DEVICES AND RANGES) - Check if the specified device is valid on the programmable controller CPU of the target station.
16433	4031H	■Device error - The specified device number is out of the range. - The start I/O number of the specified devices is invalid. - The block number of the specified device is invalid.	- Check the device number. - Check the block number (device type) and the start I/O number of the specified device. - Check with the programmable controller CPU on the target station whether the specified device, the block number, and the start I/O number are valid.
16512	4080H	■Request data error - The content of request data has an error. - The process was requested with a 64-bit version user application to a CPU module other than QCPU (Q mode) or MELSEC iQ-R series CPU modules.	- Check the content of specified request data. - When accessing CPU modules other than QCPU (Q mode) or MELSEC iQ-R series CPU modules, use the 32-bit version user application.
16685	4022H	■File related error Creating an event history file failed.	- Increase the free space on the system drive. - Reset the board. - End the program and restart the personal computer.
16837	41C5H	■File related error No event history file exists.	- Reset the board. - End the program and restart the personal computer.
18944	4A00H	■Link-related error - The network of the number set to the routing parameters does not exist. - The network is not supported by the target CPU. - The network number or station number of the target station is incorrect.	- Check the routing parameter setting. - Exchange the CPU with a CPU which supports the network. - Check the network number and the station number.
18945	4A01H		
19202	4B02H	The request is not for a CPU module.	Perform the operation for a module for which the specific function can be executed.
28416 to 28671	6F00H to 6FFFH	■Error detected by the redundant function module	Refer to the user's manual of the access target redundant function module.
-1	FFFFH (FFFFFFFFH)	■Path error - The specified path is invalid. - A function which is not supported by the specified path was executed.	- Use the path that was returned with mdOpen function. - Use a path of the communication line that supports the function. - Check if the executed function is supported by the board used.
-2	FFFEH (FFFFFFEH)	■Start device number error - The specified start device number is out of the range. - When specifying a bit device, the start device number is not multiples of 8. - The set of start device number and points in the same block that is specified for the device random read/write, is over the device range. - The set of start device number and size is over the device range. - The specified target CPU does not exist. - The specification method for the target station is incorrect.	- Check the start device number. - When specifying a bit device, specify a start device number in multiples of 8. - Check the start device number and points. - Check the start device number and size. - Check if the specified device is valid on the programmable controller CPU of the target station. - Review the target CPU. - Check if the specification methods of the network number and station number are correct.
-3	FFFDH (FFFFFFFDH)	■Device type error - The specified device type is invalid. - The set of start device number and size of the specified device type is over the device range. - The specification method for the target station is incorrect.	- Check if the device is accessible for the specified network number and station number. (Page 31 ACCESSIBLE DEVICES AND RANGES) - Check the start device number and size. - Check if the specified device is valid on the programmable controller CPU of the target station. - Check if the specification methods of the network number and station number are correct.

Error Code		Error description	Corrective action
Decimal	Hexadecimal		
-5	FFFBH (FFFFFFFBH)	■Size error - The set of start device number and size is over the device range. - An access was attempted with odd number bytes. - The specified offset is incorrect.	- Check the device size. - Check the start device number and size. - Specify even number bytes. - Check the offset.
-6	FFFAH (FFFFFFFAH)	■Number of blocks error The number of blocks specified for the device random read/write is out of the range.	Check the number of blocks.
-8	FFF8H (FFFFFFF8H)	■Channel number error The channel number specified with mdOpen function is invalid.	Check the channel number.
-12	FFF4H (FFFFFFF4H)	■Block number error The block number of the specified file register is invalid.	- Check the block number (device type) of the file register. - Check if the specified device is valid on the programmable controller CPU of the target station.
-13	FFF3H (FFFFFFF3H)	■Write protect error The block number of the specified extension file register is overlapping with the write protect area of the memory cassette.	- Check the block number (device type) of the extension file register. - Check the write protect switch of the memory cassette on the programmable controller CPU of the target station.
-16	FFF0H (FFFFFFF0H)	■Network number and station number error The specified network number or station number is out of the range.	- Check the network number and the station number. - Check the Target Setting of the utility.
-17	FFE FH (FFFFFFE FH)	■All station specification and group number specification error - All stations or group number was specified for a function other than the SEND function. - The device type with arrival acknowledgment was specified when using the SEND function with all station specification and group number specification.	- Check the network number and the station number. - Check if the function supports all station specification and group number specification. - Specify the device type without arrival acknowledgment when using the SEND function with all station specification and group number specification.
-18	FFEEH (FFFFFFEEH)	■Remote command code error A command code which is not valid for mdControl was specified.	Check the command code.
-19	FFEDH (FFFFFFEDH)	■SEND/RECV channel number error The channel number specified for the SEND/RECV function is out of the range.	Check the channel number.
-31	FFE1H (FFFFFFE1H)	■DLL load error An attempt to load DLL required to execute the function failed.	- Check whether an unsupported communication target is specified by the network number and the station number. - Reduce the size of data to be accessed or the number of blocks. - If using the random write/random read function (mdRandWEx, mdRandW, mdRandREx, mdRandR), use the batch write/batch read function (mdSendEx, mdSend, mdReceiveEx, mdReceive) instead. - The memory may be insufficient. End another running application. - End the program and restart the personal computer. - Reinstall the software package.
-32	FFE0H (FFFFFFE0H)	■Resource time-out error - The user program was forcibly terminated. - The resource is not freed within the transient timeout monitoring time because other tasks and threads are occupying the resource.	- End the user program correctly. - Retry the operation. - The memory may be insufficient. End another running application. - End the program and restart the personal computer.
-33	FFDFH (FFFFFFDFH)	■Incorrect access target error The communication target specified by the network number and the station number is a model which is not supported.	- Check whether an unsupported communication target is specified by the network number and the station number. - Check the Target Setting of the utility. - Update the software package.
-34	FFDEH (FFFFFFDEH)	■Registry access error	- Check if the parameters are set. - Reinstall the software package.
-35	FFDDH (FFFFFFDDH)		
-36	FFDCH (FFFFFFDCH)		

Error Code		Error description	Corrective action
Decimal	Hexadecimal		
-37	FFDBH (FFFFFFDBH)	■Communication initialization setting error The initial setting for communication is failed.	• Retry the operation. • The memory may be insufficient. End another running application. • End the program and restart the personal computer. • Check the free space of the memory.
-42	FFD6H (FFFFFFD6H)	■Close error The communication cannot be closed.	• Retry the operation. • End the program and restart the personal computer.
-43	FFD5H (FFFFFFD5H)	■ROM operation error A TC setting value was written to the CPU during ROM operation.	Change the TC setting value during RAM operation.
-61	FFC3H (FFFFFFC3H)	■Number of events error The number of events which is specified with mdWaitBdEvent function to set the user application to wait is out of the range.	Check the number of events that sets the user application to wait.
-62	FFC2H (FFFFFFC2H)	■Event number error The event number which is specified with mdWaitBdEvent function to set the user application to wait is out of the range.	Check the event number that sets the user application to wait.
-63	FFC1H (FFFFFFC1H)	■Event number duplicate registration error The event number which is specified with mdWaitBdEvent function to set the user application to wait is duplicated.	Specify the event number, that set the user application to wait, so as not to be duplicated.
-64	FFC0H (FFFFFFC0H)	■Timeout time error The timeout time specified with mdWaitBdEvent function is out of the range.	Check the time-out value.
-65	FFBFH (FFFFFFBFH)	■Event wait time-out error An event did not occur within the timeout time.	Retry the operation.
-66	FFBEH (FFFFFFBEH)	■Event initialization error The board or the master station (control station) was reset during the execution of mdWaitBdEvent function.	Retry the operation.
-67	FFBDH (FFFFFFBDH)	■No event setting error There is no event setting of the event number that is specified with mdWaitBdEvent function to wait the user application.	Set the event number that sets the user application to wait in the utility.
-69	FFBBH (FFFFFFBBH)	■Unsupported function execution error A function which is not supported by the software package or the driver was executed.	Check if the function is supported by the software package or the driver.
-70	FFBAH (FFFFFFBAH)	■Event duplication occurrence error An, event with the same event number occurred multiple times.	Set the interval of the event occurrence longer enough for the user program to process.
-71	FFB9H (FFFFFFB9H)	■Remote device station/remote station access error Accessing the buffer memory of a remote device station or remote station failed.	• Check the status of the network and operation status of the access station(s). • Check if the target station is a remote device station on CC-Link IE Field Network or a remote station on CC-Link IE TSN. • Check the network number and the station number. • Check the IP address. • Check if it is accessed via another network. • Check if the offset values or offset + write/read byte size is within the range of the buffer memory of the target station.
-72	FFB8H (FFFFFFB8H)	■Pointer error The address of the specified pointer is incorrect.	Check the address of the pointer.
-73	FFB7H (FFFFFFB7H)	■IP address error The specified IP address is out of the range.	Check the IP address.
-257 to -4096	FEFFH to F000H (FFFFFFFFH to FFFFF000H)	■Errors detected in the MELSECNET/H and MELSECNET/10 network system	Refer to the reference manuals of MELSECNET/10 or MELSECNET/H network systems.
-2174	F782H (FFFFF782H)	■Transient data target station number error • The target station number is incorrect. • The target station number is 0.	• Check the target station number, and perform the operation again. • If the error occurs after performing the above corrective action, please contact your local Mitsubishi Electric representative.

Error Code		Error description	Corrective action
Decimal	Hexadecimal		
-4097 to -8192	EFFFH to E000H (FFFFEFFFH) to (FFFFE000H)	■Errors detected in the CC-Link IE Controller Network system	If an error not listed in this table occurs, refer to the manual for CC-Link IE Controller Network.
-7656	E218H (FFFFE218H)	■Transient data target station number error - The target station number is incorrect. - The target station number is 0.	- Check the target station number, and perform the operation again. - If the error occurs after performing the above corrective action, please contact your local Mitsubishi Electric representative.
-7672	E208H (FFFFE208H)		
-8193 to -12288	DFFFH to D000H (FFFFDFFFH) to (FFFFD000H)	■Errors detected in the CC-Link IE Field Network system or CC-Link IE TSN system	If an error not listed in this table occurs, refer to the user's manual for a module or board on the network with the error detected.
-11683	D25DH (FFFFD25DH)	■Transient data improper The data over 960 bytes has been sent from MELSECNET/H board to CC-Link IE Field Network board.	Check if the sent data using the SEND function of MELSECNET/H board exceeds 960 bytes.
-11717	D23BH (FFFFD23BH)	■Network number error A network number other than its own network was specified with the mdRemBufReadEx function/ mdRemBufWriteEx function.	Check whether an unsupported communication target is specified by the network number.
-11746	D21EH (FFFFD21EH)	■Station number error - The specified station number is incorrect. - A process that should be requested to other station was requested to the own station, or the station number corresponds to the own station (255 (FFH)) but the network number is not 0.	Check the network number and the station number.
-12128	D0A0H (FFFFD0A0H)	■Transient data send response wait time-out error	- Check if the own station or the target station is disconnected. - Check if an unsupported communication target is specified with the network number and the station number.
-12289 to -16384	CFFFH to C000H (FFFFCFFFH) to (FFFFC000H)	■Errors detected in the Ethernet network system or CC-Link IE TSN system	If an error not listed in this table occurs, refer to the user's manual for a module or board on the network with the error detected.
-16385 to -20480	BFFFH to B000H (FFFFBFFFH) to (FFFFB000H)	■Errors detected in the CC-Link system	If an error not listed in this table occurs, refer to the user's manual for a module or board on CC-Link.
-18560	B780H (FFFFB780H)	■Module mode setting error A transient transmission was executed to the remote I/O station.	Check the network number and the station number.
-18572	B774H (FFFFB774H)	■Transient unsupported error A transient request was transmitted to the station that is not an intelligent device station.	- Check the network number and the station number. - Specify the station number for the intelligent device station. - Check the device type
-25056	9E20H (FFFF9E20H)	■Processing code error - A processing code that cannot be processed by the request destination station was set. (Request destination link module check) - The process was requested with a 64-bit version user application to a CPU module other than QCPU (Q mode) or MELSEC iQ-R series CPU modules.	- Check the request destination station number and the processing code. - Do not execute a function other than mdTypeRead function to a board in another station. - When accessing CPU modules other than QCPU (Q mode) or MELSEC iQ-R series CPU modules, use the 32-bit version user application.
-26334	9922H (FFFF9922H)	■Reset execution error/firmware update execution error - Reset was executed by another task that uses the same channel when accessing the own station or the other station. - Firmware update for a board that uses the same channel was executed when accessing the own station or the other station. - Reset operation or firmware update was executed when monitoring with the utility. - Firmware update was canceled.	- Retry the operation. - Reset the board after the firmware update is completed. - If the firmware update is canceled, reset the board and execute the firmware update again. Then, retry the operation. - Restart the personal computer.

Error Code		Error description	Corrective action
Decimal	Hexadecimal		
-26336	9920H (FFFF9920H)	■Routing request error on routing function unsupported station A routing to another loop was requested to the station which does not support the routing function.	Check the routing parameter setting.
-27902	9302H (FFFF9302H)	■Event wait time-out error An event did not occur within the timeout time specified with mdWaitBdEvent function.	• Lengthen the timeout time. • Check the condition to occur an event.
-28079	9251H (FFFF9251H)	■Channel No. reading error Reading the channel number failed.	Set the channel number again for all boards.
-28080	9250H (FFFF9250H)	Incorrect channel No. error An incorrect channel number was written.	
-28138	9216H (FFFF9216H)	■Unsupported block data assurance per station Reset or restart was performed to the CC-Link Ver.2 Board of which ROM version is 1A and the function "block data assurance per station" is enabled.	• Replace it with the ROM version 2B or later board. • Disable the block data assurance per station.
-28139	9215H (FFFF9215H)	■Link refresh error Link refresh processing did not operate normally.	• Reset the board. • The memory may be insufficient. End another running application. • End the program and restart the personal computer. • Check the free space of the memory. • Check the mounting condition of the board. • A personal computer error is suspected if the board operates normally on other personal computers. Repair or replace the personal computer. • When the same error occurs on other personal computers, replace the board. • Contact your local Mitsubishi Electric representative.
-28140	9214H (FFFF9214H)	■Incorrect mode setting error An incorrect mode was specified when setting the mode.	Check the mode.
-28141	9213H (FFFF9213H)	■System sleep error Entering sleep mode, hibernation mode, or fast startup was detected.	• End the program and restart the personal computer. • Change the setting of the power option to prevent the system from entering sleep mode, hibernation mode, or fast startup.
-28142	9212H (FFFF9212H)	■Mode error A request which cannot be used in the currently set mode was executed.	• Check if the parameters are set. • Check the currently set mode.
-28143	9211H (FFFF9211H)	■Hardware self-diagnosis error An error was detected by the hardware self-diagnosis.	• Check the system log of event viewer, and take a corrective action on the registered error. • End the program and restart the personal computer. • Take anti-noise measures for a personal computer. • Put the connector in and out after turning OFF the personal computer. • Check the mounting condition of the board. • A personal computer error is suspected if the board operates normally on other personal computers. Repair or replace the personal computer. • When the same error occurs on other personal computers, replace the board. • Contact your local Mitsubishi Electric representative.
-28144	9210H (FFFF9210H)		
-28150	920AH (FFFF920AH)	■Data link disconnected device access error An access was attempted to the device ranges of own station devices RX, RY, RWw, RWr which are assigned to the data link interrupted station or the reserved station.	• Check the specified device start number and size, or the device range of the parameter on the master station. The data write/read function can be performed even when this error occurs, but the function of the data security is not guaranteed.
-28151	9209H (FFFF9209H)	■Abnormal data reception error Abnormal response data was received.	• Check for errors in the target station and on the programmable controller CPU in the target station. If they are normal, request a process again. • Check the network status by referring to the manuals of each product.
-28153	9207H (FFFF9207H)	■Data reading error Reading data in board information failed.	• End the program and restart the personal computer. • Reinstall the software package.
-28154	9206H (FFFF9206H)	■Abnormal data reception error Abnormal request data was received.	

Error Code		Error description	Corrective action
Decimal	Hexadecimal		
-28158	9202H (FFFF9202H)	■Driver WDT error Driver WDT error is occurring. A driver WDT error may occur from the temporary system overload by the following factors. • Windows® activation process when starting the personal computer • Operation of a device driver such as a graphic board • Operation of other software applications	• Reset the board. • Restart the personal computer. • Remove the factor of system overload. • Clear "Use driver WDT function" with the utility to disable WDT. Or, extend the Driver WDT monitoring time. By changing the graphic board, an environment in which a driver WDT error does not occur may be created.
-28160	9200H (FFFF9200H)	■Hardware resource error Acquiring the hardware resource failed.	• Check the system log of event viewer, and take a corrective action on the registered error. • End the program and restart the personal computer. • Check the mounting condition of the board. • A personal computer error is suspected if the board operates normally on other personal computers. Repair or replace the personal computer. • When the same error occurs on other personal computers, replace the board.
-28611 to -28612	903DH to 903CH (FFFF903DH to FFFF903CH)	■System error	• Restart the personal computer. • Reinstall the latest software package.
-28622	9032H (FFFF9032H)	■Channel busy (dedicated instruction) error The channel specified for "Channel used by the own station" or "Target station channel" is being used by another instruction.	Wait for a little while, and retry it. Change the setting of "Channel used by the own station" or "Target station channel" in the control data.
-28634	9026H (FFFF9026H)	■Hardware self-diagnosis error An error was detected by the hardware self-diagnosis.	• Check the system log of event viewer, and take a corrective action on the registered error. • End the program and restart the personal computer. • Check the mounting condition of the board. • A personal computer error is suspected if the board operates normally on other personal computers. Repair or replace the personal computer. • When the same error occurs on other personal computers, replace the board. • Contact your local Mitsubishi Electric representative.
-28636	9024H (FFFF9024H)		

Point

When an error code is returned as a return value of the extended function ( Page 43 Function List), the error codes from -1 to -28636 will be an 8-digit value (FFFFFFFFH to FFFF9024H) in the hexadecimal format as described in the table.

APPENDIX

Appendix 1 Method for Increasing Minimum Working Set Size of Personal Computer

The following explains the method and sample programs for increasing the minimum working set size of the personal computer when an error code 77 (004DH) occurs due to the execution of the MELSEC data link library function.

Personal computer board drivers included in the following run using the minimum working set size in the memory area reserved in the user program.

- SW1DNC-CCBD2-B with version 1.08J or earlier
- SW0DNC-MNETH-B with version 21X or earlier

Some user programs may use a larger size for the minimum working set.

Therefore, if the minimum working set size for the personal computer board driver cannot be reserved, an error code 77 is returned.

In this situation, increase the minimum working set size in the user program before executing the MELSEC data link library function. (📖 Page 149 Sample programs)

The minimum working set size of 200 KB is reserved at startup of the personal computer.

When applying the sample programs introduced in this manual to the actual system, ensure the applicability and confirm that they will not cause system control problems.

Processing overview of sample program

1. Obtain the user program ID with the `GetCurrentProcessId` function.
2. Using the ID obtained in step 1, obtain the user program handle with the `OpenProcess` function.
3. The current minimum and maximum working set sizes can be obtained by executing the `GetProcessWorkingSetSize` function.
4. Set a size larger than the minimum working set obtained in step 3 and execute `SetProcessWorkingSetSize` function.
5. Release the user program handle with the `CloseHandle` function.

Sample programs

When setting with Visual Basic

■ An example when the minimum working set size is 1 MB and the maximum working set size is 3 MB

When programming with Visual Basic 5.0 or Visual Basic 6.0, change the type definition of variables (id, ph, wkmin, wkmax) from Integer to Long.

The set sizes shown here are reference sizes. Adjust the sizes according to your system.

Program example

```
' Description for using functions
Declare Function GetCurrentProcessId Lib "Kernel32.dll" () As Integer
Declare Function OpenProcess Lib "Kernel32.dll" (ByVal dwDesiredAccess As Integer,
    ByVal bInheritHandle As Integer, ByVal dwProcessId As Integer) As Integer
Declare Function GetProcessWorkingSetSize Lib "Kernel32.dll" (ByVal hProcess As Integer,
    ByRef lpMinimumWorkingSetSize As Integer, ByRef lpMaximumWorkingSetSize As Integer) As Integer
Declare Function SetProcessWorkingSetSize Lib "Kernel32.dll" (ByVal hProcess As Integer,
    ByVal dwMinimumWorkingSetSize As Integer, ByVal dwMaximumWorkingSetSize As Integer) As Integer
Declare Function CloseHandle Lib "Kernel32.dll" (ByVal hObject As Integer) As Integer
Sub ChangeWorkingSetSize()
    Dim id As Integer ' User program ID variable
    Dim ph As Integer ' User program handle variable
    Dim wkmin As Integer ' Minimum working set variable
    Dim wkmax As Integer ' Maximum working set variable
    Dim iret As Integer ' Return value
    ' Obtain the user program ID
    id = GetCurrentProcessId()
    ' Open the user program handle
    'PROCESS_SET_QUOTA = 256, PROCESS_QUERY_INFORMATION = 1024
    ph = OpenProcess(256 + 1024, 0, id)
    ' Obtain the maximum working set size and the minimum working set size for the user program
    iret = GetProcessWorkingSetSize(ph, wkmin, wkmax)
    ' Set the minimum working set size to 1 MB (1 * 1024 * 1024 = 1048576)
    wkmin = 1048576
    ' Set the maximum working set size to 3 MB (3 * 1024 * 1024 = 3145728)
    wkmax = 3145728
    ' Change the maximum working set size and the minimum working set size for the user program
    iret = SetProcessWorkingSetSize(ph, wkmin, wkmax)
    ' Close the user program handle
    iret = CloseHandle(ph)
End Sub
```

When setting with Visual C++

■An example when the minimum working set size is 1 MB and the maximum working set size is 3 MB

The set sizes shown here are reference sizes. Adjust the sizes according to your system.

Program example

```
#include <windows.h>
#define RET_ERR -1
short ChangeWorkingSetSize()
{
    DWORD dwProcessId; /*User program ID variable*/
    HANDLE hProcess; /*User program handle variable*/
    SIZE_T dwMinimumWorkingSetSize; /*Minimum working set variable*/
    SIZE_T dwMaximumWorkingSetSize; /*Maximum working set variable*/

    /*Obtain the user program ID*/
    dwProcessId = GetCurrentProcessId();

    /*Open the user program handle*/
    hProcess = OpenProcess(PROCESS_SET_QUOTA + PROCESS_QUERY_INFORMATION, FALSE, dwProcessId);
    if(hProcess == NULL){
        /*Error end*/
        return(RET_ERR);
    }

    /*Obtain the maximum working set size and the minimum working set size for the user program */
    if (GetProcessWorkingSetSize(hProcess, &dwMinimumWorkingSetSize, &dwMaximumWorkingSetSize) == 0){
        /*Error end*/
        CloseHandle(hProcess);
        return(RET_ERR);
    }

    /*Set the minimum working set size to 1 MB*/
    dwMinimumWorkingSetSize = 1 * 1024 * 1024;
    /*Set the maximum working set size to 3 MB*/
    dwMaximumWorkingSetSize = 3 * 1024 * 1024;

    /*Change the maximum working set size and the minimum working set size for the user program */
    if (SetProcessWorkingSetSize(hProcess, dwMinimumWorkingSetSize, dwMaximumWorkingSetSize) == 0){
        /*Error end*/
        CloseHandle(hProcess);
        return(RET_ERR);
    }

    /*Close the user program handle*/
    CloseHandle(hProcess);
    /*Normal return*/
    return(0);
}
```


FUNCTION INDEX

M

mdBdLedRead (reading LED information of the board)	96
mdBdModRead (reading modes of board)	94
mdBdModSet (setting modes of board)	92
mdBdRst (resetting board)	91
mdBdSwRead (reading switch status of the board)	103
mdBdVerRead (reading version information of the board)	104
mdClose (closing communication lines)	55
mdControl (remote RUN/STOP/PAUSE)	87
mdDevRst (resetting bit devices)	120
mdDevRstEx (resetting extended bit devices)	74
mdDevSet (setting bit devices)	119
mdDevSetEx (setting extended bit devices)	73
mdInit (initializing programmable controller information table)	106
mdOpen (opening communication lines)	54
mdRandR (reading devices randomly)	117
mdRandREx (reading extended devices randomly)	69
mdRandW (writing devices randomly)	115
mdRandWEx (writing extended devices randomly)	66
mdReceive (batch reading devices)	111
mdReceive (RECV function)	113
mdReceiveEx (batch reading extended devices)	61
mdReceiveEx (RECV function)	63
mdRemBufReadEx (reading data from the buffer memory of a remote device station/remote station)	77
mdRemBufReadIPEX (reading data from the buffer memory of a remote station, target station IP address specified)	81
mdRemBufWriteEx (writing data to the buffer memory of a remote device station/remote station)	75
mdRemBufWriteIPEX (writing data from the buffer memory of a remote station, target station IP address specified)	79
mdSend (batch writing devices)	107
mdSend (SEND function)	109
mdSendEx (batch writing extended devices)	56
mdSendEx (SEND function)	58
mdTypeRead (reading model names of CPU)	83
mdWaitBdEvent (waiting for event occurrence)	88

MEMO

REVISIONS

*The manual number is written at the bottom left of the back cover.

Revision date	*Manual number	Description
July 2011 to September 2015	SH(NA)-081035ENG-A to SH(NA)-081035ENG-H	Due to the transition to the e-Manual, the content of the revisions has been deleted.
May 2016	SH(NA)-081035ENG-I	Complete revision (layout change)
September 2016	SH(NA)-081035ENG-J	■Added or modified parts Section 2.3, Section 4.2, Section 4.3, Chapter 5, Chapter 6
October 2016	SH(NA)-081035ENG-K	■Added or modified parts Chapter 5, Chapter 7
January 2017	SH(NA)-081035ENG-L	■Added or modified parts Section 3.1, Section 3.2, Section 4.1, Section 4.3, Chapter 5, Chapter 7
January 2018	SH(NA)-081035ENG-M	■Added or modified part Chapter 7
June 2019	SH(NA)-081035ENG-N	■Added or modified parts Section 2.2, Section 2.3, Section 3.1, Section 3.2, Section 3.3, Section 4.3, Section 5.1, Section 5.3, Chapter 7
July 2019	SH(NA)-081035ENG-O	■Added or modified parts Section 3.1, Section 3.2
April 2020	SH(NA)-081035ENG-P	■Added or modified parts Section 4.2, Section 4.3, Section 6.8
January 2022	SH(NA)-081035ENG-Q	■Added or modified parts Section 2.2, Chapter 5
July 2022	SH(NA)-081035ENG-R	■Added or modified parts GENERIC TERMS AND ABBREVIATIONS, DISCONTINUED MODELS, Section 2.2, Section 4.2, Chapter 7
October 2023	SH(NA)-081035ENG-S	■Added models NZ81GN11-SX, NZ81GN11-T2 ■Added or modified parts RELEVANT MANUALS, GENERIC TERMS AND ABBREVIATIONS, Section 2.2, Section 2.3, Section 3.1, Section 3.2, Section 3.3, Section 4.1, Section 4.2, Section 4.3, Chapter 5, Section 5.1, Section 5.2, Section 5.3, Chapter 6, Section 6.3, Section 6.6, Chapter 7, Appendix 1
June 2024	SH(NA)-081035ENG-T	■Added or modified parts Section 4.3, Chapter 6, Section 6.3, Chapter 7

Japanese manual number: SH-081034-T

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

© 2011 MITSUBISHI ELECTRIC CORPORATION

CONDITIONS OF USE FOR THE PRODUCT

- (1) MELSEC programmable controller ("the PRODUCT") shall be used in conditions;
- i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and
 - ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.
- (2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries. MITSUBISHI ELECTRIC SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI ELECTRIC USER'S, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.
- ("Prohibited Application")
- Prohibited Applications include, but not limited to, the use of the PRODUCT in;
- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
 - Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
 - Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.
- Notwithstanding the above restrictions, Mitsubishi Electric may in its sole discretion, authorize use of the PRODUCT in one or more of the Prohibited Applications, provided that the usage of the PRODUCT is limited only for the specific applications agreed to by Mitsubishi Electric and provided further that no special quality assurance or fail-safe, redundant or other safety features which exceed the general specifications of the PRODUCTS are required. For details, please contact the Mitsubishi Electric representative in your region.
- (3) Mitsubishi Electric shall have no responsibility or liability for any problems involving programmable controller trouble and system trouble caused by DoS attacks, unauthorized access, computer viruses, and other cyberattacks.

WARRANTY

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

1. Gratis Warranty Term and Gratis Warranty Range

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

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

[Gratis Warranty Term]

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

[Gratis Warranty Range]

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

2. Onerous repair term after discontinuation of production

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

3. Overseas service

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

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

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to:

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

5. Changes in product specifications

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

INFORMATION AND SERVICES

For further information and services, please contact your local Mitsubishi Electric sales office or representative.
Visit our website to find our locations worldwide.

Locations Worldwide

mitsubishi electric Factory Automation Global Website

www.MitsubishiElectric.com/fa/about-us/overseas/

TRADEMARKS

Microsoft, Visual Basic, Visual C++, Visual C#, Visual Studio, Windows, and Windows XP are trademarks of the Microsoft group of companies.

The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.

In some cases, trademark symbols such as "™" or "®" are not specified in this manual.

COPYRIGHTS

The screens (screenshots) are used in accordance with the Microsoft Corporation guideline.

SH(NA)-081035ENG-T(2406)KWIX

MODEL: MD-FUNC-LIB-R-E

MODEL CODE: 13JV25

mitsubishi electric corporation

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA 461-8670, JAPAN

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

Specifications subject to change without notice.