

MITSUBISHI

Mitsubishi PC Interface Board

MELSEC

MELSEC Data Link Library Reference Manual

Q80BD-J61BT11N
Q81BD-J61BT11

Q80BD-J71LP21-25
Q80BD-J71LP21S-25
Q81BD-J71LP21-25
Q80BD-J71LP21G
Q80BD-J71LP21GE
Q80BD-J71BR11

Q80BD-J71GP21-SX
Q80BD-J71GP21S-SX
Q81BD-J71GP21-SX
Q81BD-J71GP21S-SX

Q81BD-J71GF11-T2

MODEL

● 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.

●CONDITIONS OF USE FOR THE PRODUCT●

- (1) Mitsubishi programmable controller ("the PRODUCT") shall be used in conditions;
- i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and
 - ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.

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

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

("Prohibited Application")

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

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

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

INTRODUCTION

Thank you for purchasing the PC interface board.

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 related manuals carefully and develop familiarity with the functions and performance of the MELSEC data link library to handle the product correctly.

Please make sure that the end users read this manual.

RELATED MANUALS

The following are the manuals relevant to this product.
Refer to the following tables when ordering required manuals.

(1) Related manuals of CC-Link Ver.2 board

Manual name <Manual number (model code)>	Description
Type Q80BD-J61BT11N/Q81BD-J61BT11 CC-Link System Master/Local Interface Board User's Manual (For SW1DNC-CCBD2-B) <SH-080527ENG, 13JR77>	Overview of system configuration, specifications, functions, handling, wiring, and troubleshooting for type Q80BD-J61BT11N /Q81BD-J61BT11 CC-Link system master/local interface board.
MELSEC-L CC-Link System Master/Local Module User's Manual <SH-080895ENG, 13JZ41>	Overview of system configuration, performance specifications, functions, handling, wiring, and troubleshooting for L series master/local modules.
CC-Link System Master/Local Module User's Manual <SH-080394E, 13JR64>	Overview of system configuration, performance specifications, functions, handling, wiring, and troubleshooting for QJ61BT11N.
CC-Link System Master/Local Module Type AJ61BT11/A1SJ61BT11 User's Manual <IB-66721, 13J872>	Overview of system configuration, performance specifications, functions, handling, wiring, and troubleshooting for AJ61BT11 and A1SJ61BT11.
CC-Link System Master/Local Module Type AJ61QBT11/A1SJ61QBT11 User's Manual <IB-66722, 13J873>	Overview of system configuration, performance specifications, functions, handling, wiring, and troubleshooting for AJ61QBT11 and A1SJ61QBT11.

(2) Related manuals of MELSECNET/H board

Manual name <Manual number (model code)>	Description
MELSECNET/H Interface Board User's Manual (For SW0DNC-MNETH-B) <SH-080128, 13JR24>	Overview of system configuration, specifications, functions, handling, wiring, and troubleshooting for MELSECNET/H interface boards.
Q Corresponding MELSECNET/H Network System Reference Manual (PLC to PLC network) <SH-080049, 13JF92>	Overview of system configuration, performance specifications, functions, handling, wiring, and troubleshooting for the MELSECNET/H network system.
For QnA/Q4AR MELSECNET/10 Network System Reference Manual <IB-66690, 13JF78>	Overview of system configuration, performance specifications, functions, handling, wiring, and troubleshooting for the MELSECNET/10 network system.

(3) Related manuals of CC-Link IE Controller Network board

Manual name <Manual number (model code)>	Description
CC-Link IE Controller Network Interface Board User's Manual (For SW1DNC-MNETG-B) <SH-080691ENG, 13JZ02>	Overview of system configuration, specifications, functions, handling, wiring, and troubleshooting for CC-Link IE Controller Network interface board.
CC-Link IE Controller Network Reference Manual <SH-080668ENG, 13JV16>	Overview of system configuration, performance specifications, functions, handling, wiring, and troubleshooting for CC-Link IE Controller Network.

(4) Related manuals of CC-Link IE Field Network board

Manual name <Manual number (model code)>	Description
CC-Link IE Field Network Interface Board User's Manual (For SW1DNC-CCIEF-B) <SH-080980ENG, 13JZ58>	Overview of system configuration, specifications, functions, handling, wiring, and troubleshooting for CC-Link IE Field Network interface board.
MELSEC-Q CC-Link IE Field Network Master/Local Module User's Manual <SH-080917ENG, 13JZ47>	Overview of CC-Link IE Field Network, and specifications, procedures before operation, system configuration, installation, wiring, settings, functions, programming, and troubleshooting for CC-Link IE Field Network master/local module.
MELSEC-L CC-Link IE Field Network Head Module User's Manual <SH-080919ENG, 13JZ48>	Overview of specifications, procedures before operation, system configuration, installation, wiring, settings, and troubleshooting for head module.
CC-Link IE Field Network Ethernet Adapter Module User's Manual <SH-080939ENG, 13JZ50>	Overview of specifications, procedures before operation, system configuration, installation, wiring, settings, and troubleshooting for Ethernet adapter module.

Remark

User's manuals of each type of boards are stored on the CD-ROM of the software package in a PDF file format.
Manuals in printed form are sold separately for a single purchase. Order a manual by quoting the manual number (model code) listed in the above table.

Memo

CONTENTS

SAFETY PRECAUTIONS	1
CONDITIONS OF USE FOR THE PRODUCT	2
INTRODUCTION	3
RELATED MANUALS	3
HOW TO READ THIS MANUAL	8
TERMS	9
CHAPTER 1 OVERVIEW	10
1.1 MELSEC Data Link Library	10
1.2 Supported Boards	10
CHAPTER 2 PROGRAMMING	11
2.1 Programming Procedure	11
2.2 Precautions when Using MELSEC Data Link Library	12
2.2.1 Precautions when programming	12
2.2.2 Precautions when accessing own station link devices and other station's programmable controller devices	14
2.3 Settings for Using Functions	16
2.3.1 Using Visual Basic®5.0, Visual Basic®6.0	16
2.3.2 Using Visual C++®5.0, Visual C++®6.0	17
2.3.3 Using Visual Basic®.NET	19
2.3.4 Using Visual C++®.NET 2003, Visual C++®2005, Visual C++®2008	20
2.3.5 Using Visual C++®2010	22
CHAPTER 3 ACCESSIBLE DEVICES AND RANGES	27
3.1 Accessible Devices	27
3.1.1 Accessible devices for CC-Link Ver.2 board	27
3.1.2 Accessible devices for MELSECNET/H board	31
3.1.3 Accessible devices for CC-Link IE Controller Network board	34
3.1.4 Accessible devices for CC-Link IE Field Network board	37
3.2 Accessible Ranges	40
3.2.1 Accessible ranges of CC-Link Ver.2 board	40
3.2.2 Accessible ranges of MELSECNET/H board	41
3.2.3 Accessible ranges of CC-Link IE Controller Network board	43
3.2.4 Accessible ranges of CC-Link IE Field Network board	44
CHAPTER 4 FUNCTIONS	45
4.1 List of Functions	45
4.2 Common Specifications of Functions	47
4.2.1 Specifying channel number	47
4.2.2 Specifying station numbers	48
4.2.3 Specifying network numbers and station numbers for extended functions	50
4.2.4 Specifying device types	51
4.3 Function Details	54
4.3.1 mdOpen (opening communication lines)	55

4.3.2	mdClose (closing communication lines)	56
4.3.3	mdSendEx (batch writing extended devices / SEND function)	57
4.3.4	mdReceiveEx (batch reading extended devices / RECV function)	61
4.3.5	mdRandWEx (writing extended devices randomly)	65
4.3.6	mdRandREx (reading extended devices randomly)	68
4.3.7	mdDevSetEx (setting extended bit devices)	71
4.3.8	mdDevRstEx (resetting extended bit devices)	73
4.3.9	mdTypeRead (reading model names of CPU)	75
4.3.10	mdControl (remote RUN/STOP/PAUSE)	78
4.3.11	mdWaitBdEvent (waiting for event occurrence)	79
4.3.12	mdBdRst (resetting board)	82
4.3.13	mdBdModSet (setting modes of board)	83
4.3.14	mdBdModRead (reading modes of board)	85
4.3.15	mdBdLedRead (read LED information of the board)	87
4.3.16	mdBdSwRead (reading switch status of the board)	93
4.3.17	mdBdVerRead (read version information of the board)	95
4.3.18	mdInit (initializing programmable controller information table)	98
4.3.19	mdSend (batch writing devices / SEND function)	99
4.3.20	mdReceive (batch read devices / RECV function)	103
4.3.21	mdRandW (writing devices randomly)	107
4.3.22	mdRandR (reading devices randomly)	110
4.3.23	mdDevSet (setting bit devices)	113
4.3.24	mdDevRst (resetting bit devices)	114

CHAPTER 5 SAMPLE PROGRAMS	115
----------------------------------	------------

5.1	Sample Programs for CC-Link Ver.2 Board	115
5.1.1	Description of sample programs	116
5.1.2	Programming language	117
5.2	Sample Programs for MELSECNET/H Board	118
5.2.1	Description of sample programs	118
5.2.2	Programming language	119
5.3	Sample Programs for CC-Link IE Controller Network Board	121
5.3.1	Description of sample programs	121
5.3.2	Programming language	122
5.4	Sample Programs for CC-Link IE Field Network Board	124
5.4.1	Description of sample programs	124
5.4.2	Programming language	125

CHAPTER 6 ERROR CODES	126
------------------------------	------------



APPENDIX	133
-----------------	------------

Appendix 1	Method for Increasing Minimum Working Set Size of Personal Computer	133
Appendix 1.1	Processing overview of sample program	133
Appendix 1.2	Sample programs	134
REVISIONS	136
Warranty	138

HOW TO READ THIS MANUAL

In this manual, pages are organized and the symbols are used as shown below.


The following page illustration is for explanation purpose only, and the content is different from the actual page.


 shows reference pages.
 shows reference manuals.

CHAPTER 2 PROGRAMMING

2.3 Settings for Using Functions

This section explains how to set the development tool to use the MELSEC data link library functions.
For programming languages supported by boards, refer to the User's Manual corresponds to the board.

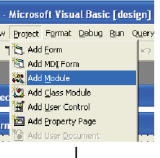
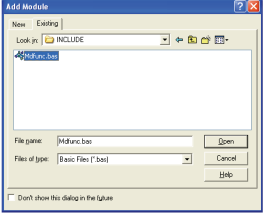
 Page 3 RELEVANT MANUALS

Point 

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®.

2.3.1 Using Visual Basic®5.0, Visual Basic®6.0

This section explains the setting operation when using Visual Basic®5.0, Visual Basic®6.0.






- 1) Start Visual Basic® and select [Project] - [Add Module].
- 2) Select <<Existing>> and select "Mdfunc.bas".
"Mdfunc.bas" is stored in the following directory at installation.
<User-specified folder> - <COMMON> - <INCLUDE>

2.3 Settings for Using Functions
2.3.1 Using Visual Basic®5.0, Visual Basic®6.0

14



The chapter of the current page is shown.

Point  shows notes that requires attention.

Remark  shows useful information.

The section of the current page is shown.

The following shows the symbols used in this manual with descriptions and examples.

Symbol	Description	Example
[]	Menu name on a menu bar	[Project]
<< >>	Tab name on a screen	<<Existing>> tab
" "	Screen name or item name on a screen	"Option" screen
	Button on a screen	 button

TERMS

This manual uses the following terms unless otherwise specified.

Term	Description
Board	Generic term for PC interface boards supported by MELSEC data link library
Utility	Generic term for utility of PC interface board supported by MELSEC data link library
CC-Link Ver.2 board	Generic term for Q80BD-J61BT11N and Q81BD-J61BT11 CC-Link system master/local interface board
MELSECNET/H board	Generic term for Q80BD-J71LP21-25, Q81BD-J71LP21-25, Q80BD-J71LP21S-25, Q80BD-J71LP21G, Q80BD-J71LP21GE, and Q80BD-J71BR11 MELSECNET/H interface board
CC-Link IE Controller Network board	Abbreviation for Q80BD-J71GP21-SX, Q80BD-J71GP21S-SX, Q81BD-J71GP21-SX, and Q81BD-J71GP21S-SX CC-Link IE Controller Network interface board
CC-Link IE Field Network board	Abbreviation for Q81BD-J71GF11-T2 CC-Link IE Field Network interface board
GX Developer	Generic product name for SW8D5C-GPPW-E, SW8D5C-GPPW-EA, SW8D5C-GPPW-EV, and SW8D5C-GPPW-EVA
GX Works2	Generic product name for SWnDNC-GXW2-E and SWnDNC-GXW2-EA (n: version)
MX Component	Generic product name for SWnD5C-ACT-E and SWnD5C-ACT-EA (n: version)

CHAPTER 1 OVERVIEW

This chapter explains overview of the MELSEC data link library.

1.1 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.

1.2 Supported Boards

The following table shows the boards supported by the MELSEC data link library.

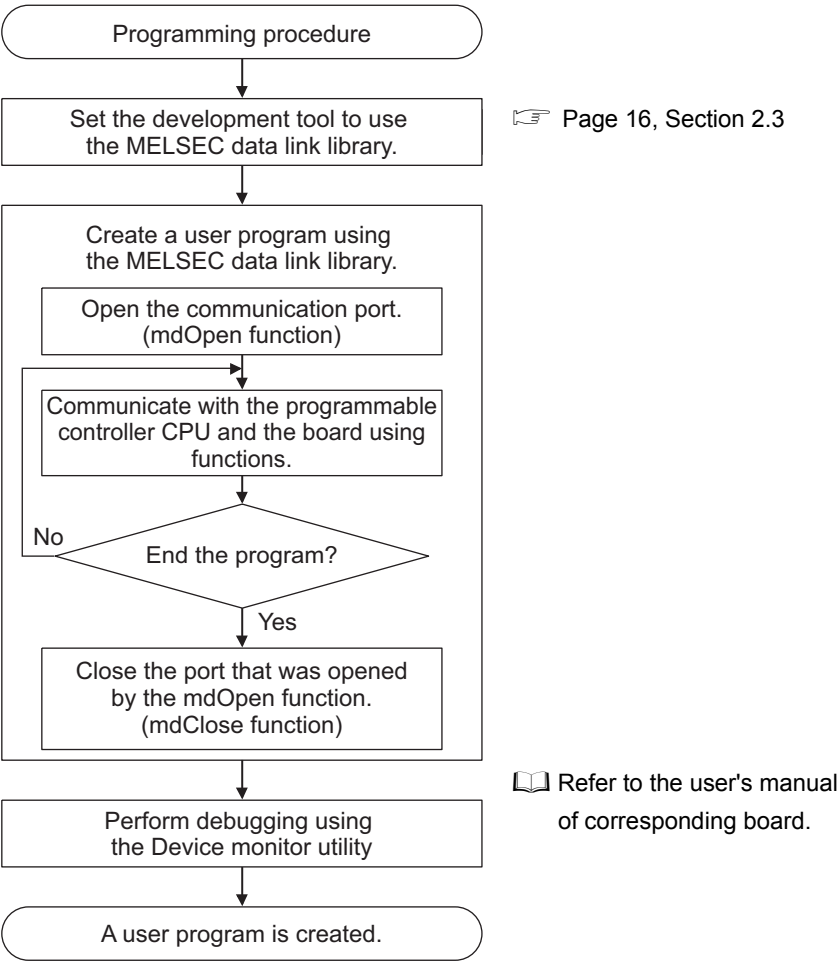
Board	Model Name
CC-Link Ver.2 board	Q80BD-J61BT11N, Q81BD-J61BT11
MELSECNET/H board	Q80BD-J71LP21-25, Q81BD-J71LP21-25, Q80BD-J71LP21S-25, Q80BD-J71LP21G, Q80BD-J71LP21GE, Q80BD-J71BR11
CC-Link IE Controller Network board	Q80BD-J71GP21-SX, Q80BD-J71GP21S-SX, Q81BD-J71GP21-SX, Q81BD-J71GP21S-SX
CC-Link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

CHAPTER 2 PROGRAMMING

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

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.



2.2 Precautions when Using MELSEC Data Link Library

2.2.1 Precautions when programming

(1) Multi-thread communications

The MELSEC data link library cannot be accessed from multiple threads within the same process.
Access the MELSEC data link library with a single thread.

(2) 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.

(3) 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.

(4) Number of stations for other station accesses

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.

(5) Forcible termination of user program

When the user program in which the MELSEC data link library operation is currently running is forcibly terminated, an error may occur to the MELSEC data link library in another user program.
This forcible termination may affect other Mitsubishi software packages (such as MX Component, GX Works2).

(6) Execution speed

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.

(7) Static type variables

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

(8) 64-bit version user program

(a) Accessing CPU modules other than QCPU (Q mode) modules

64-bit version user program cannot access CPU modules other than QCPU (Q mode) modules. Use 32-bit version user program to access CPU modules other than QCPU (Q mode) modules.

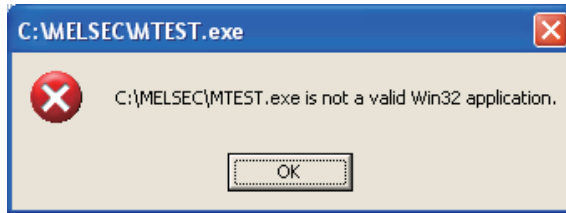
(b) Creating 64-bit version user program

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®.

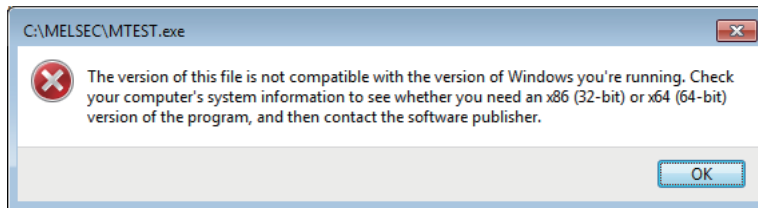
(c) Restriction when creating 64-bit version user programs

64-bit version user programs can be created on a 32-bit version operating system, however, the following dialog box appears and cannot be executed.

<When using Windows®XP (32-bit version)>



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



2.2.2 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.

(1) MELSECNET/H

(a) Accessing 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.

(b) 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).

(2) CC-Link IE Controller Network

(a) Accessing 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.

(b) 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).

(3) CC-Link IE Field Network

(a) Accessing 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.

(b) 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).

2.3 Settings for Using Functions

This section explains how to set the development tool to use the MELSEC data link library functions.
For programming languages supported by boards, refer to the user's manual corresponds to the board.

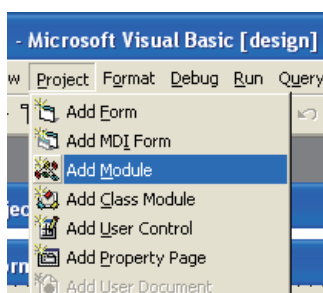
☞ Page 3 RELATED MANUALS

Point

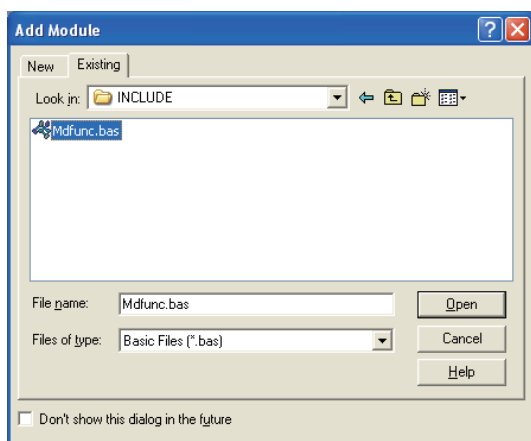
To create a 64-bit version user application, a project needs to be configured to set the 64-bit platform as a target platform.
For configuring a project and setting a target platform, refer to Help (How to: Configure Projects to Target Platforms) in Visual Studio®.

2.3.1 Using Visual Basic® 5.0, Visual Basic® 6.0

This section explains the setting operation when using Visual Basic® 5.0 or Visual Basic® 6.0.



- 1) Start Visual Basic® and select [Project] - [Add Module].

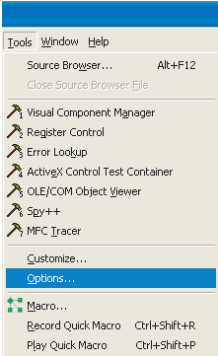


- 2) Select <<Existing>> and select "Mdfunc.bas".
"Mdfunc.bas" is stored in the following directory at installation.
<User-specified folder> - <COMMON> - <INCLUDE>

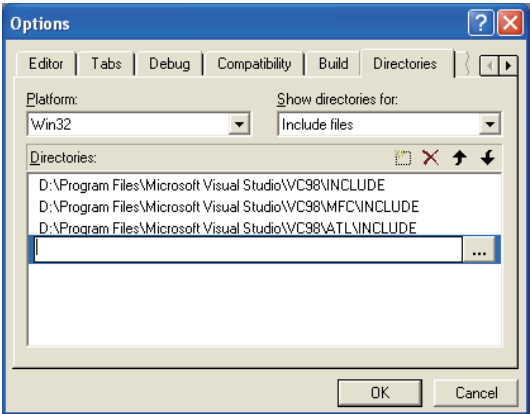
2.3.2 Using Visual C++[®] 5.0, Visual C++[®] 6.0

This section explains the setting operation when using Visual C++[®] 5.0 or Visual C++[®] 6.0.

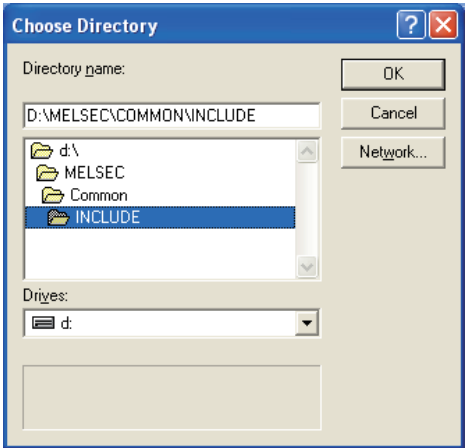
(1) Setting include files



1) Start Visual C++[®] and select [Tools] - [Options].



2) Select <<Directories>> and specify "Include files" for "Show directories for " .

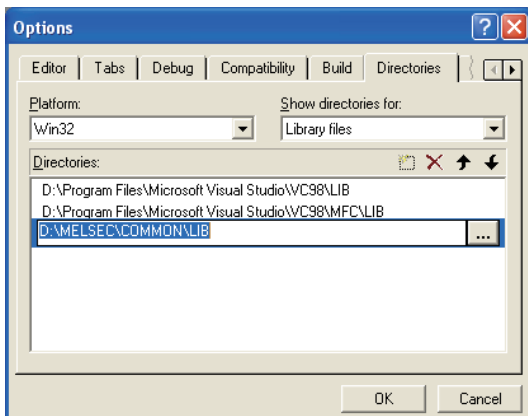
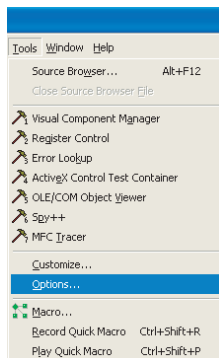


3) Double-click the item to be set and browse the folder containing include files.
"Mdfunc.h" is stored in the following directory at installation.
<User-specified folder> - <COMMON> - <INCLUDE>

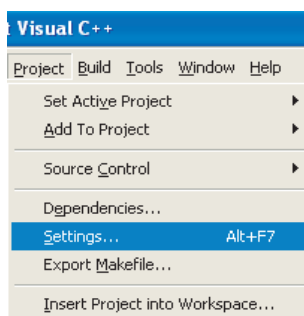
4) Add #include<Mdfunc.h> at the beginning of the program.

(2) Setting library files

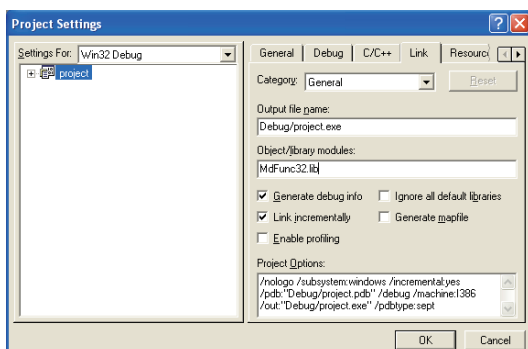
- 1) Start Visual C++® and select [Tools] - [Options].



- 2) Select <<Directories>> and specify "Library files" for "Show directories for". Browse the folder containing library files as in (1) on the previous page.
"Mdfunc32.lib" is stored in the following directory at installation.
<User-specified folder>-<COMMON>-<LIB>



- 3) Open the workspace to create a user application and select [Project] - [Settings].



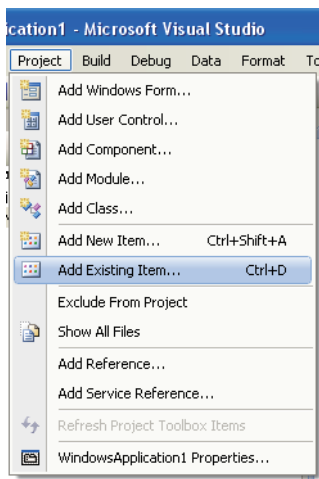
- 4) Select <<Link>> and specify "General" for the category, and enter "Mdfunc32.lib" in the "Object/library modules" field.

2.3.3 Using Visual Basic®.NET

This section explains the setting operation when using Visual Basic®.NET 2003, or when using Visual Basic® in Visual Studio® 2005, Visual Studio® 2008, or Visual Studio® 2010.

Remark

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



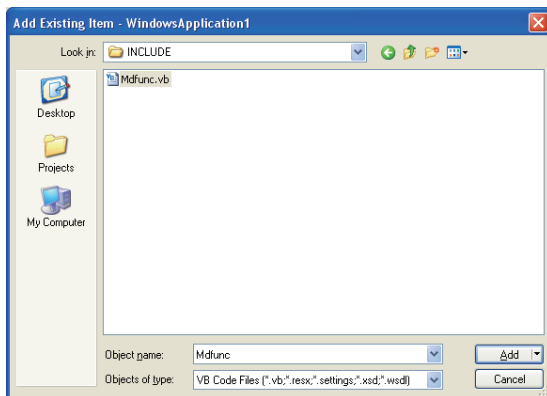
- 1) Start Visual Basic®.

For other than Visual Basic®.NET 2003:

Select [Project] - [Add Existing Item].

For Visual Basic®.NET 2003:

Select [File] - [Add Existing Item].



- 2) On the "Add Existing Item" screen, select "Mdfunc.vb".
"Mdfunc.vb" is stored in the following directory at installation.
<User-specified folder> - <COMMON> - <INCLUDE>

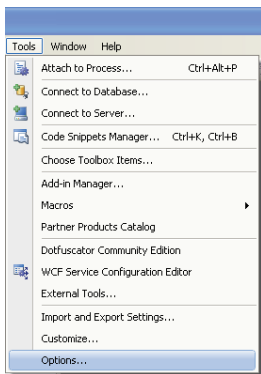
2.3.4 Using Visual C++[®].NET 2003, Visual C++[®]2005, Visual C++[®]2008

This section explains the setting operation when using Visual C++[®].NET 2003, or when using Visual C++[®] in Visual Studio[®] 2005, or Visual Studio[®] 2008.

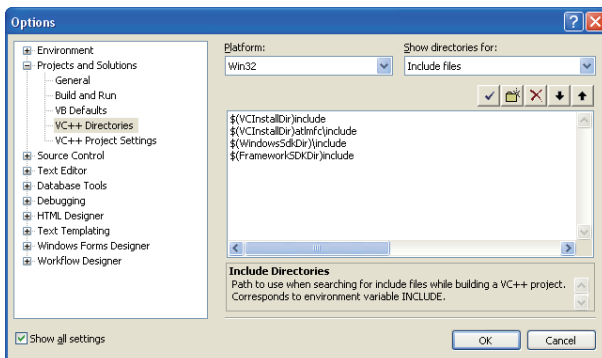
Remark

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++[®].

(1) Setting include files



- 1) Start Visual C++[®] and select [Tools] - [Options].



- 2) Select "VC++ Directories" in the folder area of the "Options" screen.

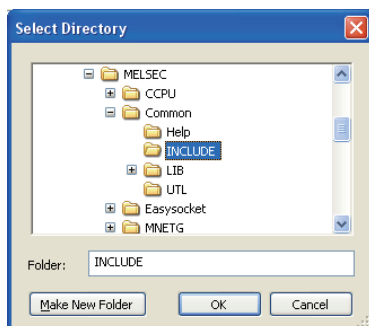
For other than Visual C++[®].NET 2003:

Select "Projects and Solutions" - "VC++ Directories".

For Visual C++[®].NET 2003:

Select "Projects" - "VC++ Directories".

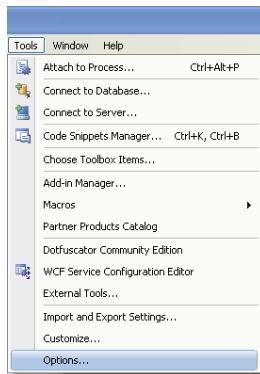
Specify "Include Files" for "Show directories for" and click .



- 3) Browse the folder containing include files.
"Mdfunc.h" is stored in the following directory at installation.
<User-specified folder> - <COMMON> - <INCLUDE>

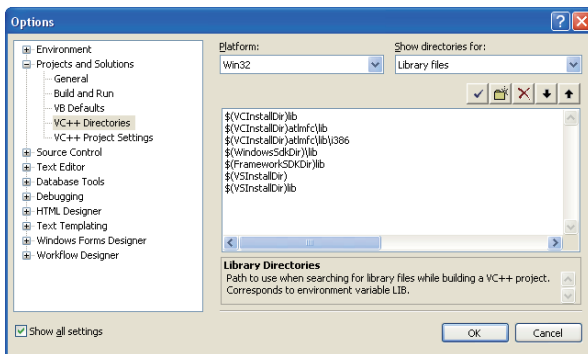
- 4) Add #include<Mdfunc.h> at the beginning of the program.

(2) Setting library files



- 1) Start Visual C++® and select [Tools] - [Options].

2



- 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".

Specify "Library files" for "Show directories for", and browse the folder containing library files as in (1) on the previous page.

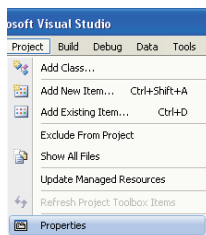
"Mdfunc32.lib" is stored in the following directory at installation.

For creating a 32-bit version user application

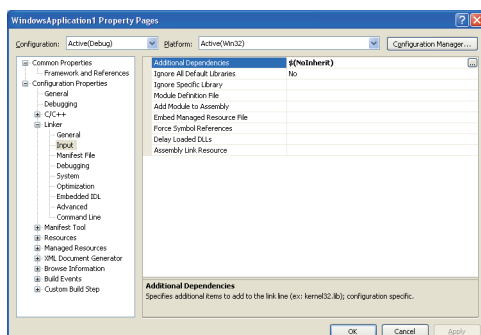
<User-specified folder> - <COMMON> - <LIB>

For creating a 64-bit version user application

<User-specified folder> - <COMMON> - <LIB> - <x64>



- 3) Open the project to create a user application and select [Project] - [project Properties].



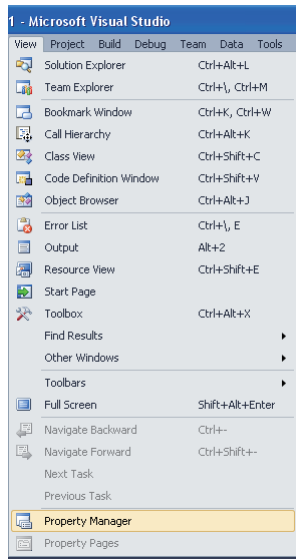
- 4) Select [Configuration Properties] - [Linker] - "Input" in the folder area of the "Property Pages" screen.

Enter "MdFunc32.lib" in the "Additional Dependencies" field.

2.3.5 Using Visual C++[®] 2010

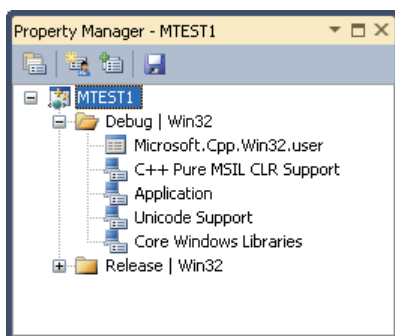
This section explains the setting operation when using Visual C++[®] in Visual Studio[®] 2010.

(1) Setting include files



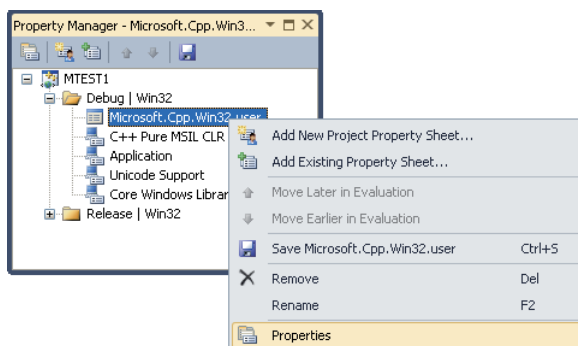
- 1) Open the project to create a user application and select [View] - [Property Manager]^{*1}.

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



- 2) Click the node (📁) of configuration and platform such as "Debug | Win32" in the property manager.

The node is expanded and the user property sheets such as Microsoft.Cpp.<platform>.user are displayed.

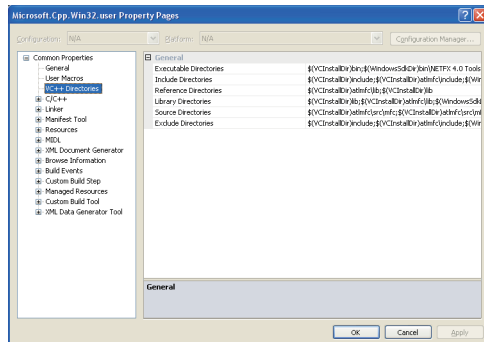


- 3) Right-click the user property sheet, and select [Properties] from the shortcut menu.

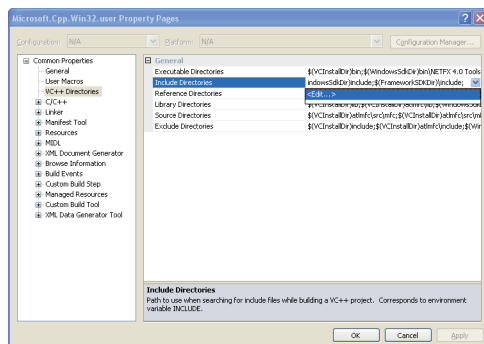
The "Property Pages" screen is displayed.


(To the next page)

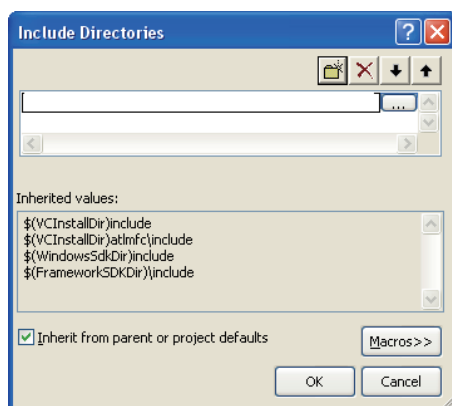
(From the previous page)





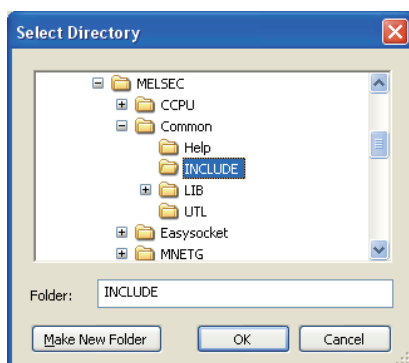
- 4) Select "Common Properties" - "VC++ Directories" on the "Property Pages" screen.



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



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

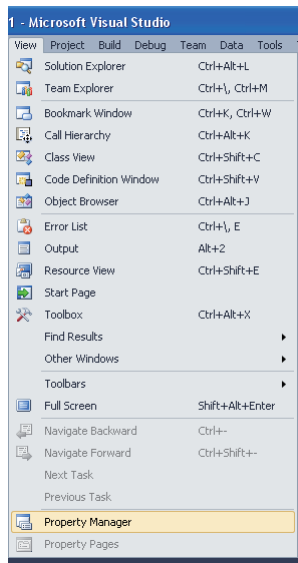


- 7) On the "Select Directory" screen, select the folder to which the include file is stored.

"Mdfunc.h" is stored in the following directory at installation.
<User-specified folder> - <COMMON> - <INCLUDE>

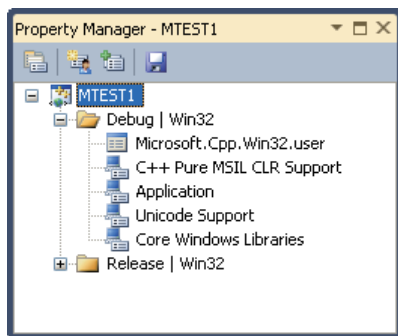
- 8) Add #include<Mdfunc.h> at the beginning of the program.

(2) Setting library files



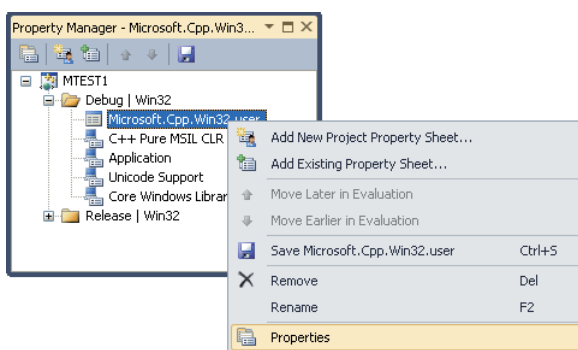
- 1) Open the project to create a user application and select [View] - [Property Manager]^{*1}.

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



- 2) Click the node (+) of configuration and platform such as "Debug | Win32" in the property manager.

The node is expanded and the user property sheets such as Microsoft.Cpp.<platform>.user are displayed.

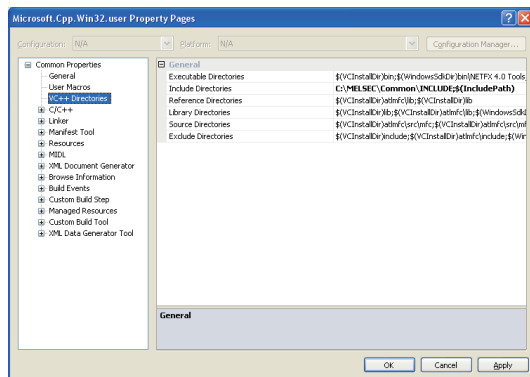


- 3) Right-click the user property sheet, and select [Properties] from the shortcut menu.

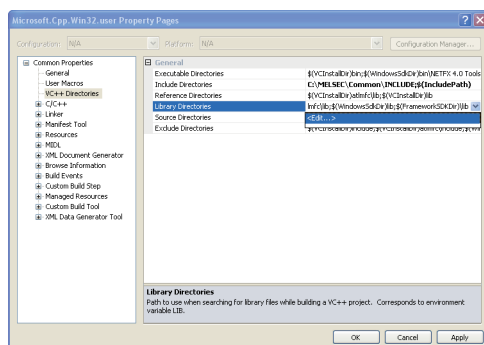
The "Property Pages" screen is displayed.


(To the next page)

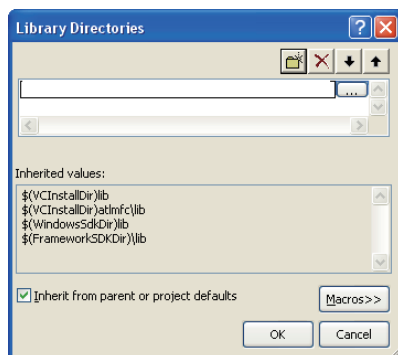
(From the previous page)





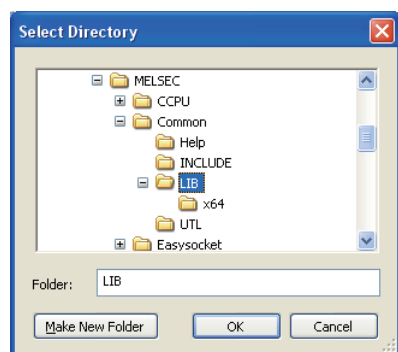
- 4) Select "Common Properties" - "VC++ Directories" on the "Property Pages" screen.



- 5) Select "Library Directories". Click  and select "<Edit...>".



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



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

"Mdfunc32.lib" is stored in the following directory at installation.

For creating a 32-bit version user application

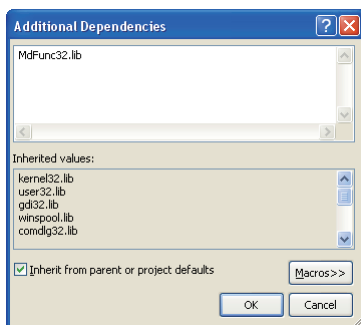
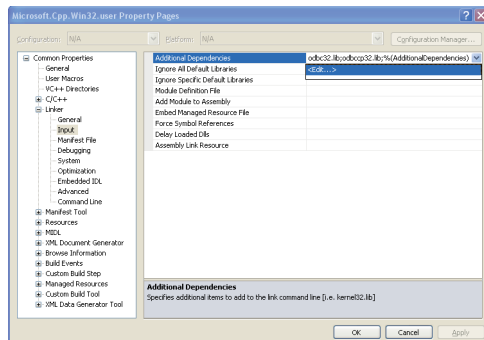
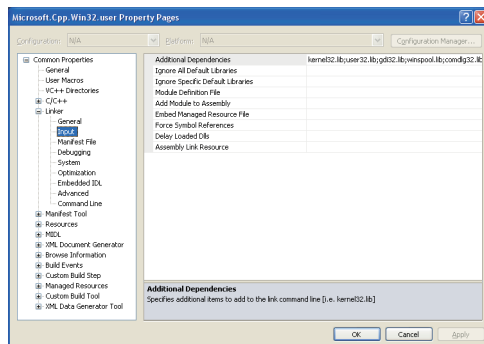
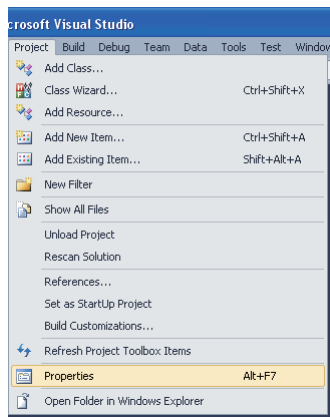
<User-specified folder> - <COMMON> - <LIB>

For creating a 64-bit version user application

<User-specified folder> - <COMMON> - <LIB> - <x64>


(To the next page)

(From the previous page)



- 8) Open the project to create a user application and select [Project] - [Properties].

- 9) In the folder area of the "Property Pages" screen, select "Configuration Properties" - "Linker" - "Input".

- 10) Select "Additional Dependencies". Click  and select "<Edit...>".

- 11) On the "Additional Dependencies" screen, enter "MdFunc32.lib".

CHAPTER 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 Accessible Devices

This section explains accessible devices for boards.

The following shows the description of access types used in the tables in this section.

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)

3.1.1 Accessible devices for CC-Link Ver.2 board

(1) When accessing own station

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

○: Accessible ×: Not accessible

Device		Access type	Accessibility
Remote input	RX	Batch	○
		Random	
Remote output	RY	Batch	○
		Random	
Link special relay	SB	Batch	○
		Random	
Link special register	SW	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	

(2) When accessing other station

The following table shows the accessible other stations.

Note that other stations cannot be accessed when the own station number is 64.

Item	Description	
A①	ACPU, QCPU(A mode)*1	A1NCPU
A②		A0J2HCPU, A1S(H)CPU, A1SJ(H)CPU, A2NCPU(-S1), A2S(H)CPU, A2C(J)
A③		A2ACPU(-S1)
A④		A2UCPU(-S1), A2USCPU(-S1), A2USHCPU-S1, Q02(H)CPU-A, Q06HCPU-A
A⑤		A3NCPU, A3ACPU
A⑥		A3UCPU
A⑦		A4UCPU
Q/QnA	QnACPU*1	Q2ACPU(-S1), Q3ACPU, Q4ACPU, Q4ARCPU, Q2ASCPU(-S1), Q2ASHCPU(-S1)
	QCPU (Q mode)	Q00JCPU, Q00CPU, Q01CPU, Q02(H)CPU, Q06HCPU, Q12HCPU, Q25HCPU, Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU, Q12PRHCPU, Q25PRHCPU, Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU
L	LCPU*1	L02CPU, L26CPU-BT
Intelligent	intelligent device station	A slave station such as AJ65BT-R2 in the CC-Link system that can perform the transient transmission.
Personal computer	—	

*1 : Can be accessed with a 32-bit version user application only.

The following table shows the accessible devices when accessing other stations.

○: Accessible ×: Not accessible

Device		Access type	Access target						
			A					L/Q/ QnA	Intelligent / Personal computer
			①	②	③④	⑤⑥	⑦		
Input	X	Batch	○	○	○	○	○	○	×
		Random							
Output	Y	Batch	○	○	○	○	○	○	×
		Random							
Latch relay	L	Batch	○	○	○	○	○	○	×
		Random							
Internal relay	M	Batch	○	○	○	○	○	○	×
		Random							
Special relay	SM	Batch	○	○	○	○	○	○	×
		Random							
Annunciator	F	Batch	○	○	○	○	○	○	×
		Random							
Timer contact	T	Batch	○	○	○	○	○	○	×
		Random							
Timer coil	T	Batch	○	○	○	○	○	○	×
		Random							

○: Accessible ×: Not accessible

Device		Access type	Access target						
			A					L/Q/ QnA	Intelligent / Personal computer
			①	②	③④	⑤⑥	⑦		
Counter contact	C	Batch	○	○	○	○	○	○	×
		Random	○	○	○	○	○	○	×
Counter coil	C	Batch	○	○	○	○	○	○	×
		Random	○	○	○	○	○	○	×
Timer current value	T	Batch	○	○	○	○	○	○	×
		Random	○	○	○	○	○	○	×
Counter current value	C	Batch	○	○	○	○	○	○	×
		Random	○	○	○	○	○	○	×
Data register	D	Batch	○	○	○	○	○	○	×
		Random	○	○	○	○	○	○	×
Special register	SD	Batch	○	○	○	○	○	○	×
		Random	○	○	○	○	○	○	×
Timer (setting value main)	T	Batch	○	○	○	○	○	×	×
		Random	×	×	×	×	×		
Timer (setting value sub1)	T	Batch	×	×	○*1	○	○	×	×
		Random	×	×	×	×	×		
Timer (setting value sub2)	T	Batch	×	×	×	×	○	×	×
		Random	×	×	×	×	×		
Timer (setting value sub3)	T	Batch	×	×	×	×	○	×	×
		Random	×	×	×	×	×		
Counter (setting value main)	C	Batch	○	○	○	○	○	×	×
		Random	×	×	×	×	×		
Counter (setting value sub1)	C	Batch	×	×	○*1	○	○	×	×
		Random	×	×	×	×	×		
Counter (setting value sub2)	C	Batch	×	×	×	×	○	×	×
		Random	×	×	×	×	×		
Counter (setting value sub3)	C	Batch	×	×	×	×	○	×	×
		Random	×	×	×	×	×		
Accumulator	A	Batch	○	○	○	○	○	×	×
		Random	○	○	○	○	○		
Index register	Z	Batch	○	○	○	○	○	○	×
		Random	○	○	○	○	○		
	V	Batch	○	○	○	○	○	×	×
		Random	○	○	○	○	○		
File register	R	Batch	×	○	○	○	○	○*2	×
		Random	×	○	○	○	○		
	ZR	Batch	×	×	×	×	×	○*2	×
		Random	×	×	×	×	×		
Extended file register	ER	Batch	×	○	○	○	○	○*2	×
		Random	×	○	○	○	○		
Link relay	B	Batch	○	○	○	○	○	○	×
		Random	○	○	○	○	○		

○: Accessible ×: Not accessible

Device		Access type	Access target						
			A					L/Q/ QnA	Intelligent / Personal computer
			①	②	③④	⑤⑥	⑦		
Link register	W	Batch	○	○	○	○	○	○	×
		Random							
Link special relay	SB	Batch	×	×	×	×	×	○	×
		Random							
Retentive timer contact	ST	Batch	×	×	×	×	×	○	×
		Random							
Retentive timer coil	ST	Batch	×	×	×	×	×	○	×
		Random							
Link special register	SW	Batch	×	×	×	×	×	○	×
		Random							
Edge relay	V	Batch	×	×	×	×	×	○	×
		Random							
Retentive timer current value	ST	Batch	×	×	×	×	×	○	×
		Random							
SEND function (with arrival acknowledgment, without arrival acknowledgment)	—	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							
Other stations* ³ (buffer memory, random access buffer, remote input, remote output, remote register)	—	Batch	○	○	○	○	○	○	○
		Random	×	×	×	×	×	×	×
Other stations* ³ (link special relay, link special register)	SB SW	Batch	○	○	○	○	○	○	○
		Random	×	×	×	×	×	×	×

*1 : Cannot be accessed with A2ACPU(-S1).

*2 : Cannot be accessed with Q00JCPU.

*3 : When accessing the buffer memory of CC-Link (the intelligent device station) module mounted on the CPUs.
Cannot be accessed when the logical station is specified.

3.1.2 Accessible devices for MELSECNET/H board

(1) When accessing own station

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

○: Accessible ×: Not accessible

Device		Access type	Accessibility
Link input	LX	Batch	○
		Random	
Link output	LY	Batch	○
		Random	
Link special relay	SB	Batch	○
		Random	
Link special register	SW	Batch	○
		Random	
Link relay	LB	Batch	○
		Random	
Link register	LW	Batch	○
		Random	
RECV function	—	Batch	○
		Random	×

(2) When accessing other station

The following table shows the accessible other stations.

Item	Description	
A①	ACPU, QCPU(A mode)*1	A1NCPU
A②		A0J2HCPU, A1S(H)CPU, A1SJ(H)CPU, A2NCPU(-S1), A2S(H)CPU, A2C(J)
A③		A2ACPU(-S1)
A④		A2UCPU(-S1), A2USCPU(-S1), A2USHCPU-S1, Q02(H)CPU-A, Q06HCPU-A
A⑤		A3NCPU, A3ACPU
A⑥		A3UCPU
A⑦		A4UCPU
Q/QnA	QnACPU	Q2ACPU(-S1), Q3ACPU, Q4ACPU, Q4ARCPU, Q2ASCPU(-S1), Q2ASHCPU(-S1)
	QCPU (Q mode)	Q00JCPU, Q00CPU, Q01CPU, Q02(H)CPU, Q06HCPU, Q12HCPU, Q25HCPU, Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU, Q12PRHCPU, Q25PRHCPU, Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU
Personal computer *1	—	

*1 : Can be accessed with a 32-bit version user application only.

The following table shows the accessible devices when accessing the other stations.

○: Accessible ×: Not accessible

Device		Access type	Access target						
			A					Q/ QnA	Personal computer
			①	②	③④	⑤⑥	⑦		
Input	X	Batch	○	○	○	○	○	○	×
		Random							
Output	Y	Batch	○	○	○	○	○	○	×
		Random							
Latch relay	L	Batch	○	○	○	○	○	○	×
		Random							
Internal relay	M	Batch	○	○	○	○	○	○	×
		Random							
Special relay	SM	Batch	○	○	○	○	○	○	×
		Random							
Annunciator	F	Batch	○	○	○	○	○	○	×
		Random							
Timer contact	T	Batch	○	○	○	○	○	○	×
		Random							
Timer coil	T	Batch	○	○	○	○	○	○	×
		Random							
Counter contact	C	Batch	○	○	○	○	○	○	×
		Random							
Counter coil	C	Batch	○	○	○	○	○	○	×
		Random							
Timer current value	T	Batch	○	○	○	○	○	○	×
		Random							
Counter current value	C	Batch	○	○	○	○	○	○	×
		Random							
Data register	D	Batch	○	○	○	○	○	○	×
		Random							
Special register	SD	Batch	○	○	○	○	○	○	×
		Random							
Timer (setting value main)	T	Batch	○	○	○	○	○	×	×
		Random	×	×	×	×	×		
Timer (setting value sub1)	T	Batch	×	×	○ ^{*1}	○	○	×	×
		Random			×	×	×		
Timer (setting value sub2)	T	Batch	×	×	×	×	○	×	×
		Random					×		
Timer (setting value sub3)	T	Batch	×	×	×	×	○	×	×
		Random					×		
Counter (setting value main)	C	Batch	○	○	○	○	○	×	×
		Random	×	×	×	×	×		
Counter (setting value sub1)	C	Batch	×	×	○ ^{*1}	○	○	×	×
		Random			×	×	×		
Counter (setting value sub2)	C	Batch	×	×	×	×	○	×	×
		Random					×		

○: Accessible ×: Not accessible

Device		Access type	Access target						
			A					Q/ QnA	Personal computer
			①	②	③④	⑤⑥	⑦		
Counter (setting value sub3)	C	Batch	×	×	×	×	○	×	×
		Random	×	×	×	×	×		
Accumulator	A	Batch	○	○	○	○	○	×	×
		Random	○	○	○	○	○		
Index register	Z	Batch	○	○	○	○	○	○	×
		Random	○	○	○	○	○		
	V	Batch	○	○	○	○	○	×	×
		Random	○	○	○	○	○		
File register	R	Batch	×	○	○	○	○	○*2	×
		Random	×	○	○	○	○		
	ZR	Batch	×	×	×	×	×	○*2	×
		Random	×	×	×	×	×		
Extended file register	ER	Batch	×	○	○	○	○	○*2	×
		Random	×	○	○	○	○		
Link relay	B	Batch	○	○	○	○	○	○	×
		Random	○	○	○	○	○		
Link register	W	Batch	○	○	○	○	○	○	×
		Random	○	○	○	○	○		
Link special relay	SB	Batch	×	×	×	×	×	○	×
		Random	×	×	×	×	×		
Retentive timer contact	ST	Batch	×	×	×	×	×	○	×
		Random	×	×	×	×	×		
Retentive timer coil	ST	Batch	×	×	×	×	×	○	×
		Random	×	×	×	×	×		
Link special register	SW	Batch	×	×	×	×	×	○	×
		Random	×	×	×	×	×		
Edge relay	V	Batch	×	×	×	×	×	○	×
		Random	×	×	×	×	×		
Retentive timer current value	ST	Batch	×	×	×	×	×	○	×
		Random	×	×	×	×	×		
SEND function (with arrival acknowledgment, without arrival acknowledgment)	—	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	×	×	×	×	×		

*1 : Cannot be accessed with A2ACPU(-S1).

*2 : Cannot be accessed with Q00JCPU.

3.1.3 Accessible devices for CC-Link IE Controller Network board

(1) When accessing own station

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

○: Accessible ×: Not accessible

Device		Access type	Accessibility
Link input	LX	Batch	○
		Random	
Link output	LY	Batch	○
		Random	
Link special relay	SB	Batch	○
		Random	
Link special register	SW	Batch	○
		Random	
Link relay	LB	Batch	○
		Random	
Link register	LW	Batch	○
		Random	
RECV function	—	Batch	○ ^{*1}
		Random	×

*1 : Supported by the 1.08J or later version of SW1DNC-MNETG-B.

(2) When accessing other station

The following table shows the accessible other stations.

Item	Description	
A①	ACPU, QCPU(A mode) ^{*1}	A1NCPU
A②		A0J2HCPU, A1S(H)CPU, A1SJ(H)CPU, A2NCPU(-S1), A2S(H)CPU, A2C(J)
A③		A2ACPU(-S1)
A④		A2UCPU(-S1), A2USCPU(-S1), A2USHCPU-S1, Q02(H)CPU-A, Q06HCPU-A
A⑤		A3NCPU, A3ACPU
A⑥		A3UCPU
A⑦		A4UCPU
Q/QnA	QnACPU ^{*1}	Q2ACPU(-S1), Q3ACPU, Q4ACPU, Q4ARCPU, Q2ASCPU(-S1), Q2ASHCPU(-S1)
	QCPU (Q mode)	Q00JCPU, Q00CPU, Q01CPU, Q02(H)CPU, Q06HCPU, Q12HCPU, Q25HCPU, Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU, Q12PRHCPU, Q25PRHCPU, Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU
Personal computer ^{*1}	—	

*1 : Can be accessed with a 32-bit version user application only.

The following table shows the accessible devices when accessing the other stations.

○: Accessible ×: Not accessible

Device		Access type	Access target						
			A					Q/ QnA	Personal computer
			①	②	③④	⑤⑥	⑦		
Input	X	Batch	○	○	○	○	○	○	×
		Random							
Output	Y	Batch	○	○	○	○	○	○	×
		Random							
Latch relay	L	Batch	○	○	○	○	○	○	×
		Random							
Internal relay	M	Batch	○	○	○	○	○	○	×
		Random							
Special relay	SM	Batch	○	○	○	○	○	○	×
		Random							
Annunciator	F	Batch	○	○	○	○	○	○	×
		Random							
Timer contact	T	Batch	○	○	○	○	○	○	×
		Random							
Timer coil	T	Batch	○	○	○	○	○	○	×
		Random							
Counter contact	C	Batch	○	○	○	○	○	○	×
		Random							
Counter coil	C	Batch	○	○	○	○	○	○	×
		Random							
Timer current value	T	Batch	○	○	○	○	○	○	×
		Random							
Counter current value	C	Batch	○	○	○	○	○	○	×
		Random							
Data register	D	Batch	○	○	○	○	○	○	×
		Random							
Special register	SD	Batch	○	○	○	○	○	○	×
		Random							
Timer (setting value main)	T	Batch	○	○	○	○	○	×	×
		Random	×	×	×	×	×		
Timer (setting value sub1)	T	Batch	×	×	○ ^{*1}	○	○	×	×
		Random			×	×	×		
Timer (setting value sub2)	T	Batch	×	×	×	×	○	×	×
		Random					×		
Timer (setting value sub3)	T	Batch	×	×	×	×	○	×	×
		Random					×		
Counter (setting value main)	C	Batch	○	○	○	○	○	×	×
		Random	×	×	×	×	×		
Counter (setting value sub1)	C	Batch	×	×	○ ^{*1}	○	○	×	×
		Random			×	×	×		
Counter (setting value sub2)	C	Batch	×	×	×	×	○	×	×
		Random					×		

○: Accessible ×: Not accessible

Device		Access type	Access target						
			A					Q/ QnA	Personal computer
			①	②	③④	⑤⑥	⑦		
Counter (setting value sub3)	C	Batch	×	×	×	×	○	×	×
		Random					×		
Accumulator	A	Batch	○	○	○	○	○	×	×
		Random							
Index register	Z	Batch	○	○	○	○	○	○	×
		Random							
	V	Batch	○	○	○	○	○	×	×
		Random							
File register	R	Batch	×	○	○	○	○	○ ^{*2}	×
		Random							
	ZR	Batch	×	×	×	×	×	○ ^{*2}	×
		Random							
Extended file register	ER	Batch	×	○	○	○	○	○ ^{*2}	×
		Random							
Link relay	B	Batch	○	○	○	○	○	○	×
		Random							
Link register	W	Batch	○	○	○	○	○	○	×
		Random							
Link special relay	SB	Batch	×	×	×	×	×	○	×
		Random							
Retentive timer contact	ST	Batch	×	×	×	×	×	○	×
		Random							
Retentive timer coil	ST	Batch	×	×	×	×	×	○	×
		Random							
Link special register	SW	Batch	×	×	×	×	×	○	×
		Random							
Edge relay	V	Batch	×	×	×	×	×	○	×
		Random							
Retentive timer current value	ST	Batch	×	×	×	×	×	○	×
		Random							
SEND function (with arrival acknowledgment, without arrival acknowledgment)	—	Batch						○ ^{*3}	○ ^{*3}
		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							

*1 : Cannot be accessed with A2ACPU(-S1).

*2 : Cannot be accessed with Q00JCPU.

*3 : Supported by the 1.08J or later version of SW1DNC-MNETG-B.

3.1.4 Accessible devices for CC-Link IE Field Network board

(1) When accessing own station

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

○: Accessible ×: Not accessible

Device		Access type	Accessibility
Remote input	RX	Batch	○
		Random	
Remote output	RY	Batch	○
		Random	
Link special relay	SB	Batch	○
		Random	
Link special register	SW	Batch	○
		Random	
Remote register (for transmission)	RWw	Batch	○
		Random	
Remote register (for reception)	RWr	Batch	○
		Random	
Buffer memory	—	Batch	○
		Random	
RECV function	—	Batch	×
		Random	

(2) When accessing other stations

The following table shows the accessible other stations.

Item	Description	
A④	ACPU, QCPU(A mode)*1	A2UCPU(-S1), A2USCPU(-S1), A2USHCPU-S1, Q02(H)CPU-A, Q06HCPU-A
A⑥		A3UCPU
A⑦		A4UCPU
Q/QnA	QnACPU	Q2ACPU(-S1), Q3ACPU, Q4ACPU, Q4ARCPU, Q2ASCPU(-S1), Q2ASHCPU(-S1)
	QCPU (Q mode)	Q00JCPU, Q00CPU, Q01CPU, Q02(H)CPU, Q06HCPU, Q12HCPU, Q25HCPU, Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU, Q12PRHCPU, Q25PRHCPU, Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU
Intelligent	Intelligent device station	NZ2GF-ETB(Ethernet adapter module), LJ72GF15-T2 (CC-Link IE Field Network head module)
Personal computer *1	—	

*1 : Can be accessed with a 32-bit version user application only.

The following table shows the accessible devices when accessing other stations.

○: Accessible ×: Not accessible

Device		Access type	Access target					
			A			Q/ QnA	Intelligent	Personal computer
			④	⑥	⑦			
Input	X	Batch	○	○	○	○	○	×
		Random						
Output	Y	Batch	○	○	○	○	○	×
		Random						
Latch relay	L	Batch	○	○	○	○	×	×
		Random						
Internal relay	M	Batch	○	○	○	○	×	×
		Random						
Special relay	SM	Batch	○	○	○	○	○	×
		Random						
Annunciator	F	Batch	○	○	○	○	×	×
		Random						
Timer contact	T	Batch	○	○	○	○	×	×
		Random						
Timer coil	T	Batch	○	○	○	○	×	×
		Random						
Counter contact	C	Batch	○	○	○	○	×	×
		Random						
Counter coil	C	Batch	○	○	○	○	×	×
		Random						
Timer current value	T	Batch	○	○	○	○	×	×
		Random						
Counter current value	C	Batch	○	○	○	○	×	×
		Random						
Data register	D	Batch	○	○	○	○	×	×
		Random						
Special register	SD	Batch	○	○	○	○	○	×
		Random						
Timer (setting value main)	T	Batch	○	○	○	×	×	×
		Random	×	×	×			
Timer (setting value sub1)	T	Batch	○ ^{*1}	○	○	×	×	×
		Random	×	×	×			
Timer (setting value sub2)	T	Batch	×	×	○	×	×	×
		Random			×			
Timer (setting value sub3)	T	Batch	×	×	○	×	×	×
		Random			×			
Counter (setting value main)	C	Batch	○	○	○	×	×	×
		Random	×	×	×			
Counter (setting value sub1)	C	Batch	○ ^{*1}	○	○	×	×	×
		Random	×	×	×			
Counter (setting value sub2)	C	Batch	×	×	○	×	×	×
		Random			×			

○: Accessible ×: Not accessible

Device		Access type	Access target					
			A			Q/ QnA	Intelligent	Personal computer
			④	⑥	⑦			
Counter (setting value sub3)	C	Batch	×	×	○	×	×	×
		Random	×	×	×			
Accumulator	A	Batch	○	○	○	×	×	×
		Random	○	○	○			
Index register	Z	Batch	○	○	○	○	×	×
		Random	○	○	○			
	V	Batch	○	○	○	×	×	×
		Random	○	○	○			
File register	R	Batch	○	○	○	○*2	×	×
		Random	○	○	○			
	ZR	Batch	×	×	×	○*2	×	×
		Random	×	×	×			
Extended file register	ER	Batch	○	○	○	○*2	×	×
		Random	○	○	○			
Link relay	B	Batch	○	○	○	○	×	×
		Random	○	○	○			
Link register	W	Batch	○	○	○	○	○	×
		Random	○	○	○			
Link special relay	SB	Batch	×	×	×	○	○	×
		Random	×	×	×			
Retentive timer contact	ST	Batch	×	×	×	○	×	×
		Random	×	×	×			
Retentive timer coil	ST	Batch	×	×	×	○	×	×
		Random	×	×	×			
Link special register	SW	Batch	×	×	×	○	○	×
		Random	×	×	×			
Edge relay	V	Batch	×	×	×	○	×	×
		Random	×	×	×			
Retentive timer current value	ST	Batch	×	×	×	○	×	×
		Random	×	×	×			
SEND function (with arrival acknowledgment, without arrival acknowledgment)	—	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	×	×	×			

*1: Cannot be accessed with A2ACPU(-S1).

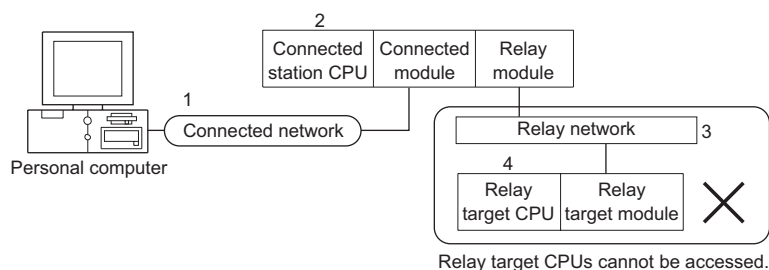
*2: Cannot be accessed with Q00JCPU.

3.2 Accessible Ranges

This section explains accessible ranges when communicating with boards.

3.2.1 Accessible ranges of CC-Link Ver.2 board

The following shows the accessible ranges via CC-Link Ver.2 board.



The own board and the connected station CPU can be accessed.

Relay target CPUs cannot be accessed.

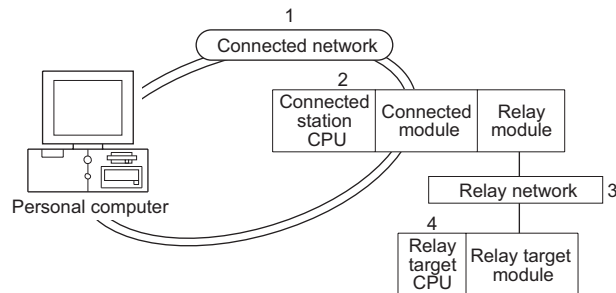
×: Not accessible

1. Connected network	2. Connected station CPU	3. Relay network	4. Relay target CPU
			LCPU, QCPU(Q mode), QnACPU, ACPU, QCPU(A mode), personal computer, intelligent device station
CC-Link	LCPU ^{*1}	CC-Link IE Field Network	×
	QCUP(Q mode)	CC-Link IE Controller Network	×
	QnACPU ^{*1}	MELSECNET/H	×
	ACPU ^{*1}	MELSECNET/10	×
	QCUP(A mode) ^{*1}	MELSECNET(II)	×
	Personal computer ^{*1}	Ethernet	×
	intelligent device station ^{*1}	Computer link	×
		CC-Link	×

^{*1}: Can be accessed with a 32-bit version user application only.

3.2.2 Accessible ranges of MELSECNET/H board

The following shows the accessible ranges via MELSECNET/H board.



The own board and the connected station CPU can be accessed.

The following table shows the accessibility of relay target CPU.

○: Accessible ×: Not accessible

1. Connected network	2. Connected station CPU	3. Relay network	4. Relay target CPU		
			QCPU (Q mode)	QnACPU, ACPU, QCPU(A mode)	Personal computer
MELSECNET/H	QCPU(Q mode) ^{*1*2}	CC-Link IE Field Network	×	×	×
		CC-Link IE Controller Network	○	×	○ ^{*3}
		MELSECNET/H	○	×	○ ^{*3}
		MELSECNET/10	○	○ ^{*3}	○ ^{*3}
		MELSECNET(II)	×	×	×
		Ethernet	×	×	×
		Computer link	×	×	×
		CC-Link	×	×	×
	Personal computer ^{*3}	CC-Link IE Field Network	×	×	×
		CC-Link IE Controller Network	×	×	×
		MELSECNET/H	×	×	×
		MELSECNET/10	×	×	×
		MELSECNET(II)	×	×	×
		Ethernet	×	×	×
		Computer link	×	×	×
		CC-Link	×	×	×

*1 : When the connected station CPU is Q00JCPU, Q00CPU, Q01CPU, Q00UJCPU, Q00UCPU, or Q01UCPU, the relay target CPU cannot be accessed.

*2 : In the redundant CPU system, the MELSECNET/H Extended mode cannot be used.

*3 : Can be accessed with a 32-bit version user application only.

○: Accessible ×: Not accessible

1. Connected network	2. Connected station CPU	3. Relay network	4. Relay target CPU		
			QCPU (Q mode)	QnACPU, ACPU, QCPU(A mode)	Personal computer
MELSECNET/10	QCPU(Q mode) ^{*1*2}	CC-Link IE Field Network	×	×	×
		CC-Link IE Controller Network	○	×	○ ^{*3}
		MELSECNET/H	○	×	○ ^{*3}
		MELSECNET/10	○	○ ^{*3}	○ ^{*3}
		MELSECNET(II)	×	×	×
		Ethernet	×	×	×
		Computer link	×	×	×
		CC-Link	×	×	×
	QnACPU ^{*3} , ACPU ^{*3} , QCPU(A mode) ^{*3}	CC-Link IE Field Network	×	×	×
		CC-Link IE Controller Network	×	×	×
		MELSECNET/H	×	×	×
		MELSECNET/10	○	○ ^{*3}	○ ^{*3}
		MELSECNET(II)	×	×	×
		Ethernet	×	×	×
		Computer link	×	×	×
		CC-Link	×	×	×
	Personal computer ^{*3}	CC-Link IE Field Network	×	×	×
		CC-Link IE Controller Network	×	×	×
		MELSECNET/H	×	×	×
		MELSECNET/10	×	×	×
		MELSECNET(II)	×	×	×
		Ethernet	×	×	×
		Computer link	×	×	×
		CC-Link	×	×	×

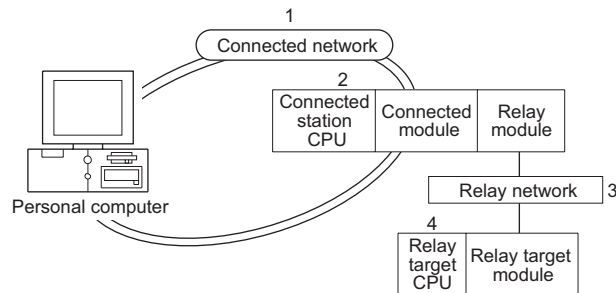
*1 : When the connected station CPU is Q00JCPU, Q00CPU, Q01CPU, Q00UJCPU, Q00UCPU, or Q01UCPU, the relay target CPU cannot be accessed.

*2 : In the redundant CPU system, the MELSECNET/H Extended mode cannot be used.

*3 : Can be accessed with a 32-bit version user application only.

3.2.3 Accessible ranges of CC-Link IE Controller Network board

The following shows the accessible ranges via CC-Link IE Controller Network board.



The own board and the connected station CPU can be accessed.

The following table shows the accessibility of relay target CPU.

○: Accessible ×: Not accessible

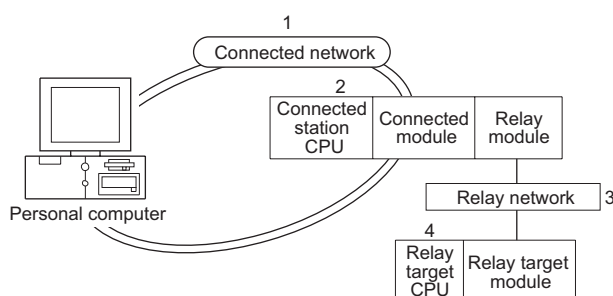
1. Connected network	2. Connected station CPU	3. Relay network	4. Relay target CPU		
			QCPU (Q mode)	ACPU QnACPU QCPU(A mode)	Personal computer
CC-Link IE Controller Network	QCPU(Q mode)*1	CC-Link IE Field Network	×	×	×
		CC-Link IE Controller Network	○	×	○*2
		MELSECNET/H	○	×	○*2
		MELSECNET/10	○	○*2	○*2
		MELSECNET(II)	×	×	×
		Ethernet	×	×	×
		Computer link	×	×	×
		CC-Link	×	×	×
	Personal computer*2	CC-Link IE Field Network	×	×	×
		CC-Link IE Controller Network	×	×	×
		MELSECNET/H	×	×	×
		MELSECNET/10	×	×	×
		MELSECNET(II)	×	×	×
		Ethernet	×	×	×
		Computer link	×	×	×
		CC-Link	×	×	×

*1 : When the connected station CPU is Q00JCPU, Q00CPU, Q01CPU, Q00UJCPU, Q00UCPU, or Q01UCPU, the relay target CPU cannot be accessed.

*2 : Can be accessed with a 32-bit version user application only.

3.2.4 Accessible ranges of CC-Link IE Field Network board

The following shows the accessible ranges via CC-Link IE Field Network board.



The own board and the connected station CPU can be accessed.

The following table shows the accessibility of relay target CPU.

○: Accessible ×: Not accessible

1. Connected network	2. Connected station CPU	3. Relay network	4. Relay target CPU			
			QCPU (Q mode)	ACPU QnACPU QCPU (A mode)	L series intelligent device station	Personal computer
CC-Link IE Field Network	QCPU(Q mode)	CC-Link IE Field Network	○	×	○ ^{*1}	○ ^{*1}
		CC-Link IE Controller Network	○	×	×	○ ^{*1}
		MELSECNET/H	○	×	×	○ ^{*1}
		MELSECNET/10	○	○ ^{*1}	×	○ ^{*1}
		MELSECNET(II)	×	×	×	×
		Ethernet	×	×	×	×
		Computer link	×	×	×	×
		CC-Link	×	×	×	×
	Personal computer ^{*1}	CC-Link IE Field Network	×	×	×	×
		CC-Link IE Controller Network	×	×	×	×
		MELSECNET/H	×	×	×	×
		MELSECNET/10	×	×	×	×
		MELSECNET(II)	×	×	×	×
		Ethernet	×	×	×	×
		Computer link	×	×	×	×
		CC-Link	×	×	×	×

^{*1} : Can be accessed with 32-bit version user application only.

CHAPTER 4 FUNCTIONS

This chapter explains the MELSEC data link library functions.

4.1 List of Functions

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

(1) List of the functions

Function name	Description	Remarks	Reference
mdOpen	Opens a communication line.	—	Page 55, Section 4.3.1
mdClose	Closes a communication line.	—	Page 56, Section 4.3.2
mdSendEx	Batch writes devices.	(Extended function ^{*3})	Page 57, Section 4.3.3
	Sends data. (SEND function) ^{*1*2}		
mdReceiveEx	Batch reads devices.	(Extended function ^{*3})	Page 61, Section 4.3.4
	Receives data. (RECV function) ^{*1*2}		
mdRandWEx	Writes devices randomly.	(Extended function ^{*3})	Page 65, Section 4.3.5
mdRandREx	Reads devices randomly.	(Extended function ^{*3})	Page 68, Section 4.3.6
mdDevSetEx	Sets a bit device.	(Extended function ^{*3})	Page 71, Section 4.3.7
mdDevRstEx	Resets a bit device.	(Extended function ^{*3})	Page 73, Section 4.3.8
mdTypeRead	Reads the type of programmable controller CPU.	—	Page 75, Section 4.3.9
mdControl	Remote operation of programmable controller CPU. (RUN/STOP/PAUSE).	—	Page 78, Section 4.3.10
mdWaitBdEvent	Waits for an event occurrence.	—	Page 79, Section 4.3.11
mdBdRst	Resets the board.	—	Page 82, Section 4.3.12
mdBdModSet	Sets the mode of the board.	—	Page 83, Section 4.3.13
mdBdModRead	Reads the mode of the board.	—	Page 85, Section 4.3.14
mdBdLedRead	Reads the LED information of the board.	—	Page 87, Section 4.3.15
mdBdSwRead	Reads the switch status of the board.	—	Page 93, Section 4.3.16
mdBdVerRead	Reads the version information of the board.	—	Page 95, Section 4.3.17
mdInit	Initializes programmable controller information table	—	Page 98, Section 4.3.18

*1 : Applicable to CC-Link IE Controller Network boards and CC-Link IE Field Network boards.

*2 : Supported by the 1.08J or later version of CC-Link IE Controller Network board.

*3 : 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) List of conventional compatible functions

Conventional compatible functions are functions used for programs created conventionally.

Function name	Description	Remarks	Reference
mdSend	Batch writes devices.	—	Page 99, Section 4.3.19
	Sends data. (SEND function) ^{*1*2}	—	
mdReceive	Batch reads devices.	—	Page 103, Section 4.3.20
	Receives data. (RECV function) ^{*1*2}	—	
mdRandW	Writes devices randomly.	—	Page 107, Section 4.3.21
mdRandR	Reads devices randomly.	—	Page 110, Section 4.3.22
mdDevSet	Sets a bit device.	—	Page 113, Section 4.3.23
mdDevRst	Resets a bit device.	—	Page 114, Section 4.3.24

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

*2 : Supported by the 1.08J or later version of CC-Link IE Controller Network board.



When using the conventional compatible functions, the accessible device numbers are from 0 to 32767.

4.2 Common Specifications of Functions

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


4.2.1 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)	

4.2.2 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  Page 50, Section 4.2.3.

(1) 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

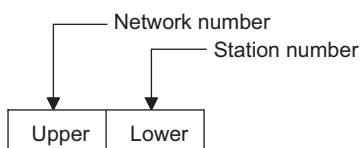
(2) MELSECNET/H board

Specification		Station number	
		Upper byte	Lower byte
Own station		255(FFH)	
Other station	Station number	Network number ^{*2} 1(01H) to 239(EFH)	1(01H) to 120(78H) ^{*3}
	Group number 1 to 32 ^{*1} ^{*4}		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 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} : Station numbers from 65(41H) to 120(78H) can be specified when using CC-Link IE Controller Network.

^{*4} : For MELSECNET/10 mode, only group numbers from 1 to 9 (129(81H) to 137(89H)) can be specified.

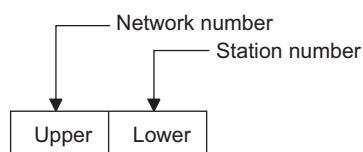
(3) CC-Link IE Controller Network board

Specification		Station number	
		Upper byte	Lower byte
Own station		255(FFH)	
Other station	Station number	Network number* ² 1(01H) to 239(EFH)	1(01H) to 120(78H)
	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>

**(4) CC-Link IE Field Network board**

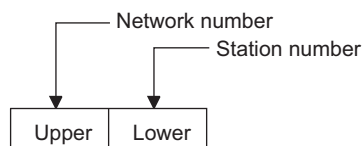
Specification		Station number	
		Upper byte	Lower byte
Own station		255(FFH)	
Other station	Station number	Network number.* ³ 1(01H) to 239(EFH)	0(00H) to 120(78H)
	Group Number 1 to 32* ¹ * ²		129(81H) to 160(A0H)
	All stations* ¹		240(F0H)
The logical station number set with the utility		0(00H) to 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 or CC-Link IE Controller Network.

*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>



4.2.3 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  Page 48, Section 4.2.2

(1) For CC-Link Ver.2 board

Specification	Network numbers	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

(2) For MELSECNET/H board

Specification		Network numbers	Station number
Own station		0(00H)	255(FFH)
Other station	Station number	1(01H) to 239(EFH)	1(01H) to 120(78H)* ¹
The logical station number set with the utility		0(00H)	0(00H) to 239(EFH)

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

(3) For CC-Link IE Controller Network board

Specification		Network numbers	Station number
Own station		0(00H)	255(FFH)
Other station	Station number	1(01H) to 239(EFH)	1(01H) to 120(78H)
	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.

(4) For CC-Link IE Field Network board

Specification		Network numbers	Station number
Own station		0(00H)	255(FFH)
Other station	Station number	1(01H) to 239(EFH)	0(00H) to 120(78H)
	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 or CC-Link IE Controller Network.

4.2.4 Specifying device types

Device types specified in the MELSEC data link library can be either a code number or a device name.

(1) Common device types

(a) Own station

Device		Device type		
		Code specification		Device name specification
		Decimal	Hexadecimal	
Link input	LX	1	0001H	DevX
Remote input	RX			
Link output	LY	2	0002H	DevY
Remote output	RY			
Special relay	SM	5	0005H	DevSM
Link special relay	SB			
Special register	SD	14	000EH	DevSD
Link special register	Sw			
Link relay	LB	23	0017H	DevB
Link register	LW	24	0018H	DevW
Remote register	RW			
Remote register (for transmission)	RWw	36	0024H	DevWw
Remote register (for reception)	RWr	37	0025H	DevWr
Buffer memory	—	50	0032H	DevSPB
RECV function	—	101	0065H	DevMAIL

(b) Other station

Device		Device type		
		Code specification		Device name specification
		Decimal	Hexadecimal	
Input	X	1	0001H	DevX
Output	Y	2	0002H	DevY
Latch relay	L	3	0003H	DevL
Internal relay	M	4	0004H	DevM
Special relay	SM	5	0005H	DevSM
Annunciator	F	6	0006H	DevF
Timer contact	T	7	0007H	DevTT
Timer coil	T	8	0008H	DevTC
Counter contact	C	9	0009H	DevCT
Counter coil	C	10	000AH	DevCC
Timer current value	T	11	000BH	DevTN
Counter current value	C	12	000CH	DevCN
Data register	D	13	000DH	DevD
Special register	SD	14	000EH	DevSD
Timer setting value main	T	15	000FH	DevTM
Timer setting value sub1	T	16	0010H	DevTS
Timer setting value sub2	T	16002	3E82H	DevTS2
Timer setting value sub3	T	16003	3E83H	DevTS3

Device		Device type		
		Code specification		Device name specification
		Decimal	Hexadecimal	
Counter setting value main	C	17	0011H	DevCM
Counter setting value sub1	C	18	0012H	DevCS
Counter setting value sub2	C	18002	4652H	DevCS2
Counter setting value sub3	C	18003	4653H	DevCS3
Accumulator	A	19	0013H	DevA
Index register	Z	20	0014H	DevZ
	V	21	0015H	DevV
File register *1	R	22	0016H	DevR
	ZR	220	00DC _H	DevZR
Extended file register *1 *2	ER	22000 to 22256	55F0 _H to 56F0 _H	DevER0 to 256
Link relay	B	23	0017H	DevB
Link register	W	24	0018H	DevW
Link special relay	SB	25	0019H	DevQSB
Retentive timer contact	ST	26	001AH	DevSTT
Retentive timer coil	ST	27	001BH	DevSTC
Link special register	SW	28	001CH	DevQSW
Edge relay	V	30	001EH	DevQV
Retentive timer current value	ST	35	0023H	DevSTN
SEND function with arrival acknowledgment	—	101	0065H	DevMAIL
SEND function without arrival acknowledgment	—	102	0066H	DevMAILNC
Link direct device link input*3	Jn\X	1001 to 1255	03E9 _H to 04E7 _H	DevLX1 to 255
Link direct device link output*3	Jn\Y	2001 to 2255	07D1 _H to 08CF _H	DevLY1 to 255
Link direct device link relay*3	Jn\B	23001 to 23255	59D9 _H to 5AD7 _H	DevLB1 to 255
Link direct device link register*3	Jn\W	24001 to 24255	5DC1 to 5EBF _H	DevLW1 to 255
Link direct device link special relay*3	Jn\SB	25001 to 25255	61A9 _H to 62A7 _H	DevLSB1 to 255
Link direct device link special register*3	Jn\SW	28001 to 28255	6D61 _H to 6E5F _H	DevLSW1 to 255
Intelligent function module device*4	Un\G	29000 to 29255	7148 _H to 7247 _H	DevSPG0 to 255

*1 : With the random read function (mdRandR, mdRandREx), execute normally even specifying devices that do not exist.(The read data is error.)

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

(2) Dedicated device types for CC-Link Ver.2 board

(a) Own station


The following is a device for accessing devices on a CC-Link Ver.2 board (own station).

For accessing other devices, refer to  Page 51, Section 4.2.4 (1)

Device	Device type		
	Code specification		Device name specification
	Decimal	Hexadecimal	
Random access buffer	33	0021H	DevMRB

(b) Other station

The following are devices for accessing devices on another station's programmable controller CC-Link module via a CC-Link Ver.2 board.

For accessing other devices, refer to  Page 51, Section 4.2.4 (1)

Device	Device type		
	Code specification		Device name specification
	Decimal	Hexadecimal	
Buffer memory*1	-32768	8000H	DevRBM
Random access buffer*1	-32736	8020H	DevRAB
Remote input*1	-32735	8021H	DevRX
Remote output*1	-32734	8022H	DevRY
Remote register*1	-32732	8024H	DevRW
Link special relay*1	-32669	8063H	DevSB
Link special register*1	-32668	8064H	DevSW

*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).

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.

①

☆

Function

This function opens a communication line by specifying a communication line channel number.

②

Format

Visual C++®

ret = mdOpen(chan,mode,path);

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 Basic®.NET

ret = mdOpen(chan,mode,path)

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

ret% = mdOpen(chan%,mode%,path%)

Integer	ret	.Return value	OUT
Integer	chan	.Channel number of communication line	IN
Integer	mode	.Dummy	IN
Long	path	.Opened line path pointer	OUT

③

Argument		Explanation
chan	Channel number of communication line	Specify the channel number of communication line. (☞ Page 46, Section 4.2.1)
mode	Dummy	Specify -1.
path	Opened line path pointer	Return the opened line path.

④

○

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, they need to be opened for each channel number.

⑤

?

Return value

Success : Return the value 0.
Error : Return the value other than 0. Refer to the list of error codes. (☞ Page 123, CHAPTER 6)

⑥

➡

Related function

mdClose()

54

① Indicates an description of the function.
② Indicates the description format in each programming language.

Item	Description
Visual C++®	A format for Microsoft® Visual C++®
Visual Basic®.NET	A format for the following Visual Basic® <ul style="list-style-type: none">• Visual Basic®.NET 2003• Visual Studio® 2005 Visual Basic®• Visual Studio® 2008 Visual Basic®• Visual Studio® 2010 Visual Basic®
Visual Basic®5.0, Visual Basic®6.0	A format for the following Visual Basic® <ul style="list-style-type: none">• Visual Basic® 5.0• Visual Basic® 6.0

③ Describes arguments of the function.
④ Describes features of the function and details of arguments.
⑤ Indicates the return values.
⑥ Indicates related functions for programming.

54

4.3.1 mdOpen (opening communication lines)



Function

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



Format

Visual C++®

```
ret = mdOpen(chan,mode,path);
```

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 Basic®.NET

```
ret = mdOpen(chan,mode,path)
```

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

```
ret% = mdOpen(chan%,mode%,path&)
```

Integer	ret	;Return value	OUT
Integer	chan	;Channel number of communication line	IN
Integer	mode	;Dummy	IN
Long	path	;Opened line path pointer	OUT

Argument		Description
chan	Channel number of communication line	Specify the channel number of communication line. (Page 47, Section 4.2.1)
mode	Dummy	Specify -1.
path	Opened line path pointer	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 : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (Page 126, CHAPTER 6)



Related function

mdClose()

4.3.2 mdClose (closing communication lines)



Function

Close a communication line by specifying a communication line path.



Format

Visual C++®

```
ret = mdClose(path);
```

short	ret;	//Return value	OUT
long	path;	//Path of channel	IN

Visual Basic®.NET

```
ret = mdClose(path)
```

Short	ret	;Return value	OUT
Integer	path	;Path of channel	IN

Visual Basic®5.0, Visual Basic®6.0

```
ret% = mdClose(path&)
```

Integer	ret	;Return value	OUT
Long	path	;Path of channel	IN

Argument		Description
path	Path of channel	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 : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (👉 Page 126, CHAPTER 6)



Related function

mdOpen()

4.3.3 mdSendEx (batch writing extended devices / SEND function)

(1) Batch writing extended devices



Function

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++®

```
ret = mdSendEx(path,netno,stno,devtyp,devno,size,data);
```

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 Basic®.NET




```
ret = mdSendEx(path,netno,stno,devtyp,devno,size,data(0))
```

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

```
ret% = mdSendEx(path&,netno%,stno%,devtyp%,devno%,size%,data%(0))
```

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

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Network number	Specify the network number of target station. ( Page 50, Section 4.2.3)
stno	Station number	Specify the station number of target station. ( Page 50, Section 4.2.3)
devtyp	Device type	Specify the type of device to which the data is written. ( Page 51, Section 4.2.4)
devno	Start device number	Specify the start device number of device to which the data is written. <ul style="list-style-type: none"> Specify the start device number with a multiple of 8 for the access to a bit device. 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	Written byte size	Specify the byte size to be written in even number. <ul style="list-style-type: none"> When the specified byte size to be written exceeds the device range (-5: size error), the applicable size is returned to "size".
data	Written data (single-precision integer array)	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.

Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. ( Page 126, CHAPTER 6)

Related function

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

(2) SEND function



Function

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



Format

Visual C++®

```
ret = mdSendEx(path,netno,stno,devtyp,devno,size,data);
```

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 Basic®.NET



```
ret = mdSendEx(path,netno,stno,devtyp,devno,size,data(0))
```

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

```
ret% = mdSendEx(path&, netno%,stno%,devtyp%,devno%,size%,data%(0))
```

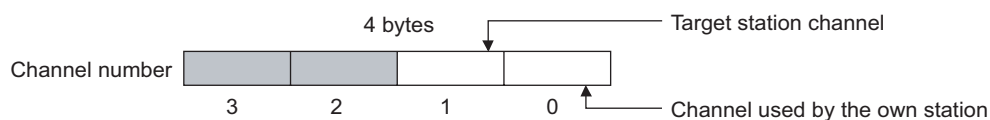
Long	ret	;Return value	OUT
Long	path	;Path of channel	IN
Long	netno	;Network number	IN
Long	stno	;Station number	IN
Long	devtyp	;Device type	IN
Long	devno	;Channel number	IN
Long	size	;Send byte size	IN/OUT
Any	data(n)	;Send data (single-precision integer array)	IN

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Network number	Specify the network number of target station. ( Page 50, Section 4.2.3)
stno	Station number	Specify the station number of target station. ( Page 50, Section 4.2.3) Logical station numbers cannot be specified.
devtyp	Device type	Specify whether to send data with arrival acknowledgment or without arrival acknowledgment. <ul style="list-style-type: none"> • With arrival acknowledgment*1 : Specify 101(65H) • Without arrival acknowledgment: Specify 102(66H)
devno	Channel number	Specify a channel used by the own station and a target station channel. For details of specifying the channels, refer to Explanation shown below.
size	Send byte size	Specify the byte size of send data in even number. <ul style="list-style-type: none"> • Specify the byte size within the range of 2 to 1920. • For MNET/10 mode of MELSECNET/H board, specify the byte size within the range of 2 to 960.
data	Send data (single-precision integer array)	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 Q/QnA.
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.	
	<ul style="list-style-type: none"> • MELSECNET/H board, CC-Link IE Controller Network board • CC-Link IE Field Network board 	1 to 8 1 to 2
channel used by the own station	Specify a channel used by the own station.	
	<ul style="list-style-type: none"> • MELSECNET/H board, CC-Link IE Controller Network board • CC-Link IE Field Network board 	1 to 8 1 to 2

Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. ( Page 126, CHAPTER 6)

Related function

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

4.3.4 mdReceiveEx (batch reading extended devices / RECV function)

(1) Batch reading extended devices



Function

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++®

```
ret = mdReceiveEx(path,netno,stno,devtyp,devno,size,data);
```

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 Basic®.NET




```
ret = mdReceiveEx(path,netno,stno,devtyp,devno,size,data(0))
```

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

```
ret% =mdReceiveEx(path&,netno%,stno%,devtyp%,devno%,size%,data%(0))
```

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


Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Network number	Specify the station number of target station. ( Page 50, Section 4.2.3)
stno	Station number	Specify the station number of target station. ( Page 50, Section 4.2.3)
devtyp	Device type	Specify the type of device from which the data is read. ( Page 51, Section 4.2.4)
devno	Start device number	Specify the start device number of device range to be read. <ul style="list-style-type: none"> Specify the start device number with a multiple of 8 for the access to a bit device. 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	Read byte size	Specify the byte size to be read in even number. When the specified byte size to be read exceeds the device range (-5: size error), the applicable size is returned to "size".
data	Read data (single-precision integer array)	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.

Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. ( Page 126, CHAPTER 6)

Related function

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

(2) RECV function



Function

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



Format

Visual C++®

```
ret = mdReceiveEx(path,netno,stno,devtyp,devno,size,data);
```

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 (single-precision integer array)	OUT

Visual Basic®.NET

```
ret = mdReceiveEx(path,netno,stno,devtyp,devno,size,data(0))
```

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(single-precision integer array)	OUT

Visual Basic®5.0, Visual Basic®6.0

```
ret% = mdReceiveEx(path&, netno%,stno%,devtyp%,devno%,size%,data%(0))
```

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(single-precision integer array)	OUT

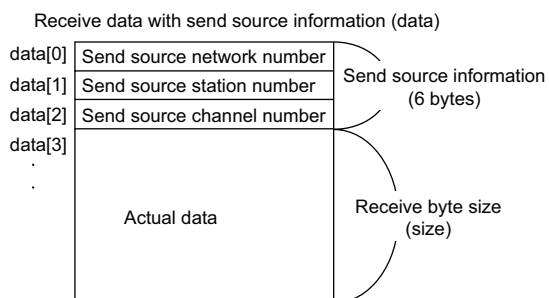
Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Network number	Specify 0(0H).
stno	Station number	Specify the own station (255(FFH)).
devtyp	Device type	Specify the RECV function(101(65H)).
devno	Channel number	Specify the own station channel number on which the received messages are stored. <ul style="list-style-type: none"> MELSECNET/H board, CC-Link IE Controller Network board 1 to 8 CC-Link IE Field Network board 1 to 2
size	Receive byte size	Specify the byte size of received data in even number. <ul style="list-style-type: none"> Specify the byte size within the range of 2 to 1920. For MNET/10 mode of MELSECNET/H board, specify the byte size within the range of 2 to 960. The size of received actual data is received.
data	Receive data with send source information (single-precision integer array)	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 Q/QnA.
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.



- 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.

Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (📖 Page 126, CHAPTER 6)

Related function

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

4.3.5 mdRandWEx (writing extended devices randomly)



Function

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



Format

Visual C++®

```
ret = mdRandWEx(path,netno,stno,dev,buf,bufsize);
```

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 Basic®.NET

```
ret = mdRandWEx(path,netno,stno,dev(0),buf(0),bufsize)
```

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

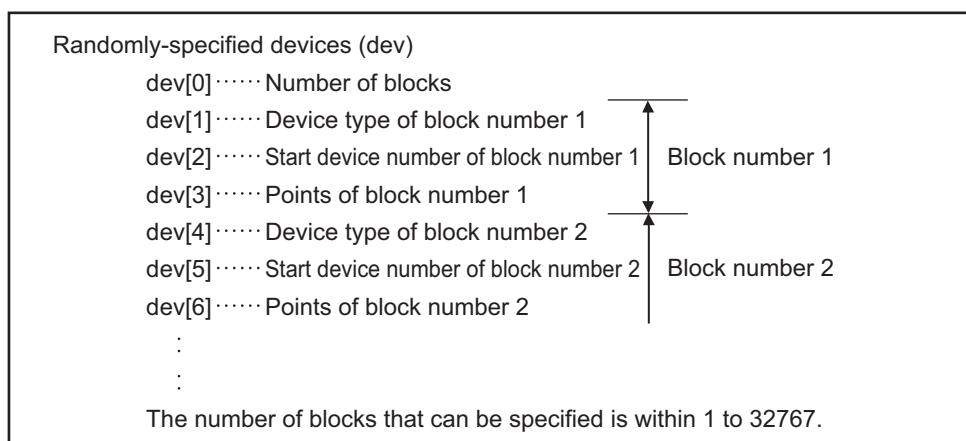
```
ret% = mdRandWEx(path&,netno%,stno%,dev%(0),buf%(0),bufsize%)
```

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

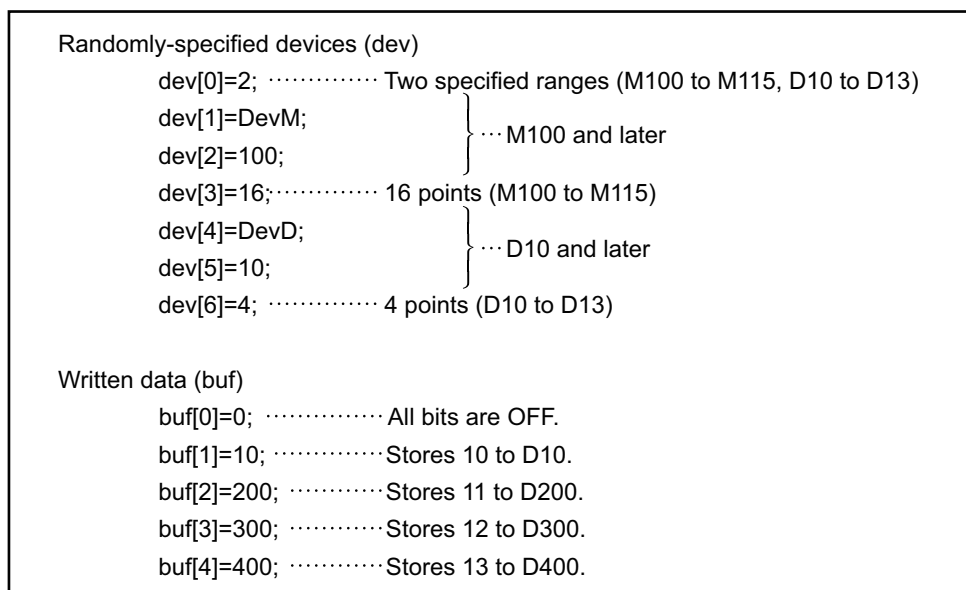
Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Network number	Specify the network number of target station. (Page 50, Section 4.2.3)
stno	Station number	Specify the station number of target station. (Page 50, Section 4.2.3)
dev	Randomly-specified device	Specify the number of blocks, the device type, the start device number, and points of device to be written. For details, refer to Explanation shown on the next page.
buf	Written data (single-precision integer array)	Specify the data to be written to the array variable of buf[0] or later.
bufsize	Dummy	—

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.
- If an error occurs when writing devices randomly to B or W of the own station in MELSECNET(II) or MELSECNET/10, blocks with errors may exist among blocks in which the write function is normally completed.
- The following explains how to specify the randomly-specified devices (dev).



- The following shows an example when writing data to multiple devices.
Example) When all bits of M100 to M115 are set to OFF, and 10, 200, 300, and 400 are written to D10, D11, D12, and D13 respectively.





Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (👉 Page 126, CHAPTER 6)



Related function

mdOpen(), mdClose(), mdRandREx()

4.3.6 mdRandREx (reading extended devices randomly)



Function

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



Format

Visual C++®

```
ret = mdRandREx(path,netno,stno,dev,buf,bufsize);
```

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 Basic®.NET



```
ret = mdRandREx(path,netno,stno,dev(0),buf(0),bufsize)
```

Integer	ret	;Return value	OUT
Integer	path	;Path of channel	IN
Integer	netno	;Network number	IN
Integer	dev(n)	;Randomly-specified device	IN
Integer	dev(n)	;Randomly selected 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

```
ret% = mdRandREx(path&,netno%,stno%,dev%(0),buf%(0),bufsize%)
```

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

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Network number	Specify the network number of target station. ( Page 50, Section 4.2.3)
stno	Station number	Specify the station number of target station. ( Page 50, Section 4.2.3)
dev	Randomly-specified device	Specify the number of blocks, the device type, the start device number, and points of device to be read. For details, refer to Explanation shown on the next page.
buf	Read data (single-precision integer array)	Store the read data to the array variable of buf[0] or later. For details of checking the read data, refer to Explanation shown on the next page.
bufsize	Number of bytes of read data	Specify the number of bytes of read data. For details of values to be specified, refer to Explanation shown on the next page.

Explanation

- The number of transient transmissions performed inside of the function changes with the specified randomly selected 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 following explains how to specify the randomly-specified devices (dev).

Randomly-specified devices (dev)

dev[0] ·····	Number of blocks	
dev[1] ·····	Device type of block number 1	
dev[2] ·····	Start device number of block number 1	Block number 1
dev[3] ·····	Points of block number 1	
dev[4] ·····	Device type of block number 2	
dev[5] ·····	Start device number of block number 2	Block number 2
dev[6] ·····	Points of block number 2	
:		
:		

The number of blocks that can be specified is within 1 to 32767.

- The following shows an example when reading multiple devices.

Example) When reading the values of M100 to M115, D10 to D13, M0 to M13, and the current value of T10.

M100 to M115 All bits are OFF.

D10 to D13 10 for D10, 200 for D11, 300 for D12, and 400 for D13

M0 to M13 All bits are ON.

Current value of T10 10 (1 second) for the current value of T10

Note that the above values are presumed values in the current status of the target devices.

Randomly-specified devices (dev)

dev[0]=4;	Four specified ranges
	(M100 to M115, D10 to D13, M0 to M13, T10)
dev[1]=DevM;	{ ... M100 and later
dev[2]=100;	
dev[3]=16;	16 points (M100 to M115)
dev[4]=DevD;	{ ... D10 and later
dev[5]=10;	
dev[6]=4;	4 points (D10 to D13)
dev[7]=DevM;	{ ... M0 and later
dev[8]=0;	
dev[9]=14;	14 points (M0 to M13)
dev[10]=DevTN;	{ ... Current and later values of T10 timer
dev[11]=10;	
dev[12]=1;	1 point (T10)

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) x 2 = 14 bytes

Enter "14" for this example.

Read data (buf)

buf[0]=0; All bits of M100 to M115 are OFF
(16 points of bit information can be stored.)

buf[1]=10; Current value of D10

buf[2]=200; Current value of D11

buf[3]=300; Current value of D12

buf[4]=400; Current value of D13

buf[5]=0x3fff; All bits of M0 to M13 are ON

buf[6]=10; Current value of T10 is 10 (1 second).



Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (📖 Page 126, CHAPTER 6)



Related function

mdOpen(), mdClose(), mdRandWEx()

4.3.7 mdDevSetEx (setting extended bit devices)



Function

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



Format

Visual C++®

```
ret = mdDevSetEx(path,netno,stno,devtyp,devno);
```

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 Basic®.NET

```
ret = mdDevSetEx(path,netno,stno,devtyp,devno)
```

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

```
ret% = mdDevSetEx(path&,netno%,stno%,devtyp%,devno%)
```

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


Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Network number	Specify the network number of target station. (Page 50, Section 4.2.3)
stno	Station number	Specify the station number of target station. (Page 50, Section 4.2.3)
devtyp	Device type	Specify the type of device to be set (ON). (Page 51, Section 4.2.4)
devno	Specified device number	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 : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. ( Page 126, CHAPTER 6)

Related function

mdOpen(), mdClose(), mdDevRstEx()

4.3.8 mdDevRstEx (resetting extended bit devices)



Function

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



Format

Visual C++®

```
ret = mdDevRstEx(path,netno,stno,devtyp,devno);
```

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 Basic®.NET

```
ret = mdDevRstEx(path,netno,stno,devtyp,devno)
```

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

```
ret% = mdDevRstEx(path&,netno%,stno%,devtyp%,devno%)
```

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


Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
netno	Network number	Specify the network number of target station. (Page 50, Section 4.2.3)
stno	Station number	Specify the station number of target station. (Page 50, Section 4.2.3)
devtyp	Device type	Specify the type of device to be reset (OFF). (Page 51, Section 4.2.4)
devno	Specified device number	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 : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. ( Page 126, CHAPTER 6)

Related function

mdOpen(), mdClose(), mdDevSetEx()

4.3.9 mdTypeRead (reading model names of CPU)



Function

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



Format

Visual C++®

```
ret = mdTypeRead(path,stno,buf);
```

short	ret;	//Return value	OUT
long	path;	//Path of channel	IN
short	stno;	//Station number	IN
short	*buf;	//Model name code	OUT

Visual Basic®.NET


```
ret = mdTypeRead(path,stno,buf)
```

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

```
ret% = mdTypeRead(path&,stno%,buf%)
```

Integer	ret	;Return value	OUT
Long	path	;Path of channel	IN
Integer	stno	;Station number	IN
Integer	buf	;Model name code	OUT

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Station number	Specify the station number of target station. ( Page 48, Section 4.2.2)
buf	Model name code	Return the model name code of CPU being read. For details of the model name code of CPU being read, refer to Explanation shown on the next page.

Explanation

The following table shows the model name codes and corresponding model names.

Model name code (Hexadecimal)	Model name
00A0H	A0J2CPU
00A1H	A1CPU, A1NCPU
00A2H	A2CPU, A2NCPU, A2SCPU
00A3H	A3CPU, A3NCPU, A1SHCPU, A1SJHCPU, A2SHCPU, A2SH1CPU
00A4H	A3HCPU, A3MCPU
0092H	A2ACPU
0093H	A2ACPU-S1
0094H	A3ACPU
0098H	A0J2HCPU, A1SCPU, A1SJCPU
009AH	A2CCPU, A2CJCPU
0082H	A2UCPU, A2USCPU
0083H	A2UCPU-S1, A2USCPU-S1
0084H	A3UCPU, A2USHCPU-S1
0085H	A4UCPU
0021H	Q2ACPU, Q2AHCPU, Q2ASCPU, Q2ASHCPU
0022H	Q2ACPU-S1, Q2AHCPU-S1, Q2ASCPU-S1, Q2ASHCPU-S1
0023H	Q3ACPU
0024H	Q4ACPU, Q4ARCPU
0090H	DOS/V personal computer
0041H	Q02(H)CPU
0042H	Q06(H)CPU
0043H	Q12(H)CPU
0044H	Q25(H)CPU
0049H	Q12PHCPU
004AH	Q25PHCPU
004BH	Q12PRHCPU
004CH	Q25PRHCPU
004DH	Q02PHCPU
004EH	Q06PHCPU
0141H	Q02CPU, 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

Model name code (Hexadecimal)	Model name
026AH	Q06UDHCPU
026BH	Q13UDHCPU
026CH	Q26UDHCPU
02E6H	Q10UDEHCPU
02E7H	Q20UDEHCPU
02E8H	Q03UDECPU
02E9H	Q04UDEHCPU
02EAH	Q06UDEHCPU
02EBH	Q13UDEHCPU
02ECH	Q26UDEHCPU
02EDH	Q50UDEHCPU
02EEH	Q100UDEHCPU
0541H	L02CPU
0548H	L26CPU-BT
0641H	LJ72GF15-T2
0642H	NZ2GF-ETB
2010H	Q172CPU, Q172CPUN(-T)
2012H	Q172HCPU
2011H	Q173CPU, Q173CPUN(-T)
2013H	Q173HCPU
2014H	Q172DCPU
2015H	Q173DCPU



Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (👉 Page 126, CHAPTER 6)



Related function

mdOpen(), mdClose()

4.3.10 mdControl (remote RUN/STOP/PAUSE)



Function

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



Format

Visual C++®

```
ret = mdControl(path,stno,buf);
```

short	ret;	//Return value	OUT
long	path;	//Path of channel	IN
short	stno;	//Station number	IN
short	buf;	//Command code	IN

Visual Basic®.NET

```
ret = mdControl(path,stno,buf)
```

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

```
ret% = mdControl(path&,stno%,buf%)
```

Integer	ret	;Return value	OUT
Long	path	;Path of channel	IN
Integer	stno	;Station number	IN
Integer	buf	;Command code	IN

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Station number	Specify the station number of target station. (Page 48, Section 4.2.2)
buf	Command code	Specify the command code (remote RUN/STOP/PAUSE) for remote operation of CPU. For details, refer to Explanation shown below.



Explanation

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

Command code	Description
0	Remote RUN
1	Remote STOP
2	Remote PAUSE



Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (Page 126, CHAPTER 6)



Related function

mdOpen(), mdClose()

4.3.11 mdWaitBdEvent (waiting for event occurrence)



Function

Wait an occurrence of event until the time out.



Format

Visual C++®

```
ret = mdWaitBdEvent(path, eventno, timeout, signaledno, details);
```

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

```
ret = mdWaitBdEvent(path, eventno(0), timeout, signaledno, details(0))
```

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

```
ret% = mdWaitBdEvent(path&,eventno%(0),timeout&,signaledno%,details%(0))
```

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

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
eventno	Waiting event number	Specify the event number of event to wait. (Use the event number set in the utility.)
timeout	Timeout value	Specify the time until time out to wait the event.
signaledno	Driven event number	Return the occurred event number.
details	Event detail information	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
0 to 2147483647(7FFFFFFFH)	Millisecond waiting

- Waiting event number (eventno) is specified as follows.

Waiting event (eventno)	
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.
Example) When waiting the event No. 1, No. 5, and No. 12 simultaneously.

Waiting event (eventno)	
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 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.

① 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)	
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

②Word device

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

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

③RECV function (CC-Link IE Field Network board)

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

Event detail information (details)	
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.



Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (Page 126, CHAPTER 6)



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.
- 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.
 - ①Bit device
 - This function ends normally.
 - "Logical add in the bit pattern of each driven event" is stored in "details[]".
 - ②Word device
 - This function returns "The driven event overlapped. -70(FFBAH)".
 - The device value when the condition of the event that occurred first is satisfied is stored in "details[0]".
- 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 126, CHAPTER 6)

4.3.12 mdBdRst (resetting board)



Function

Reset a board.



Format

Visual C++®

```
ret = mdBdRst(path);
```

short	ret;	//Return value	OUT
long	path;	//Path of channel	IN

Visual Basic®.NET

```
ret = mdBdRst(path)
```

Short	ret	;Return value	OUT
Integer	path	;Path of channel	IN

Visual Basic®5.0, Visual Basic®6.0

```
ret% = mdBdRst(path&)
```

Integer	ret	;Return value	OUT
Long	path	;Path of channel	IN

Argument		Description
path	Path of channel	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 board reset error (9922H) which notifies the board reset is returned to other applications accessing the reset board.



Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (👉 Page 126, CHAPTER 6)



Related function

mdOpen(), mdClose()

4.3.13 mdBdModSet (setting modes of board)



Function

Change the modes of a board temporarily.



Format

Visual C++®

```
ret = mdBdModSet(path,mode);
```

short	ret;	//Return value	OUT
long	path;	//Path of channel	IN
short	mode;	//Mode	IN

Visual Basic®.NET

```
ret = mdBdModSet(path,mode)
```

Short	ret	;Return value	OUT
Integer	path	;Path of channel	IN
Short	mode	;Mode	IN

Visual Basic®5.0, Visual Basic®6.0

```
ret% = mdBdModSet(path&,mode%)
```

Integer	ret	;Return value	OUT
Long	path	;Path of channel	IN
Integer	mode	;Mode	IN

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
mode	Mode	Specify the code corresponding to the mode to be executed. For details of modes and the corresponding codes, refer to Explanation shown on the next page.


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.

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

Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. ( Page 126, CHAPTER 6)

Related function

mdOpen(), mdClose(), mdBdModRead()

4.3.14 mdBdModRead (reading modes of board)



Function

Read the mode in which the board is currently operating.



Format

Visual C++®

```
ret = mdBdModRead(path,mode);
```

short	ret;	//Return value	OUT
long	path;	//Path of channel	IN
short	*mode;	//Mode	OUT

Visual Basic®.NET

```
ret = mdBdModRead(path,mode)
```

Short	ret	;Return value	OUT
Integer	path	;Path of channel	IN
Short	mode	;Mode	OUT

Visual Basic®5.0, Visual Basic®6.0

```
ret% = mdBdModRead(path&,mode%)
```

Integer	ret	;Return value	OUT
Long	path	;Path of channel	IN
Integer	mode	;Mode	OUT

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
mode	Mode	Return the corresponding code to the mode being read. For details of codes being read, refer to Explanation shown on the next page.


Explanation

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

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

Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. ( Page 126, CHAPTER 6)

Related function

mdOpen(), mdClose(), mdBdModSet()

4.3.15 mdBdLedRead (read LED information of the board)



Function

Read the LED information of the board.



Format

Visual C++®

```
ret = mdBdLedRead(path,buf);
```

short	ret;	//Return value	OUT
long	path;	//Path of channel	IN
short	buf[];	//Read data	OUT

Visual Basic®.NET

```
ret = mdBdLedRead(path,buf(0))
```

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

```
ret% = mdBdLedRead(path&,buf%(0))
```

Integer	ret	;Return value	OUT
Long	path	;Path of channel	IN
Integer	buf(n)	;Read data	OUT

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
buf	Read data	Store the LED information being read. For details of stored LED information, refer to Explanation shown on the next page.

Explanation

The following table shows the details of the LED information.

CC-Link Ver.2 Board

Element	Bit	LED	Description	Note
buf[0]	b15-9	RESERVE	—	—
	b8	ERR	Display the detection status of communication error for all stations. 1: Detected 0: Not detected	ON:1 OFF:0
	b7-1	RESERVE	—	—
	b0	RUN	Display the operating status of board. 1: Operating normally 0: Watchdog timer error	ON:1 OFF:0
buf[1]	b15-1	RESERVE	—	—
	b0	SW	Display the detection status of switch setting error. 1: Detected 0: Not detected	ON:1 OFF:0
buf[2]	b15-9	RESERVE	—	—
	b8	PRM	Display the detection status of parameter error. 1: Detected 0: Not detected	ON:1 OFF:0
	b7-1	RESERVE	—	—
	b0	M/S	Display the detection status of master station duplication error. 1: Detected 0: Not detected	ON:1 OFF:0
buf[3]	b15-9	RESERVE	—	—
	b8	LINE	Display the detection status of cable disconnection error. 1: Detected 0: Not detected	ON:1 OFF:0
	b7-1	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
buf[4]	b15-9	RESERVE	—	—
	b8	L.ERR	Display the detection status of communication error. 1: Detected 0: Not detected	ON:1 OFF:0
	b7-1	RESERVE	—	—
	b0	L.RUN	Display the data link execution status. 1: Data link in operation 0: No data link	ON:1 OFF:0
buf[5]	b15-9	RESERVE	—	—
	b8	RD	Display the data reception status. 1: Data are being received. 0: Data not received.	ON:1 OFF:0
	b7-1	RESERVE	—	—
	b0	SD	Display the data sending status. 1: Data are being transmitted. 0: Data not transmitted.	ON:1 OFF:0

MELSECNET/H Board

Element	Bit	LED	Description	Note
buf[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 are being received. 0: Data not received.	
	b9	R.LOOP RD	Display the data reception status on the reverse loop side. 1: Data are being received. 0: Data not received.	
	b8	F.LOOP SD	Display the data sending status on the forward loop side. 1: Data are being transmitted. 0: Data not transmitted.	
	b7	R.LOOP SD	Display the data sending status on the reverse loop side. 1: Data are being transmitted. 0: Data not transmitted.	
	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 watchdog timer error	

Element	Bit	LED	Description	Note
buf[1]	b15-10	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	

CC-Link IE Controller Network

Element	Bit	LED	Description	Note
buf[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 is being supplied. 0: External power supply is not supplied.	ON:1 OFF:0
	b7-14	RESERVE	—	—
	b6	RD	Display the data reception status. 1: Data are 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 detection status of parameter error. 1: Detected 0: Not detected	
	b3	ERR.	Display the error detection status. 1: Detected 0: Not detected	
	b2	SD	Display the data sending status. 1: Data are being transmitted. 0: Data not transmitted.	
	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 watchdog timer error	
buf[1]	B15-0	RESERVE	—	—


CC-Link IE Field Network

Element	Bit	LED	Description	Note
buf[0]	b15-12	RESERVE	—	—
	b11	LNK2	Display the link status of PORT2. 1: Linkup in progress 0: Linkdown in progress	ON:1 OFF:0
	b10	LNK1	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 are being received. 0: Data not received.	
	b5	SD	Display the network sending status. 1: Data are being transmitted. 0: Data not transmitted.	
	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.	
buf[1]	B15-0	RESERVE	—	—



Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. ( Page 126, CHAPTER 6)



Related function

mdOpen(), mdClose()

4.3.16 mdBdSwRead (reading switch status of the board)



Function

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



Format

Visual C++®

```
ret = mdBdSwRead(path,buf);
```

short	ret;	//Return value	OUT
long	path;	//Path of channel	IN
short	buf[];	//Read data	OUT

Visual Basic®.NET

```
ret = mdBdSwRead(path,buf(0))
```

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

```
ret% = mdBdSwRead(path&,buf%(0))
```

Integer	ret	;Return value	OUT
Long	path	;Path of channel	IN
Integer	buf(n)	;Read data	OUT

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
buf	Read data	Store the board switch status being read. For details of the stored switch status, refer to Explanation shown on the next page.


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

Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. ( Page 126, CHAPTER 6)

Related function

mdOpen(), mdClose()

4.3.17 mdBdVerRead (read version information of the board)



Function

Read the version information of the board.



Format

Visual C++®

```
ret = mdBdVerRead(path,buf);
```

short	ret;	//Return value	OUT
long	path;	//Path of channel	IN
short	buf[];	//Read data	OUT

Visual Basic®.NET

```
ret = mdBdVerRead(path,buf(0))
```

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

```
ret% = mdBdVerRead(path&,buf%(0))
```

Integer	ret	;Return value	OUT
Long	path	;Path of channel	IN
Integer	buf(n)	;Read data	OUT

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
buf	Read data	Store the version information of board being read. For details of the stored version information, refer to Explanation shown on the next page.

Explanation

The following table shows the details of the version information.

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



Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (👉 Page 126, CHAPTER 6)



Related function

mdOpen(), mdClose()

4.3.18 mdlnit (initializing programmable controller information table)



Function

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



Format

Visual C++®

```
ret = mdlnit(path);
```

short	ret;	//Return value	OUT
long	path;	//Path of channel	IN

Visual Basic®.NET

```
ret = mdlnit(path)
```

Short	ret	;Return value	OUT
Integer	path	;Path of channel	IN

Visual Basic®5.0, Visual Basic®6.0

```
ret% = mdlnit(path&)
```

Integer	ret	;Return value	OUT
Long	path	;Path of channel	IN

Argument		Description
path	Path of channel	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 mdlnit function, programmable controller device information is reacquired at the initial access to the programmable controller CPU.

Therefore, after the execution of the mdlnit function, a longer function execution time is required at the initial access.



Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (Page 126, CHAPTER 6)



Related function

mdOpen(), mdClose()

4.3.19 mdSend (batch writing devices / SEND function)

(1) Batch writing devices



Function

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++®

```
ret = mdSend(path,stno,devtyp,devno,size,data);
```

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



```
ret = mdSend(path,stno,devtyp,devno,size,data(0))
```

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

```
ret% = mdSend(path&,stno%,devtyp%,devno%,size%,data%(0))
```

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

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Station number	Specify the station number of target station. ( Page 48, Section 4.2.2)
devtyp	Device type	Specify the type of device to which the data is written. ( Page 51, Section 4.2.4)
devno	Start device number	Specify the start device number of device to which the data is written. <ul style="list-style-type: none"> Specify the start device number with a multiple of 8 for the access to a bit device. 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	Written byte size	Specify the byte size to be written in even number. <ul style="list-style-type: none"> When the specified byte size to be written exceeds the device range (-5: size error), the applicable size is returned to "size".
data	Written data (single-precision integer array)	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.

Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. ( Page 126, CHAPTER 6)

Related function

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

(2) SEND function



Function

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



Format

Visual C++®

```
ret = mdSend(path,stno,devtyp,devno,size,data);
```

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


```
ret = mdSend(path,stno,devtyp,devno,size,data(0))
```

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

```
ret% = mdSend(path&,stno%,devtyp%,devno%,size%,data%(0))
```

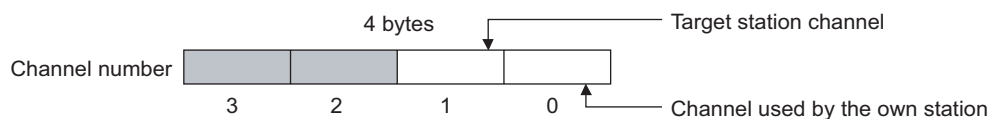
Integer	ret	;Return value	OUT
Long	path	;Path of channel	IN
Integer	stno	;Station number	IN
Integer	devtyp	;Device type	IN
Integer	devno	;Channel number	IN
Integer	size	;Send byte size	IN/OUT
Any	data(n)	;Send data (single-precision integer array)	IN

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Station number	Specify the station number of target station. ( Page 48, Section 4.2.2) Logical station numbers cannot be specified.
devtyp	Device type	Specify whether to send data with arrival acknowledgment or without arrival acknowledgment. <ul style="list-style-type: none"> • With arrival acknowledgment *1 : Specify 101(65H) • Without arrival acknowledgment : Specify 102(66H)
devno	Channel number	Specify a channel used by the own station and a target station channel. For details of specifying the channels, refer to Explanation shown below.
size	Send byte size	Specify the byte size of send data in even number. <ul style="list-style-type: none"> • Specify the byte size within the range of 2 to 1920. • For MNET/10 mode of MELSECNET/H board, specify the byte size within the range of 2 to 960.
data	Send data (single-precision integer array)	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 acknowledgement.

Explanation


- This function supports the SEND instruction of the link dedicated instruction for Q/QnA.
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. MELSECNET/H board, CC-Link IE Controller Network board	1 to 8
	CC-Link IE Field Network board	1 to 2
Channel used by the own station	Specify a channel used by the own station. MELSECNET/H board, CC-Link IE Controller Network board	1 to 8
	CC-Link IE Field Network board	1 to 2

Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. ( Page 126, CHAPTER 6)

Related function

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

4.3.20 mdReceive (batch read devices / RECV function)

(1) Batch reading devices



Function

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++®

```
ret = mdReceive(path,stno,devtyp,devno,size,data);
```

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



```
ret = mdReceive(path,stno,devtyp,devno,size,data(0))
```

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

```
ret% = mdReceive(path&,stno%,devtyp%,devno%,size%,data%(0))
```

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


Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Station number	Specify the station number of target station. ( Page 48, Section 4.2.2)
devtyp	Device type	Specify the type of device from which the data is read. ( Page 51, Section 4.2.4)
devno	Start device number	Specify the start device number of device range to be read. <ul style="list-style-type: none"> Specify the start device number with a multiple of 8 for the access to a bit device. 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	Read byte size	Specify the byte size to be read in even number. When the specified byte size to be read exceeds the device range (-5: size error), the applicable size is returned to "size".
data	Read data (single-precision integer array)	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.

Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. ( Page 126, CHAPTER 6)

Related function

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

(2) RECV function



Function

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



Format

Visual C++®

```
ret = mdReceive(path,stno,devtyp,devno,size,data);
```

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 (single-precision integer array)	OUT

Visual Basic®.NET

```
ret = mdReceive(path,stno,devtyp,devno,size,data(0))
```

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 (single-precision integer array)	OUT

Visual Basic® 5.0, Visual Basic® 6.0

```
ret% = mdReceive(path&,stno%,devtyp%,devno%,size%,data%(0))
```

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 (single-precision integer array)	OUT

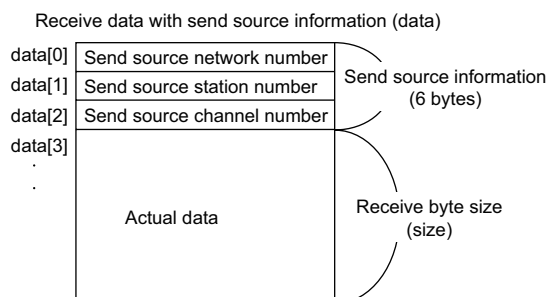
Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Station number	Specify the own station (255(FFH)).
devtyp	Device type	Specify the RECV function(101(65H)).
devno	Channel number	Specify the own station channel number on which the received messages are stored. <ul style="list-style-type: none"> MELSECNET/H board, CC-Link IE Controller Network board 1 to 8 CC-Link IE Field Network board 1 to 2
size	Receive byte size	Specify the byte size of received data in even number. <ul style="list-style-type: none"> Specify the byte size within the range of 2 to 1920. For MNET/10 mode of MELSECNET/H board, specify the byte size within the range of 2 to 960. The size of actual received data is received.
data	Receive data with send source information (single-precision integer array)	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 Q/QnA.
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.



- 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.

Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. ( Page 126, CHAPTER 6)

Related function

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

4.3.21 mdRandW (writing devices randomly)



Function

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



Format

Visual C++®

```
ret = mdRandW(path,stno,dev,buf,bufsize);
```

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

```
ret = mdRandW(path,stno,dev(0),buf(0),bufsize)
```

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

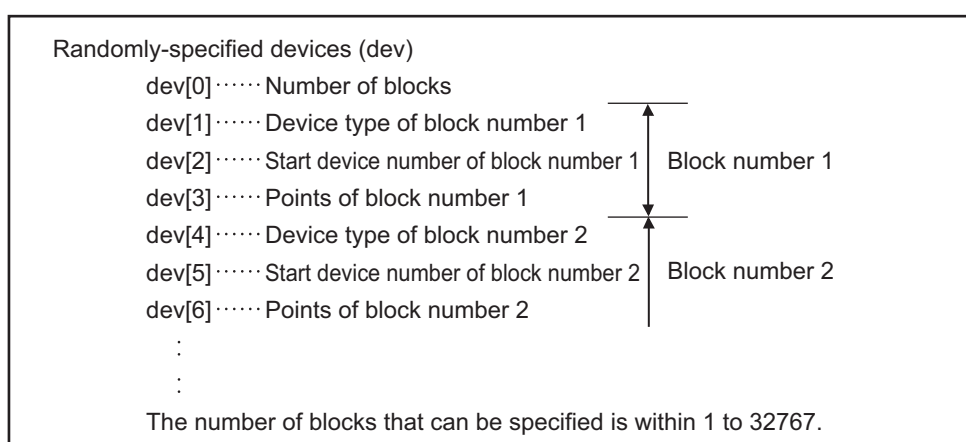
```
ret% = mdRandW(path&,stno%,dev%(0),buf%(0),bufsize%)
```

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

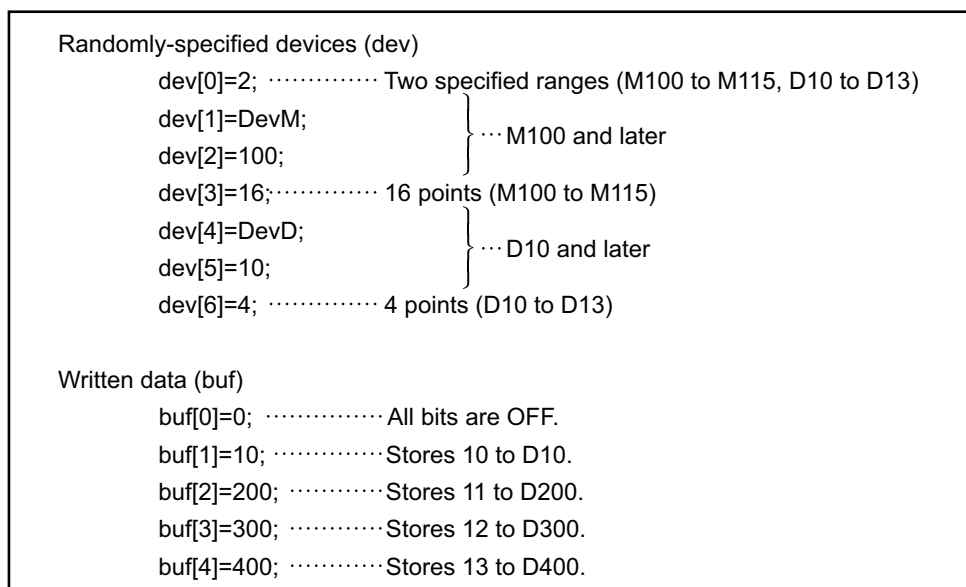
Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Station number	Specify the station number of target station. (Page 48, Section 4.2.2)
dev	Randomly-specified device	Specify the number of blocks, the device type, the start device number, and points of device to be written. For details, refer to Explanation shown on the next page.
buf	Written data (single-precision integer array)	Specify the data to be written in single-precision integer array.
bufsize	Dummy	—

○ 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.
- If an error occurs when writing devices randomly to B or W of the own station in MELSECNET(II) or MELSECNET/10, blocks with errors may exist among blocks in which the write function is normally completed.
- The following explains how to specify the randomly-specified devices (dev)



- The following shows an example when writing data to multiple devices.
Example) When all bits of M100 to M115 are set to OFF, and 10, 200, 300, and 400 are written to D10, D11, D12, and D13 respectively





Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (👉 Page 126, CHAPTER 6)



Related function

mdOpen(), mdClose(), mdRandR()

4.3.22 mdRandR (reading devices randomly)



Function

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



Format

Visual C++®

```
ret = mdRandR(path,stno,dev,buf,bufsize);
```

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


```
ret = mdRandR(path,stno,dev(0),buf(0),bufsize)
```

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

```
ret% = mdRandR(path&,stno%,dev%(0),buf%(0),bufsize%)
```

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

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Station number	Specify the station number of target station. ( Page 48, Section 4.2.2)
dev	Randomly-specified device	Specify the number of blocks, the device type, the start device number, and points of device to be read. For details, refer to Explanation shown on the next page.
buf	Read data (single-precision integer array)	Store the data being read.
bufsize	Number of bytes of read data	Specify the number of bytes of read data.

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 following explains how to specify the randomly-specified devices (dev).

Randomly-specified devices (dev)

dev[0]	Number of blocks	
dev[1]	Device type of block number 1	
dev[2]	Start device number of block number 1	Block number 1
dev[3]	Points of block number 1	
dev[4]	Device type of block number 2	
dev[5]	Start device number of block number 2	Block number 2
dev[6]	Points of block number 2	
:		
:		

The number of blocks that can be specified is within 1 to 32767.

- The following shows an example when reading multiple devices.

Example) When reading the values of M100 to M115, D10 to D13, M0 to M13, and the current value of T10.

M100 to M115 All bits are OFF.

D10 to D13 10 for D10, 200 for D11, 300 for D12, and 400 for D13

M0 to M13 All bits are ON.

Current value of T10 10 (1 second) for the current value of T10

Note that the above values are presumed values in the current status of the target devices.

Randomly-specified devices (dev)

dev[0]=4;	Four specified ranges
	(M100 to M115, D10 to D13, M0 to M13, T10)
dev[1]=DevM;	} ... M100 and later
dev[2]=100;	
dev[3]=16;	16 points (M100 to M115)
dev[4]=DevD;	} ... D10 and later
dev[5]=10;	
dev[6]=4;	4 points (D10 to D13)
dev[7]=DevM;	} ... M0 and later
dev[8]=0;	
dev[9]=14;	14 points (M0 to M13)
dev[10]=DevTN;	} ... Current and later values of T10 timer
dev[11]=10;	
dev[12]=1;	1 point (T10)

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) x 2 = 14 bytes

Enter "14" for this example.

Read data (buf)

buf[0]=0;	All bits of M100 to M115 are OFF
	(16 points of bit information can be stored.)
buf[1]=10;	Current value of D10
buf[2]=200;	Current value of D11
buf[3]=300;	Current value of D12
buf[4]=400;	Current value of D13
buf[5]=0x3fff;	All bits of M0 to M13 are ON
buf[6]=10;	Current value of T10 is 10 (1 second).



Return value

Success : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (Page 126, CHAPTER 6)



Related function

mdOpen(), mdClose(), mdRandW()

4.3.23 mdDevSet (setting bit devices)



Function

Set the bit devices on the target station (to ON).



Format

Visual C++®

```
ret = mdDevSet(path,stno,devtyp,devno);
```

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

```
ret = mdDevSet(path,stno,devtyp,devno)
```

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

```
ret% = mdDevSet(path&,stno%,devtyp%,devno%)
```

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

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Station number	Specify the station number of target station. (Page 48, Section 4.2.2)
devtyp	Device type	Specify the type of device to be set (ON). (Page 51, Section 4.2.4)
devno	Specified device number	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 : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (Page 126, CHAPTER 6)



Related function

mdOpen(), mdClose(), mdDevRst()

4.3.24 mdDevRst (resetting bit devices)



Function

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



Format

Visual C++®

```
ret = mdDevRst(path,stno,devtyp,devno);
```

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

```
ret = mdDevRst(path,stno,devtyp,devno)
```

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

```
ret% = mdDevRst(path&,stno%,devtyp%,devno%)
```

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

Argument		Description
path	Path of channel	Specify the path of channel whose communication line is opened. (Use the path which is returned when executing mdOpen.)
stno	Station number	Specify the station number of target station. (Page 48, Section 4.2.2)
devtyp	Device type	Specify the type of device to be reset (OFF). (Page 51, Section 4.2.4)
devno	Specified device number	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 : Return the value 0.

Error : Return the value other than 0. Refer to the list of error codes. (Page 126, CHAPTER 6)



Related function

mdOpen(), mdClose(), mdDevSet()

CHAPTER 5 SAMPLE PROGRAMS

This chapter explains the sample programs that are included on the CD-ROM of the software package. Sample programs are provided as a reference for creating user application programs. Use sample programs on a user's own responsibility.

5.1 Sample Programs for CC-Link Ver.2 Board

Sample programs are stored on the user-specified folder that is specified when installing the software package to a personal computer.

- <User-specified folder> - <CCBD2> - <Sample>

5.1.1 Description of sample programs

(1) Sample Programs for MELSEC data link library

The following sample programs are stored as examples of functions.

(a) Reading device data

A sample program for reading device D0 of a master station.

Folder name: MDFunction\ (folder for programming language) \DEMO

(b) Testing MELSEC data link library in general

A sample program for the MELSEC data link library in general.

Folder name: MDFunction\ (folder for programming language) \MTEST

Point


Sample programs for MTEST folder of NETVB

- This sample program cannot be used as a 64-bit version user application. In order to use this sample program on a 64-bit version operating system, select "x86" for "Target CPU", and build the program as a 32-bit version user application.
 - MSFlexGrid is used in the sample program. If MSFlexGrid is not installed, a warning occurs, however, the operation of sample program is not affected.
-

(2) Sample programs for checking data link of 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.

For details, refer to the following manual.

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

(a) Communication between a master station and a remote station

A sample program to read/write the remote I/Os for the remote I/O station.

Folder name: MasterStation\RemoteIO

(b) Communication between a master station and a remote device station

A sample program to perform the digital/analog conversion for AJ65BT-64DAV.

Folder name: MasterStation\RemoteDevice

(c) Communication between a master station and a local station

A sample program for communications between a master and a local station.

Folder name: MasterStation\Datalink, LocalStation\Datalink

(d) Communication between a master station and an intelligent device station (AJ65BT-R2)

A sample program to perform the initialization, transmission and reception of AJ65BT-R2.

Folder name: MasterStation\R2

(e) Communication between a master station and an intelligent device station (AJ65BT-D75P2-S3)

A sample program to perform the initialization, positioning, origin point return and JOG operation of AJ65BT-D75P2-S3.

Folder name: MasterStation\PositioningSystem

5.1.2 Programming language

Sample programs are stored on the following folders for each programming language to be used.

(1) NETVB folder


Sample programs for the following Visual Basic®.

- Visual Basic®.NET 2003
- Visual Studio®2005 Visual Basic®
- Visual Studio®2008 Visual Basic®
- Visual Studio®2010 Visual Basic®

Sample programs are created in Visual Basic®.NET 2003 project format.

Convert the project when using Visual Basic® other than Visual Basic®.NET 2003.


Remark

"Mdfunc.vb" is not included in the sample program. Add "Mdfunc.vb" before using the sample program.
(For the setting method  Page 19, Section 2.3.3)

(2) VB folder

Sample programs for Visual Basic®5.0 and Visual Basic®6.0.

Remark

"Mdfunc.bas" is not included in the sample program. Add "Mdfunc.bas" before using the sample program.
(For the setting method  Page 16, Section 2.3.1)

(3) VC folder


Sample programs for the following Visual C++®.

- Visual C++®5.0
- Visual C++®6.0
- Visual C++®.NET 2003
- Visual Studio®2005 Visual C++®
- Visual Studio®2008 Visual C++®
- Visual Studio®2010 Visual C++®

Sample programs are created in Visual C++®5.0 project format.

Convert the project when using Visual C++® other than Visual C++®5.0.

Remark

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  Page 17, Section 2.3.2, Page 20, Section 2.3.4, Page 22, Section 2.3.5)

5.2 Sample Programs for MELSECNET/H Board

Sample programs are stored in the user-specified folder that is specified when installing the software package to a personal computer.

- <User-specified folder> - <MNETH> - <SAMPLES>

5.2.1 Description of sample programs

(1) Sample Programs for MELSEC data link library

The following sample programs are stored as examples of functions.

(a) Reading device data

A sample program for reading device D0 of network number 1 and station number 1.

Folder name: (folder for programming language)\DEMO

(b) Testing MELSEC data link library in general

A sample program for the MELSEC data link library in general.

Folder name: (folder for programming language)\MTEST

Point

Sample programs for MTEST folder of NetVB

- This sample program cannot be used as a 64-bit version user application. In order to use this sample program on a 64-bit version operating system, select "x86" for "Target CPU", and build the program as a 32-bit version user application.
 - MSFlexGrid is used in the sample program. If MSFlexGrid is not installed, a warning occurs, however, the operation of sample program is not affected.
-

5.2.2 Programming language

Sample programs are stored on the following folders for each programming language to be used.

(1) NetVb folder


Sample programs for the following Visual Basic®.

- Visual Basic®.NET 2003
- Visual Studio® 2005 Visual Basic®
- Visual Studio® 2008 Visual Basic®
- Visual Studio® 2010 Visual Basic®

Sample programs are created in Visual Basic®.NET 2003 project format.

Convert the project when using Visual Basic® other than Visual Basic®.NET 2003.


Remark

"Mdfunc.vb" is not included in the sample program. Add "Mdfunc.vb" before using the sample program.
(For the setting method  Page 19, Section 2.3.3)

(2) VB folder

Sample program for Visual Basic® 5.0 and Visual Basic® 6.0.

Remark

"Mdfunc.bas" is not included in the sample program. Add "Mdfunc.bas" before using the sample program.
(For the setting method  Page 16, Section 2.3.1)

(3) VC folder

Sample program for the following Visual C++®.


- Visual C++® 5.0
- Visual C++® 6.0
- Visual C++®.NET 2003
- Visual Studio® 2005 Visual C++®
- Visual Studio® 2008 Visual C++®
- Visual Studio® 2010 Visual C++®

Sample programs are created in VisualC++®5.0 project format.

Convert the project when using Visual C++® other than Visual C++® 5.0.

Remark

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  Page 17, Section 2.3.2, Page 20, Section 2.3.4, Page 22, Section 2.3.5)

5.3 Sample Programs for CC-Link IE Controller Network Board

Sample programs are stored on the user-specified folder that is specified when installing the software package to a personal computer.

- <User-specified folder> - <MNETG> - <SAMPLES>

5.3.1 Description of sample programs

(1) Sample Programs for MELSEC data link library

The following sample programs are stored as examples of functions.

(a) Reading device data

A sample program for reading device D0 of network number 1 and station number 1.

Folder name: (folder for programming language)\DEMO

(b) Testing MELSEC data link library in general

A sample program for the MELSEC data link library in general.

Folder name: (folder for programming language)\MTEST

Point

Sample programs for MTEST folder of NETVB

- This sample program cannot be used as a 64-bit version user application. In order to use this sample program on a 64-bit version operating system, select "x86" for "Target CPU", and build the program as a 32-bit version user application.
- MSFlexGrid is used in the sample program. If MSFlexGrid is not installed, a warning occurs, however, the operation of sample program is not affected.

5.3.2 Programming language

Sample programs are stored on the following folders for each programming language to be used.

(1) NETVB folder


Sample programs for the following Visual Basic®.

- Visual Basic® .NET 2003
- Visual Studio® 2005 Visual Basic®
- Visual Studio® 2008 Visual Basic®
- Visual Studio® 2010 Visual Basic®

Sample programs are created in Visual Basic®.NET 2003 project format.

Convert the project when using Visual Basic® other than Visual Basic®.NET 2003.


Remark

"Mdfunc.vb" is not included in the sample program. Add "Mdfunc.vb" before using the sample program.
(For the setting method  Page 19, Section 2.3.3)

(2) VB folder

Sample programs for Visual Basic®6.0.

Remark

"Mdfunc.bas" is not included in the sample program. Add "Mdfunc.bas" before using the sample program.
(For the setting method  Page 16, Section 2.3.1)

(3) VC folder

Sample programs for the following Visual C++®.


- Visual C++®6.0
- Visual C++®.NET 2003
- Visual Studio® 2005 Visual C++®
- Visual Studio® 2008 Visual C++®
- Visual Studio® 2010 Visual C++®

Sample programs are created in VisualC++®6.0 project format.

Convert the project when using Visual C++® other than Visual C++® 6.0.

Remark

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  Page 17, Section 2.3.2, Page 20, Section 2.3.4, Page 22, Section 2.3.5)

5.4 Sample Programs for CC-Link IE Field Network Board

Sample programs are stored on the user-specified folder that is specified when installing the software package to a personal computer.

- <User-specified folder> - <CCIEF> - <SAMPLES>

5.4.1 Description of sample programs

(1) Sample Programs for MELSEC data link library

The following sample programs are stored as examples of functions.

(a) Reading device data

A sample program for reading device D0 of network number 1 and station number 1.

Folder name: (folder for programming language)\DEMO

(b) Testing MELSEC data link library in general

A sample program for the MELSEC data link library in general.

Folder name: (folder for programming language)\MTEST

Point

Sample programs for MTEST folder of NETVB

- This sample program cannot be used as a 64-bit version user application. In order to use this sample program on a 64-bit version operating system, select "x86" for "Target CPU", and build the program as a 32-bit version user application.
 - MSFlexGrid is used in the sample program. If MSFlexGrid is not installed, a warning occurs, however, the operation of sample program is not affected.
-

5.4.2 Programming language

Sample programs are stored on the following folders for each programming language to be used.

(1) NETVB folder


Sample programs for the following Visual Basic®.

- Visual Basic®.NET 2003
- Visual Studio® 2005 Visual Basic®
- Visual Studio® 2008 Visual Basic®
- Visual Studio® 2010 Visual Basic®

Sample programs are created in Visual Basic®.NET 2003 project format.

Convert the project when using Visual Basic® other than Visual Basic®.NET 2003.

Remark

"Mdfunc.vb" is not included in the sample program. Add "Mdfunc.vb" before using the sample program.
(For the setting method  Page 19, Section 2.3.3)

(2) VC folder


Sample programs for the following Visual C++®.

- Visual C++®.NET 2003
- Visual Studio® 2005 Visual C++®
- Visual Studio® 2008 Visual C++®
- Visual Studio® 2010 Visual C++®

Sample programs are created in VisualC++®.NET 2003 project format.

Convert the project when using Visual C++® other than Visual C++®.NET 2003.

Remark

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  Page 20, Section 2.3.4, Page 22, Section 2.3.5)

CHAPTER 6 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, consult your local Mitsubishi representative.

Error Code (HEX)	Error description	Corrective action
1 (0001H)	Driver not started The driver is not started.	<ul style="list-style-type: none"> • Check the channel number. • Check the driver operating condition with Device Manager. • Check the error of event viewer. • Check the board settings.
2 (0002H)	Time-out error <ul style="list-style-type: none"> • Timeout during waiting for the response of process • A CPU module other than QCPU (Q mode) is accessed from the 64-bit version user application. 	<ul style="list-style-type: none"> • 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), use the 32-bit version user application.
66 (0042H)	Channel-opened error Specified channel is already opened.	<ul style="list-style-type: none"> • Open the channel only once. • Since the correct value is stored to the path, this error can be regarded as normal status.
69 (0045H)	Unsupported function execution error A function which is not supported by the target station was executed.	<ul style="list-style-type: none"> • Check the path of channel, network number, and station number. • Check if the function is supported by the target station.
70 (0046H)	Station number error <ul style="list-style-type: none"> • 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 (255FFH) but the network number is not 0. 	<ul style="list-style-type: none"> • Check the network number and station number.
71 (0047H)	No reception data error(when RECV function) Data is not received.	<ul style="list-style-type: none"> • Check the channel number. • Check whether the data is sent on the sending station with the SEND function. • Retry the operation.
77 (004DH)	Memory reservation error/resource memory shortage error Enough memory could not be reserved.	<ul style="list-style-type: none"> • There is a possibility of a memory shortage. Terminate other application(s) currently running. • Exit the program and restart the personal computer. • Increase the minimum working set size of the personal computer. (Page 133, Appendix 1)
85 (0055H)	SEND/RECV channel number error The channel number specified with the SEND/RECV function is incorrect.	<ul style="list-style-type: none"> • 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.	<ul style="list-style-type: none"> • Retry the operation. • If the error occurs repeatedly, please consult your local Mitsubishi representative.
101 (0065H)	Routing parameter error The routing parameter is not set correctly.	<ul style="list-style-type: none"> • Check the routing parameter. • Check the specified network number.
102 (0066H)	Board Driver I/F error An attempt to send request data to the board driver is failed.	<ul style="list-style-type: none"> • Retry the operation. • Check with the function such as event log whether Windows® is operating normally. • Exit the program and restart the personal computer.

Error Code (HEX)	Error description	Corrective action
103 (0067H)	Board Driver I/F error An attempt to receive response data from the board driver is failed.	<ul style="list-style-type: none"> • Retry the operation. • Check with the function such as event log whether Windows® is operating normally. • Exit the program and restart the personal computer.
133 (0085H)	Parameter error A parameter set on the board is incorrect.	<ul style="list-style-type: none"> • Reset the board. • Correct the parameter.
4096 to 16383 (1000H to 3FFFH)	MELSEC data link library internal error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall the software package. • Consult your local Mitsubishi representative.
16384 to 20479 (4000H to 4FFFH)	Error detected by the access target CPU	<ul style="list-style-type: none"> • Refer to the user's manual of the access target CPU module.
16432 (4030H)	Device error The specified device type does not exist.	<ul style="list-style-type: none"> • Specify a device type described in the device type list. • Check if the device number is validated on the programmable controller CPU of the target station.
16433 (4031H)	Device error <ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • The process was requested with a 64-bit version user application to a CPU module other than QCPU (Q mode) module. 	<ul style="list-style-type: none"> • Check the content of specified request data. • When accessing a CPU module other than QCPU (Q mode) module, use 32-bit version user application.
18944 (4A00H)	Link-related error <ul style="list-style-type: none"> • The network of the number set to the routing parameters does not exist. • The network is not supported by the target CPU. 	<ul style="list-style-type: none"> • Check the routing parameter settings. • Exchange the CPU with a CPU which supports the network. • Check the network number and station number.
18945 (4A01H)	<ul style="list-style-type: none"> • The network number or station number of the target station is incorrect. 	
-1 (FFFFH) (FFFFFFFFH)	Path error The specified path is invalid.	<ul style="list-style-type: none"> • Use the path that was returned by the mdOpen function. • Use a path of the communication line that supports the function.
-2 (FFFEH) (FFFFFFEH)	Start device number error <ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • Check the start device number. • When specifying a bit device, specify a start device number in multiples of 8. • Check the device number and points. • Check if the specified device is validated on the programmable controller CPU of the target station.
-3 (FFFDH) (FFFFFFFDH)	Device type error The specified device type is invalid.	<ul style="list-style-type: none"> • Specify a device type described in the device type list. • Check if the specified device is validated on the programmable controller CPU of the target station.
-5 (FFFBH) (FFFFFFFBH)	Size error <ul style="list-style-type: none"> • The set of start device number and size is over the device range. • An access was attempted with odd number bytes. 	<ul style="list-style-type: none"> • Check the device size. • Check the start device number and size. • Specify even number bytes.

Error Code (HEX)	Error description	Corrective action
-6 (FFFAH) (FFFFFFFAH)	Number of blocks error The number of blocks specified for the device random read/write is out of the range.	<ul style="list-style-type: none"> Check the number of blocks.
-8 (FFF8H) (FFFFFFF8H)	Channel number error The channel number specified in the mdOpen function is invalid.	<ul style="list-style-type: none"> Check the channel number.
-12 (FFF4H) (FFFFFFF4H)	Block number error The block number of the specified file register is invalid.	<ul style="list-style-type: none"> 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.	<ul style="list-style-type: none"> 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.	<ul style="list-style-type: none"> Check the network number and the station number.
-17 (FFEFH) (FFFFFFEFH)	All station specification and group number specification error <ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> Check the network number and station number. Check if the function supports all station specification and group number specification. 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.	<ul style="list-style-type: none"> 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.	<ul style="list-style-type: none"> Check the channel number.
-31 (FFE1H) (FFFFFFE1H)	DLL load error An attempt to load DLL required to execute the function failed.	<ul style="list-style-type: none"> Reinstall the software package.
-32 (FFE0H) (FFFFFFE0H)	Resource time-out error <ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> Close the user program correctly. Retry the operation. There is a possibility of a memory shortage. Terminate other application(s) currently running. Exit 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.	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Reinstall the software package.
-35 (FFDDH) (FFFFFFDDH)		
-36 (FFDCH) (FFFFFFDCH)		


Error Code (HEX)	Error description	Corrective action
-37 (FFDBH) (FFFFFFDBH)	Communication initialization setting error The initial setting for communication is failed.	<ul style="list-style-type: none"> • Retry the operation. • There is a possibility of a memory shortage. Terminate other application(s) currently running. • Exit the program and restart the personal computer. • Check the memory availability.
-42 (FFD6H) (FFFFFFD6H)	Close error The communication cannot be closed.	<ul style="list-style-type: none"> • Retry the operation. • Exit 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.	<ul style="list-style-type: none"> • Change the TC setting value during RAM operation.
-61 (FFC3H) (FFFFFFC3H)	Number of events error The number of events which is specified in the mdWaitBdEvent function to set the user application to wait is out of the range.	<ul style="list-style-type: none"> • Check the number of events that sets the user application to wait.
-62 (FFC2H) (FFFFFFC2H)	Event number error The event number which is specified in the mdWaitBdEvent function to set the user application to wait is out of the range.	<ul style="list-style-type: none"> • Check the event number that sets the user application to wait.
-63 (FFC1H) (FFFFFFC1H)	Event number overlapped registration error The event number which is specified in the mdWaitBdEvent function to set the user application to wait is overlapped.	<ul style="list-style-type: none"> • Specify the event numbers, that set the user application to wait, not to overlap.
-64 (FFC0H) (FFFFFFC0H)	Timeout time error The timeout time specified in the mdWaitBdEvent function is out of the range.	<ul style="list-style-type: none"> • Check the time-out value.
-65 (FFBFH) (FFFFFFBFH)	Event wait time-out error The event did not occur within the timeout time.	<ul style="list-style-type: none"> • Retry the operation.
-66 (FFBEH) (FFFFFFBEH)	Event initialization error The board or the master station (control station) was reset during the execution of the mdWaitBdEvent function.	<ul style="list-style-type: none"> • Retry the operation.
-67 (FFBDH) (FFFFFFBDH)	No event setting error The event setting of the event number which is specified in the mdWaitBdEvent function to set the user application to wait does not exist.	<ul style="list-style-type: none"> • Set the event number that sets the user application to wait in the utility.
-70 (FFBAH) (FFFFFFBAH)	Event overlapped occurrence error The event with the same event number occurred multiple times.	<ul style="list-style-type: none"> • Set the interval of the event occurrence longer enough for the user program to process.
-257 (FEFFH) (FFFFFFEFH) to -4096 (F000H) (FFFFFF00H)	Errors detected in the MELSECNET/H and MELSECNET/10 network system	<ul style="list-style-type: none"> • Refer to the MELSECNET/H and MELSECNET/10 Network System Reference Manuals.
-2174 (F782H) (FFFFFF782H)	Transient data target station number error <ul style="list-style-type: none"> • The target station number is incorrect. • The target station number is 0. 	<ul style="list-style-type: none"> • Check the target station number, and perform the operation again. • If the error occurs after performing the above corrective action, please consult your local Mitsubishi representative.

Error Code (HEX)	Error description	Corrective action
-4097 (EFFFH) (FFFFEFFFH) to -8192 (E000H) (FFFFE000H)	Errors detected in the CC-Link IE Controller network system	<ul style="list-style-type: none"> Refer to the CC-Link IE Controller Network System Reference Manuals.
-7656 (E218H) (FFFFE218H)	Transient data target station number error <ul style="list-style-type: none"> The target station number is incorrect. The target station number is 0. 	<ul style="list-style-type: none"> Check the target station number, and perform the operation again. If the error occurs after performing the above corrective action, please consult your local Mitsubishi representative.
-7672 (E208H) (FFFFE208H)		
-8193 (DFFFH) (FFFFDFFFH) to -12288 (D000H) (FFFFD000H)	Errors detected in the CC-Link IE Field network system	<ul style="list-style-type: none"> Refer to the CC-Link IE Field Network Board and CC-Link IE Field Network Master/Local Module User's Manual.
-11746 (D21EH) (FFFFD21EH)	Station number error <ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> Check the network number and station number.
-12289 (CFFFH) (FFFFCFFFH) to -16384 (C000H) (FFFFC000H)	Errors detected in the Ethernet network system	<ul style="list-style-type: none"> Refer to the Ethernet Interface Module User's Manual.
-16385 (BFFFH) (FFFFBFFFH) to -20480 (B000H) (FFFFB000H)	Errors detected in the CC-Link system	<ul style="list-style-type: none"> Refer to the CC-Link system master/local board and CC-Link system master/local module user's manual.
-18560 (B780H) (FFFFB780H)	Module mode setting error A transient transmission was executed to the remote I/O station.	<ul style="list-style-type: none"> 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.	<ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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) module. 	<ul style="list-style-type: none"> Check the request destination station number and the processing code. Do not execute the function other than mdTypeRead to the other station board. When accessing a CPU module other than QCPU (Q mode) module, use 32-bit version user application.

Error Code (HEX)	Error description	Corrective action
-26334 (9922H) (FFFF9922H)	Reset error <ul style="list-style-type: none"> Reset was executed by another task that uses the same channel when accessing to the own station or the other station. Reset was executed when monitoring with the utility. 	<ul style="list-style-type: none"> Retry the operation.
-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.	<ul style="list-style-type: none"> Check the Routing Parameter Setting.
-28140 (9214H) (FFFF9214H)	Incorrect mode setting error An incorrect mode was specified when setting the mode.	<ul style="list-style-type: none"> Check the mode.
-28141 (9213H) (FFFF9213H)	System sleep error Shift to the sleep or the hibernate was detected.	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Change the setting of Power Options to prevent the system from entering sleep or hibernate.
-28142 (9212H) (FFFF9212H)	Mode error A request which cannot be used in the currently set mode was executed.	<ul style="list-style-type: none"> 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.	<ul style="list-style-type: none"> Check the system log of event viewer, and take a corrective action on the registered error. Exit 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. Consult your local Mitsubishi 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.	<ul style="list-style-type: none"> Check the specified device start number and size, or the device range of the parameter on the master station. <p>The data write/read function can be performed even when this error occurs, but the function of the data security is not guaranteed.</p>
-28151 (9209H) (FFFF9209H)	Abnormal data reception error An incorrect response data was received.	<ul style="list-style-type: none"> 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.
-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. <ul style="list-style-type: none"> Windows® activation process when starting the personal computer Operation of a device driver such as a graphic board Operation of other software applications 	<ul style="list-style-type: none"> Reset the board. Restart the personal computer. Remove the factor of system overload. Set a longer Driver WDT Monitoring Time of the utility. <p>By changing the graphic board, an environment in which a driver WDT error does not occur may be created.</p>

Error Code (HEX)	Error description	Corrective action
-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.	<ul style="list-style-type: none"> • 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 board error was detected by the hardware self-diagnosis.	<ul style="list-style-type: none"> • Check the system log of event viewer, and take a corrective action on the registered error. • Exit 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. • Consult your local Mitsubishi representative.
-28636 (9024H) (FFFF9024H)		

Remark

When an error code is returned as a return value of the extended function ( Page 45, Section 4.1), 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 with code 77(004Dн) occurs due to the execution of the MELSEC data link library function. The personal computer board driver runs using the minimum working set size in the memory area reserved in the user program. 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 134, Appendix 1.2)

The minimum working set size of 200KB 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.

Appendix 1.1 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 the SetProcessWorkingSetSize function.
- ⑤ Release the user program handle with the CloseHandle function.

A

Appendix 1.2 Sample programs

(1) When setting with Visual Basic®

(An example when the minimum working set size is 1MB and the maximum working set size is 3MB)

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.

```
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 bret As Boolean    '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, False, id)
'Obtain the maximum working set size and the minimum working set size for the user program
bret = GetProcessWorkingSetSize(ph, wkmin, wkmax)
'Set the minimum working set size to 1MB(1 * 1024 * 1024 = 1048576)
wkmin = 1048576
'Set the maximum working set size to 3MB(3 * 1024 * 1024 = 3145728)
wkmax = 3145728
'Change the maximum working set size and the minimum working set size for the user program
bret = SetProcessWorkingSetSize(ph, wkmin, wkmax)
'Close the user program handle
bret = CloseHandle(ph)
```

The set sizes shown here are reference sizes. Adjust the sizes according to your system.

(2) When setting with Visual C++®

(An example when the minimum working set size is 1MB and the maximum working set size is 3MB)

```
#define ERROR    -1
short ChangeWorkingSetSize()
{
    DWORD    dwProcessId;    /*User program ID variable*/
    HANDLE    hProcess;    /*User program handle variable*/
    DWORD    dwMinimumWorkingSetSize;    /*Minimum working set variable*/
    DWORD    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(ERROR);
    }

    /*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(ERROR);
    }

    /*Set the minimum working set size to 1MB*/
    dwMinimumWorkingSetSize = 1 * 1024 * 1024;
    /*Set the maximum working set size to 3MB*/
    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(ERROR);
    }

    /*Close the user program handle*/
    CloseHandle(hProcess);
    /*Normal return*/
    return(0);
}
```

The set sizes shown here are reference sizes. Adjust the sizes according to your system.

REVISIONS

*The manual number is written at the bottom left of the back cover.

Print date	*Manual number	Revision
Jul., 2011	SH-081035ENG-A	First edition
Apr., 2012	SH-081035ENG-B	<div>Addition</div> <p>Section 5.1.1, Section 5.1.2, Section 5.2.1, Section 5.2.2, Section 5.3.1, Section 5.3.2, Section 5.4.1, Section 5.4.2</p> <div>Correction</div> <p>TERMS, Section 2.1.1, Section 3.1.1, Section 3.1.2, Section 3.1.4, Section 3.2.1, Section 3.2.2, Section 3.2.3, Section 3.2.4, Section 5.1, Chapter 6</p>

Japanese Manual Version SH-081034-B

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

Memo

Warranty

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

1. Gratis Warranty Term and Gratis Warranty Range

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

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

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place.

Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

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

2. Onerous repair term after discontinuation of production

- (1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued.

Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.

- (2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

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

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

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

5. Changes in product specifications

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

MELSEC Data Link Library Reference Manual

MODEL	MD-FUNC-LIB-R-E
MODEL CODE	13JV25
SH(NA)-081035ENG-B(1204)KWIX	



HEAD OFFICE : TOKYO BUILDING, 2-7-3, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA, JAPAN

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

Specifications subject to change without notice.