MITSUBISHI

Type SW1D5F-CSKP-E Basic Communication Support Tool

Programming Manual

MELSEC

Mitsubishi Programmable Controller

SAFETY PRECAUTIONS

(Prior to use, please read these precautions.)

When using the type SW1D5F-CSKP-E Communication Support Tool, thoroughly read this manual and the associated manuals referenced within.
Also pay careful attention to safety and handle the product carefully. These precautions apply only to the type SW1D5F-CSKP-E Communication Support Tool. For a description of the safety precautions applicable to a general-purpose personal computer used for running the product, refer to the manuals for your computer.
These safety precautions classify the safety precautions into two categories 'DANGER' and 'CAUTION'.

◈ DANGER	**Procedures which may lead to a dangerous condition and cause death or serious injury if not carried out correctly.**
⚠ CAUTION	**Procedures which may lead to a dangerous condition and cause superficial to medium injury, or physical damage only, if not carried out correctly.**

Depending on circumstances, procedures indicated by ⚠ CAUTION may also be linked to serious results.
In any case, it is important to follows the directions for usage.

Store this manual in a safe place so that you can take it out and read it whenever necessary. Always make it available to the end user.

[DESIGN PRECAUTIONS]

◈ DANGER
● When connecting the Communication Support Tool to a programmable controller (PC) or any other control device, install an external safety circuit that keeps the entire system safe even when there are problems with the external power supply or the personal computer ● Any write operation that modifies data in the programmable controller will directly affect the mechanical control. ● The machine may move in an unexpected way if you incorrectly specify a device name or a device number. Check the specified device names and numbers before operating your machine.

About the Manuals

The table below lists other manuals provided for the type SW1D5F-CSKP-E Communication Support Tool.

Related Manuals

Title	Manual No. (Model Code)
Type SW1D5F-CSKP-E Communication Support Tool Startup Manual This manual explains features, system configuration, and installation of the communication support package. (Option)	IB-66870 (13J933)
Type SW1D5F-CSKP-E Communication Support Tool Operating Manual This manual explains how to set and operate each communication utility. (Option)	IB-66871 (13J934)
Error List Manual for General-Purpose Personal Computer Software Package This manual explains the error codes that could be returned while using a utility or a function and gives corrective actions. (Option)	IB-66873 (13JF88)

REVISIONS

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

Print Date	*Manual Number	Revision
Sep. 1998	IB(NA)-66872-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.

INTRODUCTION

Thank you for choosing the type SW1D5F-CSKP-E Communication Support Tool.

This manual describes the MELSEC Data Link Library included in the Communication Support Tool.

Before using the Communication Support Tool, carefully read the manual to familiarize yourself with its functions.

Please make this manual available to the end user.

TRADEMARK

Microsoft Windows, Visual Basic, and Visual C^{++} are registered trade marks of Microsoft Corporation, U.S.A.
Ethernet is registered trademark of XEROX Corporation.
Corporate or product names described in this manual are registered by respective owners.

CONTENTS

1. GENERAL DESCRIPTION 1- 1 to 1- 4

1.1 General Description of MELSEC Data Link Library ··········· 1- 1
1.2 Abbreviations and Generic Names Used in This Manual ··········· 1- 2
1.3 Preparation for Using Functions ··········· 1- 3
1.3.1 Communication via the Ethernet ··········· 1- 3
1.3.2 Setting to Use a Function ··········· 1- 3

2. MELSEC DATA LINK LIBRARY 2- 1 to 2-30

2.1 List of Functions ··········· 2- 1
2.2 Programming Procedure ··········· 2- 2
2.3 Items Common to All Functions ··········· 2- 3
2.3.1 Channels ··········· 2- 3
2.3.2 Station number setup ··········· 2- 4
2.3.3 Device Type ··········· 2- 5
2.3.4 Accessible Range ··········· 2- 5
2.4 Detailed Description of Functions ··········· 2- 6
2.4.1 mdOpen(communication line open) ··········· 2- 6
2.4.2 mdClose(communication line close) ··········· 2- 7
2.4.3 mdSend(batch writing to devices) ··········· 2- 8
2.4.4 mdReceive(batch reading from devices) ··········· 2-10
2.4.5 mdRandW(write to random devices) ··········· 2-11
2.4.6 mdRandR(read from random devices) ··········· 2-13
2.4.7 mdDevSet(bit device set) ··········· 2-15
2.4.8 mdDevRst(bit device reset) ··········· 2-16
2.4.9 mdTypeRead(PC CPU type read) ··········· 2-17
2.4.10 mdControl(remote RUN/STOP/PAUSE) ··········· 2-18
2.4.11 mdIint(PC device address table refreshing) ··········· 2-19
2.4.12 mdBdRst(local card reset) ··········· 2-20
2.4.13 mdBdModSet(local card mode set) ··········· 2-21
2.4.14 mdBdModRead(local card mode read) ··········· 2-22
2.4.15 mdBdLedRead(local card LED read) ··········· 2-23
2.4.16 mdBdSwRead(local card switch read) ··········· 2-25
2.4.17 mdBdVerRead(local card version read) ··········· 2-26
2.4.18 mdSend(SEND function) ··········· 2-27
2.4.19 mdReceive(RECV function) ··········· 2-29

3. SAMPLE PROGRAM 3- 1 to 3- 8

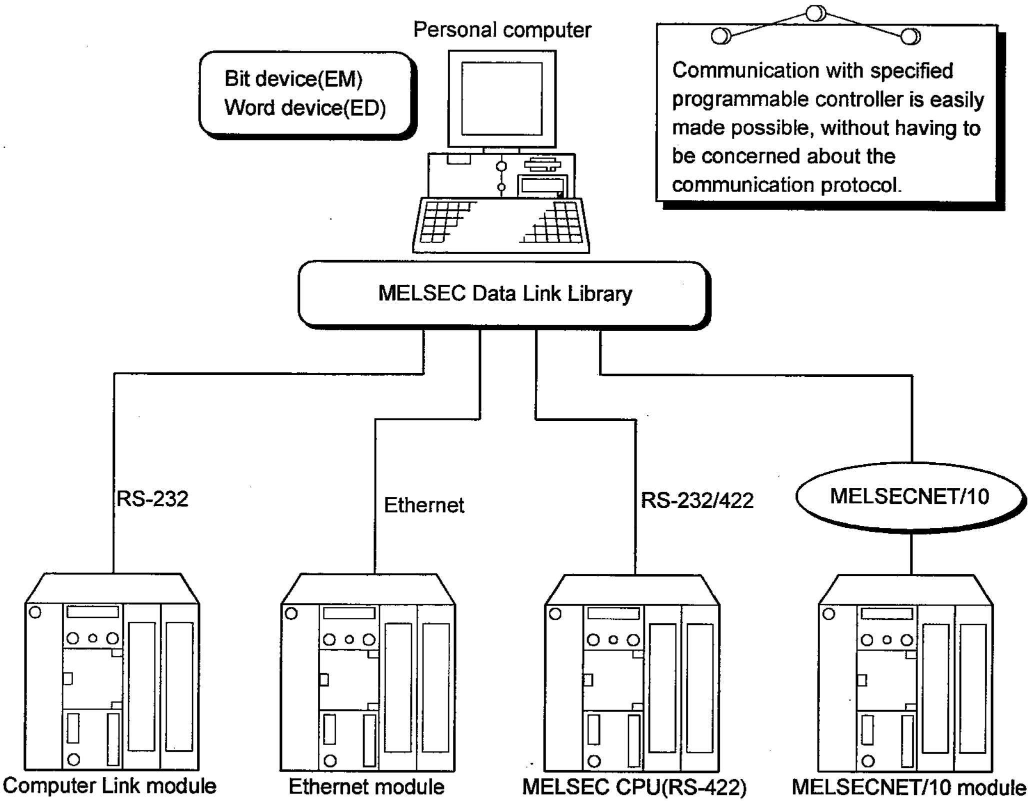
3.1 Visual Basic Sample Program ··········· 3- 1
3.2 Visual C++ Sample Program ··········· 3- 6

1. GENERAL DESCRIPTION

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

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 programmable controller 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 Abbreviations and Generic Names Used in This Manual

This manual generally uses the following abbreviations and generic names as it describes the type SW1D5F-CSKP-E Communication Support Tool.

Abbreviations/ Generic Names	Description
CSKP-E	Type SW1D5F-CSKP-E Communication Support Tool.
Windows NT 4.0	Microsoft Windows NT Workstation 4.0.
Windows 95	Microsoft Windows 95.
Windows 98	Microsoft Windows 98.
Windows	A generic name used for referring to Microsoft Windows 95, Windows 98, and Windows NT Workstation 4.0.
VB	A generic name used for referring to Microsoft Visual Basic 4.0 and Visual Basic 5.0.
VC++	Microsoft Visual C++ 4.2 and Visual C++ 5.0.
Personal computer	IBM PC/AT and its fully compatible models running DOS/V.

1.3 Preparation for Using Functions

Explanation is made for precautions in case a function is used.

1.3.1 Communication via the Ethernet

The function can be executed only when the target PC is in the RUN state.
If the PC is not in the RUN state, an error is reported.

1.3.2 Setting to Use a Function

Carry out the following setting when a function is to be used.

(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,** 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**, 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 work space 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,** 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**, then select **"Library Files"** as the directory to be displayed.

3) Open the work space 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 MELSEC DATA LINK LIBRARY

This chapter describes the specifications of the functions provided by the MELSEC Data Link Library of the CSKP-E.

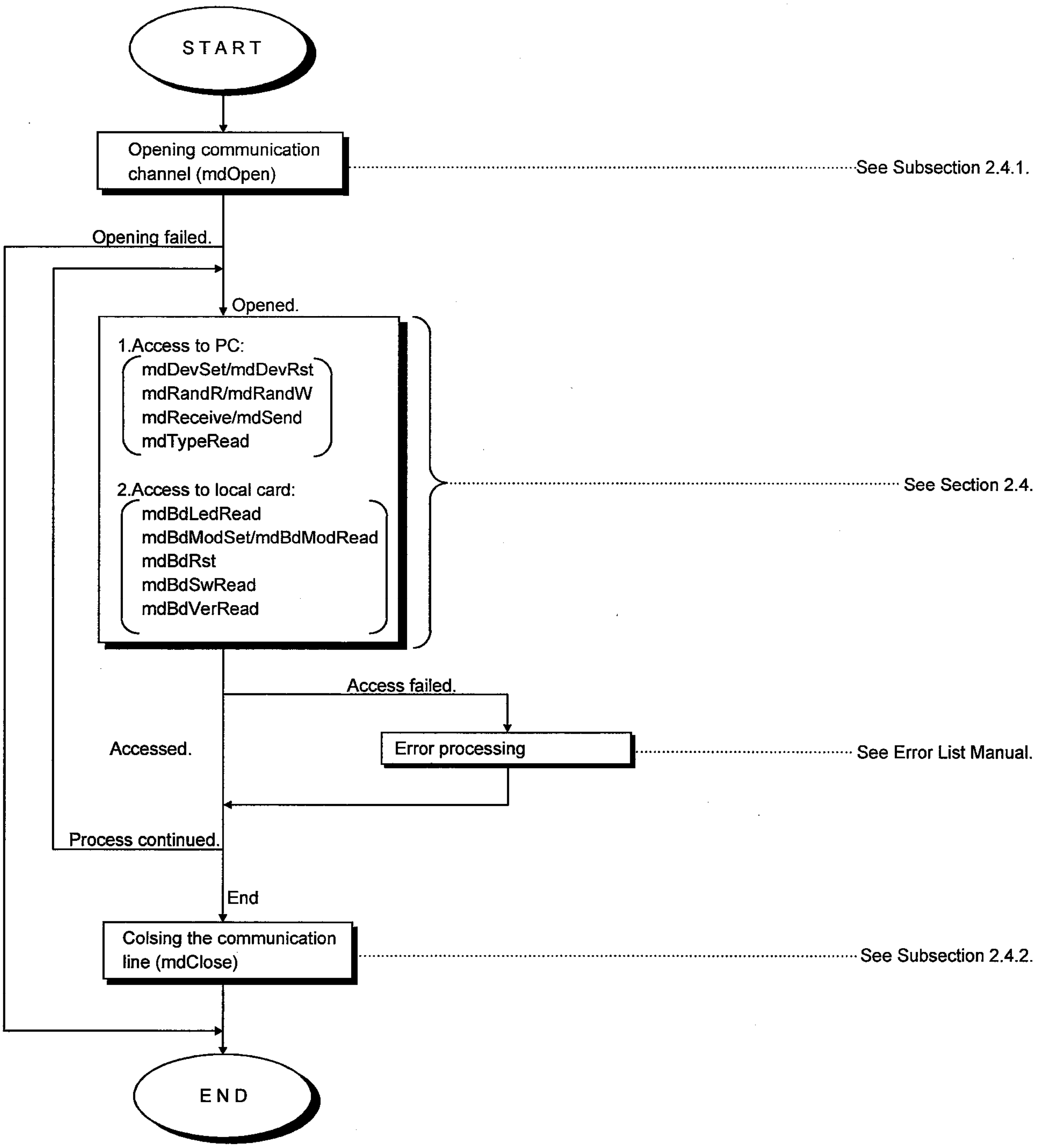
2.1 List of Functions

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

Function name	Purpose	Refer to subsection
mdOpen	Opening communication line	2.4.1
mdClose	Closing communication line	2.4.2
mdSend	Batch writing to a device	2.4.3
mdReceive	Batch reading from a device	2.4.4
mdRandW	Writing to a random device	2.4.5
mdRandR	Reading from a random device	2.4.6
mdDevSet	Setting a bit device	2.4.7
mdDevRst	Resetting a bit device	2.4.8
mdTypeRead	Reading the programmable controller CPU type	2.4.9
mdControl	Remote RUN/STOP/PAUSE	2.4.10
mdInit	Refreshing PC address	2.4.11
mdBdRst	Resetting the local card	2.4.12
mdBdModSet	Setting the mode for the local card	2.4.13
mdBdModRead	Reading the mode set to the local card	2.4.14
mdBdLedRead	Reading the states of LEDs on the local card	2.4.15
mdBdSwRead	Reading the states of switches on the local card	2.4.16
mdBdVerRead	Reading the version of the local card	2.4.17
mdSend	Data sending (SEND)	2.4.18
mdReceive	Data receiving (RECV)	2.4.19

2.2 Programming Procedure

This section explains how to code programs using functions in the MELSEC data link library



2.3 Items Common to All Functions

This section describes items that are common to all functions in the MELSEC data link library.

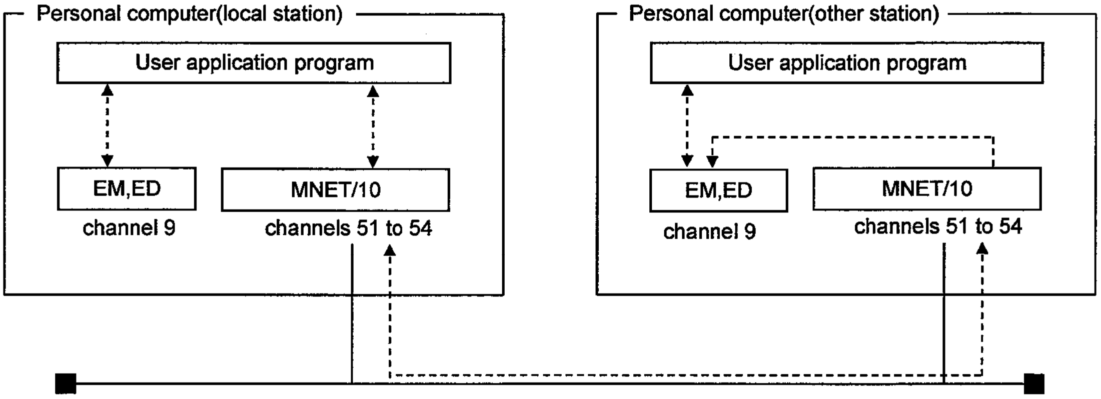
2.3.1 Channels

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

No.	Channel name	Description
9	Shared device	Used to access a shared device at the local station. (Only on Windows NT 4.0)
31 to 40	Computer link (COM1) to Computer link (COM10)	Communication via computer link Numbers starting with 31 are allocated to ports. Smaller the port number, smaller the channel number.
41 to 50	PC CPU RS-422 (COM1) to PC CPU RS-422 (COM10)	Communication with PC CPU with RS-422. Numbers starting with 41 are allocated to ports. Smaller the port number, smaller the channel number.
51 to 54	MELSECNET/10 (1 slot) to MELSECNET/10 (4 slot)	Communication via MELSECNET/10. Numbers starting with 51 are set. Smaller the card number, smaller the channel number.
61	Ethernet	Communication via the Ethernet

POINTS

- An EM or ED device can be set up on Windows NT 4.0 only using the Shared Device Server Utility. It cannot be set up on Windows 95 or 98.
- To access an EM or ED device on another personal computer (another station), specify the channel number the MELSECENET/10 card uses as the communication channel.

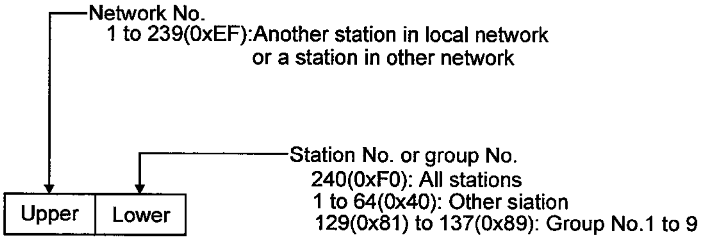


2.3.2 Station number setup

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

Communication	Station number specification
Shared device	Local station: 255(0xFF) Other station: MELSECNET/10 ····· *1
PC CPU RS-422	
MELSECNET/10	Local station: 255(0xFF) Other station: *1
Computer link	Logical station number specified with respective utilities.
Ethernet	

*1



Network No.
1 to 239(0xEF):Another station in local network or a station in other network

Station No. or group No.
240(0xF0): All stations
1 to 64(0x40): Other siation
129(0x81) to 137(0x89): Group No.1 to 9

*All station (0xF0) and group No.(0x81 to 0x89) specification are vaild only when used with the SEND function (mdSend).

2.3.3 Device Type

A device type specified when coding a function can be either a code number or a device name. For information on the accessibility of devices, see the CSKP-E Operation Manual.

Device type specification		Device	Device type specification		Device
by code	by device name		by code	by device name	
1	DevX	X	26	DevSTT	Integrating timer (contact)
2	DevY	Y	27	DevSTC	Integrating timer (coil)
3	DevL	L	28	DevQSW	QnA link special register (on QnACPU)
4	DevM	M	30	DevQV	QnA edge relay (on QnACPU)
5	DevSM	Special M(SM), SB (Link Special B for MNET/10)	35	DevSTN	Integrating timer (current value)
			101	DevMAIL	Acknowledging QnA SEND/RECV function arrival
6	DevF	F	102	DevMAILNC	Not acknowledging QnA SEND/RECV function arrival
7	DevTT	T (contact)	1001 to 1255	DevLX1 to DevLX255	Direct link input
8	DevTC	T (coil)			
9	DevCT	C (contact)			
10	DevCC	C (coil)	2001 to 2255	DevLY1 to DevLY255	Direct link output
11	DevTN	T (current value)			
12	DevCN	C (current value)			
13	DevD	D	23001 to 23255	DevLB1 to DevLB255	Direct link relay
14	DevSD	Special D(SD), SW (Link Special W for MENET/10)			
15	DevTM	T (set value, main)	24001 to 24255	DevLW1 to DevLW255	Direct link resistor
16	DevTS	T (set value, sub 1)			
16002	DevTS2	T (set value, sub 2)			
16003	DevTS3	T (set value, sub 3)	25001 to 25255	DevLSB1 to DevLSB255	Direct link special resistor (network unit side)
17	DevCM	C (set value, main)			
18	DevTS	C (set value, sub 1)			
18002	DevTS2	C (set value, sub 2)	28001 to 28255	DevLSW1 to DevLSW255	Direct link special resistor (network unit side)
18003	DevTS3	C (set value, sub 3)			
19	DevA	A			
20	DevZ	Z	29000 to 29255	DevSPG0 to DevSPG255	Special direct buffer resistor
21	DevV	V (index register)			
22	DevR	R (file register)			
22001 to 22256	DevER1 to DevER256	ER(extended register)	31000 to 31255	DevEM0 to DevEM255	EM (shared device)
23	DevB	B	32000 to 32255	DevED0 to DevED255	ED (shared device)
24	DevW	W			
25	DevQSB	QnA link special relay (on QnACPU)			

2.3.4 Accessible Range

For a description on the accessible range, refer to the CSKP-E Operation Manual.

2.4 Detailed Description of Functions

This section describes the functions in details. Each function is described in the alphabetic order of the function names.

2.4.1 mdOpen (communication line open)

□ Function

This function opens a communication line.

□ 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

□ 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.

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

□ Related function

mdClose( )

POINT

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

2.4.2 mdClose (communication line close)

□ Function

This function closes a communication line.

□ 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

□ Explanation

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

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

□ Related function

mdOpen( )

POINT

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

2.4.3 mdSend(batch writing to devices)

□ Function

This function performs batch writing to a device.

□ 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 Subsection 2.3.2.)	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,size,data);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Subsection 2.3.2.)	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

□ 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.

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

□ Related functions

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).
- When accessing a bit device, specify the head device number as follows:
  Ethernet or computer link: Multiple of 16 (0, 16, 32, ...)
  Other than above: Multiple of 8 (0, 8, 16, ...)
- 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 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.

2.4.4 mdReceive (batch reading from devices)

□ Function

This function performs a batch reading from a device.

□ 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 Subsection 2.3.2.)	IN
Integer	devtyp	Device type	IN
Integer	devno	Head device number	
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,size,data);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Subsection 2.3.2.)	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

□ 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.

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

□ 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).
- To access a bit device, specify the head device number by a multiple of 8 (0, 8, 16, so on).(For communication via the Ethernet or computer link, specify a multiple of 16.)

2.4.5 mdRandW (write to random devices)

□ Function

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

□ 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 Subsection 2.3.2.)	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 Subsection 2.3.2.)	IN
short	dev[ ];	Randomly-specified device	IN
short	buf[ ];	Data written	OUT
short	bufsize;	Dummy	IN

□ Explanation

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

Randomly-specified device (dev[ ])

dev[0] · · · ·	Number of blocks	
dev[1] · · · ·	Device type	Block No. 1
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.

□ Specifying method

Example:Setting M100 through M115 to OFF and writing 10, 200, 300, and 400 respectively to D10, D11, D12, and D13

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.
```

□ Returned value

Upon normal termination: The function returns 0.
Upon abnormal termination: The function returns a value other than 0.
(See the Error List Manual.)

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 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.
- 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.

2.4.6 mdRandR (read from random devices)

□ Function

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

□ 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 Subsection 2.3.2.)	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 Subsection 2.3.2.)	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

□ Explanation

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

Randomly-specified device (dev[ ])

dev[0]····	Number of blocks	
dev[1]····	Device type	Block No. 1
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.

□ 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 items
```

bufsize value

(buf[0] to buf[6]=7)×2=14

Data read storage area

Read

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).
```

□ Returned value

Upon normal termination: The function returns 0.
Upon abnormal termination: The function returns a value other than 0.
(See the Error List Manual.)

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).

2.4.7 mdDevSet (bit device set)

□ Function

This function sets a bit device.

□ Format

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

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	stno	Station number (See Subsection 2.3.2.)	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 Subsection 2.3.2.)	IN
short	devtyp;	Device type	IN
short	devno;	Specified device number	IN

□ Explanation

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

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

□ Related functions

mdOpen( ),mdClose( ),mdDevRst( )

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).

2.4.8 mdDevRst (bit device reset)

□ Function

This function resets a bit device.

□ Format

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

Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	stno	Station number (See Subsection 2.3.2.)	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 Subsection 2.3.2.)	IN
short	devtyp;	Device type	IN
short	devno;	Specified device number	IN

□ Explanation

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

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

□ Related functions

mdOpen( ), mdClose( ), mdDevSet( )

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).

2.4.9 mdTypeRead (PC CPU type read)

□ Function

This function reads information on the type of the programmable controller CPU.

□ Format

```
VB    : ret% = mdTypeRead(path&,stno%,buf%)
         Integer  ret     Returned value       OUT
         Long     path    Channel path          IN
         Integer  stno    Station number        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        IN
         short    buf;    Type code            OUT
```

□ Explanation

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

PC CPU type	Code	PC CPU type	Code
A1N	A1H	A2U-S1	83H
A2N,A2N-S1,A2S, A2S-S1,A1FX	A2H	A3U	84H
A3N,A1SJ,A1SH,A1SJH, A2SH,A2SH-S1	A3H	A4U	85H
A2A,A2AS,A2AS-S1, A2AS-S30,A2AS-S60, A2AS-M128	92H	Q2A,Q2AS,Q2ASH	21H
A2A-S1	93H	Q2AS-S1,Q2A-S1,Q2ASH-S1	22H
A3A	94H	Q3A	23H
A0J2H,A1S,A1S-S1,A1SJ	98H	Q4A,Q4AR	24H
A2C,A2CJ	9AH	Personal computer	90H
A2U	82H	QnA *1	20H

*1 This code is returned only when this function is executed via a QE71 unit. When the communication is made through an QE71 unit, more detailed information on the type of the QnA CPU is not available.

□ Returned value

Upon normal termination: The function returns 0.

Upon abnormal termination: The function returns a value other than 0. (See the Error List Manual.)

POINT

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

2.4.10 mdControl (remote RUN/STOP/PAUSE)

□ Function

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

□ Format

```
VB    : ret% = mdControl(path&,stno%,buf%)
        Integer  ret     Returned value                              OUT
        Long     path    Channel path                                 IN
        Integer  stno    Station number (See Subsection 2.3.2.) 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 Subsection 2.3.2.) IN
        short    buf;    Specified code                               IN
```

□ 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

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

POINTS

- If the communication driver returns an error code, the function returns that error code as is.
- An error is reported because the mdControl function provided for communication via the Ethernet or computer link does not support a remote PAUSE operation.
- An error is reported because the mdControl function issued to the Ethernet unit does not support a remote RUN/STOP operation over the own station.

2.4.11 mdInit (PC device address table refreshing)

□ Function

This function refreshes the PC device address table.

□ Format

VB : ret% = mdInit(path&)

Integer	ret	Returned value	OUT
Long	path	Channel path	IN

VC+ +: ret = mdInit(path);

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

□ Explanation

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

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

2.4.12 mdBdRst (local card reset)

□ Function

This function resets the local card.

□ 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

□ Explanation

- This function resets the local card and reads mode data in the registry to set it to the local card.
- If the local card is the management station, the function reads a link parameter in the registry to set it to the local card.

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

POINTS

- If the communication driver returns an error code, the function returns that error code as is.
- To any other application program that was accessing another station through the card reset, the function returns the card reset error code (9922H).
- The MELSECNET/10 driver clears the buffer by releasing data sent from another station with SEND.
- After being reset and until a mode is set, the local card is disconnected from data links with networks.

2.4.13 mdBdModSet (local card mode set)

□ Function

This function sets a mode for the local card.

□ 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

□ Explanation

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

Mode:

Link mode

0: on-line
(with automatic return)
2: off-line
3: forward loop test
4: reverse loop test
5: station-to-station test
(master stations)
6: station-to-station test
(slave stations)
7: self-loopback test
(internal)
8: self-loopback test
(internal)
9: hardware test

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

□ Related function

mdBdModRead( )

POINT

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

2.4.14 mdBdModRead (local card mode read)

□ Function

This function reads the mode set to the local card.

□ 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
```

□ Explanation

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

Mode:

Link mode
0: on-line
(with automatic return)
2: off-line
3: forward loop test
4: reverse loop test
5: station-to-station test
(master stations)
6: station-to-station test
(slave stations)
7: self-loopback test
(internal)
8: self-loopback test
(internal)
9: hardware test

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

□ Related function

mdBdModSet( )

POINT

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

2.4.15 mdBdLedRead (local card LED read)

□ Function

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

□ 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
```

□ Explanation

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

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

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

POINT

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

2.4.16 mdBdSwRead (local card switch read)

□ Function

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

□ 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
```

□ 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.

	Switch values	Range
buf[0]	Value set to station number switch	0 to 64
buf[1]	Value set to group number switch	0 to 9
buf[2]	Value set to network number switch	1 to 239
buf[3]	Value set to card number switch	*1
buf[4]	Value set to card identification switch	0 to 7
buf[5]	Value set to I/O address switch	*2

*1 Value of upper two digits in 2-port address
e.g. "C8H" for C8000H

*2 Value of I/O port address

□ Returned value

Upon normal termination: The function returns 0.

Upon abnormal termination: The function returns a value other than 0.
(See the Error List Manual.)

POINT

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

2.4.17 mdBdVerRead (local card version read)

□ Function

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

□ 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

□ Explanation

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

Data read	Description
00H,01H	Password: 'S' Fixed: 'G'
02H,03H	Check sum (Total from 04H to 1FH).
04H,05H	Software version 'A' to 'ZZ'
06H to 0BH	Date: (Ex) July 10, 1998 → '9' '8' '0' '7' '1' '0'
0CH to 0FH	Reserved area: 4 bytes
10H to 1FH	Software type: 16 bytes Optical double loop: "J71QLP21"
20H to 2FH	Hardware type: 16 bytes "A70BD-J71QLB23"
30H,31H	2-port memories, 1000H(4K bytes) occupied size
32H,33H	2-port attribute: 0080H fixed *1
34H,35H	Available offset: 0000H fixed
36H to 3FH	Hardware classification: 10-byte "A70BD"

*1 0080H : complete 2-port type

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

POINT

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

2.4.18 mdSend (SEND function)

□ Function

This function is used to send data.

□ Format

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

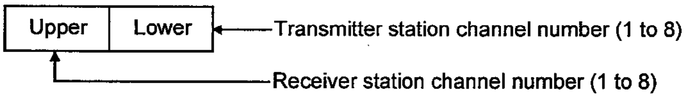
Integer	ret	Returned value	OUT
Long	path	Channel path	IN
Integer	stno	Station number (See Subsection 2.3.2.)	IN
Integer	devtyp	Device type	IN
Integer	devno	Channel number	IN
Integer	size	Size (1 to 960 bytes) of the data sent	IN/OUT
Any	data(n)	Data written(single precision integer array)	IN

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

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Subsection 2.3.2.)	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

□ 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 acknowledgment required" or **102** for "arrival acknowledgment not required.
- Specify the channel number as follows.


Upper
Lower
Transmitter station channel number (1 to 8)
Receiver station channel number (1 to 8)

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

□ Related functions

mdOpen( ), mdClose( )

POINTS

- If the communication driver returns an error code, the function returns that error code as is.
- An error will occur if a transmission is instructed specifying a channel currently in use.
- The size (number of bytes) of the data sent should be an even number.

2.4.19 mdReceive (RECV function)

□ Function

This function is used to receive data.

□ 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 Subsection 2.3.2.)	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

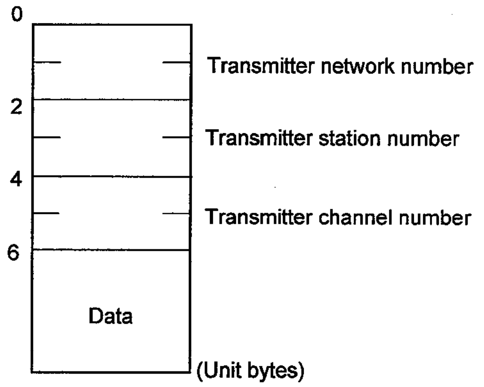
VC+ +: ret = mdReceive(path,stno,devtyp,size,data);

short	ret;	Returned value	OUT
long	path;	Channel path	IN
short	stno;	Station number (See Subsection 2.3.2.)	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

□ 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.


0
2
4
6
Transmitter network number
Transmitter station number
Transmitter channel number
Data
(Unit bytes)

□ Returned value

Upon normal termination:	The function returns 0.
Upon abnormal termination:	The function returns a value other than 0. (See the Error List Manual.)

□ Related functions

mdOpen( ), mdClose( )

POINTS

- If the communication driver returns an error code, the function returns that error code as is.
- Up to 128 pieces of received data can be stored [the total for all receiving channels (No. 1 through 8)].
  Any further data cannot be recorded in the receive data buffer.

3 SAMPLE PROGRAM

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

3.1 Visual Basic Sample Program

This is a sample program to read the data register (D) of the sequencer 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 the list of errors.
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-E Operating Manual).
In addition, carry out testing after confirming that the communication is in a normal condition.

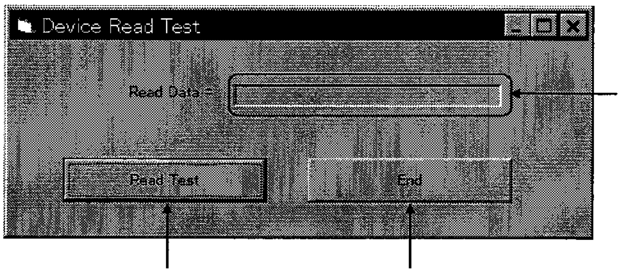
(2) Listing of sample files

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



Readout data is indicated in this area.

Readout execution button

Closing button

(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,
'** prease refer the programing 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,
'** prease refer the programing 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,
'** prease refer the programing 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 staton 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,
'** prease refer the programing 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,
'** prease refer the programing 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 staton 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,
'** prease refer the programing 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 staton 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                 'opend loop path pointer
Dim buf(128) As Integer        'read dat (single precision)

Private Sub EndBtn_Click()
    'local parameter
    Dim Ret As Integer           'returan 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 chanel
    '** open the computer link,
    '** If you want to execute open for other communicaton,
    '** prease 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 chanel
    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.

3.2 Visual C++ Sample Program

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

(1) How to use

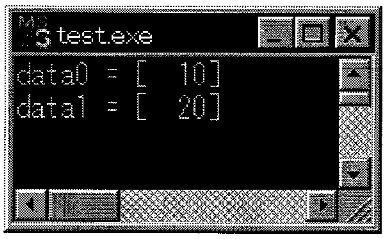
This sample program operates by creating the execution module with VC++.
When the program is executed, it implements mdOpen through the Ethernet channel.
After that, the program is implemented, and the readout data is indicated in the screen if the readout is successful.
Then, mdClose executes to close 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 the list of errors.

(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 forthe 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.



```
data0 = [   10]
data1 = [   20]
```

(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 foor 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 [%4x]\n", data[0] );     /* show data No.1   */
                        printf( "data1 [%4x]\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.

Type SW1D5F-CSKP-E Basic Communication Support Tool Programming Manual

MODEL	SW1D5F-CSKP-E-P-E
MODEL CODE	1LMS36
IB(NA)66872-A(9809)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.