

# mitsubishi

Type SW2D5F-CSKP-E Basic Communication Support Tool

---

## Programming Manual



Mitsubishi Programmable Controller



## • SAFETY INSTRUCTIONS •


(Always read these instructions before using this equipment.)

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

The instructions given in this manual are concerned with this product. For the safety instructions of the programmable controller system, please read the CPU module user's manual.



In this manual, the safety instructions are ranked as "DANGER" and "CAUTION".

 <b>DANGER</b>	Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
 <b>CAUTION</b>	Indicates that incorrect handling may cause hazardous conditions, resulting in medium or slight personal injury or physical damage.

Note that the  CAUTION level may lead to a serious consequence according to the circumstances. Always follow the instructions of both levels because they are important to personal safety.

Please save this manual to make it accessible when required and always forward it to the end user.

### [Design Instructions]

 <b>DANGER</b>
<ul style="list-style-type: none"><li>• Install a safety circuit outside of the PLC system that enables the system to operate normally even when data change or status control to an active PLC is attempted from a peripheral device. In addition, decide in advance the system countermeasures for a communication error due to cable disconnection during on-line operations to the PLC CPU from the peripheral device.</li></ul>
 <b>CAUTION</b>
<ul style="list-style-type: none"><li>• Read the manual thoroughly for safety before connecting a peripheral device to an active CPU module for on-line operation (in particular, forced output and run status change). Erroneous operation may cause equipment damage or accidents.</li></ul>

REVISIONS

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

Print Date	* Manual Number	Revision
Dec., 1998	IB (NA) 66889-A	First edition

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.

## Operating Instructions

### (1) Ethernet communication

(a) When the PLC CPU is accessed by Ethernet communication, functions may not be executed depending on the state of the PLC CPU.

#### 1) When TCP/IP is used for protocol (both E71 and QE71)

Functions can be executed only when the PLC CPU for communication is running.

When the PLC CPU is not running, an attempt to execute the functions terminates in error.

#### 2) When UDP/IP is used for protocol (E71 only)

Unless the PLC CPU is run once in advance, functions cannot be executed.

When the PLC CPU has not been run yet, an attempt to execute the function terminates in error.

(b) If the CPU goes down or the Ethernet module is reset during Ethernet communication, the communication loop will be cut off. In such a case, execute the loop close processing (mdClose), and then perform the reopen processing (mdOpen).

### (2) Access to bit devices

When a bit device is accessed via mdSend( ) and mdReceive( ), the leading device number must be set as shown in the following table.

Computer link communication	Multiple of 16 (0, 16, 32...)
Ethernet communication (TCP/IP)	
Access to RX, RY and SB in the CC-Link card (local station)	
Miscellaneous	Multiple of 8 (0, 8, 16...)

### (3) About access to FXCPU CN devices

The FXCPU CN devices with the number "200" and after cannot be accessed for writing or reading.

### (4) Use of VB 4.0 and VC++ 4.2

When using VB 4.0 or VC++ 4.2, you cannot make CC-Link communication.

## INTRODUCTION

Thank you for choosing the type SW2D5F-CSKP-E Basic Communication Support Tool.  
Before using the Basic Communication Support Tool, carefully read the manual to familiarize yourself with its functions.  
Please make this manual available to the end user.

## CONTENTS

Safety Instruction .....	A- 1
Revision .....	A- 2
Operating Instructions .....	A- 3
About Manuals .....	A- 6
How to Use This Manual .....	A- 7
About the Generic Terms and Abbreviations .....	A- 8

<b>1. OVERVIEW</b>	<b>1- 1 to 1- 2</b>
--------------------	---------------------

1.1 General Description of MELSEC Data Link Library .....	1- 1
1.2 List of Functions .....	1- 2

<b>2. MELSEC DATA LINK LIBRARY</b>	<b>2- 1 to 2- 10</b>
------------------------------------	----------------------

2.1 Setting to Use a Function .....	2- 1
2.2 Programming Procedure .....	2- 3
2.3 Channels .....	2- 5
2.4 Station Number Setup .....	2- 6
2.5 Device Type .....	2- 7
2.6 Accessible Devices and Range .....	2- 10

<b>3. MELSEC DATA LINK LIBRARY DETAILS</b>	<b>3- 1 to 3- 30</b>
--	----------------------

3.1 mdOpen (communication line open) .....	3- 2
3.2 mdClose (communication line close) .....	3- 3
3.3 mdSend (batch writing to devices) .....	3- 4
3.4 mdReceive (batch reading from devices) .....	3- 6
3.5 mdRandW (write to random devices) .....	3- 8
3.6 mdRandR (read from random devices) .....	3- 10
3.7 mdDevSet (bit device set) .....	3- 12
3.8 mdDevRst (bit device reset) .....	3- 13
3.9 mdTypeRead (PLC CPU type read) .....	3- 14
3.10 mdControl (remote RUN/STOP/PAUSE) .....	3- 16
3.11 mdInit (PLC device address table refreshing) .....	3- 17
3.12 mdBdRst (local card reset) .....	3- 18
3.13 mdBdModSet (local card mode set) .....	3- 19
3.14 mdBdModRead (local card mode read) .....	3- 20
3.15 mdBdLedRead (local card LED read) .....	3- 21

3.16 mdBdSwRead (local card switch read) .....	3- 24
3.17 mdBdVerRead (local card version read) .....	3- 25
3.18 mdSend (SEND function).....	3- 26
3.19 mdReceive (RECV function).....	3- 28

<b>4. SAMPLE PROGRAM</b>	<b>4- 1 to 4- 10</b>
--------------------------	----------------------

4.1 Visual Basic Sample Program .....	4- 1
4.2 Visual C++ Sample Program .....	4- 6

<b>5. ERROR CODES</b>	<b>5- 1 to 5- 10</b>
-----------------------	----------------------

## About Manuals

The following manuals are also related to this product.  
In necessary, order them by quoting the details in the tables below.

### Related Manuals

Manual Name	Manual Number (Model Code)
Type SW2D5F-CSKP-E Basic Communication Support Tool Operating Manual This manual describes how to set up and use each utility for communication. (Included in the product package)	IB-66888 (1LMS42)
Type A70BDE-J71QLP23/A70BDE-J71QLP23GE/A70BDE-J71QBR13 MELSECNET/10 Interface Board User's Manual (For SW2DNF-MNET10) This manual describes the features, specifications and nomenclature of the MELSECNET/10 interface card and the procedure for installing and uninstalling the driver. (Included in the product package)	IB-66894 (13JL81)
Type A80BDE-J61BT13 CC-Link Interface Board User's Manual (For SW2DNF-CCLINK) This manual describes the features, specifications and nomenclature of the CC-Link board and the procedure for installing and uninstalling the driver. (Included in the product package)	IB-66895 (13JL82)

## How to Use This Manual

"How to Use This Manual" is described by the purposes of using the functions of CSKP.

Refer to the following when using this manual.

- (1) To know the function list (Chapter 1)  
Chapter 1 gives a list of functions and explains what they will perform.
- (2) To use the functions in VB and VC++ (Section 2.1)  
Section 2.1 describes the ways of setting in VB and VC++ for use of the functions.
- (3) To know the programming procedure (Section 2.2)  
Check Section 2.2 which gives the programming procedure.
- (4) To know the channels, station numbers and device types specified for the functions (Sections 2.3 to 2.5)  
Sections 2.3 to 2.5 describes the channels, station numbers and device types.
- (5) To know the details of each function (Chapter 3)  
Chapter 3 gives the detailed explanation of each function.  
Read this chapter before creating programs.
- (6) To know how to use sample programs (Chapter 4)  
When creating programs, refer to Chapter 4 where sample programs and their usages are given.
- (7) To know the error codes displayed when using the functions (Chapter 5)  
Chapter 5 provides the error codes returned when the functions are used.  
When using the utilities, also read this chapter since error codes are returned.
- (8) To know the accessible devices and ranges  
Refer to the CSKP Operating Manual which gives the accessible devices and ranges.



## About the Generic Terms and Abbreviations

Unless stated explicitly, this manual uses the generic and abbreviations names listed in the following table to discuss the Type SW2D5F-CSKP-E Basic Communication Support Tool.

Generic/abbreviation name	Generic/abbreviation name definition
CSKP	Abbreviation for Type SW2D5F-CSKP-E Basic Communication Support Tool
Windows NT 4.0	Abbreviation for Microsoft Windows NT Workstation 4.0
Windows 95	Abbreviation for Microsoft Windows 95
Windows 98	Abbreviation for Microsoft Windows 98
Windows	Generic name for Microsoft Windows 95, Microsoft Windows 98, and Microsoft Windows NT Workstation 4.0
VB	Generic name for Microsoft Visual Basic 4.0 and Visual Basic 5.0
VC++	Generic name for Microsoft Visual C++ 4.2 and Visual C++ 5.0
Personal computer	Generic name for IBM PC/AT and its compatibles (DOS/V machines)
CC-Link G4 Module	Abbreviation for GPP Function Peripheral Connection Module Type AJ65BT-G4
GPPW	Abbreviation for GPP Function Software for Windows SW□D5C-GPPW-E/SW□D5F-GPPW-E
Ladder Logic Test Tool (LLT)	Abbreviation for Ladder Logic Test Function Tool Software for Windows SW□D5C-LLT-E/SW□D5F-LLT-E
MELSECNET/10 Card	Abbreviation for A70BDE-J71LP23GE, A70BDE-J71QLP23 and A70BDE-J71QBR13 MELSECNET/10 interface card
CC-Link Card	Abbreviation for A80BDE-J61BT13 CC-Link interface card
AnNCPUs	Generic name for A0J2HCPU, A1SCPU, A1SCPU-S1, A1SCPUC24-R2, A1SHCPU, A1SJCPU, A1SJCPU-S3, A1SJHCPU, A1SJHCPU-S8, A1NCPUs, A2CCPU, A2CCPUC24, A2CCPU24-PRF, A2CJCPU, A2NCPUs, A2NCPUs-S1, A2SCPU, A2SCPU-S1, A2SHCPU and A1FXCPU
AnACPU	Generic name for A2ACPU, A2ACPU-S1, A2ACPUP21/R21, A2ACPUP21/R21-S1, A3ACPUP21/R21, A3NCPUs, A3ACPU
AnUCPU	Generic name for A2UCPU, A2UCPU-S1, A2ASCPUs, A2ASCPUs-S1, A2ASCPUs-S30, A3UCPU and A4UCPU
QnACPU	Generic name for Q2ACPU, Q2ACPU-S1, Q2ASCPUs, Q2ASCPUs-S1, Q2ASHCPU, Q2ASHCPU-S1, Q3ACPU, Q4ACPU and Q4ARCPU
ACPU	Generic name for AnNCPUs, AnACPU and AnUCPU
FXCPU	Generic name for FX0, FX0s, FX1, FX2, FX2c, FX2N and FX2NC, Series
C24	Generic name for A1SCPUC24-R2, A1SJ71C24-PRF, A1SJ71C24-R2, A2CCPUC24, A2CCPUC24-PRF, AJ71C24-S6 and AJ71C24-S8
UC24	Generic name for AJ71UC24, AJ71UC24-PRF, A1SJ71UC24-R2 and A1SJ71UC24-PRF
QC24	Generic name for AJ71QC24, AJ71QC24-R2, AJ71QC24-R4, A1SJ71QC24, A1SJ71QC24-R2, AJ71QC24N, AJ71QC24N-R2, AJ71QC24N-R4, A1SJ71QC24N and A1SJ71QC24N-R2
E71	Generic name for AJ71E71, AJ71E71-S3, A1SJ71E71-B2, A1SJ71E71-B5, A1SJ71E71-B2-S3 and A1SJ71E71-B5-S3
QE71	Generic name for AJ71QE71, AJ71QE71-B5, A1SJ71QE71-B2 and A1SJ71QE71-B5

Microsoft Windows, Microsoft Windows NT, Microsoft Visual Basic and Microsoft Visual C++ are the registered trademarks of Microsoft Corporation.

Ethernet is the registered trademark of XEROX Corporation.

The other company names and product names are the trademarks or the registered trademarks of the pertinent companies.

1. OVERVIEW

This chapter outlines the operation of the functions provided by the MELSEC Data Link Library.

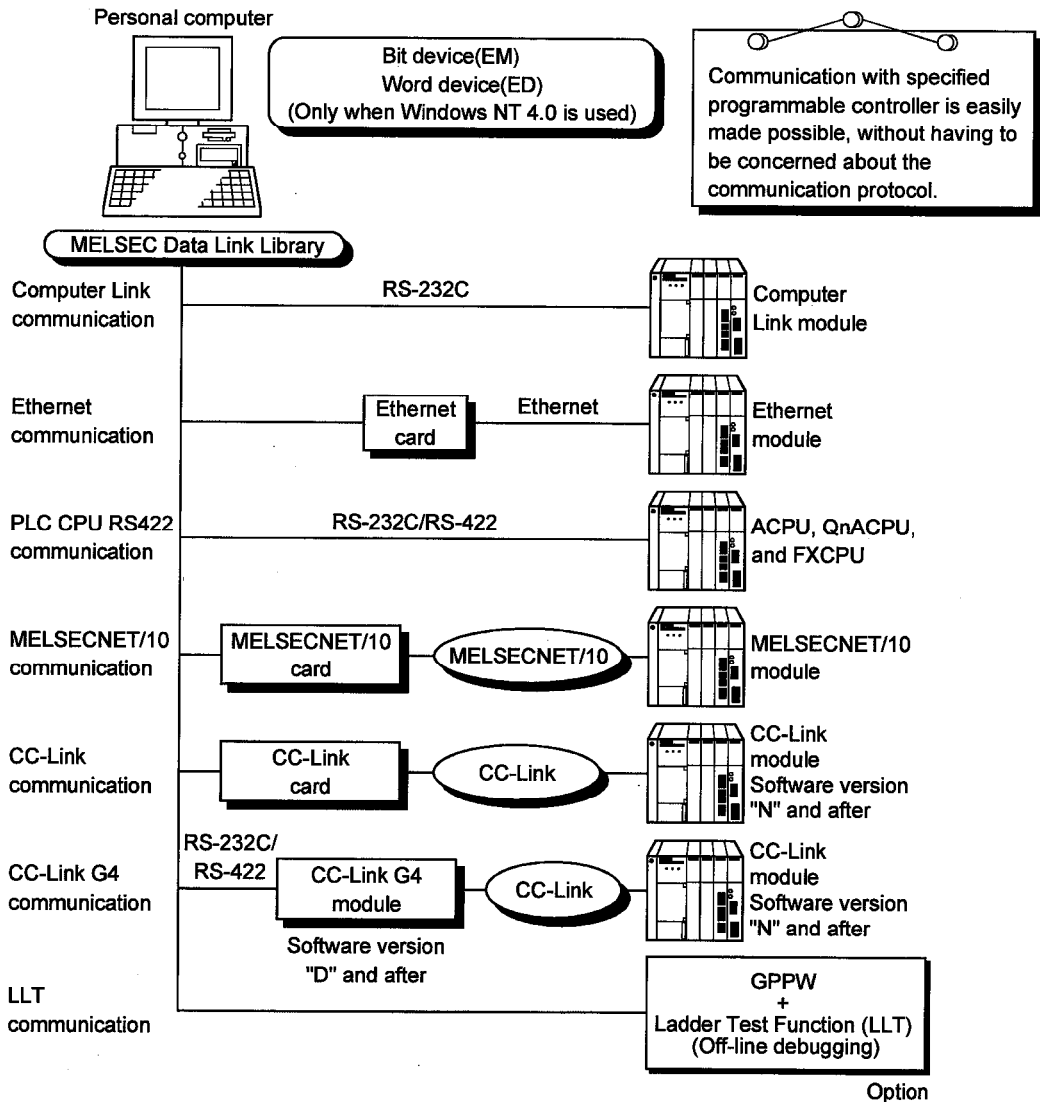
**REMARK**

To give a description of windows, this manual uses the windows under Windows 95. Note that they may differ slightly from those under Windows NT Workstation 4.0 and Windows 98.

1.1 General Description of MELSEC Data Link Library

The functions in the data link library are used for creating a user program that communicates with a PLC CPU.

Using these functions, the user can program communication procedures without being concerned about differences in hardware configurations and communication protocols at different stations.



## 1.2 List of Functions

The following functions are provided by the MELSEC Data Link Library of the CSKP.

Function name	Purpose	Reference
mdOpen	Opening communication line	Section 3.1
mdClose	Closing communication line	Section 3.2
mdSend	Batch writing to a device	Section 3.3
mdReceive	Batch reading from a device	Section 3.4
mdRandW	Writing to a random device	Section 3.5
mdRandR	Reading from a random device	Section 3.6
mdDevSet	Setting a bit device	Section 3.7
mdDevRst	Resetting a bit device	Section 3.8
mdTypeRead	Reading PLC CPU type	Section 3.9
mdControl	Remote RUN/STOP/PAUSE	Section 3.10
mdInit	Refreshing PC address	Section 3.11
mdBdRst	Resetting the local card	Section 3.12
mdBdModSet	Setting the mode for the local card	Section 3.13
mdBdModRead	Reading the mode set to the local card	Section 3.14
mdBdLedRead	Reading the states of LEDs on the local card	Section 3.15
mdBdSwRead	Reading the states of switches on the local card	Section 3.16
mdBdVerRead	Reading the version of the local card	Section 3.17
MdSend *1	Data sending (SEND)	Section 3.18
MdReceive *1	Data receiving (RECV)	Section 3.19

\*1 QnA dedicated instruction

## 2. MELSEC DATA LINK LIBRARY

This chapter describes how to make settings and code programs that use the MELSEC Data Link Library.

### 2.1 Setting to Use a Function

This section describes how to make settings when functions are used.

POINT
-------

When using Visual Basic 4.0 or Visual C++ 4.2, you cannot make CC-Link communication.
---

(1) If using Visual Basic 4.0

1) Start the Visual Basic 4.0, then select [File]-[Add File].

2) Select MDFUNC.BAS.

The file is stored at the following location during the installation:

<user-specified-directory>-<COMMON>-<INCLUDE>

(2) If using Visual Basic 5.0

1) Start the Visual Basic 5.0, then select [Project]-[Add Module].

2) Select "Existing" tab and then select "MDFUNC.BAS".

The file MDFUNC, BAS is stored at the following location during the installation:

<user -specified-directory>-<COMMON>-<INCLUDE>

(3) If using Visual C+ + 4.2

(a) In case of setting for Include File

1) Start the Visual C+ + 4.2 and select [Tools]-[Options].

2) Click the Directories tab, and then select "Include Files" as the directory to be displayed.

3) Double-click the item to be included.

The file MDFUNC.H is stored at the following location during the installation:

<user-specified-directory>-<COMMON>-<INCLUDE>

4) Add the following line to the top of the program:

```
# include<mdfunc.h>
```

(b) In case of setting for Library File

- 1) Start the Visual C++ 4.2 and select [Tools]-[Options].
- 2) Click the Directories tab, and then select "Library File" as the directory to be displayed.  
Then set the library file. Library file setting procedure is similar to the include file setting procedure explained in (a).
- 3) Open the workspace to be created and then choose [Build]-[Setting].
- 4) Click the Link tab, select "General" as the category, and "mdfunc32.lib" as the object/library modules.

(4) If using Visual C++ 5.0

(a) In case of setting for Include File

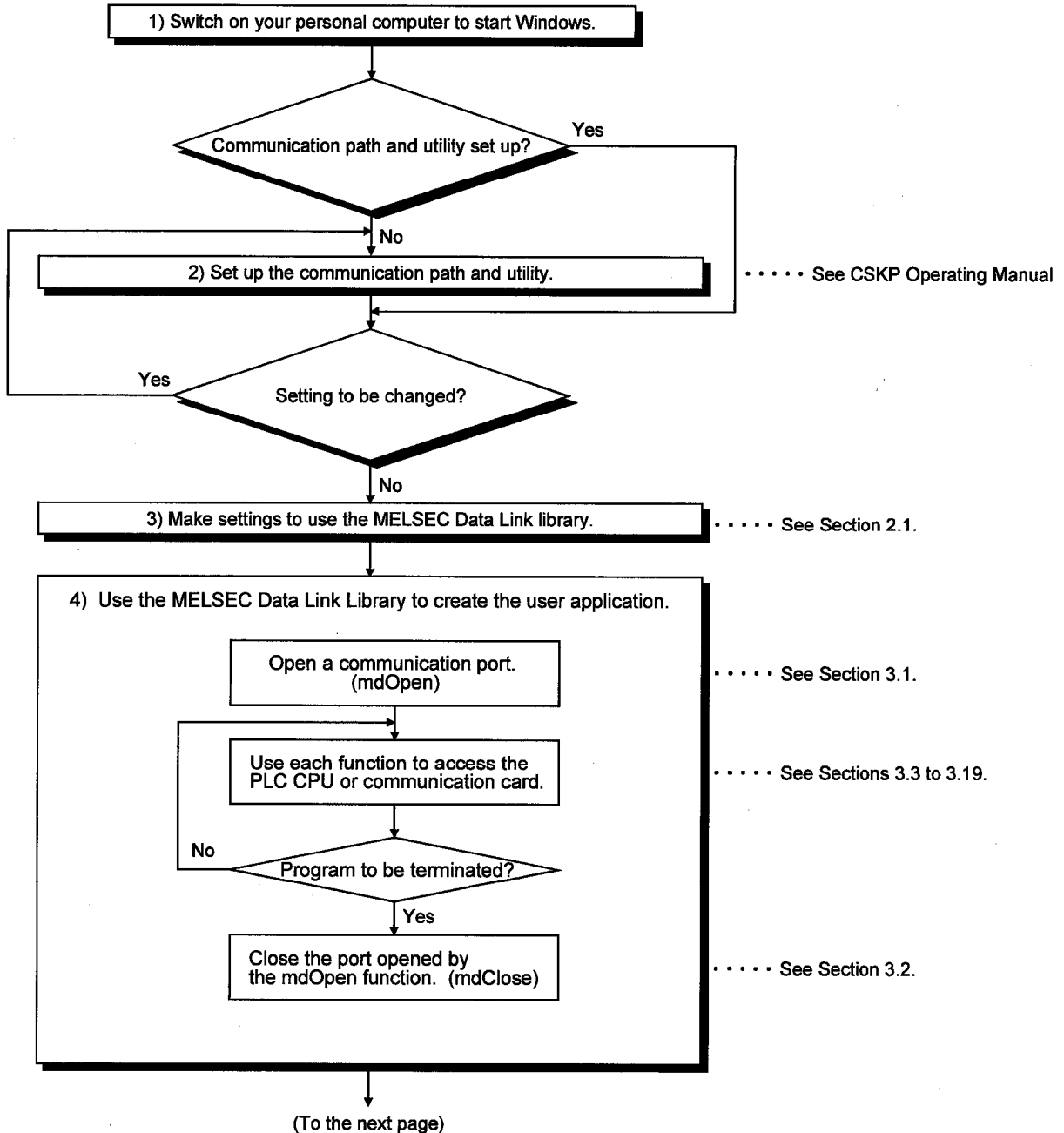
- 1) Start the Visual C++ 5.0 and select [Tools]-[Options].
- 2) Click the Directories tab, and then select "Include Files" as the directory to be displayed.
- 3) Double-Click the item to be included.  
The file "MDFUNC.H" is stored at the following location during the installation:  
<user- specified-directory>-<COMON>-<INCLUDE>
- 4) Add the following line to the top of the program:  
# include<mdfunc.h>

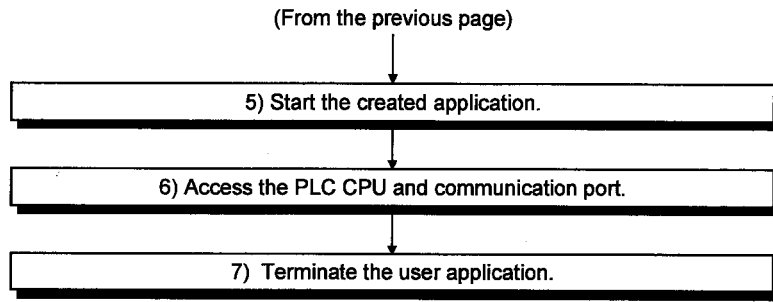
(b) In case of setting for Library File

- 1) Start the Visual C++ 5.0 and select [Tools]-[Options].
- 2) Click the Directories tab, and then select "Library Files" as the directory to be displayed.  
Then set the library file.  
Library file setting procedure is similar to the include file setting procedure explained in (a).
- 3) Open the workspace to be created and then choose [Project]-[Settings].
- 4) Click the link tab, select "General" as the category, and "mdfunc 32.lib" as the object /library modules.

2.2 Programming Procedure

This section describes how to code programs using the MELSEC Data Link Library. For descriptions, it assumes that CSKP has already been installed.





POINTS
<ul style="list-style-type: none"><li>• The communication line must be opened and closed only once by the mdOpen and mdClose functions at the start and end of the program. Communication performance deteriorates if line opening/closing takes place with each communication.</li><li>• When the PLC CPU and communication card are accessed again by the created user program, they can be accessed by the operations 5) to 7) only.</li></ul>

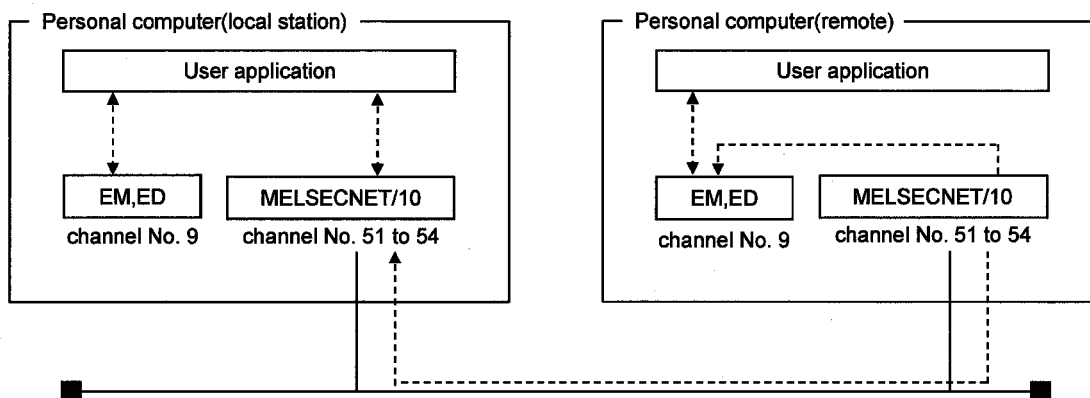
2.3 Channels

The following channels are used by the functions in the MELSEC data library.

No.	Channel name	Description
1	LLT	Used to communicate with the Ladder Logic Test Tool (LLT).
9	Shared device	Used to access a local station shared device (only when OS is Windows NT 4.0).
31 to 40	Comm. link (COM1 to COM10)	Used for access via the computer link module. Setting begins with 31 in ascending order of port numbers.
41 to 50	PLC RS-422 (COM1 to COM10)	Used for communication by direct connection to the PLC CPU. Setting begins with 41 in ascending order of port numbers.
51 to 54	MELSECNET/10 (1 slot to 4 slot)	Used for communication via the MELSECNET/10 card. Setting begins with 51 in ascending order of port numbers.
61	ETHERNET	Used for communication via the Ethernet module.
81 to 84	CC-Link (1 slot to 4 slot)	Used for communication via the CC-Link card. Set the BD No. DIP switches at the rear of the CC-Link card. Settings are made as follows by setting the switches SW1 and SW2. 81: OFF, OFF 82: ON, OFF 83: OFF, ON 84: ON, ON
91 to 100	CC-Link G4 (COM1 to COM10)	Used for communication via the CC-Link G4 Module. Setting begins with 91 in ascending order of port numbers.

**POINTS**

- A shared device (EM, ED) can be set under Windows NT 4.0 with the Shared Device Utility.  
It cannot be set under Windows 95 and Windows 98.  
However, the Windows NT 4.0 shared devices (EM, ED) can be accessed from Windows 95 or Windows 98.
- When a shared device (EM, ED) in the other personal computer (remote) is accessed, the MELSECNET/10 channel must be specified for the communication channel.

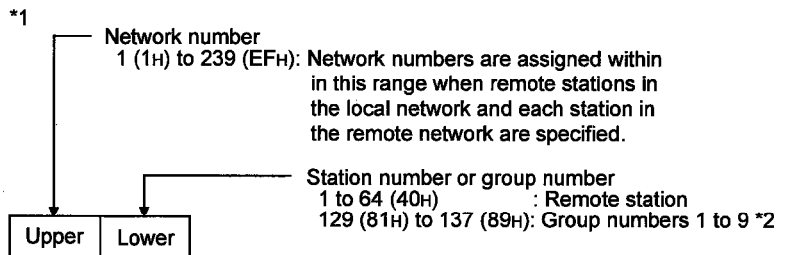




2.4 Station Number Setup

The table below shows how a station number is specified by functions.

Communication	Station number specification
Computer Link	Specifies a logical station number set by each utility.
Ethernet	
CC-Link G4	
Shared Device	Local station: 255(0xFF) Remote station: *1
RS-422	Local station: 255(0xFF) Remote station: MELSECNET(II)0 to 64(0x40) MELSECNET/10 *1
LLT	Local station: 255(0xFF)
MELSECNET/10	Local station: 255(0xFF) Remote station: *1
CC-Link	Local station: 255(0xFF) Remote station: 0 to 64(0x40) *3



\*2 Group numbers 1 to 9 (0x81 to 0x89) can be specified only when the SEND function (mdSend) function is used.

\*3 In CC-Link communication, the station number 64 cannot be specified for other than buffer memory access.

## 2.5 Device Type

Code numbers or device names can be specified as device types in the functions.

## (1) Common device type

Command device type for all communication paths

Device type			Device
by code		by device name	
Decimal	Hexadecimal		
1	1H	DevX	X
2	2H	DevY	Y
3	3H	DevL	L
4	4H	DevM	M
5	5H	DevSM	Special M(SM), SB (Link Special B for MNET/10)
6	6H	DevF	F
7	7H	DevTT	T (contact)
8	8H	DevTC	T (coil)
9	9H	DevCT	C (contact)
10	AH	DevCC	C (coil)
11	BH	DevTN	T (current value)
12	CH	DevCN	C (current value)
13	DH	DevD	D
14	EH	DevSD	Special D(SD), SW (Link Special W for MNET/10)
15	FH	DevTM	T (set value, main)
16	10H	DevTS	T (set value, sub 1)
16002	3E82 H	DevTS2	T (set value, sub 2)
16003	3E83 H	DevTS3	T (set value, sub 3)
17	11H	DevCM	C (set value, main)
18	12H	DevTS	C (set value, sub 1)
18002	4652 H	DevTS2	C (set value, sub 2)
18003	4653 H	DevTS3	C (set value, sub 3)
19	13H	DevA	A
20	14H	DevZ	Z
21	15H	DevV	V (index register)
22	16H	DevR	R (file register)
22001 to 22256	55F1H to 56F0 H	Dever1 to Dever256	ER(extended register)
23	17H	DevB	B
24	18H	DevW	W
25	19H	DevQSB	QnA link special relay (on QnACPU)

Device type			Device
by code		by device name	
Decimal	Hexadecimal		
26	1AH	DevSTT	Integrating timer (contact)
27	1BH	DevSTC	Integrating timer (coil)
28	1CH	DevQSW	QnA link special register (on QnACPU)
30	1EH	DevQV	QnA edge relay (on QnACPU)
33	21H	DevMRB	Local station random access buffer *1
35	23H	DevSTN	Integrating timer (current value)
36	24H	DevWw	Local station link register (for transmission) *1
37	25H	DevWr	Local link register (for reception) *1
40	28H	DevFS	S device of FXCPU
50	32H	DevSPB	Local station buffer memory *1
101	65H	DevMAIL	Acknowledging QnA SEND/RCV function arrival
102	66H	DevMAILNC	Not acknowledging QnA SEND/RCV function arrival
1001 to 1255	3E9H to 4E7H	DevLX1 to DevLX255	Direct link input
2001 to 2255	7D1H to 8CFH	DevLY1 to DevLY255	Direct link output
23001 to 23255	59D9H to 5AD7H	DevLB1 to DevLB255	Direct link relay
24001 to 24255	5DC1H to 5EBFH	DevLW1 to DevLW255	Direct link resistor
25001 to 25255	61A9H to 62A7H	DevLSB1 to DevLSB255	Direct link special resistor (network unit side)
28001 to 28255	6D61H to 6E5FH	DevLSW1 to DevLSW255	Direct link special resistor (network unit side)
29000 to 29255	7148H to 7247H	DevSPG0 to DevSPG255	Special direct buffer resistor
31000 to 31255	7918H to 7A17H	DevEM0 to DevEM255	EM (shared device)
32000 to 32255	7D00H to 7DFFH	DevED0 to DevED255	ED (shared device)

\*1 Dedicated device to access the buffer memory of a CC-Link card (local station).

## (2) Device type for access via the CC-Link card

This device type is a dedicated type for device access via the CC-Link card.  
It cannot be used when a device is accessed via the CC-Link card.

Device type			Device
by code		by device name	
Decimal	Hexadecimal		
1	1H	DevX	Local station RX
2	2H	DevY	Local station RY
5	5H	DevSM	Local station SB (link special B for CC-Link)
14	EH	DevSD	Local station SW (link special W for CC-Link)
33	21H	DevMRB	Local station random access buffer
36	24H	DevWw	Local station link register (for transmission)
37	25H	DevWr	Local station link register (for reception)
50	32H	DevSPB	Local station buffer memory
32768	8000 H	DevRBM	Remote station buffer memory *1
32800	8020 H	DevRAB	Remote station random access buffer *1
32801	8021 H	DevRX	Remote station RX *1
32802	8022 H	DevRY	Remote station RY *1
32804	8024 H	DevRW	Remote station link register *1
32867	8063 H	DevSB	Remote station SB (link special B for CC-Link) *1
32868	8064 H	DevSW	Remote station SW (link special W for CC-Link) *1

\*1 These devices cannot be used by the mdRandR, mdRandW, mdDevSet and mdDevRst functions.

### 2.6 Accessible Devices and Range

For details on accessible devices and ranges, see Chapter 11 in the CSKP Operating Manual.

## 3. MELSEC DATA LINK LIBRARY DETAILS

This chapter describes each function of the MELSEC Data Link Library in more detail. Functions beginning with the character string "mdBd " can be used only when the communication card is used.

In addition, the table at the upper right corner indicates whether the function can be used when the communication card is in use.

MNET/10	CC-Link
○	○

○: Can be used.

×: Cannot be used.

POINT
Only the functions described in Sections 3.1 to 3.11 can be used when communication does not use the communication card (computer link, Ethernet, and so on). The other functions cannot be used.

## 3.1 mdOpen (communication line open)

MNET/10	CC-Link
○	○

## (1) Function

This function opens a communication line.

## (2) Format

VB : ret% = mdOpen (chan&, mode%, path&)

Integer	ret	Returned value	OUT
Integer	chan	Communication line's channel number	IN
Integer	mode	Dummy (-1)	IN
Long	path	Open line bus pointer	OUT

VC++ : ret = mdOpen (chan, mode, path);

short	ret;	Returned value	OUT
short	chan;	Channel path	IN
short	mode;	Dummy (-1)	IN
long	*path;	Open line bus pointer	OUT

## (3) Explanation

- This function opens a communication channel through an initialization procedure appropriate for the channel.
- The ranges of the arguments are checked. If any error is found, the function returns an error code.

## (4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

## (5) Related function

mdClose()

## POINT

If the communication driver returns an error code, the function returns that error code as is.

3.2 mdClose (communication line close)

MNET/10	CC-Link
○	○

(1) Function

This function closes a communication line.

(2) Format

VB : ret% = mdClose (path&)

Integer	ret	Returned value	OUT
Long	path	Open line pointer	IN

VC++ : ret = mdClose (path);

short	ret;	Returned value	OUT
long	path;	Open line pointer	IN

(3) Explanation

This function closes an open channel in a way appropriate for the channel.

(4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

(5) Related function

mdOpen( )

<b>POINT</b>
If the communication driver returns an error code, the function returns that error code as is.



## 3.3 mdSend (batch writing to devices)

MNET/10	CC-Link
○	○

## (1) Function

This function performs batch writing to a device.

## (2) Format

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

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	stno	Station number (See Section 2.4.)	IN
Integer	devtyp	Device type	IN
Integer	devno	Head device number	IN
Integer	size	Size (bytes) of the data written	IN/OUT
Any	data(n)	Data written(single precision integer array)	IN

VC++ : ret = mdSend (path, stno, devtyp, devno, size, data);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Section 2.4.)	IN
short	devtyp;	Device type	IN
short	devno;	Head device number	IN
short	*size;	Size (bytes) of the data written	IN/OUT
short	data[ ];	Data written (single precision integer array)	IN

## (3) Explanation

- This function is used to write data to the specified device.
- The function checks the arguments.  
It also adds the head address and the size of the data written, both given by the arguments, to see if the sum of them indicates an address inside the memory allocated for the device.
- If the specified head address plus the size of the data written indicates an address outside the area in the memory allocated for the device, the function returns to the "size" field the maximum allowable size of the data written.

## (4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

## (5) Related functions

mdOpen( ), mdClose( )

POINTS
<ul style="list-style-type: none"><li>• If the communication driver returns an error code, the function returns that error code as is.</li><li>• When accessing the local station, specify 0xFF (255) as the station number. Specifying the actual station number will cause an error (-4).</li><li>• For bit device access, the leading device number must be specified as shown below.<ul style="list-style-type: none"><li>&lt;Specification by a multiple of 16 (0, 16, 32...)&gt;<ul style="list-style-type: none"><li>• During computer link communication</li><li>• Ethernet communication (TCP/IP)</li><li>• During access to CC-Link card (local station) RX, RY, and SB</li></ul></li><li>&lt;Specification by a multiple of 8 (0, 8, 16...)&gt;<ul style="list-style-type: none"><li>Other cases not covered above</li></ul></li></ul></li><li>• Be aware that writing data in a block to which an extended comment is assigned (extended file register) will erase the extended comment information.</li><li>• Be aware that wiring data in a block that overlaps with the Sub 2 and Sub 3 program setting areas (extended file register) will erase the Sub 2 and Sub 3 programs.</li></ul>

## 3.4 mdReceive (batch reading from devices)

MNET/10	CC-Link
○	○

## (1) Function

This function performs a batch reading from a device.

## (2) Format

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

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	stno	Station number (See Section 2.4.)	IN
Integer	devtyp	Device type	IN
Integer	devno	Head device number	IN
Integer	size	Size (bytes) of the data read	IN/OUT
Any	data(n)	Data read (single precision integer array)	OUT

VC++ : ret = mdReceive (path, stno, devtyp, devno, size, data);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Section 2.4.)	IN
short	devtyp;	Device type	IN
short	devno;	Head device number	IN
short	*size;	Size (bytes) of the data read	IN/OUT
short	data[ ];	Data read (single precision integer array)	OUT

## (3) Explanation

- This function is used to read data from the specified device.
- The function checks the arguments.  
It also adds the head address and the size of the data read, both given by the arguments, to see if the sum of them indicates an address inside the memory allocated for the device.
- If the specified head address plus the size of the data read indicates an address outside the area in the memory allocated for the device, the function returns to the "size" field the maximum allowable size of the data read.

## (4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

## (5) Related function

mdOpen( ), mdClose( )

**POINTS**

- If the communication driver returns an error code, the function returns that error code as is.
- When accessing the local station, specify 0xFF (255) as the station number. Specifying the actual station number will cause an error (-4).
- For bit device access, the leading device number must be specified as shown below.
  - <Specification by a multiple of 16 (0, 16, 32...)>
    - During computer link communication
    - Ethernet communication (TCP/IP)
    - During access to CC-Link card (local station) RX, RY, and SB
  - <Specification by a multiple of 8 (0, 8, 16...)>
  - Other cases not covered above

3.5 mdRandW (write to random devices)

MNET/10	CC-Link
○	○

(1) Function

This function is used to write data to randomly-specified devices.

(2) Format

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

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	stno	Station number (See Section 2.4.)	IN
Integer	dev(n)	Randomly-specified device	IN
Any	buf(n)	Data written	OUT
Integer	bufsize	Dummy	IN

VC++ : ret = mdRandW (path, stno, dev, buf, bufsize);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Section 2.4.)	IN
short	dev[ ];	Randomly-specified device	IN
short	buf[ ];	Data written	OUT
short	bufsize;	Dummy	IN

(3) Explanation

- This function is used to write data to randomly-specified devices.

Randomly-specified device (dev[ ])

dev[0].....	Number of blocks	} Block No. 1
dev[1].....	Device type	
dev[2].....	Head device number.	
dev[3].....	Number of points	
dev[4].....	Device type	} Block No. 2
.....	.....	

- The number of blocks should be specified with a number between 1 and 32767.

## (4) Specifying method

Example: Setting M100 through M115 to OFF and writing 10, 200, 300, and 400

Randomly-specified device

```
dev[0]=2; ..... Number of specified ranges(M100 to M115, D10 to D13)
dev[1]=DevM; } ..... M100 and after
dev[2]=100; }
dev[3]=16; ..... for the total of 16 items(M100 to M115)
dev[4]=DevD; } ..... D10 and after
dev[5]=10; }
dev[6]=4; ..... for the total of 4 items(D10 to D13)
```

Data written

```
buf[0]=0; ..... All bits are set to OFF.
buf[1]=10; ..... Stores 10 in D10.
buf[2]=200; ..... Stores 200 in D11.
buf[3]=300; ..... Stores 300 in D12.
buf[4]=400; ..... Stores 400 in D13.
```

## (5) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

## (6) Related function

mdOpen( ), mdClose( )

## POINTS

- If the communication driver returns an error code, the function returns that error code as is.
- When accessing the local station, specify 0xFF (255) as the station number. Specifying the actual station number will cause an error (-4).
- Be aware that writing data in a block to which an extended comment is assigned (extended file register) will erase the extended comment information.
- Be aware that writing data in a block that overlaps with the Sub 2 and Sub 3 program setting areas (extended file register) will erase the Sub 2 and Sub 3 programs.
- If an error happens when a random write operation is performed to "B" or "W" of the local station with the MELSECNET/10, the write operation to some of the blocks may be performed correctly while the write operation to the other blocks are performed incorrectly.

3.6 mdRandR (read from random devices)

MNET/10	CC-Link
○	○

(1) Function

This function is used for reading data from randomly-specified devices.

(2) Format

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

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	stno	Station number (See Section 2.4.)	IN
Integer	dev(n)	Randomly-specified device	IN
Any	buf(n)	Data read (single precision integer array)	OUT
Integer	bufsize	Data read storage area size (bytes)	IN

VC++ : ret = mdRandR (path, stno, dev, buf, bufsize);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Section 2.4.)	IN
short	dev[ ];	Randomly-specified device	IN
short	buf[ ];	Data read(single precision integer array)	OUT
short	bufsize;	Data read storage area size (bytes)	IN

(3) Explanation

- This function is used for reading data from randomly-specified devices.

Randomly-specified device (dev[ ])

dev[0].....	Number of blocks	}	Block No. 1
dev[1].....	Device type		
dev[2].....	Head device number.	}	Block No. 2
dev[3].....	Number of points		
dev[4].....	Device type	}	
.....	.....		

- The number of blocks should be specified with a number between 1 and 32767.

(4) Specifying method

Example: Reading current values of M100 to M115, D10 to D13, M0 to M13, and T10.

Randomly-specified device

dev[0]=4;..... Number of specified ranges  
(M100 to M115, D10 to D13, M0 to M13, T10)

dev[1]=DevM; } ..... M100 and after  
dev[2]=100; }

dev[3]=16;..... for the total of 16 items(M100 to M115)

dev[4]=DevD; } ..... D10 and after  
dev[5]=10; }

dev[6]=4;..... for the total of 4 items(D10 to D13)

dev[7]=DevM; } ..... M0 and after  
dev[8]=0; }

dev[9]=14;..... for the total of 14 items(M0 to M13)

dev[10]=DevTN; } ... T10 timer value, current value and after  
dev[11]=10; }

dev[12]=1;..... 1 item(T10)

bufsize value

(buf[0] to buf[6]=7)×2=14

Data read storage area



Data read

buf[0]=0;.....All bits between M100 and M113 are OFF.  
(16 pieces 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 between M0 and M13 are ON.

buf[6]=10;.....The current value of T10 is 10 (=1 sec).

(5) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

POINTS
<ul style="list-style-type: none"> <li>• If the communication driver returns an error code, the function returns that error code as is.</li> <li>• When accessing the local station, specify 0xFF (255) as the station number. Specifying the actual station number will cause an error (-4).</li> </ul>



## 3.7 mdDevSet (bit device set)

MNET/10	CC-Link
○	○

## (1) Function

This function sets a bit device.

## (2) Format

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

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	stno	Station number (See Section 2.4.)	IN
Integer	devtyp	Device type	IN
Integer	devno	Specified device number	IN

VC++ : ret = mdDevSet (path, stno, devtyp, devno);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Section 2.4.)	IN
short	devtyp;	Device type	IN
short	devno;	Specified device number	IN

## (3) Explanation

- This function sets (ON) the specified bit device.
- Arguments are checked.

## (4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

## (5) Related functions

mdOpen( ), mdClose( ), mdDevRst( )

POINTS
<ul style="list-style-type: none"> <li>• If the communication driver returns an error code, the function returns that error code as is.</li> <li>• When accessing the local station, specify 0xFF (255) as the station number. Specifying the actual station number will cause an error (-4).</li> </ul>

3.8 mdDevRst (bit device reset)

MNET/10	CC-Link
○	○

(1) Function

This function resets a bit device.

(2) Format

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

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	stno	Station number (See Section 2.4.)	IN
Integer	devtyp	Device type	IN
Integer	devno	Specified device number	IN

VC++ : ret = mdDevRst (path, stno, devtyp, devno);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Section 2.4.)	IN
short	devtyp;	Device type	IN
short	devno;	Specified device number	IN

(3) Explanation

- This function resets (OFF) the specified bit device.
- The arguments are checked.

(4) Returned value

Upon normal termination : The function returns 0.  
 Upon abnormal termination: The function returns a value other than 0.  
 (See Chapter 5.)

(5) Related functions

mdOpen( ), mdClose( ), mdDevSet( )

<b>POINTS</b>
<ul style="list-style-type: none"> <li>• If the communication driver returns an error code, the function returns that error code as is.</li> <li>• When accessing the local station, specify 0xFF (255) as the station number. Specifying the actual station number will cause an error (-4).</li> </ul>

3.9 mdTypeRead (PLC CPU type read)

MNET/10	CC-Link
○	○

(1) Function

This function reads information on the type of the PLC CPU.

(2) Format

VB : ret% = mdTypeRead (path&, stno%, buf%)

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	stno	Station number (See Section 2.4.)	IN
Integer	buf	Type code	OUT

VC++ : ret = mdTypeRead (path, stno, buf);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Section 2.4.)	IN
short	buf;	Type code	OUT

(3) Explanation

- This function reads information on the type of CPU in the specified PC.

PLC CPU type	Code	PLC CPU type	Code
A1N	A1H	Q2A, Q2AS, Q2ASH	21H
A2N, A2N-S1, A2S, A2S-S1, A1FX	A2H	Q2AS-S1, Q2A-S1, Q2ASH-S1	22H
A1SJ, A1SH, A1SJH, A1SJH-S8, A2SH, A2SH-S1, A3N	A3H	Q3A	23H
A2A, A2AS, A2AS-S1, A2AS-S30	92H	Q4A, Q4AR	24H
A2A-S1	93H	Personal computer	90H
A3A	94H	QnA *1	20H
A0J2H, A1S, A1S-S1, A1SJ, A1SJ-S3	98H	FX0, FX0s series	F0H
A2C, A2CJ	9AH	FX0N series	8EH
A2U	82H	FX1 series	F1H
A2U-S1	83H	FX2, FX2c series	8DH
A3U	84H	FX2N, FX2NC series	9EH
A4U	85H		

\*1 This code is returned only when this function is executed via a QE71 unit.

When the communication is made through a QE71 unit, more detailed information on the type of the QnA CPU is not available.

(4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

POINT	
If the communication driver returns an error code, the function returns that error code as is.	

3.10 mdControl (remote RUN/STOP/PAUSE)

MNET/10	CC-Link
○	○

(1) Function

This function performs a remote RUN, STOP, or PAUSE.

(2) Format

VB : ret% = mdControl (path&, stno%, buf%)

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	stno	Station number (See Section 2.4.)	IN
Integer	buf	Specified code	IN

VC++ : ret = mdControl (path, stno, buf);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Section 2.4.)	IN
short	buf;	Specified code	IN

(3) Explanation

- This function performs a remote RUN/STOP/PAUSE over the programmable controller CPU.
- The specified codes are as follows:

Instruction	Specified code
Remote RUN	0
Remote STOP	1
Remote PAUSE	2

(4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

POINTS
<ul style="list-style-type: none"> <li>• If the communication driver returns an error code, the function returns that error code as is.</li> <li>• When the access target CPU is any of the FX series other than the FX2N and FX2NC, you cannot use the mdControl function.</li> <li>• In any of the following cases, an error is returned since remote PAUSE is not supported.                             <ul style="list-style-type: none"> <li>• Ethernet communication</li> <li>• Computer link communication</li> <li>• The access target is any of the FX2N series, FX2NC series and Ladder Logic Test Tool (LLT).</li> </ul> </li> <li>• When TCP/IP is used for protocol, the mdControl function for E71 causes an error because remote run/stop to the local station is not supported.</li> </ul>

## 3.11 mdlnit (PLC device address table refreshing)

MNET/10	CC-Link
○	○

## (1) Function

This function refreshes the PLC device address table.

## (2) Format

VB : ret% = mdlnit (path&)

Integer	ret	Returned value	OUT
Long	path	Channel path	IN

VC++ : ret = mdlnit (path);

short	ret;	Returned value	OUT
long	path;	Channel path	IN

## (3) Explanation

This function refreshes the PLC device address table (MELSEC data link library internal data).

## (4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

3.12 mdBdRst (local card reset)

MNET/10	CC-Link
○	○

(1) Function

This function resets the local card.

(2) Format

VB : ret% = mdBdRst(path&)

Integer	ret	Returned value	OUT
Long	path	Channel path	IN

VC++ : ret = mdBdRst(path);

short	ret;	Returned value	OUT
long	path;	Channel path	IN

(3) Explanation

A local card is reset, and then data is read out from the register and set in the card again.

(4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

<b>POINTS</b>
<ul style="list-style-type: none"> <li>• If the communication driver returns an error code, the function returns that error code as is.</li> <li>• To any other application program that was accessing a remote station through the card reset, the function returns the card reset error code (9922H).</li> <li>• The MELSECNET/10 or CC-Link driver releases the SEND data transmitted from a remote station, then clears the buffer.</li> <li>• After being reset and until a mode is set, the local card is disconnected from data links with networks.</li> </ul>

3.13 mdBdModSet (local card mode set)

MNET/10	CC-Link
○	○

(1) Function

This function sets a mode for the local card.

(2) Format

VB : ret% = mdBdModSet (path&, mode%)

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	mode	Mode	IN

VC++: ret = mdBdModSet (path, mode);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	mode;	Mode	IN

(3) Explanation

This function resets and then sets a mode for the local card.

(a) Mode:

MELSECNET/10	CC-Link
<ul style="list-style-type: none"> <li>• Link mode</li> <li>0: on-line (with automatic return)</li> <li>2: off-line</li> <li>3: forward loop test</li> <li>4: reverse loop test</li> <li>5: station-to-station test (master stations)</li> <li>6: station-to-station test (slave stations)</li> <li>7: self-loopback test</li> <li>8: self-loopback test (internal)</li> <li>9: hardware test</li> </ul>	<ul style="list-style-type: none"> <li>• Link mode</li> <li>0: on-line (with automatic return)</li> <li>2: off-line</li> <li>3: Data link test</li> <li>4: Remote station test</li> <li>5: Setting parameter check mode</li> <li>6: Single module H/W operation check</li> <li>7: Setting not possible</li> </ul>

(4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

(5) Related function

mdBdModRead( )

<b>POINT</b>
If the communication driver returns an error code, the function returns that error code as is.



3.14 mdBdModRead (local card mode read)

MNET/10	CC-Link
○	○

(1) Function

This function reads the mode set to the local card.

(2) Format

VB : ret% = mdBdModRead (path&, mode%)

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	mode	Mode	OUT

VC++ : ret = mdBdModRead (path, mode);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	*mode;	Mode	OUT

(3) Explanation

This function reads the registry to find out the mode set to the local card.

(a) Mode:

MELSECNET/10	CC-Link
<ul style="list-style-type: none"> <li>• Link mode</li> <li>0: on-line (with automatic return)</li> <li>2: off-line</li> <li>3: forward loop test</li> <li>4: reverse loop test</li> <li>5: station-to-station test (master stations)</li> <li>6: station-to-station test (slave stations)</li> <li>7: self-loopback test</li> <li>8: self-loopback test (internal)</li> <li>9: hardware test</li> </ul>	<ul style="list-style-type: none"> <li>• Link mode</li> <li>0: on-line (with automatic return)</li> <li>2: off-line</li> <li>3: Data link test</li> <li>4: Remote station test</li> <li>5: Setting parameter check mode</li> <li>6: Single module H/W operation check</li> <li>7: Setting not possible</li> </ul>

(4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

(5) Related function

mdBdModSet( )

<b>POINT</b>
If the communication driver returns an error code, the function returns that error code as is.

## 3.15 mdBdLedRead (local card LED read)

MNET/10	CC-Link
○	○

## (1) Function

This function reads the states of the LEDs on the local card.

## (2) Format

VB : ret% = mdBdLedRead (path&, buf%(0))

Integer	ret	Returned value	OUT
Long	path	Open line pointer	IN
Integer	buf(n)	Data read	OUT

VC++ : ret = mdBdLedRead (path, buf);

short	ret;	Returned value	OUT
long	path;	Open line pointer	IN
short	buf[ ];	Data read	OUT

## (3) Explanation

This function reads the states (lit/unlit) of the LEDs on the local card.

(a) MELSECNET/10 card

	Bit	Description	Remark
buf[0]	0	UNDER RUN forward loop	Lit: 0 Unlit: 1
	1	DATA forward loop	
	2	TIME forward loop	
	3	ABORT.IN-FR forward loop	
	4	OVER RUN forward loop	
	5	CRC forward loop	
	6	PRM error	
	7	M/S error	
	8	UNDER RUN reverse loop	
	9	DATA reverse loop	
	10	TIME reverse loop	
	11	ABORT.IN-FR reverse loop	
	12	OVER RUN reverse loop	
	13	CRC reverse loop	
	14	F.LOOP	
	15	R.LOOP	
buf[1]	0	T.PASS	Lit: 0 Unlit: 1
	1	D.LINK	
	2	S.MNG	
	3	MNG	
	4	SW error	
	5	DUAL	
	6	REMORT	
	7	PC	
	8	Disconnection in forward loop detected	
	9	Forward loop forcible error detected	
	10	Forward loop luminous energy alarm signal	
	11	Always 1	
	12	Disconnection in reverse loop detected	
	13	Reverse loop forced error detected	
	14	Reverse loop luminous energy alarm signal	
	15	Always 1	
buf[2]		Card operation status	Card abnormal: 0 Card operating: 1

(b) CC-Link card

	Bit	Description	Remark
buf[0]	0	RUN (Module normal run/error)	Lit: 1  Unlit: 0
	8	ERR (Communication error)	
buf[1]	0	CPU RW (During communication)	
	8	SW (Switch setting error)	
buf[2]	0	M/S (Master station duplicate error)	
	8	PRM (Parameter error)	
buf[3]	0	TIME (Time over)	
	8	LINE (Cable disconnection error)	
buf[4]	0	L.RUN (During data link)	
	8	L.ERR (Data link error)	
buf[5]	0	SD (During data transmission)	
	8	RD (During data reception)	

(4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

<b>POINT</b>
If the communication driver returns an error code, the function returns that error code as is.

3.16 mdBdSwRead (local card switch read)

MNET/10	CC-Link
○	○

(1) Function

This function reads the states of switches on the local card.

(2) Format

VB : ret% = mdBdSwRead (path&, buf%(0))

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	buf(n)	Data read	OUT

VC++ : ret = mdBdSwRead (path, buf);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	buf[ ];	Data read	OUT

(3) Explanation

This function gives the information on the following by reading the states of switches on the local card: station number, card number, card identification, and I/O address.

(a) MELSECNET/10 and CC-Link card

	Switch values	Range	
		MELSECNET/10	CC-Link
buf[0]	Value set to station number switch	1 to 64	
buf[1]	Value set to group number switch	0 to 9	Fixed to 0
buf[2]	Value set to network number switch	1 to 239	Fixed to 0
buf[3]	Value set to card number switch	*1	Fixed to 0
buf[4]	Value set to card identification switch	0 to 7	Fixed to 0
buf[5]	Value set to I/O address switch	Personal Computer *2	Fixed to 0

\*1 Value of upper two digits in 2-port address  
e.g. "C8H" for C800H

\*2 Value of I/O port address

(4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

<b>POINT</b>
If the communication driver returns an error code, the function returns that error code as is.

3.17 mdBdVerRead (local card version read)

MNET/10	CC-Link
○	○

(1) Function

This function reads information on the version of the local card.

(2) Format

VB : ret% = mdBdVerRead (path&, buf%(0))

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	buf(n)	Data read	OUT

VC++ : ret = mdBdVerRead (path, buf);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	buf[ ];	Data read	OUT

(3) Explanation

This function reads information on the version of the local card.

Data read	Description	Details	
		MELSECNET/10	CC-Link
00H, 01H	Password	Fixed to S or G	
02H, 03H	Check sum	Total from 04H to 1FH	
04H, 05H	Software version	'A' to 'ZZ'	
06H to 0BH	Date (year, month, day)	(Example) July 10, 1998 → '9' '8' '0' '7' '1' '0'	
0CH to 0FH	Reserved area (4 bytes)	—	
10H to 1FH	Software type (16 bytes)	"J71QLP23" "J71QBR13"	"J61BT13"
20H to 2FH	Hardware type (16 bytes)	"A70BD-J71QLP23" "A70BD-J71QBR13"	"A80BD-J61BT13"
30H, 31H	2-port memories occupied size	1000H (4K) bytes	4000H (16K) bytes
32H, 33H	2-port attribute	0080H fixed *1	
34H, 35H	Available offset	0000H fixed	
36H to 3FH	Hardware classification (10-bytes)	Personal computer: "A70BD", "A80BD"	

\*1 0080H: complete 2-port type

(4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

<b>POINT</b>
If the communication driver returns an error code, the function returns that error code as is.

3.18 mdSend (SEND function)

MNET/10	CC-Link
○	○

(1) Function

This function is used to send data.

(2) Format

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

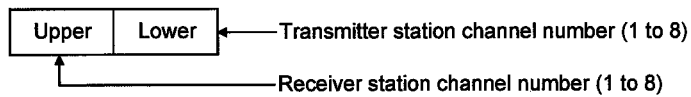
Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	stno	Station number (See Section 2.4.)	IN
Integer	devtyp	Device type	IN
Integer	devno	Channel number	IN
Integer	size	Size (1 to 960 bytes) of the data sent	IN/OUT
Any	data(n)	Data written (single precision integer array)	IN

VC++ : ret = mdSend (path, stno, devtyp, devno, size, data);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Section 2.4.)	IN
short	devtyp;	Device type	IN
short	devno;	Channel number	IN
short	*size;	Size (1 to 960 bytes) of the data sent	IN/OUT
short	data[ ];	Data written (single precision integer array)	IN

(3) Explanation

- This function supports the RECV instruction, an exclusive instruction for the QnA MELSECNET/10 network system.
- As the device type, specify 101 for "arrival acknowledgement required" or 102 for "arrival acknowledgement not required".
- Specify the channel number as follows.



## (4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.

(See Chapter 5.)

## (5) Related functions

mdOpen( ), mdClose( )

POINTS
<ul style="list-style-type: none"><li>• If the communication driver returns an error code, the function returns that error code as is.</li><li>• An error will occur if a transmission is instructed specifying a channel currently in use.</li><li>• The size (number of bytes) of the data sent should be an even number.</li></ul>



3.19 mdReceive (RECV function)

MNET/10	CC-Link
○	○

(1) Function

This function is used to receive data.

(2) Format

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

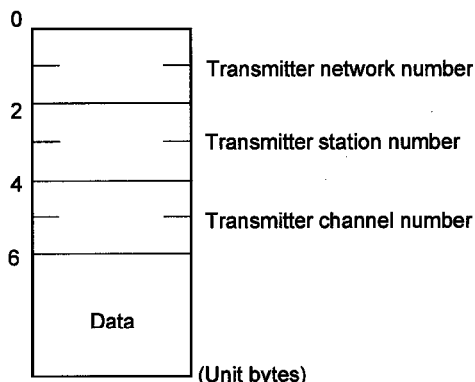
Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	stno	Station number (See Section 2.4.)	IN
Integer	devtyp	Device type	IN
Integer	devno	Channel number	IN
Integer	size	Size (bytes) of the data received	IN/OUT
Any	data(n)	Data received (single precision integer array)	OUT

VC++ : ret = mdReceive (path, stno, devtyp, size, data);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Section 2.4.)	IN
short	devtyp;	Device type	IN
short	devno;	Channel number	IN
short	*size;	Size (bytes) of the data received	IN/OUT
short	data[ ];	Data received (single precision integer array)	OUT

(3) Explanation

- This function supports the RECV instruction, an exclusive instruction for the QnA MELSECNET/10 network system.
- Specify 101 as the device type.
- Specify FFh as the station number.
- Specify the channel number as follows.  
number for the channel used for receiving data (1 to 8)
- The data received is stored in the receiving buffer as follows.



## (4) Returned value

Upon normal termination : The function returns 0.

Upon abnormal termination: The function returns a value other than 0.  
(See Chapter 5.)

## (5) Related functions

mdOpen( ), mdClose( )

POINTS
--------

- |  |
|--|
| <ul style="list-style-type: none"><li>• If the communication driver returns an error code, the function returns that error code as is.</li><li>• Up to 128 pieces of received data can be stored [the total for all receiving channels (No.1 to 8)].<br/>Any further data cannot be recorded in the receive data buffer.</li></ul> |
|--|



## 4, SAMPLE PROGRAM

This chapter provides sample programs in the VB and VC+ +.

## 4.1 Visual Basic Sample Program

This is a sample program to read the data register (D) of PLC CPU through a computer link channel.

This sample program was created using Visual Basic 4.0 (32-bit version).

## (1) How to use

When a form is loaded, mdOpen is implemented through the computer link channel.

When you click the "Read Test" button on the display, mdReceive executes, and when it succeeds in a readout, the readout data is indicated in the Read Data indication column.

When the "End" button is clicked, mdClose executes to close the test program. If an error occurs when a function is executed, an error message with an error code is indicated.

Eliminate the error cause, consulting Chapter 5.

If an error occurs when the "Read Test" button is clicked, clear the Read Data indication column.

Before executing this test program, assign the information to the logic code 0 using the computer link utility (See CSKP Operating Manual).

In addition, carry out testing after confirming that the communication is in a normal condition.

## (2) Listing of sample files

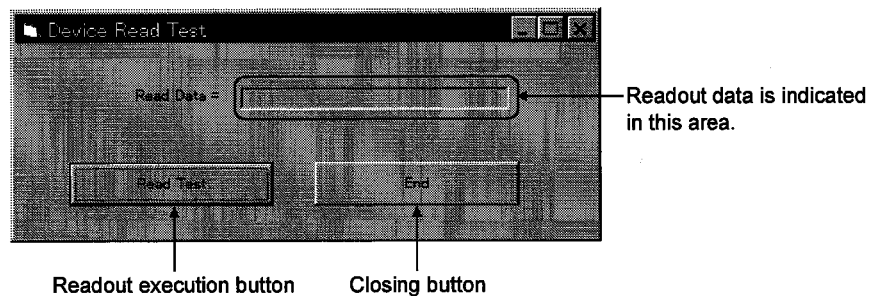
Sample files are installed in the following folders by default:

C:\MELSEC\COMMON\SAMPLE\VB\DemoPro. vbp    Project file  
 C:\MELSEC\COMMON\SAMPLE\VB\Demo. frm        Source file  
 C:\MELSEC\COMMON\SAMPLE\VB\Mdfunc. bas    Header file

In addition, when an Ethernet channel is used, a sample file of the ladder program for the sequencer CPU for mounting an Ethernet unit is installed below the C:\MELSEC\COMMON\SAMPLE\GPP\ folder.

(For further details, see Ethernet Unit Manual)

## (3) Screen



## (4) Program

```

*****
' MELSEC Communication Function Test Program
*****

'Definition
** you replace the comment for computer link and
** use the comment for other communications.
** You change the logical station number.
** This program reads the data for device type D0.
** If you want to execute read for other device,
** please refer the programming manual and change the device name,device number.
'Computer link(COM1)
Const CHAN_UC24QC24 = 31      'channel number for Computer link(COM1)
Const MODE_DUMMY = -1      'mode(dummy) (select -1)
Const STNO_LOGIC = 0       'logical station number(0)
Const DEVTYPE_D = 13      'device name (D)
Const DEVNO_0 = 0         'device number
Const SIZE_2_BYTES = 2    '2 bytes

** If you want to test the program for Ethernet Communication,
** you replace the comment for Ethernet Communication and
** use the comment for other communications.
** You change the logical station number.
** This program reads the data for device type D0.
** If you want to execute read for other device,
** please refer the programming manual and change the device name,device number.
'Ethernet Communication
'Const CHAN_ETHERNET = 61    'channel number for Ethernet Communication
'Const MODE_DUMMY = -1     'mode(dummy) (select -1)
'Const STNO_LOGIC = 0      'logical station number(0)
'Const DEVTYPE_D = 13     'device name (D)
'Const DEVNO_0 = 0        'device number
'Const SIZE_2_BYTES = 2   '2 bytes

```

\*\*\* If you want to test the program for PLC RS-422 Communication,  
\*\*\* you replace the comment for PLC RS-422 Communication and  
\*\*\* use the comment for other communications.  
\*\*\* You change the network number and the station number.  
\*\*\* This program reads the data for device type D0.  
\*\*\* If you want to execute read for other device,  
\*\*\* please refer the programming manual and change the device name,device number.

'PLC RS-422 Communication(COM1)  
'Const CHAN\_RS422 = 41       'channel number for PLC RS-422 Communication  
'Const MODE\_DUMMY = -1       'mode(dummy) (select -1)  
'Const STNO\_SELFSTATION = 255 'local station number(network number and station number,or stationnumber)  
'Const DEVTYPE\_D = 13       'device name (D)  
'Const DEVNO\_0 = 0       'device number  
'Const SIZE\_2\_BYTES = 2       '2 bytes

\*\*\* If you want to test the program for MELSECNET(II) Communication,  
\*\*\* you replace the comment for MELSECNET(II) Communication and  
\*\*\* use the comment for other communications.  
\*\*\* You change the station number.  
\*\*\* This program reads the data for device type W0.  
\*\*\* If you want to execute read for other device,  
\*\*\* please refer the programming manual and change the device name,device number.

'MELSECNET(II) Communication  
'Const CHAN\_MNET2 = 21       'channel number for MELSECNET(II) Communication  
'Const MODE\_DUMMY = -1       'mode(dummy) (select -1)  
'Const STNO\_SELFSTATION = 255 'local station number  
'Const DEVTYPE\_W = 24       'device name (W)  
'Const DEVNO\_0 = 0       'device number  
'Const SIZE\_2\_BYTES = 2       '2 bytes

\*\*\* If you want to test the program for MELSECNET/10 Communication,  
\*\*\* you replace the comment for MELSECNET/10 Communication and  
\*\*\* use the comment for other communications.  
\*\*\* You change the network number and the station number.  
\*\*\* This program reads the data for device type W0.  
\*\*\* If you want to execute read for other device,  
\*\*\* please refer the programming manual and change the device name,device number.

'MELSECNET/10 Communication  
'Const CHAN\_MNET10 = 51       'channel number for MELSECNET/10 Communication  
'Const MODE\_DUMMY = -1       'mode(dummy) (select -1)  
'Const STNO\_SELFSTATION = 255 'local station number(network number and station number)  
'Const DEVTYPE\_W = 24       'device name (W)  
'Const DEVNO\_0 = 0       'device number  
'Const SIZE\_2\_BYTES = 2       '2 bytes

```

*** If you want to test the program for Shared device Communication,
*** you replace the comment for Shared device Communication and
*** use the comment for other communications.
*** You change the network number and the station number.
*** This program reads the data for device type ED0.
*** If you want to execute read for other device,
*** please refer the programming manual and change the device name,device number.
'Shared device Communication
'Const CHAN_EMED = 9      'channel number for Shared device Communication
'Const MODE_DUMMY = -1  'mode(dummy) (select -1)
'Const STNO_SELFSTATION = 255 'local station number(network number and station number,or stationnumber)
'Const DEVTYPE_ED = 32000 'device name (ED0)
'Const DEVNO_0 = 0      'device number
'Const SIZE_2_BYTES = 2 '2 bytew

'common parameter
Dim Path As Long      'open loop path pointer
Dim buf(128) As Integer 'read dat (single precision)

Private Sub EndBtn_Click()
  'local parameter
  Dim Ret As Integer  'return value

  'close the channel
  Ret = mdClose(Path)
  If (Ret <> 0) Then
    MsgBox "Channel Close Error : " & "Error Code = " & Ret
  End If
End

End Sub

Private Sub Form_Load()
  'local parameter
  Dim Chan As Integer  'channel number
  Dim Mode As Integer  'mode
  Dim Ret As Integer   'return value

  'open the channel
  *** open the computer link,
  *** If you want to execute open for other communication,
  *** please change the channel number.
  Chan = CHAN_UC24QC24
  Mode = MODE_DUMMY

```

```
Ret = mdOpen(Chan, Mode, Path)
If (Ret <> 0) And (Ret <> 66) Then
    MsgBox "Channel Open Error : " & "Error Code = " & Ret
End
End If

End Sub
```

```
Private Sub ReadBtn_Click()
    'local parameter
    Dim Stno As Integer      'station number
    Dim Devtyp As Integer   'device name
    Dim devno As Integer    'device number
    Dim size As Integer     'date size (bytes)
    Dim Ret As Integer      'returan value

    'read the channel
    Stno = STNO_LOGIC
    Devtyp = DEVTYPE_D
    devno = DEVNO_0
    size = SIZE_2_BYTES

    Ret = mdReceive(Path, Stno, Devtyp, devno, size, buf(0))
    If Ret = 0 Then
        'show data
        RdDat.Caption = buf(0)
    Else
        MsgBox "Read Error : " & "Error Code = " & Ret
        'clear
        RdDat.Caption = " "
    End If

End Sub
```

- (5) In case it is used through other channels  
Change to the channel and the information on the device for testing the constant-declaration area, and execute after changing an argument value to a function in the program.



## 4.2 Visual C++ Sample Program

This sample program is capable of reading the data register of PLC CPU in an Ethernet channel.

## (1) How to use

This sample program operates by creating the execution module with VC++. When the program is executed, it implements mdOpen through the Ethernet channel.

After that, the program is implemented, and the readout data is indicated in the screen if the readout is successful.

Then, mdClose executes to close the program.

If an error occurs when a function is implemented, an error message with an error code is indicated.

Eliminate the error cause, consulting Chapter 5.

Before executing this test program, assign information to the logical station number 0 with the Ethernet utility (refer to the CSKP Operating Manual).

Also, confirm in advance that communication condition is normal.

## (2) Listing of sample files

Sample files are installed in the following folder by default:

C:\MELSEC\COMMON\SAMPLE\VC\SmpE71.C Source file

In addition, when an Ethernet channel is used, a sample file of the ladder program for the sequencer CPU for mounting an Ethernet unit is installed below the C:\MELSEC\COMMON\SAMPLE\VB\GPP\ folder.

(For further details, see Ethernet Unit Manual)

## (3) Screen

A sample output is indicated when a sample program is implemented.

Readout data is indicated as shown in the following screen.



## (4) Program

```

/*****
* MELSEC Communication Function Test Program          *
* <FileName>SmpE71.c                                *
*                                                    *
* This program is sample for Ethernet communication. *
*                                                    *
*****/

/*****
*          Include                                  *
*****/
#include <stdio.h>
#include <windows.h>                                /* Windows          */
#include "mdfunc.h"                                /* MELSEC Data Link Library */

/*****
*          Definition                              *
*****/
#define      CHSN_ETHERNET      61                /* channel number for Ethernet communication */
#define      MODE_DUMMY         -1                /* mode(dummy) (select -1) */
#define      STNO_ROGIC         0                 /* logical station number */
#define      DEVTYPE_D          13                /* device name (D) */
#define      DEVNO_0            0                 /* device number */
#define      SIZE_R_BYTES      4                 /* bytes for read data */
#define      DATA_INITIAL     0                 /* initial data for read eria*/

/*****
* You can read PLC data that logical station number is 0.
* You have to set the logical station number.
* The logical station number should be set in the Ethernet Utility.
*****/

void main()
{
    /* mdopen parameter */
    long   path;                /* opened loop path pointer */
    short  chan;                /* channel number for selected communication*/
    short  mode;                /* mode (dummy) (select -1) */
    short  oret;                /* return value from function of mdopen*/

```

```

/* mdreceive parameter */
short  stno;           /* station number */
short  devtyp;        /* device name */
short  devno;        /* device number */
short  size;         /* data size (bytes) */
/*
short  data[2];      /* read data (single precision) */
short  rret;        /* return value from function for mdreceive */

/* mdclose parameter */
short  cret;        /* return value from function for mdclose */

/* set mdopen parameter */
chan   = CHSN_ETHERNET; /* channel number for Ethernet communication */
mode   = MODE_DUMMY;    /* mode (dummy) */
/* open the channel */
oret = mdOpen( chan, mode, &path );
if( oret != 0 ){
    /* In case of error, it responds the error. */
    printf( "mdopen error[%04x]\n", oret );
}else{

    /* In case of success, it resumes next. */
    /* set mdreceive parameter */
    stno   = STNO_ROGIC; /* logical station number */
    devtyp = DEVTYPE_D; /* device name (D) */
    devno  = DEVNO_0;    /* device number 0 and 1 */
    size   = SIZE_R_BYTES; /* 4 bytes */
    data[0] = DATA_INITIAL; /* Initialize */
    data[1] = DATA_INITIAL; /* Initialize */
    /* It reads the data for selected device type. */
    rret = mdreceive( path, stno, devtyp, devno, &size, &data );
    if( rret != 0 ){
        /* In case of error, it responds the error. */
        printf( "mdreceive error[%04x]\n", rret );
    }else{
        /* show data that is received from PLC */
        printf( "data0 [%04x]\n", data[0] ); /* show data No.1 */
        printf( "data1 [%04x]\n", data[1] ); /* show data No.2 */
    }
}

```

```
/* open the channel          */
cret = mdclose( path );
if( cret != 0 ){
    /* In case of error, it responds the error. */
    printf( "mdclose error[%04x] \n", cret );
}
}
}
```

(5) In case it is used through other channels

Change to the channel and the information on the device for testing the constant-declaration area, and execute after changing an argument value to a function in the program.



## 5. ERROR CODES

This chapter summarizes the error codes issued when the MELSEC Data Link Library is used.

## POINT

When the return code issued is not listed in the following table, refer to the error code list included in the CPU, module or interface (I/F) card manual.

Error code A value enclosed by parentheses is hexadecimal.	Error description	Action to be taken
1	Driver not started Driver not started yet. Interrupt number or I/O address identical to that in the other card.	Correct the error at the start of driver. Check the settings on the card.
2	Card reply error Time-out in waiting for a replay to be made for processing.	Check whether the card has been mounted firmly. Check whether the destination (for communication) has been set correctly with the utility.
65 (41)	Channel error The specified channel number has not been registered yet.	Specify a correct channel number.
66 (42)	OPEN error A specified channel has already been opened.	Open the channel only once.
67 (43)	CLOSE error A specified channel has already been closed.	Close the channel only once.
68 (44)	PATH error A line has not been opened yet for a specified path.	Specify a number used in opening the path.
69 (45)	Processing code error A processing code not supported has been issued.	Use a processing code supported.
70 (46)	Station number specification error A specified station number is not correct. A processing request for a remote station has been made for a local station or a station number is for the local station (0xFF) but the network number is not 0.	Correct the station number of an application program.
71 (47)	Received data error (at RECV request) Data has not been received yet.	Wait for data reception.
72 (48)	Mode setting wait No mode has been set yet.	Set the mode.

Error code A value enclosed by parentheses is hexadecimal.	Error description	Action to be taken
73 (49)	Mode error A request has been issued to a remote station even though the set mode is not ON-LINE.	Set the mode to ON-LINE or cancel the request.
	Interrupt number error An interrupt number is the same as that of the other card.	Check the settings for the card.
	I/O address error An I/O address is the same as that of the other card.	
77 (4D)	Memory reservation error Memory cannot be reserved.	Because the cause may be memory shortage, close other applications. Check that the system is operating normally. Restart the system.
78 (4E)	Mode setting time-out error Though mode setting has been attempted, the mode could not be set for time-out.	Check if 2-port memory is used by the other card, and then restart the system. When the same error recurs, a hardware (H/W) error is assumed.
80 (50)	Failed to map a shared memory address	Check whether a shared memory address is the same as that of the other card.
85 (55)	Channel number error (at RECV request) Channel number error	Check a channel number when a RECV request has been issued.
100 (64)	A local station card is being accessed. An access request has been issue for a local station being accessed.	Retry.
101 (65)	Routing parameter error No routing parameter has been specified.	Correct the routing parameter.
102 (66)	Data transmission error Data transmission failed.	Retry. Confirm that the system is operating normally. Restart the system.
103 (67)	Data reception error Data reception failed.	
128 (80)	Read byte error The specified number of bytes to be read is not within a range.	Specify the number of bytes within a range.
129 (81)	Device type error The specified device type is invalid.	Check the device type.
130 (82)	Device number error A specified device number is not within a range. A device number is not a multiple of 8 when a bit device is specified.	Check the device number.
131 (83)	Device count error A specified count is not within a device range. A count is not a multiple of 8 when a bit device is specified.	Check the size.

Error code A value enclosed by parentheses is hexadecimal.	Error description	Action to be taken
132 (84)	Write byte error The specified number of bytes to be written is not within a range.	Specify the number of bytes within a range.
133 (85)	Link parameter error The link parameter has been damaged. The total number of substations specified in the link parameter has been reset to 0.	Specify the link parameter again.
135 (87)	Remote RUN/STOP/PAUSE specification error A specified RUN, STOP or PAUSE value is not a value from 0 to 2.	Specify a value from 0 to 2.
136 (88)	Random write specification error A specified random write value is not a value from 0 to 2.	
137 (89)	Processing cancelled The next processing request has been issued before the preceding processing is complete.	Terminate the preceding processing, then make a request again.
210 (D2)	Disabled in the RUN state A sequence P shift request has been issued in the RUN state.	Stop the CPU, then make a request again.
212 (D4)	Processing cancelled A new request has been issued in the RUN state.	Review the request.
215 (D7)	Received data length error The received data length or received byte length is not within a range.	Retry. Check the cable.
	Request data buffer length over The length of requested data exceeded a request data area.	Reduce the request data size.
216 (D8)	Protocol error The protocol is invalid. There is no request code.	Check the cable.
217 (D9)	Address error The address is not within an access range.	Check the request data.
219 (DB)	Write error No data can be written.	
224 (E0)	PCNO. error There is no requested destination (station).	Correct the station number.
225 (E1)	Processing mode error A requesting ACPU has issued a processing code that cannot be handled. (The requesting ACPU checks the processing code.)	Review the processing code of the requesting ACPU.



Error code A value enclosed by parentheses is hexadecimal.	Error description	Action to be taken
226 (E2)	Special module specification error A specified special module is not a module that can be handled.	Correct the YNO.
227 (E3)	Other data error Data (such as request data address, leading step, number of shifts, etc.) is invalid.	Correct the request data.
228 (E4)	Link specification error A requesting station has issued a processing code that cannot be handled. (The requesting link module checks the processing code.)	Check the requesting station number and processing code.
232 (E8)	Remote error A keyword specified in the remote RUN/STOP/PAUSE request is not consistent.	Searches for a source that performed a remote stop/pause operation to a destination ACPU.
233 (E9)	Link time-over A requesting station has disconnected a link during processing.	Restore the link.
234 (EA)	Special module BUSY In general data transmission, the receive buffer of a destination is full or it is not ready for data reception.	Check the special module hardware.
236 (EC)	Destination BUSY In general data transmission, the receive buffer of a destination is full or it is not ready for data reception.	Make a request when the destination is ready for data reception.
240 (F0)	Link error A request has been issued to a station whose link has been disconnected.	Restore the link.
241 (F1)	Special module bus error A specified special module is not ready for processing.	Check the special module hardware.
242 (F2)	Special module time-over A specified special module does not reply.	Check the special module hardware.
1280 (500)	Local memory access error in the local card	Check the switch settings on the local card, and then change the memory address to an address range (area) which is not influenced by the other cards. When 8-bit memory access has been set, change it to 16-bit memory access.
1281 (501)	I/O card access disabled	Check the I/O port address. Conduct the self loop-back test of the card to check the hardware.

Error code A value enclosed by parentheses is hexadecimal.	Error description	Action to be taken
16432 (4030)	The specified device type does not exist.	Check the specified device type.
16433 (4031)	A specified device number is not within a range.	Check the specified device number.
16448 (4040)	A specified module does not exist.	Don't issue an error request to a specified special module.
16449 (4041)	The number of device points is not within a range.	Check the leading address and number of access points for access within a range.
16450 (4042)	A target module has a fault.	Check whether the specified module is operating normally.
16451 (4043)	A module does not exist in the specified position.	Check the leading I/O number of a specified module.
40577 (9E81)	Device type error A device type for a requesting station is invalid. (The requesting station checks the device type.)	Check the device type.
40578 (9E82)	Device number error A device number specified for a requesting station is not within a range. When a bit device is specified, the device number is not a multiple of 8. (The requesting station checks the number of points.)	Check the device number.
40579 (9E83)	Device point count error The number of device points specified for a requesting station is not within a range. When a bit device is specified, the number of device points is not a multiple of 8. (The requesting station checks the number of device points.)	Check the size.
-1 (FFFF)	Path error A specified path is invalid. A specified communication line is down in a shared device server.	Use the path returned by the mdOpen function. When this error occurs in the shared device server, check the communication line itself.
-2 (FFFE)	Device number error A specified device number is not within a range. When a bit device is specified, the device number is not a multiple of 8.	Check the leading device number of a specified device.

Error code A value enclosed by parentheses is hexadecimal.	Error description	Action to be taken
-3 (FFFD)	Device type error A specified device type is invalid.	Check whether the specified device type is included in a list of device types.
-4 (FFFC)	CPU error An invalid station has been specified.	Check the status of a communicating station. Check the specified station number.
-5 (FFFB)	Size error The total of device number and size is not within a range. Access has been attempted in odd bytes.	Check the specified size.
-6 (FFFA)	Block count error The number of blocks specified in dev[0] (for random reading or writing of devices) is not within a range.	Check the number of blocks specified in dev[0].
-8 (FFF8)	Channel number error The channel number specified in the mdOpen function is invalid.	Check the specified channel number.
-11 (FFF5)	Buffer area shortage The read area size specified in the read data storage array variable is too small.	Check the read size and read data storage size.
-12 (FFF4)	Block error A specified block number of the extended file register is invalid.	Check the extended file register block number (device type).
-13 (FFF3)	Write protect error A specified block number of the extended file register is overlapping the write-protect area of the memory cassette.	Check the extended file register block number (device type). Check the write-protect DIPswitch of the memory cassette to be accessed.
-14 (FFF2)	Memory cassette error The memory cassette on the accessed CPU has not been mounted or an invalid memory cassette has been mounted.	Check the memory cassette to be accessed.
-15 (FFF1)	Read area length error The read area size specified in the read data storage array variable is too small.	Check the read size and read data storage size.
-16 (FFF0)	Station number or network number error A specified station number or network number is not within a range.	Check the specified station number or network number.
-17 (FFEF)	Option (all stations or group number) specification error The option "All stations" or "Group number" has been specified for a function where the option is not available.	Check whether the option "All stations" or "Group number" is available with the function.

Error code A value enclosed by parentheses is hexadecimal.	Error description	Action to be taken
-18 (FFEE)	Remote specification error A specified code is not available.	Check the specified code.
-19 (FFED)	SEND/RECV channel number error A channel number specified for the SEND/RECV function is not within a range.	Check the specified channel number.
-31 (FFE1)	DLL load error The DLL necessary for function execution failed.	Set up the package again.
-32 (FFE0)	A resource is being occupied by another task or thread, and it cannot be released within 30 seconds.	Confirm that the resource is not being occupied by the other task or thread.
-33 (FFDF)	Access invalid error A specified destination for communication is not correct.	Check whether a correct destination for communication has been set with the utility. Check whether the destination is correct. (Refer to the CSKP Operating Manual for details.)
-34 (FFDE)	Registry open error Registry open processing failed.	
-35 (FFDD)	Registry read error Registry reading failed.	Check whether a correct destination for communication has been set with the utility.
-36 (FFDC)	Registry write error Registry writing failed.	
-37 (FFDB)	Communication initialization error Initialization for communication failed.	
-38 (FFDA)	Ethernet communication setting error Failed to make settings for Ethernet communication.	Retry. Check whether a correct destination for communication has been set with the utility. Because memory shortage is assumed, terminate the other applications. Check whether the system is operating normally. Restart the system.
-39 (FFD9)	COM communication setting error Failed to make settings for COM communication.	
-41 (FFD7)	COM control error. Failed to control COM communication.	Retry. Check whether the system is operating normally.
-42 (FFD6)	Close error Failed to close communication.	

Error code A value enclosed by parentheses is hexadecimal.	Error description	Action to be taken
-43 (FFD5)	ROM operation error TO set value was written to the CPU during ROM operation.	Change the TO set value during ROM operation.
-44 (FFD4)	LLT communication setting error Setting for making LLT communication failed.	Retry. Check the utility for correct setting of the communication target. Since memory may be insufficient, exit from the other running applications. Check whether the system is operating properly. Restart the system.
-45 (FFD3)	Ethernet control error Proper control cannot be exercised for Ethernet communication.	Retry. Check whether the system is operating properly. Restart the system.
-50 (FFCE)	Open path upper-limit over The number of open paths has exceeded the upper limit (32)	Close some paths.
-51 (FFCD)	Exclusive control error An exclusive control error occurred	Retry. Check whether the system is operating normally.
-478 to -3839 (F101 to FE22)	Refer to the QnA MELSECNET/10 Network System Reference Manual or MELSECNET/10 Network System Reference Manual (PC-PC network) for details.	
-2174 (F782)	Destination station number specification error A specified station number for a destination is for a local station.	Correct the specified station number.
-24957 (9E83)	Number of device points The specified number of device points for a destination is not within a range. When a bit device is specified, the number of bit devices is not a multiple of 8. (The destination link module checks the number of device points.)	Check the size.
-24958 (9E82)	Device number error A specified device number for a destination is not within a range. When a bit device is specified, the device number is not a multiple of 8. (The destination link module checks the device number.)	Check the device number.
-24959 (9E81)	Device type error The device type specified for a destination is invalid. (The destination link module checks the device type.)	Check the device type.

## 5. ERROR CODES

MELSEC

Error code A value enclosed by parentheses is hexadecimal.	Error description	Action to be taken
-25056 (9E20)	Processing code error A specified processing code cannot be handled by a request processing station. (The destination checks the processing code.)	Check the destination station number and processing code.
-26334 (9922)	Card reset error A card has been reset by the other process using the same channel while the other station is being accessed.	Retry.
-26336 (9920)	Other loop request error A routing request has been issued to the other loop.	Change the destination (to which the routing request is to be issued) to AnUCPU or QnACPU.
-28150 (920A)	Data link disconnection error	A local station link device has been accessed while the data link is disconnected.
-28156 (9204)	2-port memory handshake error	Remove the other option card(s).
-28158 (9202)	WDT error	Reset the card. Restart the personal computer.



# Type SW2D5F-CSKP-E Basic Communication Support Tool Programming Manual

MODEL	SW2D5F-CSKP-E-P-E
MODEL CODE	1LMS43
IB(NA)66889-A(9812)MEE	

 **MITSUBISHI ELECTRIC CORPORATION**

HEAD OFFICE : MITSUBISHI DENKI BLDG MARUNOUCHI TOKYO 100-8310 TELEX : J24532 CABLE MELCO TOKYO  
NAGOYA WORKS : 1-14, YADA-MINAMI 5, HIGASHI-KU, NAGOYA, JAPAN

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

Specifications subject to change without notice.